Certificateless Encryption Schemes Strongly Secure in the Standard Model

Alexander W. Dent[1], Benoît Libert[2], and Kenneth G. Paterson[1]

[1] Information Security Group,
Royal Holloway, University of London (United Kingdom)
[2] UCL, Microelectronics Laboratory, Crypto Group (Belgium)

Abstract. This paper presents the first constructions for certificateless encryption (CLE) schemes that are provably secure against strong adversaries in the standard model. It includes both a generic construction for a strongly secure CLE scheme from any passively secure scheme as well as a concrete construction based on the Waters identity-based encryption scheme.

Keywords. certificateless encryption, standard model, strong security.

1 Introduction

Certificateless public key cryptography (CL-PKC), as proposed by Al-Riyami and Paterson [1], represents an interesting and potentially useful balance between identity-based cryptography and public key cryptography based on certificates. It eliminates the key escrow associated with identity-based cryptography without requiring the introduction of certificates, which pose many operational difficulties in PKIs. The main idea of CL-PKC is that a user Alice combines two key components to form her private key: one component (the partial private key, PPK) is generated by a Key Generation Centre (KGC) using a master secret, and another component (the secret value) is generated by the user herself. The user also publishes a public key derived from her secret value; a party who wishes to encrypt to Alice only needs to have Alice's identity and public key along with the KGC's public parameters. One novel aspect of CL-PKC is the modelling of adversaries who are capable of replacing the public keys of users with keys of their choice. This is necessary because there are no certificates to authenticate users' public keys in CL-PKC.

The topic of certificateless cryptography has undergone quite rapid development, with many schemes being proposed for encryption (CLE) [1, 3, 6, 12, 25] and signatures (CLS) [1, 20, 22, 33, 36]. One notable feature has been the development of a number of alternative security models for CLE that are substantially weaker than the original model of [1]. These different models are summarised by Dent [14]. In the model of [1], the attacker is of one of two types. The Type I attacker models an "outsider" adversary, who can replace the public keys of users, obtain PPKs and private keys, and make decryption queries. The Type II attacker models an "honest-but-curious" KGC who is given the master secret (and can therefore generate any PPK), can obtain private keys and make decryption queries, but is trusted not to replace any public keys. (We actually use a slightly stronger model of security for Type II attackers, in which the attacker can replace public keys providing that they do not allow the attacker to trivially break the scheme.)

In their original security model, Al-Riyami and Paterson chose to make the Type I adversary as strong as possible, insisting in their model that a challenger should correctly respond to decryption queries even if the public key of a user had been replaced. This is called a Strong Type I attacker in [14]. Currently, the only published CLE schemes that have been proven secure against strong Type I adversaries [1, 25] make use of the random oracle model [4]. Notably, Libert and Quisquater [25] provide a generic construction which converts a CLE scheme secure against passive adversaries (who do not have access to a decryption oracle) into a scheme secure against strong adversaries, using a Fujisaki-Okamoto-style conversion [17]. This conversion allows decryption queries to be handled using a form of knowledge extraction, but does require the use of random oracles.

Related Work

In 2003, Gentry [19] introduced a different but related concept named certificate based encryption (CBE). 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.

Subsequent works [34, 32] considered the relations between identity-based (IBE), certificate based (CBE) and certificateless encryption schemes (CLE) and established a result of essential equivalence [34] between the three primitives. The generic transformations of [34, 32] do not use random oracles but those results do not hold in the full security model developed in [1] for CLE schemes; indeed, they were even shown not to hold in relaxed CLE models [18].

In [15], Dodis and Katz described generic methods to construct IND-CCA secure multiple-encryption schemes from public key encryption schemes which are individually IND-CCA. They proved that their methods apply to the design of certificate-based encryption schemes [19] and yield CBE schemes without random oracles. Because of the strong properties required of decryption oracles in [1], these techniques do not directly apply in the present context. In security proofs, the technical difficulty is that the simulator does not know the secret value of entities whose public key was replaced. In other words, the constructions of [15] are not designed to handle decryption queries for arbitrary public keys chosen "on-the-fly" by adversaries who may not even know the matching secret as in the present context.

Other authors [26] have also recently attempted to address the problem of designing certificateless cryptosystems (or related primitives) in the standard model. However their results are not presented in the full model of [1]. In particular, the recent work of Huang and Wong [21] constructs a certificateless encryption scheme that is secure in the standard model but does not permit a Strong Type I adversary.

Finally, a recently initiated research direction considers authorities that maliciously generate system-wide parameters [2]. As we shall see, the model of [2] makes it even more difficult to devise schemes that are provably secure in the standard model. Neither of the schemes we present are secure against adversaries that maliciously generate the system-wide parameters.

Our Contributions

We make two contributions which resolve questions raised by the above debate concerning CLE security models.

Firstly, we present a generic construction for strongly secure CLE. Our construction uses any CLE scheme and any normal public key encryption (PKE) scheme as components, but these only need to be secure against passive adversaries. In contrast to [25], our construction does not intrinsically require the use of random oracles. Instead, we use an extension of the techniques of Naor-Yung [27] and Sahai [29]; however, some additional ideas are needed to handle decryption queries for adversarially-selected public keys. As it makes use of non-interactive zero-knowledge (NIZK) proofs for general statements in NP, our generic construction cannot be regarded as being practical.

Secondly, we provide the first concrete and efficient construction for a CLE scheme that is secure in the standard model against strong adversaries. In fact, our scheme is secure against both Strong Type I attackers and Strong Type II adversaries. The latter represents a natural strengthening of the original Type II adversary introduced in [1]. The construction is based upon the Waters identity-based encryption (IBE) scheme, modifying this scheme using ideas from [1]. The scheme enjoys relatively short public keys and ciphertexts; its security is based on the hardness of a slight and natural generalisation of the DBDH problem.

Why Consider Strong Decryption Oracles?

There has been some debate on whether the Strong Type I and Strong Type II security models correctly model the security capabilities of an attacker against a certificateless encryption scheme [1, 6, 12, 21]. A full discussion of this issue is given in the survey by Dent [14]. It can be argued that an attacker should be given access to an oracle if it supplies information that an attacker might be able to obtain in real life. For example, a decryption oracle provides information about a message that an attacker might be able to obtain by observing how a system behaves after receiving and decrypting a ciphertext or by bribing/threatening the user who received a ciphertext. In certificateless encryption, it is necessary to model the adversary's ability to fool a sender into using the wrong public key when encrypting a message, because public keys are not supported by certificates. This is done by allowing the adversary to replace public keys at will in the model. But there is no reason to suppose that a recipient would use anything other than its own, original private key when decrypting. So there is no practical reason to require that a decryption oracle for a replaced public key should be available to the attacker.

However, we still believe that the results of this paper are of theoretical interest to the research community, even if they are not practically relevant. There are several reasons for this:

- The strong models have been widely used in the previous papers and the question of whether it is possible to construct a scheme that is secure in the Strong Type I and Strong Type II models without using the random oracle methodology has been widely discussed. Indeed, it has even been conjectured that it was impossible to construct schemes that are both Strong Type I and Strong Type II secure in the standard model. In this paper, we show this conjecture to be false.
- Even if the strong model is not of practical interest, security in this model does guarantee security in the weaker, but more practically relevant, security models. Hence, at a basic level, this paper can be seen to be proving the security of several certificateless encryption schemes in the standard model (assuming honest-but-curious KGCs). Of particular interest is the generic construction presented in Section 3, which demonstrates that certificateless encryption schemes can be constructed from generic assumptions.
- Lastly, our work demonstrates that it is possible for a polynomial-time scheme to be secure in a model that allows the attacker access to oracles that compute non-polynomial-time functions (in this case computing the decryptions of ciphertexts created using arbitrary public keys). We believe that the idea of considering the security of schemes in non-polynomial-time models to be theoretically interesting.

2 Preliminaries

2.1 Notation

We use the following notation. Let $\emptyset$ denote the empty bitstring. If $\mathcal{A}$ is a deterministic algorithm, then $y \leftarrow \mathcal{A}(x)$ denotes the assignment to y of the output of $\mathcal{A}$ when run on the input x. If $\mathcal{A}$ is a randomised algorithm, then $y \xleftarrow{\$} \mathcal{A}(x)$ the assignment to y of the output of $\mathcal{A}$ when run on the input x with a fresh random tape. We let $y \leftarrow A(x; r)$ denote the assignment to y of the output of $\mathcal{A}$ when run on the input x with the random tape r. If $\mathcal{A}$ is a probabilistic polynomial-time (PPT) algorithm, then we may assume that r is of polynomial length. If S is a finite set, then $y \xleftarrow{\$} S$ denotes the random generation of an element $x \in S$ using the uniform distribution. A function $\nu : \mathbb{N} \rightarrow [0, 1]$ is said to be *negligible* if for all $c \in \mathbb{N}$ there exists a $k_c \in \mathbb{N}$ such that $\nu(k) < k^{-c}$ for all $k > k_c$.

2.2 Certificateless Encryption Schemes

The notion of a certificateless encryption scheme was introduced by Al-Riyami and Paterson [1]. A certificateless public-key encryption scheme is defined by seven probabilistic, polynomial-time algorithms:

- `Setup`: takes as input a security parameter 1^k and returns the master private key *msk* and the master public key *mpk*. This algorithm is run by a KGC to initially set up a certificateless system.
- `Extract`: takes as input the master public key *mpk*, the master private key *msk*, and an identifier $\mathsf{ID} \in \{0,1\}^*$. It outputs a partial private key d_{ID}. This algorithm is run by a KGC once for each user, and the corresponding partial private key is distributed to that user in a suitably secure manner.
- `SetSec`: given the master public key *mpk* and an entity's identifier ID as input, and outputs a secret value x_{ID} for that identity. This algorithm is run once by the user.
- `SetPriv`: takes as input the master public key *mpk*, an entity's partial private key d_{ID} and an entity's secret value x_{ID}. It outputs the full private key sk_{ID} for that user. This algorithm is run once by the user.
- `SetPub`: given the master public key *mpk* and an entity's secret value x_{ID}, this algorithm outputs a public key $pk_{\mathsf{ID}} \in \mathcal{PK}$ for that user. This algorithm is run once by the user and the resulting public key is widely and freely distributed. The public-key space $\mathcal{PK}$ is defined using *mpk* and is assumed to be publicly recognisable: given *mpk*, public keys having a matching private key should be easily distinguishable from ill-formed public keys.
- `Encrypt`: this algorithm takes as input the master public key *mpk*, a user's identity ID, a user's public key $pk_{\mathsf{ID}} \in \mathcal{PK}$ and a message $m \in \mathcal{M}$. It outputs either a ciphertext $C \in \mathcal{C}$ or the error symbol $\perp$.
- `Decrypt`: this algorithm takes as input the master public key *mpk*, a user's private key sk_{ID} and a ciphertext $C \in \mathcal{C}$. It returns either a message $m \in \mathcal{M}$ or the error symbol $\perp$.

We insist that all certificateless encryption schemes satisfy the obvious correctness conditions (that decryption "undoes" encryption).

Dent [14] has surveyed the numerous different security models proposed for certificateless encryption. In this paper, we will only be concerned with the Strong Type I and Strong Type II security definitions. Both of these security models consider attack games that extend the standard IND-CCA attack game for public-key encryption. In both games, we are concerned with the difference in probability

$$Adv_{\mathcal{A}}^{\texttt{CL-CCA-X}}(k) = |Pr[Expt_{\mathcal{A}}^{\texttt{CL-CCA-X}}(0,k)=1] - Pr[Expt_{\mathcal{A}}^{\texttt{CL-CCA-X}}(1,k)=1]|$$

for $\texttt{X} \in \{\texttt{I},\texttt{II}\}$ where $\mathcal{A}$ is any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and the experiment $Expt_{\mathcal{A}}^{\texttt{CL-CCA-X}}(b,k)$ is defined as:

$$\begin{array}{l}
Expt_{\mathcal{A}}^{\texttt{CL-CCA-X}}(b,k): \\
(mpk, msk) \xleftarrow{\$} \texttt{Setup}(1^k) \\
(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk, aux) \\
C^* \xleftarrow{\$} \texttt{Encrypt}(m_b, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk) \\
b' \xleftarrow{\$} \mathcal{A}_2(C^*, state) \\
\text{Output } b'
\end{array}$$

We insist that $\mathcal{A}_1$ outputs messages (m_0, m_1) such that $|m_0| = |m_1|$. The Type I security model ($\mathtt{X} = \mathtt{I}$) and the Type II security model ($\mathtt{X} = \mathtt{II}$) are distinguished by the value aux and the oracles to which the attacker has access. The Type I model is meant to represent an outside attacker and so $aux = \emptyset$. The Type II model captures the actions of an honest-but-curious KGC and so $aux = msk$. We consider the following oracles:

- **Request public key:** the attacker supplies an identity ID and the oracle returns the public key pk_{ID} for that identity. If pk_{ID} has not previously been defined, the oracle generates it.
- **Replace public key:** the attacker supplies an identity ID and a public key $pk_{\mathsf{ID}} \in \mathcal{PK}$, and the oracle replaces any previously generated public key for ID with pk_{ID}. Such a query is only allowed for correctly shaped new keys. Recall that the model of [1] requires the well-formedness of pk_{ID} (and the existence of a secret value) to be publicly checkable.
- **Extract partial private key:** the attacker supplies an identity ID and the oracle returns the partial private key d_{ID} for that identity.
- **Extract private key:** the attacker supplies an identity ID and the oracle responds with the full private key sk_{ID} for that identity.
- **Strong decrypt (or decrypt):** the attacker supplies an identity ID and a ciphertext C, and the oracle responds by constructing a private key sk_{ID} that corresponds to the identity ID and its associated public key. The oracle returns the decryption of C under this private key. Note that the oracle has to respond to decryption oracle queries even if the public key for the identity has been replaced.

