Certified Encryption Revisited

P. Farshim[1] and B. Warinschi[2]

[1] Departamento de Informática, Universidade do Minho,
Campus de Gualtar, 4710-057 Braga, Portugal.
farshim@di.uminho.pt
[2] Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road,
Bristol BS8 1UB, United Kingdom.
bogdan@cs.bris.ac.uk

Abstract. The notion of certified encryption had recently been suggested as a suitable setting for analyzing the security of encryption against adversaries that tamper with the key-registration process. The flexible syntax afforded by certified encryption suggests that identity-based and certificateless encryption schemes can be analyzed using the models for certified encryption. In this paper we explore the relationships between security models for these two primitives and that for certified encryption. We obtain the following results.

We show that an identity-based encryption scheme is secure if and only if it is secure when viewed as a certified encryption scheme. This result holds under the (unavoidable) restriction that registration occurs over private channels. In the case of certificateless encryption we observe that a similar result cannot hold. The reason is that existent models explicitly account for attacks against the non-monolithic structure of the secret keys whereas certified encryption models treat secret keys as whole entities. We propose an extension for certified encryption where the adversary is allowed to partially modify the secret keys of honest parties. The extension that we propose is very general and may lead to unsatisfiable notions. Nevertheless, we exhibit one instantiation for which we can prove the desired result: a certificateless encryption is secure if and only if its associated certified encryption scheme is secure.

As part of our analysis, and a result of separate interest we confirm the folklore belief that for both IBE and CLE, security in the single-user setting (as captured by existent models) is equivalent to security in the multi-user setting.

Keywords. Identity-Based Encryption, Certificateless Encryption, Certified Encryption, Security Models, Corrupt Decryption.

1 Introduction

Background. Research on public-key encryption has tacitly assumed that important preconditions for widespread use of the primitive can somehow be achieved. These conditions include the existence of trusted third parties that register keys, the existence of secure public directories and, of course, social acceptance. The typical envisioned solution is the use of digital certificates which can be somehow made transparent to the user. An immediate benefit of this separation of concerns is that research can concentrate on the more important/technical/difficult aspects of the primitive without the burden of explicitly considering the use of certificates. The downside is that attacks on the primitive that take advantage somehow by the steps of the protocol used to register keys are not directly captured. In turn, this may lead to insecurities in protocols where encryption is used as a building block [19].

Recent proposals that aim to address the difficulties associated to certificate management are identity-based encryption (IBE) [18, 8] and certificateless encryption (CLE) [1, 9, 3, 11]. For both of the primitives, encryption can be carried out without actually verifying a certificate: unless party ID carries out a registration process with the certification authorities, he/she would not be able to decrypt ciphertexts addressed to ID. The importance of the registration protocol to the security of

the primitives is in this case clear, and indeed, attacks that use registration are captured by the models for these primitives.

A systematic way of analyzing the security of public-key encryption schemes in the presence of adversaries that can interfere with the key-registration process has recently been put forth by Boldyreva *et al.* [7]. They give a general and flexible syntax for encryption schemes and a rigorous security model that captures a variety of attacks that involve the registration process. The authors call schemes secure in their model *certified encryption* (CE) schemes. An interesting observation is that the syntax of certified encryption is so general that (with small modifications) both identity-based and certificateless encryption can be viewed as certified encryption schemes. Unfortunately this syntactic fit does not shed any light on the relation between the security models for certified encryption and those for IBE/CLE. In an ideal situation, security for an IBE/CLE scheme in the sense defined by certified encryption would be equivalent to security in some standard security model for IBE/CLE, but no formal result in this sense is known. The goal of this paper is to clarify the relations between certified encryption and standard models for IBE/CLE encryption. Our results are as follows.

IBE AS CERTIFIED ENCRYPTION. We first explore the connection between IBE and certified encryption. We show how to transform an arbitrary IBE scheme IBE into a certified encryption one $\mathsf{IBE\text{-}2\text{-}CE}(\mathsf{IBE})$, by providing an appropriate registration protocol. Such a protocol is explicitly required by the syntax of certified encryption but is only implicitly defined in IBE literature. To avoid trivial attacks, the secret key which the certification authority produces to register a user needs to be sent over private channels.

We then show that IBE satisfies indistinguishability against adaptively chosen identity and ciphertext attacks if $\mathsf{IBE\text{-}2\text{-}CE}(\mathsf{IBE})$ is a secure certified encryption scheme, if registration takes place over private channels. If registration took place over public channels, the resulting certified encryption scheme would be trivially insecure. This result shows that the security model for certified encryption captures all attacks against an IBE scheme that are also captured by existing literature.

A useful step in our analysis, and a result of independent interest, is a proof that confirms in the IBE setting single-user security and multi-user security are equivalent notions. This result had been known for standard public key encryption [4] but the theorems in that paper do not immediately extend to the IBE setting.

CLE AS CERTIFIED ENCRYPTION WITH CORRUPT DECRYPTION. Next, we concentrate on certificateless encryption. Results for this case are complicated by the multiple deeply related but subtly different security models that have been developed for this primitive [1, 9, 3, 11]. For our analysis we have settled on the stable models of Al-Riyami and Paterson [1] as formalized by Dent [11]. Our results extend via appropriate modifications to the other models as well.

Recall that to avoid the key-escrow problem specific to the IBE settings, in CLE schemes the secret key of parties is obtained from two basic components: one component is produced by the certification authority and is tied to the identity of the party, the second component is usually an actual secret key created by the user himself. Since secret keys are not monolithic, security models for certificateless encryption have to account for a variety of attacks which target part of the key. For example, the adversary is allowed to learn half of the key (say the component given by the CA) but should not learn any information as long as the rest of the secret key remains secret.

In the certified encryption model of [7] secret keys are viewed as monolithic entities which can be either corrupt or non-corrupt, but no partial corruption is possible. In consequence, not all attacks against CLE schemes are captured by the certified encryption models and a result similar to that for IBE is not immediately possible. Since the above attacks reflect real-life practical concerns, it is important to explore extensions of the CE model that takes these attacks into consideration. An important contribution of this paper is one such extension.

Since we work in a very general setting (in particular there is no required form for the secret keys) we provide the adversary with *indirect* access to the secret keys of parties via an abstract class of functions $\mathcal{F}$ that parameterizes the security experiment. The adversary may use these functions to modify the value of a secret key when used for decryptions. That is, the adversary can see decryptions under $f(sk)$ where $f \in \mathcal{F}$ and sk is a key of some honest party. The notion resembles that of security under related keys attacks for symmetric primitives [6]. We call the resulting model *certified encryption with corrupt decryption*.

Although this is not the most general extension one could imagine, for example $f(sk)$ could be provided directly to the adversary, the resulting model suffices to capture all attacks against certificateless encryption. In fact, the model is so strong that no certificateless scheme would be secure in this model for rather trivial reasons (which we discuss later in the paper). We therefore provide a slight relaxation which allows us to prove the desired implications: scheme CLE is secure in a standard CLE model if and only if the associated scheme $\mathsf{CLE\text{-}2\text{-}CE}(\mathsf{CLE})$ is a secure certified encryption scheme. Our result relies on a lemma that states that for certificateless encryption, security in single-user settings is equivalent to that in multi-user settings.

Related work. The importance of including the key-registration process in the analysis has been first pointed out by Shoup [19] and exemplified by Kaliski [15]. The registration protocol is explicitly modeled in several papers where security relies on parties possessing the secret keys of registered public keys, e.g. [14, 5].

The first efficient and provably-secure identity-based encryption scheme, was introduced by Boneh and Franklin in [8]. An alternative scheme was given by Sakai and Kasahara [17]. The security of both of these schemes relies on random oracles. Waters [20], and later Gentry [13], proposed practical IBE schemes which are secure in the standard model of computation.

Following the original work of Al-Ryiami and Paterson [1], who proposed the concept of certificateless encryption, many other constructions and several variations of the primitive have been proposed. Cheng and Comley (CC) in [9] simplified the syntax of the primitive by integrating the full secret key algorithm into the decryption procedure. They also extended the security model which allows an adversary to extract the locally computed secret values of users. Baek, Safavi-Naini and Susilo (BSS) [3] further simplified the CLE definition by letting the users generate their public keys based on their partial private keys obtained from the certification authority. This allows for CLE schemes that do not rely on bilinear maps on elliptic curves. Furthermore, a notion of security known as Denial-of-Decryption [16], can only be achieved in the BSS formulation. In this attack, an adversary replaces the public key of a user with a value which results in valid ciphertexts, but the receiver upon decryption recovers invalid or different plaintext than those which where intended to be communicated. A good survey of certificateless public-key encryption schemes and security models has recently been provided by Dent [11].

Paper Organization. In Section 2 we recall the notion of certified encryption model and motivate and explain our extension. In Section 3 we recall the relevant security notions for identity-based encryption, and prove that certified encryption models can be used to analyze IBE schemes. In Section 4 we recall the definitions and security models for certificateless encryption, motivate a relaxation for certified encryption and prove the equivalence between certified and certificateless encryption models.

2 Certified Encryption

In this section we present our extension to the notion of certified encryption of Boldyreva *et al.* [7]. We start with recalling that model, and then explain the modification that we propose.

Certified Encryption with Honest Decryption

As discussed in the introduction, this security notion for encryption explicitly takes into account the registration process and captures secrecy of plaintexts even when the adversary can tamper with the registration process.

Syntax. Formally, a certified encryption scheme is defined via a five-tuple of polynomial-time algorithms as follows.

1. $\mathsf{Setup}_{\mathsf{CE}}(1^k)$ is a probabilistic *parameter-generation* algorithm. It takes input 1^k, where k is the security parameter, and outputs some parameters I, available to all parties. For the sake of readability we omit I from the input of the parties.
2. $\mathbb{G}_{\mathsf{CE}}(I)$ is a probabilistic *key-generation* algorithm. It takes input a set of parameters I, and outputs a pair $(\mathsf{SK}_{\mathsf{CA}}, \mathsf{PK}_{\mathsf{CA}})$ consisting of a secret key and a matching public key for the certification authority.
3. $(\mathbb{C}_{\mathsf{CE}}, \mathbb{U}_{\mathsf{CE}})$ is a pair of interactive probabilistic algorithms forming the (two-party) public-key *registration protocol*. Algorithm $\mathbb{C}_{\mathsf{CE}}$ takes input a secret key $\mathsf{SK}_{\mathsf{CA}}$. Algorithm $\mathbb{U}_{\mathsf{CE}}$ takes input the identity ID of a user and the public key $\mathsf{PK}_{\mathsf{CA}}$ corresponding to $\mathsf{SK}_{\mathsf{CA}}$. As result of the interaction, the output of $\mathbb{C}_{\mathsf{CE}}$ is $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert})$, where PK is a public key and cert is an issued certificate. The local output of $\mathbb{U}_{\mathsf{CE}}$ is $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert})$, where SK is a secret key that user uses to decrypt ciphertexts. We write
$$((\mathsf{ID}, \mathsf{PK}, \mathsf{cert}), (\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert})) \leftarrow (\mathbb{C}_{\mathsf{CE}}(\mathsf{SK}_{\mathsf{CA}}), \mathbb{U}_{\mathsf{CE}}(\mathsf{ID}, \mathsf{PK}_{\mathsf{CA}}))$$
for the result of this interaction. Either party can quit the execution prematurely, in which case the output of the party is set to $\perp$.
4. $\mathbb{E}_{\mathsf{CE}}(\mathsf{m}, \mathsf{ID}, \mathsf{PK}, \mathsf{cert}, \mathsf{PK}_{\mathsf{CA}})$ is a probabilistic *encryption* algorithm that takes input a message $\mathsf{m} \in \mathbb{M}_{\mathsf{CE}}(I)$, a user's identity ID, a public encryption key PK, a certificate cert, and the authority's public key $\mathsf{PK}_{\mathsf{CA}}$, and outputs a ciphertext $\mathsf{c} \in \{0,1\}^* \cup \{\perp\}$.
5. $\mathbb{D}_{\mathsf{CE}}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert}, \mathsf{PK}_{\mathsf{CA}})$ is a *deterministic* decryption algorithm which takes as input a ciphertext c, a user's identity ID, a secret decryption key SK, a certificate cert, the authority's public key $\mathsf{PK}_{\mathsf{CA}}$, and outputs $\mathsf{m} \in \mathbb{M}_{\mathsf{CE}}(I) \cup \{\perp\}$.

Correctness. A certified encryption scheme is *correct* if the decryption algorithm is the inverse of the encryption algorithm. I.e. for any $\mathsf{ID} \in \{0,1\}^*$ and any $\mathsf{m} \in \mathbb{M}_{\mathsf{CE}}(I)$:

$$\begin{aligned}\Pr[\mathsf{m}' = \mathsf{m} \mid\ & I \leftarrow \mathsf{Setup}_{\mathsf{CE}}(1^k); (\mathsf{SK}_{\mathsf{CA}}, \mathsf{PK}_{\mathsf{CA}}) \leftarrow \mathbb{G}_{\mathsf{CE}}(I); \\ & ((\mathsf{ID}, \mathsf{PK}, \mathsf{cert}), (\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert})) \leftarrow (\mathbb{C}_{\mathsf{CE}}(\mathsf{SK}_{\mathsf{CA}}), \mathbb{U}_{\mathsf{CE}}(\mathsf{ID}, \mathsf{PK}_{\mathsf{CA}})); \\ & \mathsf{c} \leftarrow \mathbb{E}_{\mathsf{CE}}(\mathsf{m}, \mathsf{ID}, \mathsf{PK}, \mathsf{cert}, \mathsf{PK}_{\mathsf{CA}}); \mathsf{m}' \leftarrow \mathbb{D}_{\mathsf{CE}}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert}, \mathsf{PK}_{\mathsf{CA}})] = 1\end{aligned}$$

Security. The security of a certified encryption scheme CE is defined through the experiments in Figure 1. Experiment $\mathbf{Exp}^{\mathsf{mCE\text{-}CCA\text{-}I}}_{\mathsf{CE},\mathcal{A},b}(1^k)$ models the situation where the certification authority is honest. Experiment $\mathbf{Exp}^{\mathsf{mCE\text{-}CCA\text{-}M}}_{\mathsf{CE},\mathcal{A},b}(1^k)$ models the situation where the authority is corrupt. Both experiments involve an adversary $\mathcal{A}$ (which may run in multiple stages) and is parameterized by bit b and maintain two lists $\mathsf{RegListPub}$ and $\mathsf{RegListSec}$ used to store public and secret information pertaining to users. The adversary can read the content of $\mathsf{RegListPub}$. The adversaries in the experiments have access to a set of oracles $\mathcal{O}$ that provides the adversaries with the following capabilities. Each oracle corresponds to a type of queries that the adversary can issue. These queries are the following:

– $\mathsf{Register}(\mathsf{ID}, \mathsf{L})$: When this query is issued, with parameters some identity ID and a label $\mathsf{L} \in \{\mathsf{honest}, \mathsf{corrupt}\}$ the oracle answers as follows. If $\mathsf{L} = \mathsf{honest}$ then the registration protocol is executed internally by the oracle, i.e.

$$((\mathsf{ID}', \mathsf{PK}', \mathsf{cert}'), (\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert})) \leftarrow (\mathbb{C}_{\mathsf{CE}}(\mathsf{SK}_{\mathsf{CA}}), \mathbb{U}_{\mathsf{CE}}(\mathsf{ID}, \mathsf{PK}_{\mathsf{CA}}).$$

The entry $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert})$ is stored in $\mathsf{RegListPub}$ and $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert})$ is stored in $\mathsf{RegListSec}$. If $\mathsf{L} = \mathsf{corrupt}$ then the registration protocol is executed with the adversary playing the role of the ID (i.e. running a corrupt version of $\mathbb{U}_{\mathsf{CE}}$). At the end of the execution, when $\mathbb{C}_{\mathsf{CE}}$ outputs some values $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert})$, the tuple $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert})$ in stored in $\mathsf{RegListPub}$.
– $\mathsf{Encrypt}(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}, \mathsf{PK}, \mathsf{cert})$: On input two messages m_0 and m_1 and an a tuple $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert})$, this oracle returns the ciphertext $\mathbb{E}_{\mathsf{CLE}}(\mathsf{m}_b, \mathsf{ID}, \mathsf{PK}, \mathsf{cert}, \mathsf{PK}_{\mathsf{CA}})$.
– $\mathsf{Decrypt}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{cert})$: If the entry $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert})$ for some SK occurs in $\mathsf{RegListSec}$, then the oracle returns $\mathbb{D}_{\mathsf{CE}}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert}, \mathsf{PK}_{\mathsf{CA}})$. Otherwise, it returns $\perp$.

$\mathbf{Exp}^{\mathsf{mCE\text{-}CCA\text{-}I}}_{\mathsf{CE},\mathcal{A},b}(1^k)$
1. $I \leftarrow \mathsf{Setup}_{\mathsf{CE}}(1^k)$
2. $(\mathsf{SK}_{\mathsf{CA}}, \mathsf{PK}_{\mathsf{CA}}) \leftarrow \mathbb{G}_{\mathsf{CE}}(I)$
3. $b' \leftarrow \mathcal{A}_1^{\mathcal{O}}(I, \mathsf{PK}_{\mathsf{CA}}, \mathsf{st})$
4. Return b'

$\mathbf{Exp}^{\mathsf{mCE\text{-}CCA\text{-}M}}_{\mathsf{CE},\mathcal{A},b}(1^k)$
1. $I \leftarrow \mathsf{Setup}_{\mathsf{CE}}(1^k)$
2. $(\mathsf{PK}_{\mathsf{CA}}, \mathsf{st}) \leftarrow \mathcal{A}_0(I)$
3. $b' \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathsf{st})$
4. Return b'

Fig. 1. Experiments for defining security for a certified encryption scheme CE against chosen-ciphertext attacks, in the private channels model. Here st is some state information. Adversary $\mathcal{A}$ is required not to cause any of the following events. 1) Calling $\mathsf{Register}$ with $\mathsf{L} = \mathsf{corrupt}$ if CA was corrupted; 2) Calling $\mathsf{Encrypt}$ on two messages with unequal lengths, or using a tuple $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert}) \in \mathsf{RegListPub}$ if it also placed a $\mathsf{Register}(\mathsf{ID}, \mathsf{corrupt})$ query, or if CA is corrupt, with a tuple $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert}) \notin \mathsf{RegListPub}$; 3) Placing $\mathsf{Decrypt}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{cert})$ with the ciphertext c previously received from the left-right encryption oracle on a query involving $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert})$.

We say that a certified encryption scheme CE is secure against type x adversaries ($\mathsf{x} = \mathsf{I}$ is for the case when the CA is honest throughout the attack, and $\mathsf{x} = \mathsf{M}$ indicates that the CA is corrupted) with chosen-ciphertext capabilities, if the advantage of any probabilistic polynomial-time (ppt) adversary $\mathcal{A}$, defined by:

$$\mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}x}}_{\mathsf{CE},\mathcal{A}}(k) := \Pr\left[\mathbf{Exp}^{\mathsf{mCE\text{-}CCA\text{-}x}}_{\mathsf{CE},\mathcal{A},1}(1^k) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{mCE\text{-}CCA\text{-}x}}_{\mathsf{CE},\mathcal{A},0}(1^k) = 1\right]$$

is a negligible function.

Remark 1 All of the security notions regarding encryption schemes in this paper use the indistinguishability definitional paradigm. To unclutter notation we do not explicitly indicate this in our notation. Also, in the experiment above we use mCE in our notation (as opposed to CE) to indicate that our model for certified encryption schemes is multi-user (the adversary has multiple left-right queries for potentially different identities/public keys).

Remark 2 Our model is concerned specifically with schemes where registration takes place over secure channels: the transcript of the registration protocol executed by an honest party is not returned to the adversary. A model for public channels can then be immediately obtained by giving these transcripts to the adversary.