Definition 1. *A CLE scheme is Strong Type I secure if, for every PPT adversary $\mathcal{A}$ that respects the following oracle constraints*

- *$\mathcal{A}$ cannot extract the private key for the identity ID^* at any time,*
- *$\mathcal{A}$ cannot extract the private key of any identity for which it has replaced the public key,*
- *$\mathcal{A}$ cannot extract the partial private key of ID^* if $\mathcal{A}$ replaced the public key pk_{ID^*} before the challenge was issued,*
- *$\mathcal{A}_2$ cannot query the strong decrypt oracle on the challenge ciphertext C^* for the identity ID^* unless the public key pk_{ID^*} used to create the challenge ciphertext has been replaced,*

we have that $Adv_{\mathcal{A}}^{\mathtt{CL\text{-}CCA\text{-}I}}(k)$ is negligible. In this model, $aux = \emptyset$.

Definition 2. *A CLE scheme is Strong Type II secure if, for every PPT adversary $\mathcal{A}$ that respects the following oracle constraints*

- *$\mathcal{A}$ cannot extract the private key for the identity ID^* at any time,*
- *$\mathcal{A}$ cannot extract the private key of any identity for which it has replaced the public key,*
- *$\mathcal{A}$ does not query the partial private key oracle (since it can compute them itself given msk),*

– *$\mathcal{A}_1$ cannot output a challenge identity ID^* for which it has replaced the public key,*
– *$\mathcal{A}_2$ cannot query the strong decrypt oracle on the challenge ciphertext C^* for the identity ID^* unless the public key pk_{ID^*} used to create the challenge ciphertext has been replaced.*

we have that $Adv_{\mathcal{A}}^{CL\text{-}CCA\text{-}II}(k)$ is negligible. In the Type II model, we have $aux = msk$, i.e. $\mathcal{A}_1$ takes the master private key as an additional input.

We note that the definition of Type II security only covers honest-but-curious KGCs, as originally defined by Al-Riyami and Paterson [1]. An alternative definition, proposed by Au *et al.* [2], attempts to model security against a KGC that can maliciously generate its master public and private keys. We note that our schemes are not secure in this model. Nevertheless, we claim that the original security model still captures a significant level of security and that the design of secure standard model schemes fitting the original definitions represents a significant step forward in the theory of certificateless encryption. We do not find it unrealistic to assume that KGCs are honest at key generation time and erase relevant crucial information in case they are later broken into. Furthermore, it is difficult to see how a scheme can be proven secure against malicious key generation centres and outside attackers in the standard model and with strong decryption oracles using known proof techniques. The recent work of Huang and Wong [21] proves the security of a scheme against malicious KGCs in the standard model but does not permit a Strong Type I adversary, so the construction of such a scheme should still be considered an open problem.

A certificateless encryption scheme is said to be strongly secure if it is both Strong Type I and Strong Type II secure. A certificateless encryption scheme is said to be passively secure if it is Strong Type I and Strong Type II secure against adversaries who make no decryption oracle queries.

3 Generic Construction

In this section we develop a generic construction of a strongly secure certificateless encryption scheme from a passively secure certificateless encryption scheme, a passively secure public key encryption scheme, and a non-interactive zero-knowledge proof system. We do this by adapting the ideas of Naor-Yung [27] and Sahai [29] to the certificateless setting. The requirement that the simulator be able to decrypt ciphertexts encrypted using arbitrary public keys makes the construction slightly more complicated than in the public-key encryption case.

We first recall the notion of an NP language and that of simulation-sound non-interactive zero-knowledge proof system. Our requirements are similar to those of Sahai [29], but slightly more demanding.

Definition 3. *A language $L \in \{0,1\}^*$ is an* NP *language ($L \in$ NP) if there exists a (deterministic) Turing machine R that is polynomial-time with respect to its first input and satisfies:*

$$x \in L \Longleftrightarrow \exists w \in \{0,1\}^* \text{ such that } R(x,w) = 1$$

We require a NIZK proof system that is statistically sound, computationally simulation-sound and computationally zero-knowledge. We require statistical soundness because (at one point in the proof) we will be forced to simulate a decryption oracle that can provide functionality that cannot be computed in polynomial-time, i.e. decrypting ciphertexts that are encrypted under adversarially chosen public keys.

Definition 4. *A statistically sound, computationally simulation-sound, and computationally zero knowledge non-interactive zero-knowledge proof system (NIZK) for a language $L \in \mathrm{NP}$ is a tuple $\Pi = (f, P, V, S_1, S_2)$ where f is a polynomial and P, V, S_1 and S_2 are probabilistic, polynomial-time Turing machines that satisfy the following conditions:*

- ***Complete:*** *For all $x \in L$ and all w such that $R(x, w) = 1$, and for all strings $\sigma \in \{0,1\}^{f(k)}$, we have that $V(x, \pi, \sigma) = 1$ for all $\pi \xleftarrow{\$} P(x, w, \sigma)$.*
- ***Simulation complete:*** *For all $x \in \{0,1\}^*$ and all strings $(\sigma, \kappa) \xleftarrow{\$} S_1(1^k)$, we have that $V(x, \pi, \sigma) = 1$ for all $\pi \xleftarrow{\$} S_2(x, \kappa)$. κ can be thought of as a secret key that allows S_2 to produce false proofs.*
- ***Statistically sound****: Almost all common random strings σ should not allow any false theorem to be proven. In other words,*

  $$Pr[\exists\, x \in \{0,1\}^* \setminus L\ \exists\, \pi \in \{0,1\}^* \text{ such that } V(x, \pi, \sigma) = 1]$$

  is negligible as a function of the security parameter k where the probability is taken over the choice of $\sigma \xleftarrow{\$} \{0,1\}^{f(k)}$.
- ***Simulation sound:*** *For all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ we have that $Adv_{\mathcal{A}}^{\texttt{NZIK-SS}}(k) = Pr[Expt_{\mathcal{A}}^{SS}(k) = 1]$ is negligible as a function of k, where*

$Expt_{\mathcal{A}}^{SS}(k)$:	
$(\sigma, \kappa) \xleftarrow{\$} \mathcal{S}_1(1^k)$	*Output 1 if and only if:*
$(x, state) \xleftarrow{\$} \mathcal{A}_1(1^k, \sigma)$	• $(x', \pi') \neq (x, \pi)$
$\pi \xleftarrow{\$} S_2(x, \kappa)$	• $x' \notin L$
$(x', \pi') \xleftarrow{\$} \mathcal{A}_2(\pi, state)$	• $V(x', \pi', \sigma) = 1$

- ***Zero knowledge:*** *For all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ we have that*

  $$Adv_{\mathcal{B}}^{\texttt{NIZK-ZK}}(k) = |Pr[Expt_{\mathcal{A}}(k) = 1] - Pr[Expt_{\mathcal{A}}^{S}(k) = 1]|$$

  is negligible as a function of k, where

$Expt_{\mathcal{A}}(k)$:	*$Expt_{\mathcal{A}}^{S}(k)$:*
$\sigma \xleftarrow{\$} \{0,1\}^{f(k)}$	$(\sigma, \kappa) \xleftarrow{\$} \mathcal{S}_1(1^k)$
$(x, w, state) \xleftarrow{\$} \mathcal{A}_1(1^k, \sigma)$	$(x, w, state) \xleftarrow{\$} \mathcal{A}_1(1^k, \sigma)$
If $R(x, w) = 0$, *then* $\pi \leftarrow \emptyset$	*If* $R(x, w) = 0$, *then* $\pi \leftarrow \emptyset$
Otherwise $\pi \xleftarrow{\$} P(x, w, \sigma)$	*Otherwise* $\pi \xleftarrow{\$} S_2(x, \kappa)$
Return $\mathcal{A}_2(\pi, state)$	*Return* $\mathcal{A}_2(\pi, state)$

Sahai [29] uses a (single theorem) computationally sound and computationally zero-knowledge NIZK proof system to construct a (multiple theorem) computationally sound, computationally simulation-sound and computationally zero-knowledge NIZK proof system. This construction assumes that one-way permutations exist. A brief examination of the proof verifies that we can construct a statistically sound, computationally simulation-sound NIZK proof system from a statistically sound NIZK proof system. Furthermore, it is not difficult to verify that statistically sound NIZK proof systems can be constructed for any NP language using the techniques of Feige, Lapidot and Shamir [16] under the assumption that certified trapdoor permutations exist. This condition is relaxed by Bellare and Yung [5] to require only that trapdoor permutations exist. Therefore we can construct suitably secure NIZK proof systems under the assumption that trapdoor permutations exist. Our construction will also make use of a passively-secure encryption scheme.

Definition 5. *A triple of PPT algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is an encryption scheme if (1) $\mathcal{G}$ takes as input a security parameter 1^k and outputs a public key pk and a private key sk; (2) $\mathcal{E}$ takes as input a message $m \in \mathcal{M}$ and a public key pk, and outputs a ciphertext $C \in \mathcal{C}$; and (3) $\mathcal{D}$ takes as input a ciphertext $C \in C$ and a private key sk, and outputs either a message $m \in \mathcal{M}$ or the error symbol $\perp$. This encryption scheme is said to be passively secure if the difference in probabilities*

$$Adv_{\mathcal{A}}^{\text{PKE-CPA}}(k) = |Pr[Expt_{\mathcal{A}}^{\text{PKE-CPA}}(0,k) = 1] - Pr[Expt_{\mathcal{A}}^{\text{PKE-CPA}}(1,k) = 1]|$$

is negligible for every probabilistic, polynomial-time attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. The experiment $Expt_{\mathcal{A}}^{\text{PKE-CPA}}(b,k)$ is defined as

$$\begin{array}{l} Expt_{\mathcal{A}}^{\text{PKE-CPA}}(b,k): \\ (pk, sk) \xleftarrow{\$} \mathcal{G}(1^k) \\ (m_0, m_1, state) \xleftarrow{\$} \mathcal{A}_1(1^k, pk) \\ C^* \xleftarrow{\$} \mathcal{E}(m_b, pk) \\ Return\ \mathcal{A}_2(C^*, state) \end{array}$$

where we insist that $|m_0| = |m_1|$.

We construct a strongly secure CLE scheme from a passively secure one and *two* distinct instances of a public-key encryption scheme. We use the NIZK proof system to prove that these independently generated ciphertexts all encrypt the same message. Let ($\texttt{Setup}, \texttt{Extract}, \texttt{SetSec}, \texttt{SetPriv}, \texttt{SetPub}, \texttt{Encrypt}, \texttt{Decrypt}$) be a passively secure CLE scheme and $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a passively secure public-key encryption scheme. Furthermore, let (f, P, V, S_1, S_2) be a statistically sound and computationally simulation-sound NIZK proof system for the language

$$\begin{aligned} L = \{(C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3) \,|\, \exists\, (m, r_1, r_2, r_3) \\ \text{such that } C_1 = \texttt{Encrypt}(m, pk, \mathsf{ID}, mpk_1; r_1) \\ \wedge\, C_2 = \mathcal{E}(m, mpk_2; r_2) \wedge C_3 = \mathcal{E}(m, mpk_3; r_3)\} \end{aligned}$$

Let ($\mathtt{Setup}'$, $\mathtt{Extract}$, $\mathtt{SetSec}$, $\mathtt{SetPriv}$, $\mathtt{SetPub}$, $\mathtt{Encrypt}'$, $\mathtt{Decrypt}'$) be the certificateless encryption scheme derived from the passively secure scheme and the algorithms given in Figure 1. We assume that users' public key pk and identity ID are included in their full private key sk. We also assume (for simplicity and without loss of generality) that the random tapes used by each of the algorithms is of length $poly(k)$.