Remark 3 The original model of Boldyreva *et al.* only considered one experiment where the adversary chooses in the beginning to corrupt the CA or not. For clarity, we chose to have two separate experiments, one for each possibility. A scheme is secure in the sense of [7] if it is secure for against both type I and type M adversaries.

Certified Encryption with Corrupted Decryption

In this section we describe the extension that we propose to the certified encryption model. We are motivated by the observation that an important class of attacks is not captured by the model above. Indeed, the adversary lacks the ability to tamper the secret keys of honest parties since once an honest party is registered the adversary can only involve that party's secret key in decryptions (via the decryption oracle) but can do nothing more. Yet, scenarios of practical interest where such attacks may occur can be easily envisioned. Consider for example the setting of certificateless encryption where the secret keys of party ID has two components SK and D. The first component is generated by the user and the second one by a trusted authority. One can imagine a setting where the two parts of the key are stored separately (e.g. D a trusted card and SK stored on a computer) and the adversary may tamper with one of the two (by gaining access to only one of the two devices). Decryptions performed by the user now involve the modified key.

Next we present an extension of the certified encryption model which includes extra power for the adversary. We are trying to be as general as possible (and not concentrate for example on the case where secret keys are as in certificateless encryption). We allow the adversary to apply any function f from a set $\mathcal{F}$ to the secret keys of parties, before the keys are used for decryption. Formally, we replace the query $\mathsf{Decrypt}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{cert})$ that adversary $\mathcal{A}$ is allowed to issue in with a new query $\mathsf{MalDecrypt}(f, \mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{cert})$ with $f \in \mathcal{F}$. When this query is issued, the oracle searches the list $\mathsf{RegListSec}$ for an entry $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert}, \mathsf{SK})$. If no such entry exists the oracle returns $\perp$. Otherwise the oracle returns m computed as: $\mathsf{m} \leftarrow \mathbb{D}_{\mathsf{CE}}(\mathsf{c}, f(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert}), \mathsf{PK}_{\mathsf{CA}})$.

The resulting model is quite flexible in the abilities that the adversary can gain. However, due to the high level of abstraction it is difficult to describe which are the trivial attacks that the adversary is not allowed to perform. In this paper, and for the specific function sets $\mathcal{F}$ that we use, the following restrictions on this oracle is imposed.

- $\mathcal{A}$ is not allowed to query the $\mathsf{MalDecrypt}$ oracle on $(f, \mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{cert})$ with $f = \mathrm{Id}$ and a ciphertext c which was previously received from the left-right encryption oracle on a query involving $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert})$.
- $\mathcal{A}$ is not allowed to query the $\mathsf{MalDecrypt}$ oracle on $(f, \mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{cert})$ with an f such that $f(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{cert}) = (\mathsf{ID}, \mathsf{PK}, \mathsf{SK}', \mathsf{cert})$ with $\mathsf{SK} \neq \mathsf{SK}'$ and ciphertext c previously received from the left-right encryption oracle on a query involving $(\mathsf{ID}, \mathsf{PK}, \mathsf{cert})$.

The second condition is imposed as the CLE model does not allow replacing the public key of the challenge value with the *same* value as that used to generate the challenge ciphertext but providing a new SK.

We say that scheme CE is secure against chosen-ciphertext attacks with $\mathcal{F}$-decryptions if for any ppt adversary $\mathcal{A}$, its advantage, defined by:

$$\mathbf{Adv}^{\mathsf{mCE}\text{-}\mathcal{F}\text{-}\mathsf{CCA}\text{-}\mathsf{x}}_{\mathsf{CE},\mathcal{A}}(k) := \Pr\left[\mathbf{Exp}^{\mathsf{mCE}\text{-}\mathcal{F}\text{-}\mathsf{CCA}\text{-}\mathsf{x}}_{\mathsf{CE},\mathcal{A},1}(1^k) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{mCE}\text{-}\mathcal{F}\text{-}\mathsf{CCA}\text{-}\mathsf{x}}_{\mathsf{CE},\mathcal{A},0}(1^k) = 1\right]$$

is a negligible function.

Remark 1 Our model generalizes the one of the previous section: if $\mathcal{F}$ contains only the identity function, the extended model is not more stringent than the previous one. Similarly, if we set $\mathcal{F}$ to be the empty set we obtain CPA (chosen-plaintext) attacks.

Remark 2 The functions in $\mathcal{F}$ are fixed, and do not adaptively change. One interesting direction is to consider a stronger model where the functions have as parameter the local state of the adversary. For example, this would allow f to depend on the secret key of the CA when the CA is corrupt.

Remark 3 The extended model allows the adversary to modify the keys that honest parties use

for decryption, but does not allow it to learn these keys (or parts of these keys) adaptively. Another plausible way to strengthen the model that we propose would be to add adversarial capabilities that account for these possibilities. We do not pursue this direction in this paper since the model that we have is sufficient to capture security for both IBE and CLE schemes.

3 Identity-Based Encryption as Certified Encryption

In this section we recall the syntax and the relevant security models for identity-based encryption. Our presentation mainly follows [8]. We also introduce a multi-user security model and prove the folklore result that the single-user security is equivalent to multi-user security. Then, we show that an IBE scheme secure in the certified encryption sense is secure in a standard model for identity-based encryption.

Syntax. In the identity-based setting, parties use their unstructured identities as their own public keys, and have associated secret keys generated by a trusted third party. Formally, an identity-based encryption scheme IBE is specified by five polynomial-time algorithms as follows.

- $\mathsf{Setup}_{\mathsf{IBE}}(1^k)$: A probabilistic *setup* algorithm which takes as input the security parameter 1^k and returns the domain parameters I of cryptosystem. This parameter includes descriptions of: groups underlying the scheme, message space $\mathbb{M}_{\mathsf{IBE}}(I)$ and ciphertext space $\mathbb{C}_{\mathsf{IBE}}(I)$. We assume that I is available to all parties and do not include it explicitly in the various algorithms that define the scheme.
- $\mathbb{G}_{\mathsf{IBE}}(I)$: A probabilistic *key-generation* algorithm which takes as input the domain parameters I and returns a master secret key, master public key pair $(\mathsf{Msk}, \mathsf{Mpk})$.
- $\mathbb{X}_{\mathsf{IBE}}(\mathsf{ID}, \mathsf{Msk})$: A probabilistic *private key extraction* algorithm. It takes as input the master secret key Msk and an identity $\mathsf{ID} \in \{0,1\}^*$, and returns the associated private key D.
- $\mathbb{E}_{\mathsf{IBE}}(\mathsf{m}, \mathsf{ID}, \mathsf{Mpk})$: A probabilistic *encryption* algorithm. On input a message $\mathsf{m} \in \mathbb{M}_{\mathsf{IBE}}(I)$, an identifier ID, and the master public key Mpk, this algorithm outputs a ciphertext $\mathsf{c} \in \mathbb{C}_{\mathsf{IBE}}(I)$.
- $\mathbb{D}_{\mathsf{IBE}}(\mathsf{c}, \mathsf{ID}, \mathsf{D})$: A deterministic *decryption* algorithm. On input of a ciphertext c and a private key D this algorithm outputs a message $\mathsf{m} \in \mathbb{M}_{\mathsf{IBE}}(I)$ or a failure symbol $\perp$.

Correctness. An identity-based encryption scheme is *correct* if

$$\Pr[\mathsf{m}' = \mathsf{m} \mid I \leftarrow \mathsf{Setup}_{\mathsf{IBE}}(1^k); (\mathsf{Msk}, \mathsf{Mpk}) \leftarrow \mathbb{G}_{\mathsf{IBE}}(I); \mathsf{D} \leftarrow \mathbb{X}_{\mathsf{IBE}}(\mathsf{ID}, \mathsf{Msk}); \mathsf{c} \leftarrow \mathbb{E}_{\mathsf{IBE}}(\mathsf{m}, \mathsf{ID}, \mathsf{Mpk}); \mathsf{m}' \leftarrow \mathbb{D}_{\mathsf{IBE}}(\mathsf{c}, \mathsf{ID}, \mathsf{D})] = 1$$

for any $\mathsf{ID} \in \{0,1\}^*$ and any $\mathsf{m} \in \mathbb{M}_{\mathsf{IBE}}(I)$.

3.1 Single-user Security for IBE schemes

Security. Security of the identity-based encryption scheme IBE is defined, as usual, via indistinguishability games. In Figure 2 we describe experiments for chosen-ciphertext attacks. The adversary in those experiments have access to a set of oracles $\mathcal{O}$ that formalizes its various capabilities. Both experiments involve an adversary $\mathcal{A}$ and are parameterized by a bit $b \in \{0,1\}$ which the adversary aims to determine. The adversary has access to a key extraction oracle $\mathsf{Extract}$ and a decryption oracle $\mathsf{Decrypt}$, both parameterized by $(I, \mathsf{Mpk}, \mathsf{Msk})$. The oracle process the following queries:

- $\mathsf{Extract}(\mathsf{ID})$: This oracle on input an identity ID returns $\mathbb{X}_{\mathsf{IBE}}(\mathsf{ID}, \mathsf{Msk})$.
- $\mathsf{Decrypt}(\mathsf{c}, \mathsf{ID})$: This oracle on input a ciphertext/identity pair $(\mathsf{c}, \mathsf{ID})$ answers with $\mathbb{D}_{\mathsf{IBE}}(\mathsf{c}, \mathsf{ID}, \mathsf{D})$ where $\mathsf{D} \leftarrow \mathbb{X}_{\mathsf{IBE}}(\mathsf{ID}, \mathsf{Msk})$.

We say that scheme IBE is secure against chosen-ciphertext attacks in the single-user setting (or $\mathsf{ID\text{-}CCA}$ secure) if the advantage of any ppt adversary $\mathcal{A}$ defined by

$$\mathbf{Adv}^{\mathsf{ID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(k) := \Pr\left[\mathbf{Exp}^{\mathsf{ID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A},1}(1^k) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{ID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A},0}(1^k) = 1\right]$$

is a negligible function.