$\mathtt{Setup}'(1^k)$:
$(mpk_1, msk_1) \xleftarrow{\$} \mathtt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$\sigma \xleftarrow{\$} \{0,1\}^{f(k)}$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
Output (mpk', msk')

$\mathtt{Encrypt}'(m, pk, \mathsf{ID}, mpk')$:
$r_1,\ r_2,\ r_3 \xleftarrow{\$} \{0,1\}^{poly(k)}$
$C_1 \xleftarrow{\$} \mathtt{Encrypt}(m, pk, \mathsf{ID}, mpk_1; r_1)$
$C_2 \xleftarrow{\$} \mathcal{E}(m, mpk_2; r_2)$
$C_3 \xleftarrow{\$} \mathcal{E}(m, mpk_3; r_3)$
$x \leftarrow (C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3)$
$\pi \xleftarrow{\$} P(x, (m, r_1, r_2, r_3), \sigma)$
$C \leftarrow (C_1, C_2, C_3, \pi)$
Output C

$\mathtt{Decrypt}'(C, sk, mpk')$:
$x \leftarrow (C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3)$
If $V(x, \pi, \sigma) \neq 1$ then output $\perp$
Otherwise set $m \xleftarrow{\$} \mathtt{Decrypt}(C_1, sk, mpk)$
Output m

Fig. 1. A construction for a strongly secure certificateless encryption scheme

Theorem 1. *If*

- ($\mathtt{Setup}$, $\mathtt{Extract}$, $\mathtt{SetSec}$, $\mathtt{SetPriv}$, $\mathtt{SetPub}$, $\mathtt{Encrypt}$, $\mathtt{Decrypt}$) *is a passively secure certificateless encryption scheme,*
- $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ *is a passively secure public-key encryption scheme,*
- (f, P, V, S_1, S_2) *is a statistically sound, computationally simulation-sound and computationally zero-knowledge NIZK proof system for the* NP *language*

$$L = \{(C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3) \mid \exists\,(m, r_1, r_2, r_3) \text{ such that } C_1 = \mathtt{Encrypt}(m, pk, \mathsf{ID}, mpk_1; r_1) \wedge C_2 = \mathcal{E}(m, mpk_2; r_2) \wedge C_3 = \mathcal{E}(m, mpk_3; r_3)\}$$

then the certificateless encryption scheme given in Figure 1 is secure in the Strong Type I and Strong Type II models.

The proof is given in Appendix A. It depends upon the fact that the master private key msk' does not contain the decryption keys for the public-key encryption schemes (msk_2, msk_3) or the simulation key κ for the NIZK proof system. We stress that this proof only works against Strong Type II adversaries who follow the setup procedure precisely, including the secure deletion of (msk_2, msk_3) and

κ. The scheme can be trivially broken by a KGC that can generate the master public key in an adversarial way. In the standard model, it remains an open problem to construct a scheme that is strongly secure against adversaries who can generate the master public key.

Remark 1. This construction can also be thought of as using a NIZK proof to bind the encryption of a message under a passively secure certificateless encryption scheme to the encryption of the same message under an IND-CCA2 secure encryption scheme. In the specific case of the construction that we have proposed, the IND-CCA2 encryption scheme is the Sahai [29] construction of an IND-CCA2 encryption scheme from two passively secure encryption schemes and a (separate) NIZK proof system. The proofs of security can easily be adapted to the case where an arbitrary IND-CCA2 secure encryption scheme is used.

Remark 2. We note that we may construct passively secure encryption schemes and suitably secure NIZK proof systems for any NP language from trapdoor one-way permutations [29]. Furthermore, we may construct passively secure CLE schemes from passively secure public-key encryption schemes and passively secure identity-based encryption schemes [25]. Hence, we can conclude that strongly secure certificateless encryption schemes exist provided that NIZK proof systems and passively secure identity-based encryption schemes exist. It is an open problem to show that a passively secure identity-based encryption scheme can be constructed from any recognised minimal assumption. Since it is possible to construct NIZK proof systems [10] and passively secure identity-based encryption schemes [31] under the DBDH assumption, we can conclude that there exists a strongly secure certificateless encryption schemes under the DBDH assumption alone.

Remark 3. Two public-key encryption scheme are required in order to provide security against attackers with access to a strong decryption oracle. In weaker security models, where the attacker does not have access to a strong decryption oracle, a single public-key encryption scheme suffices.

4 Concrete Construction

Our concrete construction for CLE uses *bilinear map groups*, i.e. groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p for which there is an efficiently computable mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties:

1. bilinearity: $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$ and $a, b \in \mathbb{Z}$;
2. non-degeneracy: $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$.

In such groups, we require the intractability of the following decisional problem that was suggested for the first time in [7] as a natural variant of the DBDH and DDH problems.

Definition 6. *The* Decision **3**-Party Diffie-Hellman Problem *(3-DDH) is to decide if $T = g^{abc}$ given $(g^a, g^b, g^c, T) \in \mathbb{G}^4$. Formally, we define the advantage of* *a PPT algorithm $\mathcal{A}$ as*

$$Adv_{\mathcal{A}}^{3\text{-}DDH}(k) = \begin{array}{l} |Pr[1 \xleftarrow{\$} \mathcal{A}(g^a, g^b, g^c, T) \mid T \xleftarrow{\$} g^{abc} \ \wedge \ a, b, c \xleftarrow{\$} \mathbb{Z}_p^*] \\ \qquad -Pr[1 \xleftarrow{\$} \mathcal{A}(g^a, g^b, g^c, T) \mid T \xleftarrow{\$} \mathbb{G} \ \wedge \ a, b, c \xleftarrow{\$} \mathbb{Z}_p^*]| \end{array}$$

We will assume that $Adv_{\mathcal{A}}^{3\text{-}DDH}(k)$ is negligible for all PPT algorithms $\mathcal{A}$.

Our scheme is easily adapted to work in the more general setting of prime-order groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ (instantiable from ordinary elliptic curve unlike the symmetric configuration that requires supersingular curves), in which case we need to use the obvious variant of the above hardness assumption. We also require a hash function H drawn from a family of collision resistant hash functions.

Definition 7. *A hash function $H \xleftarrow{\$} \mathcal{H}(k)$ is* collision resistant *if for all PPT algorithms $\mathcal{A}$ the advantage*

$$Adv_{\mathcal{A}}^{CR}(k) = Pr[H(x) = H(y) \wedge x \neq y \mid (x, y) \xleftarrow{\$} \mathcal{A}(1^k, H) \wedge H \xleftarrow{\$} \mathcal{H}(k)]$$

is negligible as a function of the security parameter.

Our scheme is an extension of the chosen-ciphertext secure IBE obtained by applying ideas from Boyen, Mei and Waters [9] to the 2-level hierarchical extension of the Waters IBE.

`Setup`$(1^k, n)$**:** Let $(\mathbb{G}, \mathbb{G}_T)$ be bilinear map groups of order $p > 2^k$ and let g be a generator for $\mathbb{G}$. Set $g_1 = g^\gamma$, for a random $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$, and pick a group element $g_2 \xleftarrow{\$} \mathbb{G}$ and vectors $(u', u_1, \ldots, u_n)$, $(v', v_1, \ldots, v_n) \xleftarrow{\$} \mathbb{G}^{n+1}$. We note that these vectors define the hash functions

$$F_u(\mathsf{ID}) = u' \prod_{i=1}^{n} u_j^{i_j} \qquad \text{and} \qquad F_v(w) = v' \prod_{i=1}^{n} v_j^{w_j}$$

where $\mathsf{ID} = i_1 i_2 \ldots i_n$ and $w = w_1 w_2 \ldots w_n$. We also select a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^n$. The master public key is

$$mpk \leftarrow (g, g_1, g_2, u', u_1, \ldots, u_n, v', v_1, \ldots, v_n)$$

and the master secret[3] is $msk \leftarrow g_2^\gamma$.

`Extract`$(mpk, \gamma, \mathsf{ID})$**:** Pick $r \xleftarrow{\$} \mathbb{Z}_p^*$ and return $d_{\mathsf{ID}} \leftarrow (d_1, d_2) = (g_2^\gamma \cdot F_u(\mathsf{ID})^r, g^r)$.

`SetSec`(mpk)**:** Return a randomly chosen secret value $x_{\mathsf{ID}} \xleftarrow{\$} \mathbb{Z}_p^*$.

`SetPub`(x_{ID}, mpk)**:** Return $pk_{\mathsf{ID}} \leftarrow (X, Y) = (g^{x_{\mathsf{ID}}}, g_1^{x_{\mathsf{ID}}})$.

`SetPriv`$(x_{\mathsf{ID}}, d_{\mathsf{ID}}, mpk)$**:** Parse d_{ID} into (d_1, d_2), choose $r' \xleftarrow{\$} \mathbb{Z}_p^*$ and set the private key to

$$sk_{\mathsf{ID}} \leftarrow (s_1, s_2) = (d_1^{x_{\mathsf{ID}}} \cdot F_u(\mathsf{ID})^{r'}, d_2^{x_{\mathsf{ID}}} \cdot g^{r'}) = (g_2^{\gamma x_{\mathsf{ID}}} \cdot F_u(\mathsf{ID})^t, g^t)$$

with $t = r x_{\mathsf{ID}} + r'$.

[3] In order to ensure security against Type II attacks according to definition 2, the discrete logarithms of elements g_2, $u', u_1, \ldots, u_n, v', v_1, \ldots, v_n$ w.r.t. the base g are not part of the master secret and should be deleted after key generation by the KGC.

$\mathtt{Encrypt}(m, pk_{\mathsf{ID}}, \mathsf{ID}, mpk)$**:** To encrypt $m \in \mathbb{G}_T$, parse $\mathsf{pk}_{\mathsf{ID}}$ as (X, Y), then check that it has the right shape (i.e. that $e(X, g_1)/e(g, Y) = 1_{\mathbb{G}_T}$). If so, choose $s \xleftarrow{\$} \mathbb{Z}_p^*$ and compute

$$C = (C_0, C_1, C_2, C_3) \leftarrow \Big(m \cdot e(Y, g_2)^s, g^s, F_u(\mathsf{ID})^s, F_v(w)^s\Big)$$

where $w \leftarrow H(C_0, C_1, C_2, \mathsf{ID}, pk_{\mathsf{ID}})$.

$\mathtt{Decrypt}(C, sk_{\mathsf{ID}}, mpk)$**:** Parse C as (C_0, C_1, C_2, C_3) and the private key sk_{ID} as (s_1, s_2). Check that

$$e\big(C_1, F_u(\mathsf{ID}) \cdot F_v(w)\big) = e\big(g, C_2 \cdot C_3\big)$$

where $w \leftarrow H(C_0, C_1, C_2, \mathsf{ID}, pk_{\mathsf{ID}})$, and reject C if those conditions do not hold. Otherwise, return

$$m \leftarrow C_0 \cdot \frac{e(C_2, s_2)}{e(C_1, s_1)}$$

To check the completeness, we note that private keys (s_1, s_2) satisfy

$$e(g, s_1) = e(Y, g_2) \cdot e(F_u(\mathsf{ID}), s_2) \qquad \text{and so} \qquad e(C_1, s_1) = e(Y, g_2)^s \cdot e(C_2, s_2)\,.$$

To speed up the decryption algorithm using ideas from [23], we observe that the receiver can randomly choose $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and directly return

$$m = C_0 \cdot \frac{e(C_2, s_2 \cdot g^{\alpha}) \cdot e(C_3, g^{\alpha})}{e(C_1, s_1 \cdot F_u(\mathsf{ID})^{\alpha} \cdot F_v(w)^{\alpha})}$$

which is the actual plaintext if C was properly encrypted and a random element of $\mathbb{G}_T$ otherwise. The well-formedness of C is thus implicitly checked and a product of three pairings suffices to decipher the message. This is sufficient to satisfy our security models; however, it should be noted that this system has the disadvantage of outputting a random message when presented with an invalid ciphertext. This may be a problem in some applications. In the same way, the public key validation can be made implicit at encryption: given $pk_{\mathsf{ID}} = (X, Y)$, the sender picks $\beta \xleftarrow{\$} \mathbb{Z}_p^*$ and computes $C_0 = m \cdot e(Y, g_2^s \cdot g^{s\beta})/e(X, g_1^{s\beta})$ which actually encrypts m whenever pk_{ID} has the correct shape and results in an invalid ciphertext otherwise.

We have the following security results for this concrete scheme:

Theorem 2. *Suppose $\mathcal{A}$ is a Strong Type I adversary that runs in time t, makes at most q_d decryption queries, q_{ppk} partial private key queries, and q_{pk} private key queries. Then there exists*

- *an adversary $\mathcal{A}'$ against the 3-DDH problem that has advantage $Adv_{\mathcal{A}'}^{\text{3-DDH}}(k)$ and runs in time $O(t) + O(\epsilon^{-2} \ln \delta^{-1})$ for sufficiently small ϵ and δ, and*
- *an adversary $\mathcal{A}''$ against the collision resistance of the hash function H that runs in time $O(t)$ and has advantage $Adv_{\mathcal{A}''}^{CR}(k)$*

such that the advantage of $\mathcal{A}$ is bounded by

$$Adv_{\mathcal{A}}^{\text{CL-CCA-I}}(k) < 8(q_{ppk} + q_{pk})q_d(n+1)^2 \cdot (8 \cdot Adv_{\mathcal{A}'}^{\text{3-DDH}}(k) + \delta) + Adv_{\mathcal{A}''}^{\text{CR}}(k)\,.$$

The proof of this theorem is given in Appendix B; it uses ideas from [9, 31]. Namely, the mapping F_v is chosen so as to have $F_v(w) = g_2^{J_v(w)} g^{K_v(w)}$, for certain functions J_v and K_v, in the simulation of the attack environment. Hence, for any valid ciphertext $C = (C_0, C_1, C_2, C_3)$, we have $C_1 = g^s$ and $C_3 = F_v(w)^s$, for some $s \in \mathbb{Z}_p^*$, and the simulator can extract

$$g_2^s = (C_3 / C_1^{K_v(w)})^{1/J_v(w)}$$

whenever $J_v(w) \neq 0 \bmod p$. Hence, the simulator can compute $e(Y, g_2)^s$ regardless of whether the public key $pk = (X, Y)$ was replaced or not.

Theorem 3. *Suppose $\mathcal{A}$ is a Strong Type II adversary that runs in time t and makes at most q_d decryption queries and q_{pk} private key queries. Then there exists*

- *an adversary $\mathcal{A}'$ against the 3-DDH problem that has advantage $Adv_{\mathcal{A}'}^{\text{3-DDH}}(k)$ and runs in time $O(t) + O(\epsilon^{-2} \ln \delta^{-1})$ for sufficiently small ϵ and δ, and*
- *an adversary $\mathcal{A}''$ against the collision resistance of the hash function H that runs in time $O(t)$ and has advantage $Adv_{\mathcal{A}''}^{\text{CR}}(k)$*

such that the advantage of $\mathcal{A}$ is bounded by

$$Adv_{\mathcal{A}}^{\text{CL-CCA-II}}(k) < 8q_{pk}q_d(n+1)^2 \cdot (8 \cdot Adv_{\mathcal{A}'}^{\text{3-DDH}}(k) + \delta) + Adv_{\mathcal{A}''}^{\text{CR}}(k)\,.$$

The proof of this theorem is given in Appendix B and uses similar ideas to the proof of Theorem 2.

The reductions given in the proofs of Theorems 2 and 3 leave definite room for improvement since chosen-ciphertext security is achieved by applying the Boyen-Mei-Waters techniques [9] to a 2-level HIBE.

One solution to improve the reduction is to use the Canetti-Halevi-Katz [11] or Boneh-Katz [8] techniques that significantly lengthen ciphertexts and/or introduce additional assumptions for the security of the scheme. If we borrow ideas from [35] and generate the checksum value $C_3 = F(w)^s$ using a chameleon hash function [24] in instead of Waters' "hash", an interesting tradeoff can be achieved. In the above variant, a single element of $\mathbb{Z}_p^*$ (acting as random coins used to compute of the chameleon hash function) should be appended to ciphertexts and the degradation factor q_d is avoided in both reductions. Using a chameleon hash function built upon Pedersen's discrete-logarithm-based trapdoor commitment [28], the resulting combination does not imply any additional intractability assumption for the security of the final scheme.

Acknowledgements

The authors would like to thank Douglas Wikström for an initial conversation about whether it would be possible to construct strong certificateless encryption using Naor-Yung style techniques, and Eike Kiltz for several discussions on artificial aborts. The authors would also like to thank the PKC referees and David Galindo for their helpful comments. The second author acknowledges the Belgian National Fund for Scientific Research (F.R.S.-F.N.R.S.) for their support. This work was supported in part by the European Commission under Contract IST-2002-507932 ECRYPT.

References

1. S. S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In C. S. Laih, editor, *Advances in Cryptology – Asiacrypt 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 452–473. Springer-Verlag, 2003.
2. M. H. Au, J. Chen, J. K. Liu, Y. Mu, D. S. Wong, and G. Yang. Malicious KGC attack in certificateless cryptography. In *Proc. ACM Symposium on Information, Computer and Communications Security*. ACM Press, 2007.
3. J. Baek, R. Safavi-Naini, and W. Susilo. Certificateless public key encryption without pairing. In J. Zhou and J. Lopez, editors, *Proceedings of the 8th International Conference on Information Security (ISC 2005)*, volume 3650 of *Lecture Notes in Computer Science*, pages 134–148. Springer-Verlag, 2005.
4. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of the First ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
5. M. Bellare and M. Yung. Certifying permutations: Non-interactive zero-knowledge based on any trapdoor permutation. *Journal of Cryptology*, 9(1):149–166, 1996.
6. K. Bentahar, P. Farshim, J. Malone-Lee, and N. P. Smart. Generic constructions of identity-based and certificateless KEMs. Available from `http://eprint.iacr.org/2005/058`, 2005.
7. D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003.
8. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In A. Menezes, editor, *Topics in Cryptology - CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer-Verlag, 2005.
9. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *Proc. of the 12th ACM Conference on Computer and Communications Security*, pages 320–329, 2005.
10. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In E. Biham, editor, *Advances in Cryptology – Eurocrypt 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer-Verlag, 2003. Full version available at `http://eprint.iacr.org/2003/083`.
11. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – Eurocrypt 2003*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer-Verlag, 2004.

12. Z. Cheng and R. Comley. Efficient certificateless public key encryption. Available from `http://eprint.iacr.org/2005/012/`, 2005.
13. A. W. Dent. A note on game-hopping proofs. Available from `http://eprint.iacr.org/2006/260`, 2006.
14. A. W. Dent. A survey of certificateless encryption schemes and security models. Available from `http://eprint.iacr.org/2006/211`, 2006.
15. Y. Dodis and J. Katz. Chosen-ciphertext security of multiple encryption. In J. Kilian, editor, *Theory of Cryptography – TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 188–209. Springer-Verlag, 2005.
16. U. Feige, D. Lapidot, and A. Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SAIM Journal on Computing*, 29(1):1–28, 1999.
17. E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimal cost. In H. Imai and Y. Zheng, editors, *Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68. Springer-Verlag, 1999.
18. D. Galindo, P. Morillo, and C. Ràfols. Breaking Yum and Lee generic constructions of certificate-less and certificate-based encryption schemes. In A. S. Atzeni and A. Lioy, editors, *Public Key Infrastructure: Third European PKI Workshop (EuroPKI 2006)*, volume 4043 of *Lecture Notes in Computer Science*, pages 81–91. Springer-Verlag, 2006.
19. C. Gentry. Certificate-based encryption and the certificate revocation problem. In *Eurorypt'03*, volume 2656 of *LNCS*, pages 272–293. Springer, 2003.
20. B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng. Key replacement attack against a generic construction of certificateless signature. In L. M. Batten and R. Safavi-Naini, editors, *11th Australasian Conference on Information Security and Privacy (ACISP 2006)*, volume 4058 of *Lecture Notes in Computer Science*, pages 235–246. Springer-Verlag, 2006.
21. Q. Huang and D. S. Wong. Generic certificateless encryption in the standard model. In A. Miyaji, H. Kikuchi, and K. Rannenberg, editors, *Advances in Information and Computer Security (IWSEC 2007)*, volume 4752 of *Lecture Notes in Computer Science*, pages 278–291. Springer-Verlag, 2007.
22. X. Huang, W. Susilo, Y. Mu, and F. Zhang. On the security of certificateless signature schemes from Asiacrypt 2003. In Y. Li and Y. Mu, editors, *Cryptology and Network Security – CANS 2005*, volume 3810 of *Lecture Notes in Computer Science*, pages 13–25. Springer-Verlag, 2005.
23. E. Kiltz and D. Galindo. Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. In L. M. Batten and R. Safavi-Naini, editors, *Australasian Conference in Information Security and Privacy (ACISP 2006)*, volume 4058 of *Lecture Notes in Computer Science*, pages 336–347. Springer-Verlag, 2006.
24. H. Krawczyk and T. Rabin. Chameleon signatures. In *In the Proceedings of the Network and Distributed Systems Symposium (NDSS 2000)*, pages 143–154, 2000.
25. B. Libert and J.-J. Quisquater. On constructing certificateless cryptosystems from identity based encryption. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, *Public Key Cryptography – PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 474–490. Springer-Verlag, 2006.
26. J. K. Liu, M. H. Au, and W. Susilo. Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In *Proc. ACM Symposium on Information, Computer and Communications Security*. ACM Press, 2007.

27. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proc. 22nd Symposium on the Theory of Computing, STOC 1990*, pages 427–437. ACM, 1990.
28. T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *Advance in Cryptology – Crypto'91*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer-Verlag, 1991.
29. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99*, pages 543–553. IEEE Computer Society, 1999.
30. V. Shoup. Sequences of games: A tool for taming complexity in security proofs. Available from `http://eprint.iacr.org/2004/332/`, 2004.
31. B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer-Verlag, 2005.
32. D. H. Yum and P. J. Lee. Generic construction of certificateless encryption. In Antonio Laganà *et al.*, editor, *Computational Science and Its Applications ICCSA 2004: Part I*, volume 3043 of *Lecture Notes in Computer Science*, pages 802–811. Springer-Verlag, 2004.
33. D. H. Yum and P. J. Lee. Generic construction of certificateless signature. In H. Wang, J. Pieprzyk, and V. Varadharajan, editors, *9th Australasian Conference in Information Security and Privacy (ACISP 2004)*, volume 3108 of *Lecture Notes in Computer Science*, pages 200–211. Springer-Verlag, 2004.
34. D. H. Yum and P. J. Lee. Identity-based cryptography in public key management. In S. K. Katsikas, S. Gritzalis, and J. Lopez, editors, *Public Key Infrastructure: First European PKI Workshop (EuroPKI 2004)*, volume 3093 of *Lecture Notes in Computer Science*, pages 71–84. Springer-Verlag, 2004.
35. R. Zhang. Tweaking TBE/IBE to PKE transforms with chameleon hash functions. In J. Katz and M. Yung, editors, *Applied Cryptography and Network Security*, volume 4521 of *Lecture Notes in Computer Science*, pages 323–339. Springer, 2007.
36. Z. Zhang, D. S. Wong, J. Xu, and D. Feng. Certificateless public-key signature: Security model and efficient construction. In J. Zhou, M. Yung, and F. Bao, editors, *Applied Cryptography and Network Security*, volume 3989 of *Lecture Notes in Computer Science*, pages 293–308. Springer-Verlag, 2006.

A Proof of Theorem 1

We will use standard game hopping techniques. The proof is simple, but it requires a large number of game hops. In the language of Shoup [30], almost every transition will be a "transition based on indistinguishability". We seek to bound the advantage of an arbitrary probabilistic polynomial-time attacker attacker $\mathcal{A}$

$$Adv_{\mathcal{A}}^{\mathtt{CL\text{-}CCA\text{-}X}}(k) = |\Pr[Expt_{\mathcal{A}}^{\mathtt{CL\text{-}CCA\text{-}X}}(0,k) = 1] - \Pr[Expt_{\mathcal{A}}^{\mathtt{CL\text{-}CCA\text{-}X}}(1,k) = 1]|$$

for $\mathtt{X} \in \{\mathtt{I}, \mathtt{II}\}$. The proof strategy is the same for both Type I and Type II security. First, we re-write $Expt_{\mathcal{A}}^{\mathtt{CL\text{-}CCA\text{-}X}}(b,k)$ as $Expt_{\mathcal{A}}^{\mathtt{CL\text{-}CCA\text{-}X}}(\alpha,\beta,\gamma,k)$ where

$Expt^{\texttt{CL-CCA-X}}_{\mathcal{A}}(\alpha, \beta, \gamma, k)$:
$(mpk_1, msk_1) \xleftarrow{\$} \texttt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$\sigma \xleftarrow{\$} \{0,1\}^{f(k)}$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$
$r_1^*, r_2^*, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \texttt{Encrypt}(m_\alpha, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_\beta, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_\gamma, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2, C_3^*, mpk_3)$
$w^* \leftarrow (m_b, r_1^*, r_2^*, r_3^*)$
$\pi^* \xleftarrow{\$} P(x^*, w^*, \sigma)$
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$

Hence,

$$Expt^{\texttt{CL-CCA-X}}_{\mathcal{A}}(b, k) = Expt^{\texttt{CL-CCA-X}}_{\mathcal{A}}(b, b, b, k) .$$

The proof strategy is given in Figure 2. It is clear that if we can show that the difference between the success probabilities in successive games is negligible, then we will have shown that the difference in probabilities between the first and last game is negligible, and can therefore conclude that the proposed certificateless encryption scheme is secure.

Game	Description	Transition
Game 1	This game is identical to $Expt^{\texttt{CL-CCA-X}}_{\mathcal{A}}(0, k)$	
Game 2	We change the way in which the decryption oracle works. To decrypt a ciphertext (C_1, C_2, C_3, π), instead of searching for the unique message which encrypts to the given ciphertext (a non-polynomial-time operation), we simply decrypt the ciphertext C_3 using msk_3.	Statistically sound
Game 3	We simulate the NIZK proof in the challenge ciphertext rather than generating a proper NIZK proof. This will allow us later to produce false proofs.	Zero knowledge
Game 4	We change the value of α from 0 to 1.	Type `X` security
Game 5	We change the value of β from 0 to 1.	Security of the encryption scheme
Game 6	It would be nice if, at this stage, we could change γ from 0 to 1. However, if we were to try to do this, we would run into a problem as we would not be able to simulate the decryption oracle. Hence, we change the way the decryption oracle works. A ciphertext (C_1, C_2, C_3, π) is now decrypted by decrypting the ciphertext C_2 using the msk_2.	Simulation-sound
Game 7	We change the value of γ from 0 to 1.	Security of the encryption scheme
Game 8	We produce the NIZK proof in the challenge ciphertext using the correct algorithm.	Zero knowledge
Game 9	We change the way the decryption oracle works back to the correct method, i.e. the oracle searches for the unique value m gives the ciphertext when encrypted. This games is now identical to $Expt^{\texttt{CL-CCA-X}}_{\mathcal{A}}(1, k)$	Statistically sound

Fig. 2. The proof strategy for proving Theorem 1

Remark. The construction of secure public key encryption schemes presented by Sahai [29] only requires the combination of two passively secure components with a NIZK. It may be wondered why our construction requires three passively secure components (two public-key encryption schemes and a certificateless encryption scheme). The use of two public-key encryption schemes in our CLE construction allows us to simulate the strong decryption oracles. If we were to use a single public-key encryption scheme, then we would be unable to simulate the strong decryption oracle when we make the game hop based on the security of that encryption scheme. This is because we cannot decrypt arbitrary ciphertexts using the private key of the certificateless encryption scheme. It should be noted that if we only require security in a model without strong decryption oracles, then a single public-key encryption scheme suffices.