$\mathbf{Exp}^{\mathsf{ID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A},b}(1^k)$
1. $I \leftarrow \mathsf{Setup}_{\mathsf{IBE}}(1^k)$
2. $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow \mathbb{G}_{\mathsf{IBE}}(I)$
3. $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}^*, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathcal{O}}(I, \mathsf{Mpk})$
4. $\mathsf{c}^* \leftarrow \mathbb{E}_{\mathsf{IBE}}(\mathsf{m}_b, \mathsf{ID}^*, \mathsf{Mpk})$
5. $b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(\mathsf{c}^*, \mathsf{st})$
6. Return b'

$\mathbf{Exp}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A},b}(1^k)$
1. $I \leftarrow \mathsf{Setup}_{\mathsf{IBE}}(1^k)$
2. $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow \mathbb{G}_{\mathsf{IBE}}(I)$
3. $b' \leftarrow \mathcal{A}^{\mathcal{O}}(I, \mathsf{Mpk})$
4. Return b'

Fig. 2. Experiments for defining security of identity-based scheme IBE against chosen-ciphertext attacks. In both experiments, st is some state information. We require that messages m_0 and m_1 be of equal length. The model on the left if for the single-user setting. Here, we require that adversary $\mathcal{A}$ does not cause any of the following events. 1) Querying $\mathsf{Extract}$ on ID^*; 2) $\mathcal{A}_2$ querying $\mathsf{Decrypt}(\mathsf{c}^*, \mathsf{ID}^*)$. The model on the right is for the multi-user setting. Here we require that adversary $\mathcal{A}$ does not cause any of the following events. 1) Calling $\mathsf{Encrypt}$ on two messages with unequal lengths; 2) Querying $\mathsf{Extract}$ on any identity sent to the left-right encryption oracle; 3) $\mathcal{A}$ querying $\mathsf{Decrypt}$ on a pair $(\mathsf{c}, \mathsf{ID})$ with c previously received from the left-right encryption oracle on a query involving ID.

3.2 Multi-user Security for IBE Schemes

The security model for IBE schemes that we presented in above was a single-user one: the adversary targets only a single identity to attack and receives only one challenge ciphertext under this identity. It is more realistic to analyze the security of a scheme in the *multi-user* setting, where many challenge ciphertexts on many adversarially-chosen identities are available. We formalize this model via the experiments shown in Figure 2 on the right.

The experiment involves an adversary $\mathcal{A}$ and is parameterized by a bit $b \in \{0, 1\}$ which the adversary aims to determine. In addition to the oracles $\mathsf{Extract}$ and $\mathsf{Decrypt}$ (as defined for the single-user setting) the adversary also has access to a left-right encryption oracle with access to $(I, \mathsf{Mpk}, \mathsf{Msk})$. This oracle behaves as follows. On a query $\mathsf{Encrypt}(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID})$, with m_0 and m_1 bit-strings of equal length, and identity ID, the oracle returns the ciphertext $\mathbb{E}_{\mathsf{IBE}}(\mathsf{m}_b, \mathsf{ID}, \mathsf{Mpk})$.

We say that an identity-based encryption scheme IBE is secure against chosen-ciphertext attacks in the multi-user setting (or $\mathsf{mID\text{-}CCA}$ secure) if the advantage of any ppt adversary $\mathcal{A}$, defined by:

$$\mathbf{Adv}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(k) := \Pr\left[\mathbf{Exp}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A},1}(1^k) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A},0}(1^k) = 1\right]$$

is a negligible function. Although security in the multi-user setting seems a stronger requirement we prove that, as for the case of standard public-key encryption, it is equivalent to security in the single-user setting.

Lemma 1. *Identity base encryption scheme* IBE *is secure in the multi-user setting if and only if it is secure in the single-user setting.*

The proof is by a standard hybrid argument and is given in Appendix A. We show that for any ppt adversary $\mathcal{A}$ against an IBE scheme IBE in the multi-user sense, there exists a ppt adversary $\mathcal{B}$ against the scheme in the single-user setting such that:

$$\mathbf{Adv}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(k) \leq Q_{\mathsf{ID}}(k) \cdot Q_{\mathsf{E}}(k) \cdot \mathbf{Adv}^{\mathsf{ID\text{-}CCA}}_{\mathsf{IBE},\mathcal{B}}(k).$$

Here $Q_{\mathsf{ID}}(k)$ denotes the number of different identities sent to the left-right encryption oracle, and $Q_{\mathsf{E}}(k)$ denotes the maximum number of left-right encryption queries per identity.

3.3 The IBE-2-CE Transformation

In this section we explain how any identity-based encryption scheme naturally gives rise to an *associated* identity-based certified encryption scheme. Then, we prove that if the resulting scheme is a secure certified encryption scheme, then the original scheme was a secure identity-based encryption scheme.

Fix an arbitrary identity-based encryption scheme $\mathsf{IBE} = (\mathsf{Setup}_{\mathsf{IBE}}, \mathbb{G}_{\mathsf{IBE}}, \mathbb{X}_{\mathsf{IBE}}, \mathbb{E}_{\mathsf{IBE}}, \mathbb{D}_{\mathsf{IBE}})$. The associated certified encryption scheme $\mathsf{IBE\text{-}2\text{-}CE}(\mathsf{IBE}) = (\mathsf{Setup}_{\mathsf{CE}}, \mathbb{G}_{\mathsf{CE}}, (\mathbb{U}_{\mathsf{CE}}, \mathbb{C}_{\mathsf{CE}}), \mathbb{E}_{\mathsf{CE}}, \mathbb{D}_{\mathsf{CE}})$ is as follows. Algorithms $\mathsf{Setup}_{\mathsf{CE}}$, $\mathbb{G}_{\mathsf{CE}}$, $\mathbb{E}_{\mathsf{CE}}$ and $\mathbb{D}_{\mathsf{CE}}$ are identical to $\mathsf{Setup}_{\mathsf{IBE}}$, $\mathbb{G}_{\mathsf{IBE}}$, $\mathbb{E}_{\mathsf{IBE}}$ and $\mathbb{D}_{\mathsf{IBE}}$, respectively, and the registration protocol is defined as follows.

1. $\mathbb{U}_{\mathsf{CE}}(\mathsf{ID}, \mathsf{PK}_{\mathsf{CA}})$: Sends ID to the CA;
2. $\mathbb{C}_{\mathsf{CE}}(\mathsf{SK}_{\mathsf{CA}})$: Receives ID, runs $\mathsf{D} \leftarrow \mathbb{X}_{\mathsf{IBE}}(\mathsf{ID}, \mathsf{SK}_{\mathsf{CA}})$, and sends D to user ID. It outputs $(\mathsf{ID}, \epsilon, \epsilon)$ locally and terminates;
3. $\mathbb{U}_{\mathsf{CE}}(\mathsf{ID}, \mathsf{PK}_{\mathsf{CA}})$: Receives D, outputs $(\mathsf{ID}, \epsilon, \mathsf{D}, \epsilon)$ locally and terminates.

The registration process consists in the user sending his identity ID to the certification authority who extracts the secret key associated to ID and sends it to the user. The first main result of this paper is that identity-based encryption scheme can be analyzed in the models for certified encryption.

Theorem 1. *Let* IBE *be an arbitrary identity-based encryption scheme and set* $\mathsf{CE} := \mathsf{IBE\text{-}2\text{-}CE}(\mathsf{IBE})$. *Then* IBE *is secure if and only if* CE *is a secure certified-encryption scheme with CA honest.*

The proof of the theorem can be found in Appendix B where we show that for any ppt adversary $\mathcal{A}$ against IBE in the single-user setting there exists a ppt type I adversary $\mathcal{B}$ against CE such that:

$$\mathbf{Adv}^{\mathsf{ID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(k) \leq Q_{\mathsf{ID}}(k) \cdot \mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}I}}_{\mathsf{CE},\mathcal{B}}(k).$$

Here $Q_{\mathsf{ID}}(k)$ denotes the number of distinct identities queried during the ID-CCA experiment, that is the number of distinct identities sent to the $\mathsf{Extract}$ or the $\mathsf{Decrypt}$ oracle, together with ID^*. Conversely, for any ppt adversary $\mathcal{A}$ against CE there exists ppt adversary $\mathcal{B}$ against IBE in the multi-user setting such that:

$$\mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}I}}_{\mathsf{CE},\mathcal{A}}(k) \leq \mathbf{Adv}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{B}}(k).$$

We note that the security model for IBE schemes that we have presented has an extraction oracle which computes a secret key with fresh random coins for an identity which is re-submitted to this oracle. This is in line with the certified security model where a user can invoke multiple runs of the register protocol on an identity. In the setting where only a single private can be extracted, the certified model should be modified so that it only allows the users to register once.

4 Certificateless Encryption as Certified Encryption

In this section we recall the syntax and the relevant security models for certificateless encryption. The security models that we use are those of Al-Ryiami and Paterson [1], as described by Dent [11]. We also introduce a multi-user security model for the primitive, and prove the folklore theorem that single-user security is equivalent to multi-user security. Towards proving that certificateless encryption can be viewed and analyzed using the certified encryption model we introduce a slight weakening of the latter. We then prove that a certificateless scheme is secure if and only if it is secure as a certified encryption scheme.

Syntax. In a certificateless encryption scheme users' public keys consist of their identity and a user-generated public value. A recipient uses two partial secret values, corresponding to it identity and public value, to decrypt ciphertexts. Formally, a certificateless encryption scheme CLE is specified by six polynomial-time algorithms as follows.

1. $\mathsf{Setup}_{\mathsf{CLE}}(1^k)$. A probabilistic *setup* algorithm, which takes as input the security parameter 1^k and returns the descriptions of underlying groups, message space $\mathbb{M}_{\mathsf{CLE}}(I)$ and ciphertext space $\mathbb{C}_{\mathsf{CLE}}(I)$. This algorithm is executed by the key-generation center (KGC), which publishes I. We assume that I is available to all parties and do not include it explicitly in the various algorithms that define the scheme.
2. $\mathbb{G}_{\mathsf{CLE}}(I)$. A probabilistic algorithm for*KGC key-generation* which on input the domain parameters I, outputs a master secret key, master public key pair $(\mathsf{Msk}, \mathsf{Mpk})$.
3. $\mathbb{X}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{Msk})$. A probabilistic algorithm for *partial private key extraction* which takes as input an identifier string $\mathsf{ID} \in \{0,1\}^*$, the master secret key Msk, and returns a partial secret key D. This algorithm is run by the KGC, after verifying the user's identity.
4. $\mathbb{U}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{Mpk})$ A probabilistic algorithm for *user key-generation* which takes an identity and the master public key, and outputs a secret value SK and a public key PK. This algorithm is run by a user to obtain a public key and a secret value which can be used to construct a full private key. The public key is published without certification.
5. $\mathbb{S}_{\mathsf{CLE}}(\mathsf{SK}, \mathsf{D}, \mathsf{Mpk})$. A probabilistic algorithm for *full secret key extraction* which takes a secret value SK and a partial private key D as well as the master public key and returns a full secret key S.
6. $\mathbb{E}_{\mathsf{CLE}}(\mathsf{m}, \mathsf{ID}, \mathsf{PK}, \mathsf{Mpk})$. This is the probabilistic *encryption* algorithm. On input of a message $\mathsf{m} \in \mathbb{M}_{\mathsf{CLE}}(I)$, receiver's identifier ID, the receiver's public key PK, and the master public key Mpk, this algorithm outputs a ciphertext $\mathsf{c} \in \mathbb{C}_{\mathsf{CLE}}(I)$ or an error symbol $\perp$.
7. $\mathbb{D}_{\mathsf{CLE}}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{S}, \mathsf{Mpk})$. This is the deterministic decryption algorithm. On input of a ciphertext c, an identity ID, a public key PK, the receiver's full private key S, and Mpk this algorithm outputs a message m or a failure symbol $\perp$.

Correctness. A certificateless encryption scheme is called *correct* if

$$\Pr[\mathsf{m}' = \mathsf{m} | I \leftarrow \mathsf{Setup}_{\mathsf{CLE}}(1^k); (\mathsf{Msk}, \mathsf{Mpk}) \leftarrow \mathbb{G}_{\mathsf{CLE}}(I); \mathsf{D} \leftarrow \mathbb{X}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{Msk}); (\mathsf{SK}, \mathsf{PK}) \leftarrow \mathbb{U}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{Mpk}); \mathsf{S} \leftarrow \mathbb{S}_{\mathsf{CLE}}(\mathsf{SK}, \mathsf{D}, \mathsf{Mpk}); \mathsf{c} \leftarrow \mathbb{E}_{\mathsf{CLE}}(\mathsf{m}, \mathsf{ID}, \mathsf{PK}, \mathsf{Mpk}); \mathsf{m}' \leftarrow \mathbb{D}_{\mathsf{CLE}}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{S}, \mathsf{Mpk})] = 1$$

for any $\mathsf{ID} \in \{0,1\}^*$ and any $\mathsf{m} \in \mathbb{M}_{\mathsf{CLE}}(I)$.

4.1 Single-user Security for CLE Schemes

We next recall two different security models for the privacy of encrypted plaintexts. In line with previous literature we classify the attackers as of type I, and M. Intuitively, the attackers in these models correspond to the following usage scenarios for a certificateless encryption scheme. A type I attack corresponds to the case where the key-generation center is honest; a type M attack corresponds to the case where the key-generation center is dishonest, and in particular, may generate its public/secret key pair not following the prescribed key-generation algorithm. In both models the adversaries are allowed to obtain partial secret keys (i.e. D that correspond to identity ID), secret values (i.e. SK that correspond to identity ID), replace public keys of parties, etc. For a more elaborate discussion on the various existent models for certificateless encryption we refer the reader to [11].

The experiments that define security against these attackers are summarized in Figure 3.

Both experiments are parameterized by bit $b \in \{0,1\}$ and involve an adversary $\mathcal{A}$. The adversaries in those experiments have access to a set of oracle $\mathcal{O}$ that formalizes its various capabilities. The oracle maintains internally two lists Real and Fake. The entries of both lists are of the form $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK})$ and maintain the secret keys that correspond to the public key of identity ID. List Real keeps track of the "real" public/secret keys of parties and its entries are not modified during the execution. The list Fake maintains keys associated to identities, but we allow the adversary to

$\mathbf{Exp}^{\mathsf{CL\text{-}CCA\text{-}I}}_{\mathsf{CLE},\mathcal{A},b}(1^k)$
1. $I \leftarrow \mathsf{Setup}(1^k)$
2. $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow \mathbb{G}_{\mathsf{CLE}}(I)$
3. $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}^*, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathcal{O}}(I, \mathsf{Mpk})$
4. $\mathsf{c}^* \leftarrow \mathbb{E}_{\mathsf{CLE}}(\mathsf{m}_b, \mathsf{ID}^*, \mathsf{PK}^*, \mathsf{Mpk})$
5. $b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(\mathsf{c}^*, \mathsf{st})$
6. Return b'

$\mathbf{Exp}^{\mathsf{CL\text{-}CCA\text{-}M}}_{\mathsf{CLE},\mathcal{A},b}(1^k)$
1. $(I) \leftarrow \mathsf{Setup}_{\mathsf{CLE}}(1^k)$
2. $(\mathsf{Mpk}, \mathsf{st}) \leftarrow \mathcal{A}_0(I)$
3. $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}^*, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathsf{st})$
4. $\mathsf{c}^* \leftarrow \mathbb{E}_{\mathsf{CLE}}(\mathsf{m}_b, \mathsf{ID}^*, \mathsf{PK}^*, \mathsf{Mpk})$
5. $b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(\mathsf{c}^*, \mathsf{st})$
6. Return b'

Fig. 3. Experiments for defining type I and M security for certificateless encryption scheme CLE. Here st is some state information and we require the messages m_0 and m_1 to be of equal length. Encryption in stage 5 is performed with respect to the public key PK^* associated with ID^* on the Fake list. In all of the above experiments the adversary $\mathcal{A}$ is required not to cause any of the following events. 1) It places both an ExtractFSK query on ID^*; 2) $\mathcal{A}_2$ placing a Decrypt query on $(\mathsf{L}, \mathsf{c}^*, \mathsf{ID}^*)$ and the current public key of ID^* retrieved from the list associated to L is PK^*.

modify them. In all experiments the oracles are parameterized by $(I, \mathsf{Mpk}, \mathsf{Msk})$, where we assume $\mathsf{Msk} = \perp$ in type M models. The queries that the adversary can make to the oracle are as follows.

- ReqPK(ID, D): This query has as parameter an identity ID and a partial private key D and is processed as follows. In experiment against a type I adversary the query is processed as follows: The oracle checks the list Real for an entry of the form $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}', \mathsf{S})$. If such an entry exists it returns PK. Otherwise, it 1) Executes $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathbb{U}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{Mpk})$; 2) Sets $\mathsf{D} \leftarrow \mathbb{X}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{Msk})$; 3) Computes $\mathsf{S} \leftarrow \mathbb{S}_{\mathsf{CLE}}(\mathsf{SK}, \mathsf{D}, \mathsf{Mpk})$; and 4) Adds $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}, \mathsf{S})$ to both the Real and Fake lists. It then returns PK to the adversary. Note that the provided D is not used in this case.
  In the experiment against a type M adversary the query is processed as follows: The oracle checks the list Real for an entry of the form $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}', \mathsf{S})$. If such an entry exists, it replaces D' with D and returns PK. Otherwise, it 1) Executes $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathbb{U}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{Mpk})$; 2) Computes $\mathsf{S} \leftarrow \mathbb{S}_{\mathsf{CLE}}(\mathsf{SK}, \mathsf{D}, \mathsf{Mpk})$; and 3) Adds $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}, \mathsf{S})$ to both the Real and Fake lists. It returns PK to the adversary.
- ReplacePK(ID, PK, SK): This query has as parameters an identity ID, a public key PK, and a secret key SK. When such a query is issued, the oracle searches the Fake lit for an entry $(\mathsf{ID}, \mathsf{PK}', \mathsf{SK}', \mathsf{D}, \mathsf{S})$ and replaces it with $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}, \mathsf{S})$. We assume, without loss of generality, that the adversary has previously performed a ReqPK query involving ID so that such an entry always exits.
- ExtractPSK(ID): When such a query is issued, the oracle returns the D component from the Real list. We assume ReqPK has already been called on ID.
- ExtractFSK(ID): When such a query is issued, the oracle returns the S component from the Real list. We assume ReqPK has already been called on ID.
- Decrypt(L, c, ID): This oracle has as parameters $\mathsf{L} \in \{\mathsf{fake}, \mathsf{real}\}$, an identity ID and a ciphertext c. On input $(\mathsf{real}, \mathsf{c}, \mathsf{ID})$ searches the list Real for an entry $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}, \mathsf{S})$. and on input $(\mathsf{fake}, \mathsf{c}, \mathsf{ID})$ the oracle finds an entry $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}, \mathsf{S})$ in Fake. It outputs $\mathsf{m} \leftarrow \mathbb{D}_{\mathsf{CLE}}(\mathsf{c}, \mathsf{ID}, \mathsf{PK}, \mathsf{S}, \mathsf{Mpk})$. We assume that prior to any query to the decryption oracle that contains ID, the adversary made at some point a query ReqPK(ID, D).

The experiments that we consider are summarized in Figure 3. In addition to the restrictions that we outline there, we define three specific classes of adversaries. Type I and M adversaries correspond to adversaries in the literature. Type I* adversary is a variant of type I which is useful in deriving our later results. These classes are defined by the following additional restrictions on the behavior of the adversary:

- *Type* I: Adversary Both $\mathcal{A}_1$ does not issue a $\mathsf{ReplacePK}(\mathsf{ID}^*, \mathsf{PK}, \mathsf{SK})$ query and $\mathcal{A}$ does not place an $\mathsf{ExtractPSK}(\mathsf{ID}^*)$ query;
- *Type* I*: Adversary $\mathcal{A}$ does not issue an $\mathsf{ExtractPSK}(\mathsf{ID}^*)$ query;

– *Type* M: Adversary $\mathcal{A}$ does not issue a ReplacePK or an ExtractPSK query.