Theorem 4. *Let*

- $\Pi = (\mathtt{Setup}, \mathtt{Extract}, \mathtt{SetSec}, \mathtt{SetPriv}, \mathtt{SetPub}, \mathtt{Encrypt}, \mathtt{Decrypt})$ *be a passively secure certificateless encryption scheme,*
- $\Gamma = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ *be a passively secure public-key encryption scheme,*
- $\Sigma = (f, P, V, S_1, S_2)$ *be a statistically sound, computationally simulation-sound and computationally zero-knowledge NIZK proof system for the language*

$$L = \{(C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3) \mid \exists\, (m, r_1, r_2, r_3) \\ \text{such that } C_1 = \mathtt{Encrypt}(m, pk, \mathsf{ID}, mpk_1; r_1) \\ \wedge\, C_2 = \mathcal{E}(m, mpk_2; r_2) \wedge C_3 = \mathcal{E}(m, mpk_3; r_3)\}\,,$$

- *and let* $\Pi' = (\mathtt{Setup}', \mathtt{Extract}, \mathtt{SetSec}, \mathtt{SetPriv}, \mathtt{SetPub}, \mathtt{Encrypt}', \mathtt{Decrypt}')$ *be the certificateless encryption scheme defined by* Π *and the algorithm contained in Figure 1.*

If $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *is any probabilistic, polynomial-time adversary in Strong Type I security model, then*

$$Adv_{\mathcal{A}}^{\mathit{CL\text{-}CCA\text{-}I}}(k) \le 2Adv^{\mathit{NIZK\text{-}Sim}}(k) + 2Adv_{\mathcal{B}}^{\mathit{NIZK\text{-}ZK}}(k) \\ + 2Adv_{\mathcal{B}}^{\mathit{PKE\text{-}CPA}}(k) + Adv_{\mathcal{B}}^{\mathit{NIZK\text{-}SS}}(k) + Adv_{\mathcal{B}}^{\mathit{CL\text{-}CPA\text{-}I}}(k)\,.$$

If $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *is an probabilistic, polynomial-time adversary in the Strong Type II security model, then adversary in Strong Type I security model, then*

$$Adv_{\mathcal{A}}^{\mathit{CL\text{-}CCA\text{-}II}}(k) \le 2Adv^{\mathit{NIZK\text{-}Sim}}(k) + 2Adv_{\mathcal{B}}^{\mathit{NIZK\text{-}ZK}}(k) \\ + 2Adv_{\mathcal{B}}^{\mathit{PKE\text{-}CPA}}(k) + Adv_{\mathcal{B}}^{\mathit{NIZK\text{-}SS}}(k) + Adv_{\mathcal{B}}^{\mathit{CL\text{-}CPA\text{-}II}}(k)\,.$$

Proof The proof uses standard game hopping techniques.

Game 1 and Game 2 – Simulating the decryption oracle (part 1)

Let Game 1 be identical to $Expt_{\mathcal{A}}^{\mathtt{CL\text{-}CCA\text{-}X}}(0, k)$.

Let Game 2 be identical to Game 1 except that we change the way in which the decryption oracle handles decryption queries. In Game 2, the decryption oracle uses the following algorithm to decrypt ciphertexts:

$Decrypt'(C, sk, mpk)$:
$x \leftarrow (C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3)$
If $V(x, \pi, \sigma) \neq 1$ then output $\perp$
Otherwise set $m \xleftarrow{\$} \mathcal{D}(C_3, msk_3)$
Output m

Furthermore, let

$$Adv^{\texttt{NIZK-Sim}}(k) = \Pr[\exists\, x \in \{0,1\}^* \setminus L\ \exists\, \pi \in \{0,1\}^* \text{ such that } V(x,\pi,\sigma) = 1 | \sigma \xleftarrow{\$} \{0,1\}^{f(k)}]\,,$$

which is negligible as a function of k by the assumption that Σ is a statistically sound NIZK proof system. We will need to use the following basic lemma:

Lemma 1. *Let A, B and E be events in some probability space and suppose that* $\Pr[A|\neg E] = \Pr[B|\neg E]$. *Then* $|\Pr[A] - \Pr[B]| \leq \Pr[E]$.

We may now state and prove the lemma which captures the difficulty associated with this game hop.

Lemma 2.

$$|\Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 1] - \Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 2]| \leq Adv^{\textit{NIZK-Sim}}(k)$$

Proof Game 1 and Game 2 proceed identically unless $\mathcal{A}$ submits a decryption query (C_1, C_2, C_3, π) for which $x \leftarrow (C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3)$ satisfies $V(x, \pi, \sigma) = 1$ but $\texttt{Decrypt}(C_1, sk_{\mathsf{ID}}, mpk) \neq \mathcal{D}(C_3, msk_3)$. Let E be the event that this occurs, and let A and B be the events that $\mathcal{A}$ outputs 1 in Game 1 and Game 2 respectively. Note if E occurs, then $x \notin L$; hence, $\Pr[E] \leq Adv^{\texttt{NIZK-Sim}}$. The result can now be proven by applying Lemma 1. □

Game 3 – Simulating the challenge ciphertext (part 1)

We next replace the proof π^* used in the challenge ciphertext with a simulated proof. Let Game 3 be identical to Game 2 except that the proof π^* in the challenge encryption is produced using the simulated proof algorithms, rather than the real proof algorithm. In other words, we define Game 3 to be:

Game $3(k)$:
$(mpk_1, msk_1) \xleftarrow{\$} \texttt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$(\sigma, \kappa) \xleftarrow{\$} S_1(1^k)$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$
$r_1^*, r_2^*, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \texttt{Encrypt}(m_b, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_b, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_b, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2, C_3^*, mpk_3)$
$\pi^* \xleftarrow{\$} S_2(x^*, \kappa)$
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$

Lemma 3.

$$|\Pr[\mathcal{A} \textit{ outputs } 1|\mathcal{A} \textit{ plays Game } 2] - \Pr[\mathcal{A} \textit{ outputs } 1|\mathcal{A} \textit{ plays Game } 3]| \leq Adv_{\mathcal{B}}^{\texttt{NIZK-ZK}}(k)$$

where $\mathcal{B}$ is an algorithm that runs in approximately the same time as $\mathcal{A}$.

Proof Consider the following probabilistic, polynomial-time algorithm $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against the zero knowledge property of the NIZK proof system Σ:

$\mathcal{B}_1(1^k, \sigma)$:
$(mpk_1, msk_1) \xleftarrow{\$} \texttt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$
$r_1^*, r_2^*, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \texttt{Encrypt}(m_0, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2, C_3^*, mpk_3)$
$w^* \leftarrow (m_b, r_1, r_2, r_3)$
Output $(x^*, w^*, state)$

$\mathcal{B}_2(\pi^*, state)$:
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$
If $b' = b$ then output 1
Otherwise output 0

It is easy to see that $\mathcal{B}$ can handle all of $\mathcal{A}$'s oracle queries trivially (using its knowledge of msk') except for the decryption oracle queries. This it handles by first checking whether $V(x, \pi, \sigma) = 1$ and (if so) responding with $\mathcal{D}(C_3, msk_3)$. Therefore, it is easy to see that

$$\Pr[\mathcal{A} \text{ outputs } 1|\mathcal{A} \text{ plays Game } 2] = \Pr[Expt_{\mathcal{B}}(k) = 1]$$

and

$$\Pr[\mathcal{A} \text{ outputs } 1|\mathcal{A} \text{ plays Game } 3] = \Pr[Expt_{\mathcal{B}}^{S}(k) = 1]\,.$$

Hence, the lemma holds by the definition of computational zero knowledge. □

Game 4 – The passive Type X security of the certificateless scheme

Let Game 4 be identical to Game 3 except that, when we construct the challenge ciphertext, the `Encrypt` algorithm is applied to m_1 rather than m_0. In other words,

Game $4(k)$:
$(mpk_1, msk_1) \xleftarrow{\$} \texttt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$(\sigma, \kappa) \xleftarrow{\$} S_1(1^k)$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$
$r_1^*, r_2^*, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \texttt{Encrypt}(m_1, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2, C_3^*, mpk_3)$
$\pi^* \xleftarrow{\$} S_2(x^*, \kappa)$
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$

Let $Adv_{\mathcal{B}}^{\texttt{CL-CPA-X}}(k) = Adv_{\mathcal{B}}^{\texttt{CL-CCA-X}}(k)$ for all probabilistic, polynomial-time adversaries $\mathcal{B}$ which make no decryption oracle queries. This value will be negligible of the certificateless scheme in question is passively secure.

Lemma 4.

$$|\Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 3] - \Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 4]| \leq Adv_{\mathcal{B}}^{\textit{CL-CPA-X}}(k)$$

where $\mathcal{B}$ is an algorithm that runs in approximately the same time as $\mathcal{A}$.

Proof Consider the following probabilistic, polynomial-time adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against the passive security of the certificateless encryption scheme Π:

$\mathcal{B}_1(1^k, mpk_1)$:
$(\sigma, \kappa) \xleftarrow{\$} S_1(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$

$\mathcal{B}_2(C_1^*, state)$:
$r_2^*, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2, C_3^*, mpk_3)$
$\pi^* \xleftarrow{\$} S_2(x^*, \kappa)$
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$

Note that $\mathcal{B}$ may answer all of $\mathcal{A}$'s oracle queries using its own oracles, except for $\mathcal{A}$'s decryption oracle queries. $\mathcal{B}$ may answer $\mathcal{A}$'s decryption oracle queries directly using msk_3. Therefore, it is easy to see that

$$\Pr[\mathcal{A} \text{ outputs } 1 | \mathcal{A} \text{ plays Game } 3] = \Pr[Expt_{\mathcal{B}}^{\texttt{CL-CPA-X}}(0, k) = 1]$$

and

$$\Pr[\mathcal{A} \text{ outputs } 1 | \mathcal{A} \text{ plays Game } 4] = \Pr[Expt_{\mathcal{B}}^{\texttt{CL-CPA-X}}(1, k) = 1] .$$

Hence, the lemma holds by the definition of a passively secure certificateless encryption scheme. $\square$

Game 5 – The passive security of the encryption scheme (part 1)

Let Game 5 be identical to Game 4 except that, when we construct the challenge ciphertext, the first instance of the public key encryption algorithm $\mathcal{E}$ is applied to m_1 rather than m_0. In other words,

Game 5(k):
$(mpk_1, msk_1) \xleftarrow{\$} \texttt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$(\sigma, \kappa) \xleftarrow{\$} S_1(1^k)$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$
$r_1^*, r_2^*, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \texttt{Encrypt}(m_1, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_1, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2, C_3^*, mpk_3)$
$\pi^* \xleftarrow{\$} S_2(x^*, \kappa)$
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$

Lemma 5.

$$|\Pr[\mathcal{A} \textit{ outputs } 1|\mathcal{A} \textit{ plays Game } 4] - \Pr[\mathcal{A} \textit{ outputs } 1|\mathcal{A} \textit{ plays Game } 5]| \leq Adv_{\mathcal{B}}^{\textit{PKE-CPA}}(k)$$

where $\mathcal{B}$ is an algorithm that runs in approximately the same time as $\mathcal{A}$.

Proof Consider the following probabilistic, polynomial-time adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against the passive security of the public-key encryption scheme $\mathit{\Gamma}$:

$\mathcal{B}_1(1^k, mpk_2)$:
$(\sigma, \kappa) \xleftarrow{\$} S_1(1^k)$
$(mpk_1, msk_1) \xleftarrow{\$} \texttt{Setup}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$

$\mathcal{B}_2(C_2^*, state)$:
$r_1^*, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \texttt{Encrypt}(m_1, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2, C_3^*, mpk_3)$
$\pi^* \xleftarrow{\$} S_2(x^*, \kappa)$
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$

Note that $\mathcal{B}$ can answer all of $\mathcal{A}$'s oracle queries using its knowledge of msk_1 and msk_3. Therefore, it is easy to see that

$$\Pr[\mathcal{A} \text{ outputs } 1|\mathcal{A} \text{ plays Game } 4] = \Pr[Expt_{\mathcal{B}}^{\texttt{PKE-CPA}}(0,k) = 1]$$

and

$$\Pr[\mathcal{A} \text{ outputs } 1|\mathcal{A} \text{ plays Game } 5] = \Pr[Expt_{\mathcal{B}}^{\texttt{PKE-CPA}}(1,k) = 1]\,.$$

Hence, the lemma holds by the definition of a passively secure public-key encryption scheme. □

Game 6 – Simulating the decryption algorithm (part 2)

It would be nice if, at this stage, we could use the same argument as in Game 5 to move to a game in which m_1 was encrypted by the second instance of the public-key encryption scheme. Unfortunately, we cannot do this at the moment. The reason for this is that, in the adversarial algorithm $\mathcal{B}$ which we used to relate Game 4 and Game 5, we used our knowledge of msk_3 in order to simulate the decryption algorithm. Therefore, if we were to try and change the second instance of public-key encryption scheme to encrypt m_1, then we would not know msk_3 and therefore would not be able to simulate the decryption algorithm. In order to solve this problem, we change the decryption algorithm so that it decrypts ciphertexts using msk_2 rather than msk_3.