For $\mathsf{x} \in \{\mathsf{I}, \mathsf{I}^\star, \mathsf{M}\}$, we say that a certificateless encryption scheme CLE is type x secure in the single-user CCA setting if for all ppt adversaries $\mathcal{A}$ its advantage defined by

$$\mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A}}(k) := \Pr\left[\mathbf{Exp}^{\mathsf{CL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A},1}(1^k) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{CL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A},0}(1^k) = 1\right]$$

is a negligible function.

4.2 Multi-user Security for CLE Schemes

Analogously to the identity-based setting, we extend the security models for certificateless encryption schemes to multi-user scenario in two different dimensions. First, we consider a setting where parties may possess more than one public key, and second, the adversary may receive multiple challenge ciphertexts. The experiments defining security are give in Figure 4.

$\mathbf{Exp}^{\mathsf{mCL\text{-}CCA\text{-}I}}_{\mathsf{CLE},\mathcal{A},b}(1^k)$
1. $I \leftarrow \mathsf{Setup}_{\mathsf{CLE}}(1^k)$
2. $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow \mathbb{G}_{\mathsf{CLE}}(I)$
3. $b' \leftarrow \mathcal{A}^{\mathcal{O}}(I, \mathsf{Mpk})$
4. Return b'

$\mathbf{Exp}^{\mathsf{mCL\text{-}CCA\text{-}M}}_{\mathsf{CLE},\mathcal{A},b}(1^k)$
1. $I \leftarrow \mathsf{Setup}_{\mathsf{CLE}}(1^k)$
2. $(\mathsf{Mpk}, \mathsf{st}) \leftarrow \mathcal{A}_0(I)$
3. $b' \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathsf{st})$
4. Return b'

Fig. 4. Experiments for defining type I and M security for a certificateless encryption scheme CLE in the multi-user setting. Here st is some state information. For the experiment against type I adversary $\mathcal{A}$, the adversary is required not to cause any of the following events. 1) Calling Encrypt on two messages with unequal lengths; 2) Calling ExtractFSK on an identity submitted to the Encrypt oracle; 3) Calling Decrypt query on $(\mathsf{L}, \mathsf{c}, \mathsf{ID}, \mathsf{PK})$ with c previously received from a left-right encryption query involving the pair $(\mathsf{ID}, \mathsf{PK})$ and such that the public key associated to ID on the list characterized by L is still PK.

We give separate experiments for type I and type M attackers. Both experiments involve an adversary $\mathcal{A}$ and are parameterized by bit $b \in \{0, 1\}$ that the adversary aims to determine. The oracles available to the adversary include those in the single-user setting. In addition, the adversary has access to a left-right encryption oracle, which is parameterized by b and has access to $(I, \mathsf{Mpk}, \mathsf{Msk})$. When the left-right oracle receives a query $\mathsf{Encrypt}(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}, \mathsf{PK})$ with m_0 and m_1 equal length messages, this oracle returns the ciphertext $\mathbb{E}_{\mathsf{CLE}}(\mathsf{m}_b, \mathsf{ID}, \mathsf{PK}, \mathsf{Mpk})$. Also, the queries to the ReqPK oracle trigger the generation of a new secret value/private key each time an identity is submitted, even if this identity had been previously submitted as a query.

We say that a certificateless encryption scheme CLE is type x secure for $\mathsf{x} \in \{\mathsf{I}, \mathsf{I}^\star, \mathsf{M}\}$ in the multi-user CCA setting if for all ppt adversaries $\mathcal{A}$ of type x, its advantage defined by

$$\mathbf{Adv}^{\mathsf{mCL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A}}(k) := \Pr\left[\mathbf{Exp}^{\mathsf{mCL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A},1}(1^k) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{mCL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A},0}(1^k) = 1\right]$$

is a negligible function.

The power afforded to the adversary in the multi-user extension seems quite extensive. Nevertheless, we can still show that security in the single-user setting is equivalent to security in the multi-user setting. Since one implication is obvious, we only state and prove the more difficult direction.

Lemma 2. *Let* CLE *be an arbitrary certificateless encryption scheme secure against type* x *adversaries in the single-user* CCA *setting (for some* $\mathsf{x} \in \{\mathsf{I}, \mathsf{I}^\star, \mathsf{M}\}$*). Then,* CLE *is secure against type* x *adversaries in the multi-user setting* CCA*.*

We prove the lemma in Appendix C where we show that for any ppt type x adversary $\mathcal{A}$ against a CLE scheme CLE in the multi-user setting, there exists a ppt type x adversary $\mathcal{B}$ against the scheme in the single-user setting such that:

$$\mathbf{Adv}^{\mathsf{mCL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A}}(k) \leq Q_{\mathsf{ID}}(k) \cdot Q_{\mathsf{PK}}(k) \cdot Q_{\mathsf{E}}(k) \cdot \mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{B}}(k).$$

Here $Q_{\mathsf{ID}}(k)$ is the number of distinct identities sent to the left-right encryption oracle, $Q_{\mathsf{PK}}(k)$ the maximum number of public key replacement query on an identity, and $Q_{\mathsf{E}}(k)$ the maximum number of left-right encryption queries on an identity/public key pair.

4.3 The CLE-2-CE Transformation

We now move towards proving that CLE schemes can be analyzed using CE models. First, we provide a syntactic transformation that associates a CE scheme to a CLE scheme. Just as for IBE encryption, the only needed change is a registration protocol; the rest of the algorithms of the scheme remain essentially the same.

Given a CLE scheme we define its associated CE scheme $\mathsf{CE} = \mathsf{CLE\text{-}2\text{-}CE}(\mathsf{CLE})$ by setting $\mathsf{Setup}_{\mathsf{CE}}$, $\mathbb{G}_{\mathsf{CE}}$, $\mathbb{E}_{\mathsf{CE}}$ and $\mathbb{D}_{\mathsf{CE}}$ to be $\mathsf{Setup}_{\mathsf{CLE}}$, $\mathbb{G}_{\mathsf{CLE}}$, $\mathbb{E}_{\mathsf{CLE}}$ and $\mathbb{D}_{\mathsf{CLE}}$ algorithms of CLE respectively. The registration protocol is defined as follows. In the registration protocol the user sends his identity to the CA who computes the partial key that corresponds to ID which he then sends back to ID. The user then generates a public key/secret key pair and sends the public key back to the CA. More formally, the registration protocol is as follows:

1. $\mathbb{U}_{\mathsf{CE}}(\mathsf{ID}, \mathsf{PK}_{\mathsf{CA}})$: Sends ID to the CA;
2. $\mathbb{C}_{\mathsf{CE}}(\mathsf{SK}_{\mathsf{CA}})$: Receives ID, runs $\mathsf{D} \leftarrow \mathbb{X}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{SK}_{\mathsf{CA}})$ and sends D to user ID;
3. $\mathbb{U}_{\mathsf{CE}}(\mathsf{ID}, \mathsf{PK}_{\mathsf{CA}})$: Receives D, runs $(\mathsf{SK}, \mathsf{PK}) \leftarrow \mathbb{U}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{PK}_{\mathsf{CA}})$ and $\mathsf{S} \leftarrow \mathbb{S}_{\mathsf{CLE}}(\mathsf{SK}, \mathsf{D}, \mathsf{Mpk})$. It sends PK to the CA and outputs $(\mathsf{ID}, \mathsf{PK}, \mathsf{S}, \epsilon)$ locally and terminates;
4. $\mathbb{C}_{\mathsf{CE}}(\mathsf{SK}_{\mathsf{CA}})$: Receives PK and outputs $(\mathsf{ID}, \mathsf{PK}, \epsilon)$ locally and terminates;

The main results of this section relate the security of CLE to that of $\mathsf{CLE\text{-}2\text{-}CE}(\mathsf{CLE})$. We start with the case when the attacker against the scheme is of type M, as in this case we get perfect equivalence between the models:

Theorem 2. *Let* CLE *be an arbitrary certificateless encryption scheme and set* $\mathsf{CE} := \mathsf{CLE\text{-}2\text{-}CE}(\mathsf{CLE})$. *Then* CLE *is secure against type* M *adversaries in the single-user* CCA *model if and only if* CE *is secure certified encryption scheme against adversaries in the* CCA *model which corrupt the CA.*

The proof, which we give in Appendix E, shows that for any ppt type M adversary $\mathcal{A}$ against CLE there exists a type M adversary $\mathcal{B}$ against CE such that type

$$\mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}M}}_{\mathsf{CLE},\mathcal{A}}(k) \leq Q_{\mathsf{ID}}(k) \cdot \mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}M}}_{\mathsf{CE},\mathcal{B}}(k).$$

Conversely, we have that for any adversary $\mathcal{A}$ of type M against CE there exists an type M adversary $\mathcal{B}$ against CLE such that:

$$\mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}M}}_{\mathsf{CE},\mathcal{A}}(k) \leq \mathbf{Adv}^{\mathsf{mCL\text{-}CCA\text{-}M}}_{\mathsf{CLE},\mathcal{B}}(k).$$

Unfortunately, for the case of honest CA (that is, type I adversaries) the situation is more complex and a similar theorem does not immediately hold. To clarify some of the difficulties, notice that via the transformation $\mathsf{CLE\text{-}2\text{-}CE}$ (and in fact via any other similar transformation), the resulting CE scheme can be trivially attacked using the powers that the adversary has in the CE model. The adversary, interacting with an honest CA does the following. It starts the execution of the registration protocol for some identity ID and stops (prematurely) before executing step (4) of the

protocol. That is, the adversary computes $(\mathsf{PK}, \mathsf{SK})$ as prescribed, but does not send PK to the certification authority. The adversary then calls the left-right encryption oracle on a message pair and $(\mathsf{ID}, \mathsf{PK})$. Notice that since the CA did not complete the protocol, the tuple $(\mathsf{ID}, \mathsf{PK})$ does not occur in the list $\mathsf{RegListSec}$ (which would have rendered the query invalid). However, the adversary has the right decryption key and thus can immediately win the game.