Let Game 6 be identical to Game 5 except that the decryption algorithm used by the decryption oracle is changed to

$Decrypt'(C, sk, mpk)$:
$x \leftarrow (C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3)$
If $V(x, \pi, \sigma) \neq 1$ then output $\perp$
Otherwise set $m \xleftarrow{\$} \mathcal{D}(C_2, msk_2)$
Output m

Lemma 6.

$$|\Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 5] - \Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 6]| \leq Adv_{\mathcal{B}}^{\textit{NIZK-SS}}(k)$$

where $\mathcal{B}$ is an algorithm that runs in approximately the same time as $\mathcal{A}$.

Proof We apply Lemma 1 for this result. Let A and B be the events that $\mathcal{A}$ outputs 1 in Game 5 and Game 6 respectively. Let E be the event that $\mathcal{A}$ submits a ciphertext (C_1, C_2, C_3, π) to the decryption oracle such that $\mathcal{D}(C_2, msk_2) \neq \mathcal{D}(C_3, msk_3)$ but $x \leftarrow (C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3)$ satisfies $V(x, \pi, \sigma) = 1$. Note that if E does not occur, then we must have that $\mathcal{D}(C_2, msk_2) = \mathcal{D}(C_3, msk_3)$ and so Game 5 and and Game 6 proceed identically. Furthermore, note that if E does occur then we have found a pair (x, π) such that $x \notin L$ but $V(x, \pi, \sigma) = 1$. Now, by Lemma 1, we have that

$$|\Pr[\mathcal{A} \text{ outputs } 1 | \mathcal{A} \text{ plays Game } 5] - \Pr[\mathcal{A} \text{ outputs } 1 | \mathcal{A} \text{ plays Game } 6]| \leq \Pr[E]\,.$$

We now construct an adversary against the simulation-soundness of the NZIK proof system whose advantage is related to $\Pr[E]$. Consider the probabilistic, polynomial-time adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ where $\mathcal{B}_1(1^k, \sigma)$ acts as follows:

$\mathcal{B}_1(1^k, \sigma)$:
$(mpk_1, msk_1) \xleftarrow{\$} \mathtt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$
$r_1^*,\, r_2,\, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \mathtt{Encrypt}(m_1, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_1, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_0, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2,$
$\quad C_3^*, mpk_3)$
Output $(x^*, state)$

$\mathcal{B}_2(\pi^*, state)$:
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$
Output $(\emptyset, \emptyset)$

For all oracle queries except decryption oracle queries, $\mathcal{B}$ responds to $\mathcal{A}$ correctly using its knowledge of msk'. If $\mathcal{A}$ queries the decryption oracle with a ciphertext (C_1, C_2, C_3, π) then $\mathcal{B}$ computes $x \leftarrow (C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3)$ and checks that $V(x, \pi, \sigma) = 1$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ checks that $\mathcal{D}(C_2, msk_2) = \mathcal{D}(C_3, msk_3)$. If not, $\mathcal{B}$ outputs (x, π) as a false proof and halts. Otherwise, $\mathcal{B}$ returns $\mathcal{D}(C_2, msk_2)$ to $\mathcal{A}$.

It is clear to see that if E occurs, then $\mathcal{B}$ will output a false proof and so break the simulation-soundness of the NIZK proof system. In other words, $\Pr[E] \leq Adv_{\mathcal{B}}^{\mathtt{NIZK\text{-}SS}}(k)$. This completes the proof. □

Game 7 – The passive security of the encryption scheme (part 2)

This game hop is similar to the hop between Game 4 and Game 5. Let Game 7 be identical to Game 6 except that when we construct the challenge ciphertext, the second instance of the public key encryption algorithm $\mathcal{E}$ is applied to m_1 rather than m_0. In other words,

Game $7(k)$:
$(mpk_1, msk_1) \xleftarrow{\$} \mathtt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$(\sigma, \kappa) \xleftarrow{\$} S_1(1^k)$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$
$r_1^*,\, r_2^*,\, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \mathtt{Encrypt}(m_1, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_1, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_1, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2,$
$\quad C_3^*, mpk_3)$
$\pi^* \xleftarrow{\$} S_2(x^*, \kappa)$
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$

Lemma 7.

$$|\Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 6] - \Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 7]| \leq Adv_{\mathcal{B}}^{\mathit{PKE\text{-}CPA}}(k)$$

where $\mathcal{B}$ is an algorithm that runs in approximately the same time as $\mathcal{A}$.

This lemma is proven in exactly the same way as Lemma 5.

Game 8 – Simulating the challenge ciphertext (part 2)

We now start to restore the certain elements in the game. We begin by removing the simulated proof. Let Game 8 be identical to Game 7 except that we use the real proof algorithm to create the challenge ciphertext. In other words,

Game 8(k):
$(mpk_1, msk_1) \xleftarrow{\$} \texttt{Setup}(1^k)$
$(mpk_2, msk_2) \xleftarrow{\$} \mathcal{G}(1^k)$
$(mpk_3, msk_3) \xleftarrow{\$} \mathcal{G}(1^k)$
$\sigma \xleftarrow{\$} \{0,1\}^{f(k)}$
$mpk' \leftarrow (mpk_1, mpk_2, mpk_3, \sigma)$
$msk' \leftarrow msk_1$
$(m_0, m_1, \mathsf{ID}^*, state) \xleftarrow{\$} \mathcal{A}_1(1^k, mpk')$
$r_1^*, r_2^*, r_3^* \xleftarrow{\$} \{0,1\}^\infty$
$C_1^* \xleftarrow{\$} \texttt{Encrypt}(m_1, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1; r_1^*)$
$C_2^* \xleftarrow{\$} \mathcal{E}(m_1, mpk_2; r_2^*)$
$C_3^* \xleftarrow{\$} \mathcal{E}(m_1, mpk_3; r_3^*)$
$x^* \leftarrow (C_1^*, pk_{\mathsf{ID}^*}, \mathsf{ID}^*, mpk_1, C_2^*, mpk_2, C_3^*, mpk_3)$
$w^* \leftarrow (m_b, r_1^*, r_2^*, r_3^*)$
$\pi^* \xleftarrow{\$} P(x^*, w^*, \sigma)$
$C^* \leftarrow (C_1^*, C_2^*, C_3^*, \pi^*)$
$b' \xleftarrow{\$} \mathcal{A}_2(C^*, state)$

Lemma 8.

$$|\Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 7] - \Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 8]| \leq Adv_{\mathcal{B}}^{\texttt{NIZK-ZK}}(k)$$

where $\mathcal{B}$ is an algorithm that runs in approximately the same time as $\mathcal{A}$.

This lemma is proven in exactly the same way as Lemma 3.

Game 9 – Simulating decryption (part 3)

Lastly, we restore the decryption algorithm to its original state. Let Game 9 be identical to Game 8 except that the decryption algorithm now works properly:

Decrypt$'(C, sk, mpk)$:
$x \leftarrow (C_1, pk, \mathsf{ID}, mpk_1, C_2, mpk_2, C_3, mpk_3)$
If $V(x, \pi, \sigma) \neq 1$ then output $\perp$
Find the unique message m such that $C = \texttt{Encrypt}(m, pk_{\mathsf{ID}}, \mathsf{ID}, mpk)$
Output m

Lemma 9.

$$|\Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 8] - \Pr[\mathcal{A} \textit{ outputs } 1 | \mathcal{A} \textit{ plays Game } 9]| \leq Adv^{\texttt{NIZK-Sim}}(k)$$

This lemma is proven in exactly the same way as Lemma 2. However, Game 9 is identical to $Expt_{\mathcal{A}}^{\texttt{CL-CCA-X}}(1, k)$. Combining all of the results of the lemmas give us that

$$\begin{aligned}
Adv_{\mathcal{A}}^{\texttt{CL-CCA-X}}(k) &= |\Pr[Expt_{\mathcal{A}}^{\texttt{CL-CCA-X}}(0,k)=1] - \Pr[Expt_{\mathcal{A}}^{\texttt{CL-CCA-X}}(1,k)=1]| \\
&= |\Pr[\mathcal{A} \text{ outputs } 1|\mathcal{A} \text{ plays Game } 1] \\
&\quad - \Pr[\mathcal{A} \text{ outputs } 1|\mathcal{A} \text{ plays Game } 9]| \\
&\leq 2Adv^{\texttt{NIZK-Sim}}(k) + 2Adv_{\mathcal{B}}^{\texttt{NIZK-ZK}}(k) + 2Adv_{\mathcal{B}}^{\texttt{PKE-CPA}}(k) \\
&\quad + Adv_{\mathcal{B}}^{\texttt{NIZK-SS}}(k) + Adv_{\mathcal{B}}^{\texttt{CL-CPA-X}}(k)\,.
\end{aligned}$$

Hence, $Adv_{\mathcal{A}}^{\texttt{CL-CCA-X}}(k)$ is negligible as a function of the security parameter, and Π' is Strong Type I and Strong Type II secure. □

B Proof of Theorem 2 and Theorem 3

Proof. (of Theorem 2). The proof proceeds by a sequence of games. All games involve a Type I attacker $\mathcal{A}_I$ who attempts to guess a hidden bit d for which she eventually outputs a guess d'. For all i, we call S_i the event that $\mathcal{A}_i$ is successful (i.e. that $d' = d$) in Game i and we denote by $Adv_i = |\Pr[S_i] - 1/2|$ the advantage of $\mathcal{A}_I$.

We use the following game hopping technique suggested by Dent [13]. Suppose Game i is such a game where $\mathcal{A}_I$ wins with probability S_i. Consider an event E that may occur during $\mathcal{A}_I$'s execution such that:

- E is detectable by the simulator;
- E is independent of S_i;
- Game i and Game $i+1$ are identical unless E occurs, in which case the game halts and outputs random bit.

Then we have $Adv_{i+1} = |\Pr[S_{i+1}] - \frac{1}{2}| = \Pr[\neg E]|\Pr[S_{i+1}] - \frac{1}{2}| = \Pr[\neg E] \cdot Adv_i$.

Game 1: In this game, $\mathcal{A}_I$ is interacting with the actual attack environment. Namely, $\mathcal{B}$ generates the master key, the public parameters and the initial user's public keys and secret values following the specification of the scheme. We also assume that the environment $\mathcal{B}$ can answer decryption queries without knowing the matching secret values for changed public keys. In this real attack, let $\mathsf{PPK} = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_{q_{ppk}}\}$ denote the inputs of partial private key queries and $\mathsf{PK} = \{\mathsf{ID}'_1, \ldots, \mathsf{ID}'_{q_{pk}}\}$ be the set of identities queried for private key extraction. Let also $\mathsf{D} = \{w_1, \ldots, w_{q_d}\}$ be the set of strings $w_j = H(C_0, C_1, C_2, \mathsf{ID}_j, pk_j)$ involved in decryption queries. Finally, let $(\mathsf{ID}^\star, pk_{\mathsf{ID}^\star})$ denote the target identity/public key pair involved in the challenge phase and let $C^\star = (C_0^\star, C_1^\star, C_2^\star, C_3^\star)$ be the returned challenge ciphertext and $w^\star = H(C_0^\star, C_1^\star, C_2^\star, \mathsf{ID}^\star, pk^\star)$.

Game 2: Here, we change the generation of the master public key. The attack environment picks $a, b \xleftarrow{\$} \mathbb{Z}_p^*$ to set $g_1 = g^a, g_2 = g^b$. It also picks $\kappa_u, \kappa_v \in \{0, \ldots, n\}$. Let τ_u and τ_v be integers such that $\tau_u(n+1), \tau_v(n+1) < p$. We will define these values explicitly later on. The environment randomly selects $x'_u \xleftarrow{\$} \mathbb{Z}_{\tau_u}, x'_v \xleftarrow{\$} \mathbb{Z}_{\tau_v}$ and vectors $(x_{u,1}, \ldots, x_{u,n})$, $(x_{v,1}, \ldots, x_{v,n})$ of elements with $x_{u,j} \in \mathbb{Z}_{\tau_u}, x_{v,j} \in \mathbb{Z}_{\tau_v}$ for all j. It also draws $y'_u, y'_v \xleftarrow{\$} \mathbb{Z}_p$ and vectors

$(y_{u,1},\ldots,y_{u,n})$, $(y_{v,1},\ldots,y_{v,n})$ with $y_{u,j},y_{v,j} \xleftarrow{\$} \mathbb{Z}_p$ for all j. The remaining master public key elements are chosen to be

$$u' = g_2^{x'_u-\kappa_u\tau_u} g^{y'_u} \qquad u_j = g_2^{x_{u,j}} g^{y_{u,j}} \text{ for } 1 \le j \le n \tag{1}$$

$$v' = g_2^{x'_v-\kappa_v\tau_v} g^{y'_v} \qquad v_j = g_2^{x_{v,j}} g^{y_{v,j}} \text{ for } 1 \le j \le n. \tag{2}$$

This change obviously does not affect the distribution of the master public key. Hence, $\Pr[S_1] = \Pr[S_2]$ and $Adv_1 = Adv_2$.