What happens here is that although from the point of view of the adversary, the key PK had been registered, the experiment does not (and cannot) capture this situation since it does not get access to PK. Since encryptions under such keys can be trivially decrypted by the adversary, this attack motivates a weaker, but still reasonable security model for certified encryption. The class of attackers that we consider have the right to issue left-right encryption queries only for identities that had not been corrupt. We call these kind of adversaries *weak* CE adversaries.

An additional important observation is that to simulate the queries of the CLE setting a CE adversary needs somehow access to the internal structure of the secret keys of parties. We provide such access using corrupt decryption oracles (as defined in Section 2). We show that we can the obtain a characterization of security against type I adversaries in the CLE sense via security in certified encryption models.

Recall that an adversary with corrupted decryption capabilities can choose functions in a set $\mathcal{F}$ to be applied to the secret keys of parties, before these secret keys are used to decrypt ciphertexts of the adversary's choice. For our purposes, we consider the set:

$$\mathcal{F} := \{(\mathrm{Id}, \mathrm{Id}, f, \mathrm{Id}) : f = \mathrm{Id} \text{ or } f = f_{\mathsf{SK}} : (\mathsf{SK}', \mathsf{D}') \mapsto (\mathsf{SK}, \mathsf{D}') \text{ for some } \mathsf{SK} \in \{0,1\}^*\}.$$

which consists essentially of functions that allow changing the first component SK' of the secret key of a party with any other secret key SK that the adversary chooses. We can the use adversary with $\mathcal{F}$-decryption to obtain the desired link. We start with a relation between type I^* adversaries against CLE and weak adversaries with $\mathcal{F}$-decryptions.

Proposition 1. *Let* CLE *be a certificateless encryption scheme and set* $\mathsf{CE} := \mathsf{CLE\text{-}2\text{-}CE}(\mathsf{CLE})$. *Then* CLE *is secure against type* I^* *adversaries in the single-user* CCA *model if and only if* CE *is secure against weak type* I *adversaries with* $\mathcal{F}$*-decryptions.*

The proof, which we give in Appendix E, shows that for any ppt type $\mathsf{I}^\star$ (resp. type M) adversary $\mathcal{A}$ against CLE in the single-user setting there exists a ppt adversary $\mathcal{B}$ against CE in the $\mathcal{F}$-extended model which does not corrupt (resp. corrupts) the CA such that:

$$\mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}I}^\star}_{\mathsf{CLE},\mathcal{A}}(k) \leq Q_{\mathsf{ID}}(k) \cdot \mathbf{Adv}^{\mathsf{mCE\text{-}}\mathcal{F}\text{-}\mathsf{CCA\text{-}I}^-}_{\mathsf{CE},\mathcal{B}}(k),$$

Here $Q_{\mathsf{ID}}(k)$ denotes the maximum number of identities queried to the experiment.

Conversely, for any ppt weak CE adversary $\mathcal{A}$ against CE with $\mathcal{F}$-decryptions there exists a ppt type $\mathsf{I}^\star$ adversary $\mathcal{B}$ against CLE in the multi-user setting such that:

$$\mathbf{Adv}^{\mathsf{mCE\text{-}}\mathcal{F}\text{-}\mathsf{CCA\text{-}I}^-}_{\mathsf{CE},\mathcal{A}}(k) \leq \mathbf{Adv}^{\mathsf{mCL\text{-}CCA\text{-}I}^\star}_{\mathsf{CLE},\mathcal{B}}(k).$$

To obtain a relation between security against type I adversaries in the CLE model and security in the sense of certified encryption we first establish a link between models where the adversary can obtain information on D using corruption capabilities and models where the adversary corrupts the master secret key which he then uses to obtain D on his own. The proof of the Lemma is in Appendix D.

Lemma 3. *For any ppt type* I *adversary* $\mathcal{A}$*, there exists ppt adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *in type* $\mathsf{I}^\star$ *and type* M *models respectively such that:*

$$\mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}I}}_{\mathsf{CLE},\mathcal{A}}(k) \leq \mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}I}^\star}_{\mathsf{CLE},\mathcal{B}_1}(k) + \mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}M}}_{\mathsf{CLE},\mathcal{B}_2}(k),$$

The following theorem is the main result of this section. Informally it says that security of certificateless schemes can be analyzed using certified encryption models, extended with corrupt decryptions. The proof of the theorem follows from Proposition 1 and Lemma 3.

Theorem 3. CLE *scheme is secure against type* I *and type* M *attackers if and only if* $\mathsf{CLE\text{-}2\text{-}CE}(\mathsf{CLE})$ *is secure against type* I *weak CE adversaries with* $\mathcal{F}$*-decryption and also secure against type* M *adversaries.*

As described earlier in the paper, several variations of the CLE schemes have been proposed. Our results should extend to those settings. In particular, we sketch in Appendix F how to deal with the models of [9, 3].

Acknowledgments

The work carried out by the first author was supported in part by the Scientific and Technological Research Council of Turkey (TÜBİTAK) while at Middle East Technical University. The authors has been supported in part by the European Commission through the IST Programme under Contract IST-2007-216646 ECRYPT II grant. The information in this document reflects only the authors' views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. S.S. Al-Riyami and K.G. Paterson. Certificateless Public-Key Cryptography. In *Advances in Cryptology – ASIACRYPT 2003*, LNCS 2894:452–473, Springer-Verlag, 2003.
2. M.H. Au, J. Chen, J.K. Liu, Y. Mu, D.S. Wong and G. Yang. Malicious KGC Attacks in Certificateless Cryptography. In *ACM Symposium on Information, Computer and Communications Security*, pages 302–311, March 2007.
3. J. Baek, R. Safavi-Naini and W. Susilo. Certificateless Public Key Encryption Without Pairing. In *Proceedings of the 8th International Conference on Information Security (ISC 2005)*, LNCS 3650:134-148, Springer-Verlag, 2005.
4. M. Bellare, A. Boldyreva and S. Micali. Public-Key Encryption in a Multi-User Setting: Security Proofs and Improvements. In *Advances in Cryptology – EUROCRYPT 2000*, LNCS 1807:259–274, Springer-Verlag, 2000.
5. M. Bellare, A. Boldyreva and J. Staddon. Multi-Recipient Encryption Schemes: Security Notions and Randomness Re-Use. In *Public Key Cryptography – PKC 2003*, LNCS 2567:85–99, Springer-Verlag, 2003.
6. M. Bellare and T. Kohno. A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications. In *Advances in Cryptology – CRYPTO 2003*, LNCS 2656:491–506, Springer-Verlag, 2003.
7. A. Boldyreva, M. Fischlin, A. Palacio and B. Warinschi. A Closer Look at PKI: Security and Efficiency. In *Public Key Cryptography – PKC 2007*, LNCS 4450:458–475, Springer-Verlag, 2007.
8. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In *SIAM Journal on Computing*, 32:586–615, 2003.
9. Z. Cheng and R. Comley. Efficient Certificateless Public Key Encryption. In *Cryptology ePrint Archive*, Report 2005/012, 2005.
10. A.W. Dent. A Note On Game-Hopping Proofs. In *Cryptology ePrint Archive*, Report 2006/260, 2006.
11. A.W. Dent. A Survey of Certificateless Encryption Schemes and Security Models. In *International Journal of Information Security*, 7(5):349–377, Springer-Verlag, 2008.
12. C. Gentry. Certificate-Based Encryption and the Certificate Revocation Problem. In *Advances in Cryptology – EUROCRYPT 2003*, LNCS 2656:272–293, Springer-Verlag, 2003.
13. C. Gentry. Practical Identity-Based Encryption without Random Oracles. In *Advances in Cryptology – EUROCRYPT 2006*, LNCS 4004:445–464, Springer-Verlag, 2006.
14. J. Herzog, M. Liskov and S. Micali. Plaintext Awareness via Key Registration. In *Advances in Cryptology – CRYPTO 2003*, LNCS 2729:548–564, Springer-Verlag, 2003.
15. B.Kaliski. An Unknown Key-Share Attack on the MQV Key Agreement Protocol. In *ACM Transactions on Information and System Security – TISSEC*, 4(3):275–288, 2001.
16. 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 *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*, pages 273–283, ACM Press, 2007.

17. R. Sakai and M. Kasahara. ID-Based Cryptosystems with Pairing on Elliptic Curve. In *2003 Symposium on Cryptography and Information Security – SCIS 2003*, 2003.
18. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Advances in Cryptology – CRYPTO '84*, LNCS 196:47–53, Springer-Verlag, 1985.
19. V. Shoup. On Formal Models for Secure Key Exchange. *IBM Research Report RZ 3120*, 1999.
20. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology – EUROCRYPT 2005*, LNCS 3494:114–127, Springer-Verlag, 2005.

A Multi-User Security of IBE Schemes

Proof. The proof follows that for standard public-key encryption given in [4]. We begin by designing the following sequence of experiments parameterized by and $\ell \in \{0, \ldots, Q_{\mathsf{ID}} Q_{\mathsf{E}}\}$[3].

$\mathbf{Exp}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A},\ell}(1^k)$
- If $\ell = 0$ then set $(c, r) \leftarrow (0, 0)$.
- If $\ell > 0$ then (c, r) is such that $(c-1)Q_{\mathsf{E}} + r = \ell$ with $1 \le r \le Q_{\mathsf{E}}$ and $1 \le c \le Q_{\mathsf{ID}}$.
- Run $\mathcal{A}$ as follows.
  Answer an encryption query $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID})$ on user ID with index $1 \le i \le Q_{\mathsf{ID}}$ as follows.
  - If $i < c$ the left message m_0 is encrypted.
  - If $i > c$ the right message m_1 is encrypted.
  - If $i = c$ then
    * $ctr \leftarrow ctr + 1$
    * If $ctr \le r$ then the left message m_0 is encrypted.
    * If $ctr > r$ then the right message m_1 is encrypted.