Game 3: This game is identical to Game 2 except that the environment halts if the attacker submits a decryption query (C, ID, pk) for a well-formed ciphertext $C = (C_0, C_1, C_2, C_3)$ where w is either equal to the same value as a previously submitted ciphertext or w is equal to $w^\star$ in the post challenge phase. For such a legal decryption query, we have $C \neq C^\star$ or $(\mathsf{ID}, pk) \neq (\mathsf{ID}^\star, pk^\star)$. In either case, this implies a collision for H. Hence, we can construct an algorithm $\mathcal{A}''$ such $|\Pr[S_1] - \Pr[S_2]| \le Adv^{\mathrm{CR}}_{\mathcal{A}''}(k)$.

It may initially be thought that it is not possible to build the algorithm $\mathcal{A}''$ in such a way that it runs in polynomial-time, as the algorithm has to simulate the responses of the strong decryption oracle, which is a non-polynomial-time function. However, we note that the strong decryption oracle is only required to decrypt ciphertexts under the action of a well-formed public key (i.e. a public key $pk_{\mathsf{ID}} = (X, Y)$ where $e(g_1, X) = e(g, Y)$), therefore we can assume that $X = g^{x_{\mathsf{ID}}}$ and $Y = g_1^{x_{\mathsf{ID}}}$ for some (possibly unknown) value x_{ID}. For such a public key we may form a private key $s_1 = g_2^{\gamma x_{\mathsf{ID}}} F_u(\mathsf{ID})^t$ and $s_2 = g^t$ by choosing a value of $g_2 = g^z$ and computing $s_1 = X^{\gamma z} F_u(\mathsf{ID})^t$. Hence, the private key can always be computed and the decryption oracle will work correctly.

Game 4: We modify the environment so that it flips a coin $c_{mode} \xleftarrow{\$} \{0,1\}$ at the outset of the game. If $c_{mode} = 0$, it bets that $\mathcal{A}_I$ will choose to be challenged on an identity of replaced public key (and never extracts the matching partial private key). If $c_{mode} = 1$, it expects $\mathcal{A}_I$ to rather extract the partial private key of the target entity at some point.

At the challenge phase, if $c_{mode} = 0$ and $\mathcal{A}_I$ has not replaced the challenge public key, then $\mathcal{B}$ aborts and simulates $\mathcal{A}$'s output as $d' \xleftarrow{\$} \{0,1\}$. Similarly, $\mathcal{B}$ aborts if $c_{mode} = 1$ and $\mathcal{A}_I$ has replaced the challenge public key. Since c_{mode} is completely hidden from $\mathcal{A}_I$, this new abortion rule applies with probability 1/2. The Dent game hopping argument [13] yields $Adv_4 = \frac{1}{2} \cdot Adv_3$.

Games 5 **and** 6: We alter the generation of the challenge ciphertext. To this end, we consider values from (1)-(2) which allow defining functions

$$J_u(\mathsf{ID}) = x'_u + \sum_{j=1}^{n} i_j x_{u,j} - \kappa_u\tau_u, \qquad K_u(\mathsf{ID}) = y'_u + \sum_{j=1}^{n} i_j y_{u,j},$$

$$J_v(w) = x'_v + \sum_{j=1}^{n} w_j x_{v,j} - \kappa_v\tau_v, \qquad K_v(w) = y'_v + \sum_{j=1}^{n} w_j y_{v,j},$$

taking as input n-bit strings $\mathsf{ID} = i_1 \ldots i_n$ and $w = w_1 \ldots w_n$. For any strings $\mathsf{ID}, w \in \{0,1\}^n$, $F_u(\mathsf{ID}) = u' \cdot \prod_{j=1}^n u_j^{i_j} = g_2^{J_u(\mathsf{ID})} \cdot g^{K_u(\mathsf{ID})}$ and $F_v(w) = v' \cdot$

$\prod_{j=1}^n v_j^{w_j} = g_2^{J_v(w)} \cdot g^{K_v(w)}$. Game 5 is the same as Game 4 except that, after $\mathcal{A}_I$ outputs her guess d' for d, the environment $\mathcal{B}$ checks whether $J_u(\mathsf{ID}^\star) = J_v(w^\star) = 0 \bmod p$. If $J_u(\mathsf{ID}^\star) \neq 0 \bmod p$ or $J_v(w^\star) \neq 0 \bmod p$, $\mathcal{B}$ aborts and simulates $\mathcal{A}_I$'s output using a random $d' \xleftarrow{\$} \{0,1\}$. Since values $(x'_u, x_{u,1}, \ldots, x_{u,n})$ and $(x'_v, x_{v,1}, \ldots, x_{v,n})$ are information theoretically hidden from $\mathcal{A}_I$, it can only produce $\mathsf{ID}^\star$ so that $J_u(\mathsf{ID}^\star) = 0 \bmod p$ by chance. Therefore

$$\begin{aligned}\Pr[J_u(\mathsf{ID}^\star) = 0 \bmod p] &= \Pr[J_u(\mathsf{ID}^\star) = 0 \bmod p | J_u(\mathsf{ID}^\star) = 0 \bmod \tau_u] \\ &\quad \cdot \Pr[J_u(\mathsf{ID}^\star) = 0 \bmod \tau_u] \\ &= \frac{1}{\tau_u(n+1)}\end{aligned}$$

and we similarly have $\Pr[J_v(w^\star) = 0 \bmod p] = \frac{1}{\tau_v(n+1)}$ since the event $J_v(w^\star) = 0 \bmod p$ is easily seen to occur by pure chance. Hence, the game hopping argument of [13] yields $Adv_5 = Adv_4/\tau_u\tau_v(n+1)^2$.

In Game 6, we actually modify the way the challenge ciphertext is constructed. The environment $\mathcal{B}$ introduces a new variable $c \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $C_1^\star = g^c$. Let $pk_{\mathsf{ID}^\star} = (X^\star, Y^\star)$ be entity $\mathsf{ID}^\star$'s public key at the challenge phase. The environment flips a coin $d^\star \xleftarrow{\$} \{0,1\}$ and computes

$$\begin{aligned}C_0^\star &= m_{d^\star} \cdot e(Y^\star, g_2)^c \\ C_2^\star &= C_1^{\star K_u(\mathsf{ID}^\star)} = (g^c)^{K_u(\mathsf{ID}^\star)} \\ C_3^\star &= C_1^{\star K_v(w^\star)} = (g^c)^{K_v(w^\star)}\end{aligned}$$

where $w^\star = H(C_0^\star, C_1^\star, C_2^\star, \mathsf{ID}^\star, pk_{\mathsf{ID}^\star})$. The returned ciphertext $(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$ has the correct distribution since $J_u(\mathsf{ID}^\star) = J_v(w^\star) = 0 \bmod p$. We clearly have $Adv_6 = Adv_5$.

In Games 7 and 8, we modify the treatment of private key, partial private key and decryption queries.

Game 7: We change Game 6 so that, after $\mathcal{A}_I$ outputs her guess d', the environment $\mathcal{B}$ checks if one of the following events occurs:

- $c_{mode} = 0$ and $J_u(\mathsf{ID}_i) = 0 \bmod \tau_u$ for some $\mathsf{ID}_i \in \mathsf{PPK}$ with $i \in \{1, \ldots, q_{ppk}\}$.
- $J_u(\mathsf{ID}_j) = 0 \bmod \tau_u$ for some $\mathsf{ID}_j \in \mathsf{PK}$ where $j \in \{1, \ldots, q_{pk}\}$.
- $J_v(w_l) = 0 \bmod \tau_v$ for some $w_l \in \mathsf{D}$ where $l \in \{1, \ldots, q_d\}$.

Let E be the event that any of these conditions hold. It would be nice, at this point, if we could apply the Dent game hopping lemma. It is easy to see that E is recognisable, but we cannot be sure that E is independent of S_6. It might be the case that there exists two sequences of oracle queries for which $Pr[E]$ is significantly different for each sequence and that $\mathcal{A}$ can choose to use one sequence in a manner that somehow depends upon the challenge message m_d.

We avoid this problem by using the "re-normalisation" technique of Waters [31]. We derive a non-negligible lower bound for the probability that $\neg E$ occurs for any set of oracle queries, estimate the probability that E occurs for the particular set of oracle queries that occurred during the execution of $\mathcal{A}$, and add "artificial aborts" to make sure that $\mathcal{A}$ aborts with exactly the probability given by the lower bound.

We begin by deriving the theoretical lower bound. For simplicity, we only consider the case where $c_{mode} = 1$. The case where $c_{mode} = 0$ is similar.

$$\begin{aligned}
\Pr[\neg E] = \Pr[&\bigwedge_{\mathsf{ID} \in \mathsf{PK}} J_u(\mathsf{ID}) \neq 0 \bmod \tau_u \bigwedge_{w \in \mathsf{D}} J_v(w) \neq 0 \bmod \tau_v \\
&\qquad\qquad \mid J_u(\mathsf{ID}^\star) = 0 \bmod \tau_u \wedge J_v(w^\star) = 0 \bmod \tau_v] \\
= \Pr[&\bigwedge_{\mathsf{ID} \in \mathsf{PK}} J_u(\mathsf{ID}) \neq 0 \bmod \tau_u \mid J_u(\mathsf{ID}^\star) = 0 \bmod \tau_u] \\
&\qquad\qquad \cdot \Pr[\bigwedge_{w \in \mathsf{D}} J_v(w) \neq 0 \bmod \tau_v \mid J_v(w^\star) = 0 \bmod \tau_v]
\end{aligned}$$

We may handle each of these two terms independently.

$$\begin{aligned}
\Pr[&\bigwedge_{\mathsf{ID} \in \mathsf{PK}} J_u(\mathsf{ID}) \neq 0 \bmod \tau_u \mid J_u(\mathsf{ID}^\star) = 0 \bmod \tau_u] \\
&= 1 - \Pr[\bigvee_{\mathsf{ID} \in \mathsf{PK}} J_u(\mathsf{ID}) = 0 \bmod \tau_u \mid J_u(\mathsf{ID}^\star) = 0 \bmod \tau_u] \\
&\geq 1 - \sum_{\mathsf{ID} \in \mathsf{PK}} \Pr[J_u(\mathsf{ID}) = 0 \bmod \tau_u \mid J_u(\mathsf{ID}^\star) = 0 \bmod \tau_u] \\
&= 1 - \frac{q_{pk}}{\tau_u}
\end{aligned}$$

The other term is handled similarly. We therefore have that

$$Pr[\neg E] \geq \begin{cases} (1 - \frac{q_d}{\tau_v})(1 - \frac{q_{pk}}{\tau_u}) & \text{if } c_{mode} = 1 \\ (1 - \frac{q_d}{\tau_v})(1 - \frac{q_{ppk} + q_{pk}}{\tau_u}) & \text{if } c_{mode} = 0 \end{cases}$$

If we set $\tau_v = 2q_d$, $\tau_u = 2q_{pk}$ if $c_{mode} = 1$ and $\tau_u = 2q_{ppk} + 2q_{pk}$ if $c_{mode} = 0$, then we have $Pr[\neg E] \geq 1/4$.

As we've mentioned, this is only a theoretical lower bound for the abort probability. We wish to arrange it so that the abort probability is exactly $1/4$. To this end, we estimate the probability that the sequence of oracle queries that $\mathcal{A}$ has made will cause an abort by repeatedly picking values for x'_u, $x_{u,j}$, x'_v and $x_{v,j}$ and testing to see whether these values will cause an abort for the sequence of oracle queries that $\mathcal{A}$ made. Note that this does not involve re-running the attacker $\mathcal{A}$, but instead merely checking to see whether the simulator aborts with the set of oracle queries that $\mathcal{A}$ made during its first execution. Note also that we do not constrain the values of x'_u, $x_{u,j}$, x'_v and $x_{v,j}$ to "fit" the public key value — we may assume that the y values are chosen so that the public key elements are as in the original execution of $\mathcal{A}$. Let η' be the probability that we do not have to abort for the given sequence of queries made by $\mathcal{A}$, i.e. $\eta' = Pr[\neg E]$. Let η'' be the probability estimate for η' given by the repeated sampling of the x values. The Chernoff bound implies that for $\epsilon \geq 0$, $\delta \geq 0$ and $O(\epsilon^{-2} \ln \delta^{-1})$ samples, we have that $Pr[|\eta' - \eta''| \geq \epsilon] \leq \delta$. We have already shown that $\eta' \geq 1/4$. If $\mathcal{A}$'s execution did not abort, we force an artificial abort with probability $(\eta'' - 1/4)/\eta''$ (whenever $\eta'' \geq 1/4$). In such a situation, the environment assumes that $\mathcal{A}$ output a random value d'. The probability that an abort occurs is now given by:

$$
\begin{aligned}
Pr[\textsc{Abort}] = Pr[\textsc{Natural Abort}] &+ Pr[\textsc{Artificial Abort}] \\
&= (1 - \eta') + \frac{\eta'' - 1/4}{\eta''}\eta' \\
&= 1 - \eta' + (\eta'' - 1/4)\frac{\eta'}{\eta''} \\
&\leq 1 - \eta' + (\eta'' - 1/4)\frac{\eta'}{\eta' - \epsilon} \\
&= 1 - \eta' + (\eta'' - 1/4)(1 + \frac{\epsilon}{\eta' - \epsilon}) \\
&\leq 1 - \eta' + (\eta'' - 1/4)(1 + \frac{4\epsilon}{1 - 4\epsilon}) \qquad \text{as } \eta' \geq 1/4 \\
&\leq 1 - \eta' + (\eta'' - 1/4)(1 + 5\epsilon) \qquad \text{as } \tfrac{1}{1-4\epsilon} \leq 5/4 \text{ for sufficiently small } \epsilon \\
&\leq 1 - \eta' + (\eta' + \epsilon - 1/4)(1 + 5\epsilon) \\
&\leq 3/4 + 6\epsilon + 5\epsilon^2
\end{aligned}
$$

Hence, an abort does not occur with probability at least $1/4 - O(\epsilon)$ providing $|\eta' - \eta''| \leq \epsilon$ (which itself occurs with probability $1 - \delta$). We first say that an error occurs if $|\eta' - \eta''| > \epsilon$ (which adds a constant δ term) and then apply Dent's game hopping lemma. We conclude that $Adv_7 \geq (Adv_6 - \delta)(1/4 - O(\epsilon))$. For sufficiently small ϵ, we may conclude that $Adv_7 \geq (Adv_6 - \delta)/8$.

Game 8**:** We effectively change the treatment of $\mathcal{A}_I$'s queries. Let $A = g^a$ be a random element such that a is unknown to $\mathcal{B}$. We first change the generation of the master public key. Depending on c_{mode}, g_1 is generated in different ways.

- If $c_{mode} = 0$, $\mathcal{B}$ sets $g_1 = A$ (and does not know the master secret a).
- If $c_{mode} = 1$, it sets $g_1 = g^\gamma$ for a randomly chosen $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$ which is kept for later use.

Queries: once started, $\mathcal{A}_I$ issues queries whose treatment may depend on c_{mode}.

- **Request public key** queries for an identity ID:
  - If $c_{mode} = 0$, $\mathcal{B}$ randomly picks $x_{\mathsf{ID}} \xleftarrow{\$} \mathbb{Z}_p^*$, returns $pk_{\mathsf{ID}} \leftarrow (g^{x_{\mathsf{ID}}}, g_1^{x_{\mathsf{ID}}})$.
  - If $c_{mode} = 1$, $\mathcal{B}$ picks $x_{\mathsf{ID}} \xleftarrow{\$} \mathbb{Z}_p^*$ and returns $pk_{\mathsf{ID}} \leftarrow (A^{x_{\mathsf{ID}}}, A^{\gamma x_{\mathsf{ID}}})$.
- **Replace public key** queries for an input $(\mathsf{ID}, (\tilde{X}, \tilde{Y}))$: $\mathcal{B}$ ensures that $(\tilde{X}, \tilde{Y})$ has the correct shape and performs the replacement.
- **Extract partial private key** queries on an identity ID:

- if $c_{mode} = 0$, $\mathcal{B}$ aborts if $J_u(\mathsf{ID}) = 0 \bmod \tau_u$ as in the previous game. Otherwise (observe that $J_u(\mathsf{ID}) \neq 0 \bmod \tau_u$ implies $J_u(\mathsf{ID}) \neq 0 \bmod p$), it draws $r \xleftarrow{\$} \mathbb{Z}_p^*$ and returns $d_A = (d_1, d_2)$ where

$$d_1 \leftarrow F_u(\mathsf{ID})^r \cdot g_1^{-\frac{K_u(\mathsf{ID})}{J_u(\mathsf{ID})}} = g_2^a \cdot F_u(\mathsf{ID})^{\tilde{r}} \qquad d_2 \leftarrow g^r \cdot g_1^{-\frac{1}{J_u(\mathsf{ID})}} = g^{\tilde{r}}$$

  where $\tilde{r} = r - \frac{a}{J_u(\mathsf{ID})}$.
- if $c_{mode} = 1$, $\mathcal{B}$ uses the master key $msk = \gamma$ to compute partial private keys following the specification of the scheme.

– **Extract private key** queries on an input ID: as previously, $\mathcal{B}$ aborts if $J_u(\mathsf{ID}) \neq 0 \bmod \tau_u$. If not, we necessarily have $J_u(\mathsf{ID}) \neq 0 \bmod p$. Let $pk_{\mathsf{ID}} = (X, Y)$ be the (unreplaced) public key for ID: the environment draws $t \xleftarrow{\$} \mathbb{Z}_p^*$ and computes

$$\begin{aligned} sk_{\mathsf{ID}} = (s_1, s_2) &= \left(F_u(\mathsf{ID})^t \cdot Y^{-\frac{K_u(\mathsf{ID})}{J_u(\mathsf{ID})}}, g^t \cdot Y^{-1/J_u(\mathsf{ID})}\right) \\ &= \begin{cases} \left(g_2^{ax_{\mathsf{ID}}} \cdot F_u(\mathsf{ID})^{\tilde{t}}, g^{\tilde{t}}\right) & \text{with } \tilde{t} = t - \frac{ax_{\mathsf{ID}}}{J_u(\mathsf{ID})} \quad \text{if } c_{mode} = 0 \\ \left(g_2^{a\gamma x_{\mathsf{ID}}} \cdot F_u(\mathsf{ID})^{\tilde{t}}, g^{\tilde{t}}\right) & \text{with } \tilde{t} = t - \frac{a\gamma x_{\mathsf{ID}}}{J_u(\mathsf{ID})} \quad \text{if } c_{mode} = 1 \end{cases} \end{aligned}$$

  (Recall that in the case $c_{mode} = 0$ the secret value is x_{ID} and the implicitly defined master key value is a. In the case $c_{mode} = 1$ the implicitly defined secret value is ax_{ID} and γ is the master key.)

– **Decryption** queries for a valid ciphertext $C = (C_0, C_1, C_2, C_3)$ encrypted for an entity ID using $pk_{\mathsf{ID}} = (X, Y)$ (which may have been replaced by the attacker): let $w = H(C_0, C_1, C_2, \mathsf{ID}, pk)$. As in the previous game, $\mathcal{B}$ aborts and chooses a random $d' \xleftarrow{\$} \{0,1\}$ if $J_v(w) = 0 \bmod \tau_v$. Otherwise, $J_v(w) \neq 0 \bmod p$ and we have $C_3 = (g_2^{J_v(w)} g^{K_v(w)})^s$ and $C_1 = g^s$ for some $s \in \mathbb{Z}_p^*$. Hence, $\mathcal{B}$ is able to extract

$$g_2^s = \left(C_3 / C_1^{K_v(w)}\right)^{1/J_v(w)}$$

  which allows $\mathcal{B}$ to compute $e(Y, g_2)^s$ and $m = C_0 / e(Y, g_2)^s$ regardless of whether (X, Y) is the original public key or not.

We observe that the altered generation of the master key does not prevent $\mathcal{B}$ to answer $\mathcal{A}_I$'s queries as in Game 7. This implies $Adv_8 = Adv_7$.

Game 9: We again alter the generation of the challenge ciphertext. For the variables $b, c \xleftarrow{\$} \mathbb{Z}_p^*$ respectively introduced in Games 2 and 6, let $C_1^\star = g^c$ and $T = A^{bc}$.

- If $c_{mode} = 0$, let $pk_{\mathsf{ID}^\star} = (X^\star, Y^\star)$ be entity $\mathsf{ID}^\star$'s current public key. For a binary coin $d^\star \xleftarrow{\$} \{0,1\}$, $\mathcal{B}$ computes

$$C_0^\star = m_{d^\star} \cdot e(X^\star, T) \tag{3}$$

which equals $C_0^\star = m_{d^\star} \cdot e(g^{x^\star}, T) = m_{d^\star} \cdot e(g^{ax^\star}, g^{bc}) = m_{d^\star} \cdot e(Y^\star, g_2)^c$. Then, $\mathcal{B}$ computes $w^\star = H(C_0^\star, C_1^\star, C_2^\star, \mathsf{ID}^\star, pk_{\mathsf{ID}^\star})$. If $J_v(w^\star) \neq 0 \bmod p$, $\mathcal{B}$ aborts as previously. Otherwise, it defines $C_2^\star = (g^c)^{K_u(\mathsf{ID}^\star)}$ and $C_3^\star = (g^c)^{K_v(w^\star)}$ and returns $(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$.

- If $c_{mode} = 1$, $\mathcal{B}$ retrieves the value $x_{\mathsf{ID}^\star}$ s.t. $pk_{\mathsf{ID}^\star} = (A^{x_{\mathsf{ID}^\star}}, A^{\gamma x_{\mathsf{ID}^\star}})$, flips a coin $d^\star \xleftarrow{\$} \{0,1\}$, computes

$$C_0^\star = m_{d^\star} \cdot e(g, T)^{\gamma x_{\mathsf{ID}^\star}}, \tag{4}$$

computes $w^\star = H(C_0^\star, C_1^\star, C_2^\star, \mathsf{ID}^\star, pk_{\mathsf{ID}^\star})$. As previously, $\mathcal{B}$ aborts if $J_v(w^\star) \neq 0 \bmod p$. Otherwise, it sets $C_2^\star = (g^c)^{K_u(\mathsf{ID}^\star)}$ and $C_3^\star = (g^c)^{K_v(w^\star)}$ and returns $(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$.

As long as $J_v(w) = 0 \bmod p$, these changes do not affect the distribution of the challenge ciphertext and $Adv_9 = Adv_8$.

Game 10: We again alter the challenge phase. This time, the environment "forgets" the values b, c and simply retains $g_2 = g^b$ and $C_1^\star = g^c$. The challenge ciphertext is constructed following (3) and (4) as in Game 9 but using a randomly chosen $T \xleftarrow{\$} \mathbb{G}$ this time. In Game 10, the whole simulation only depends on the values g^a, g^b, g^c and the simulator does not use a, b, c at all. The transition between Game 9 and Game 10 is clearly based on indistinguishability: both games are equal unless there exists a PPT algorithm $\mathcal{A}'$ that distinguishes $T = g^{abc}$ from random. Therefore, we have $|\Pr[S_9] - \Pr[S_{10}]| \leq Adv_{\mathcal{A}'}^{\text{3-DDH}}(k)$. Besides $C_0^\star$ now perfectly hides $m_{d^\star}$ and is completely independent from $d^\star$. Therefore $\Pr[S_{10}] = 1/2$.

This completes the game hopping. We now combine the various inequalities arising in our game hopping steps. We have

$$Adv_7 = Adv_8 = Adv_9 \leq Adv_{\mathcal{A}'}^{\text{3-DDH}}(k)$$

and

$$Adv_5 = Adv_6 \leq 8 \cdot Adv_7 + \delta.$$

Since $Adv_5 = Adv_4/(\tau_u \tau_v (n+1)^2)$, $\tau_u \leq 2(q_{ppk} + q_{pk})$ and $\tau_v = 2q_d$, we get

$$Adv_4 < 8(q_{ppk} + q_{pk}) q_d (n+1)^2 \cdot (8 \cdot Adv_{\mathcal{A}'}^{\text{3-DDH}}(k) + \delta).$$

We also have $Adv_3 = 2 \cdot Adv_4$ and

$$\begin{aligned} Adv_1 = Adv_2 &= |\Pr[S_2] - \frac{1}{2}| \leq |\Pr[S_2] - \Pr[S_3]| + |\Pr[S_3] - \frac{1}{2}| \\ &= Adv_{\mathcal{A}''}^{\text{CR}}(k) + Adv_3. \end{aligned}$$

Combining the above, we finally obtain

$$Adv_1 < 8(q_{ppk} + q_{pk}) q_d (n+1)^2 \cdot (8 \cdot Adv_{\mathcal{A}'}^{\text{3-DDH}}(k) + \delta) + Adv_{\mathcal{A}''}^{\text{CR}}(k)$$

which completes the proof. □

Proof. (of Theorem 3). The proof is very similar to that of Theorem 2. The differences are in Games 8, 9 and 10. Recall that the adversary $\mathcal{A}_{II}$ never makes partial private key queries and receives the master key $msk := \gamma$ at the beginning of the game. In Game 8, the environment proceeds as in the case $c_{mode} = 1$ in the proof of Theorem 2 and hands $msk = \gamma$ to $\mathcal{A}_{II}$. All queries are handled as in the cases $c_{mode} = 1$ and the challenge ciphertext is computed following equation (4) in Game 9. □