  Extraction and decryption queries are answered truthfully.
- Return the bit d that $\mathcal{A}$ eventually outputs.

Here c and r should be thought as the "user number" and "encryption query number" defined by parameter ℓ. If we let P_ℓ denote the probability that the output is 1 in the ℓ-th experiment, we see that:

$$\mathbf{Adv}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(k) = P_0 - P_{Q_{\mathsf{ID}} Q_{\mathsf{E}}}.$$

We now describe an adversary $\mathcal{B}$ attacking the scheme in the single user IND-atk security game.

- Pick $\ell \in \{1, \ldots, Q_{\mathsf{ID}} Q_{\mathsf{E}}\}$ uniformly at random.
- Set (c, r) such that $(c-1)Q_{\mathsf{E}} + r = \ell$ with $1 \le r \le Q_{\mathsf{E}}$ and $1 \le c \le Q_{\mathsf{ID}}$.
- Run $\mathcal{A}$ as follows.
  Answer an encryption query $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID})$ on user ID with index $1 \le i \le Q_{\mathsf{ID}}$ as follows.
  - If $i < c$ return an encryption of m_0.
  - If $i > c$ return an encryption of m_1.
  - If $i = c$ then
    * $ctr \leftarrow ctr + 1$
    * If $ctr < r$ return an encryption of m_0.
    * If $ctr = r$ call $\mathcal{B}$'s challenge oracle and return the result.
    * If $ctr > r$ return an encryption of m_1.

  Extraction and decryption queries are answered using the equivalent oracles provided to $\mathcal{B}$ in the single user-game.
- Return the bit d that $\mathcal{A}$ eventually outputs.

[3] We drop the parameter k for the sake of readability. We also associated identity ID with an index i which indicates ID is the i-th non-repeat identity queried to the experiment.

Note that the restrictions on the oracles in the multi-user game implies those in the single-user game and hence the above simulation is consistent according to the rules of ID-CCA game.

Let us denote the bit hidden in the single-user encryption oracle by b. We have that

$$\Pr[d = 1|b = 1] = \sum_{i=1}^{Q_{\mathsf{ID}}Q_{\mathsf{E}}} \Pr[d = 1 \wedge \ell = i|b = 1]$$
$$\Pr[d = 1|b = 0] = \sum_{i=1}^{Q_{\mathsf{ID}}Q_{\mathsf{E}}} \Pr[d = 1 \wedge \ell = i|b = 0].$$

Note that conditioned on a fixed value of $\ell = i$, when $b = 0$, algorithm $\mathcal{A}$ is run by $\mathcal{B}$ according to the environment in the i-th experiment and when $b = 1$, it is run in the $(i-1)$-st experiment environment. The advantage of $\mathcal{B}$ in the single user game is the difference between the above probabilities. Noting the cancellation in the telescoping sum and given the fact that ℓ is chosen uniformly at random we obtain.

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{ID\text{-}CCA}}_{\mathsf{IBE},\mathcal{B}}(k) &= \Pr[d = 1|b = 1] - \Pr[d = 1|b = 0] \\
&= \Pr[d = 1 \wedge \ell = 1|b = 1] - \Pr[d = 1 \wedge \ell = Q_{\mathsf{ID}}Q_{\mathsf{E}}|b = 0] \\
&= 1/(Q_{\mathsf{ID}}Q_{\mathsf{E}}) \cdot (\Pr[d = 1|b = 1 \wedge \ell = 1] - \Pr[d = 1|b = 0 \wedge \ell = Q_{\mathsf{ID}}Q_{\mathsf{E}}]) \\
&= 1/(Q_{\mathsf{ID}}Q_{\mathsf{E}}) \cdot (P_0 - P_{Q_{\mathsf{ID}}Q_{\mathsf{E}}}) \\
&= 1/(Q_{\mathsf{ID}}Q_{\mathsf{E}}) \cdot \mathbf{Adv}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(k).
\end{aligned}$$

□

B The IBE-2-CE Transformation

Proof. Let $\mathcal{A}$ be an adversary against the scheme in the ID-CCA model.

Game_0 : This game is identical to the ID-CCA game:

$$\mathbf{Adv}^{\mathsf{Game}_0}_{\mathsf{IBE},\mathcal{A}}(k) = \mathbf{Adv}^{\mathsf{ID\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(k)$$

Game_1 : This game chooses an index ℓ uniformly at random from $\{1, \ldots, Q_{\mathsf{ID}}\}$. The following restriction is introduced in this game.

– $\mathcal{A}_1$ is not allowed to output an identity ID^* whose index is not ℓ.

A straightforward argument [10] shows that:

$$\mathbf{Adv}^{\mathsf{Game}_1}_{\mathsf{IBE},\mathcal{A}}(k) = (1/Q_{\mathsf{ID}}) \cdot \mathbf{Adv}^{\mathsf{Game}_0}_{\mathsf{IBE},\mathcal{A}}(k)$$

We now construct an adversary $\mathcal{B}$ against the associated certified scheme in the private mCE-CCA-I model. This algorithm operates as follows.

– It chooses an index ℓ uniformly at random from $\{1, \ldots, Q_{\mathsf{ID}}\}$
– It obtains the pair (I, Mpk) and passes it onto $\mathcal{A}$.
– When $\mathcal{A}_1$ places a $\mathsf{Extract}(\mathsf{ID})$ query, if the index of ID is not ℓ, algorithm $\mathcal{B}$ places the query $\mathsf{Register}(\mathsf{ID}, \mathsf{corrupt})$. It obtains the corresponding user secret key D, and returns this to the adversary $\mathcal{A}$. Note that by the rules of the game an identity with index ℓ cannot be queried to this oracle.

– When $\mathcal{A}_1$ places a $\mathsf{Decrypt}(\mathsf{c}, \mathsf{ID})$ query, if the index of ID is not ℓ, algorithm $\mathcal{B}$ first places the query $\mathsf{Register}(\mathsf{ID}, \mathsf{corrupt})$ if it has not already done so. It then uses the knowledge of the user secret key to answer the query. If the index of ID is ℓ, algorithm $\mathcal{B}$ places the same query to the decryption oracle provided to it, querying $\mathsf{Register}(\mathsf{ID}, \mathsf{honest})$ first if it has not already done so.
– When $\mathcal{A}_1$ outputs $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}^*)$, by the rules of the game we know that the index of ID^* is ℓ. Algorithm $\mathcal{B}$ places the same query to its left-right encryption oracle, registering the user as honest first if needed. Note that by the rules of the $\mathsf{ID\text{-}CCA}$ game ID^* is never queried to the extraction oracle, and hence ID^* is an honestly registered user in the $\mathsf{mCE\text{-}CCA\text{-}I}$ game.
– Decryption queries made by $\mathcal{A}_2$ are answered as in phase one: When the identity involved is ID^* the provided decryption oracle is called. For other ID's algorithm $\mathcal{B}$ answers using the knowledge of secret keys. Note that the restrictions on the decryption oracles in the two experiments are compatible.
– Eventually $\mathcal{A}$ outputs a bit b, which $\mathcal{B}$ also returns as its own guess.

The simulation above is perfect according to the rules of Game_1. Moreover the hidden bit in the certified game corresponds to that in the identity-based game. Therefore:

$$\mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}I}}_{\mathsf{CE},\mathcal{B}}(k) = \mathbf{Adv}^{\mathsf{Game}_1}_{\mathsf{IBE},\mathcal{A}}(k).$$

Conversely suppose that there is an $\mathsf{mCE\text{-}CCA\text{-}I}$ adversary $\mathcal{A}$ against the transformed certified scheme. We construct an $\mathsf{mID\text{-}CCA}$ adversary $\mathcal{B}$ against the underlying IBE scheme as follows.

– Adversary $\mathcal{B}$ obtains (I, Mpk) and passes it onto $\mathcal{A}$.
– When $\mathcal{A}$ places a query $\mathsf{Register}(\mathsf{ID}, \mathsf{L})$ algorithm $\mathcal{B}$ proceeds as follows.
  • If $\mathsf{L} = \mathsf{corrupt}$, algorithm $\mathcal{B}$ places the query $\mathsf{Extract}(\mathsf{ID})$ to obtain D. This enables $\mathcal{B}$ to execute the registration protocol on ID with $\mathcal{A}$.
  • If $\mathsf{L} = \mathsf{honest}$, algorithm $\mathcal{B}$ takes no action.
– When $\mathcal{A}$ calls its left-right encryption oracle on $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID})$, algorithm $\mathcal{B}$ places the same query to its equivalent left-right encryption oracle. Since ID must be uncorrupted according to the rules of the certified model, ID has not been queried to the extraction oracle and hence $\mathcal{B}$'s query is legal.
– When $\mathcal{A}$ calls its decryption oracle on $(\mathsf{c}, \mathsf{ID})$, algorithm $\mathcal{B}$ places the same query to its decryption oracle. Again it is easily checked that the restrictions are compatible.
– Eventually $\mathcal{A}$ outputs a bit b, which $\mathcal{B}$ also returns as its own guess.

The simulation presented above is perfect according to the rules of $\mathsf{mCE\text{-}CCA\text{-}I}$ experiment. Moreover the hidden bit in the IBE game corresponds to that in the certified game. We therefore have:

$$\mathbf{Adv}^{\mathsf{mID\text{-}CCA}}_{\mathsf{IBE},\mathcal{B}}(k) = \mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}I}}_{\mathsf{CE},\mathcal{A}}(k).$$

□

C Multi-User Security of CLE Schemes

Proof. The proof is a hybrid argument similar to that given in Appendix B, involving and extra dimension which takes the public keys into account. We only prove type I and M simultaneously. In what follows c should be though as the identity number, g as the public key number and r as the encryption query number corresponding to ℓ. We begin by considering the following experiments parameterized by $0 \leq \ell \leq Q_{\mathsf{ID}} Q_{\mathsf{PK}} Q_{\mathsf{E}}$.

$\mathbf{Exp}^{\mathsf{mCL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A},\ell}(1^k)$

- If $\ell = 0$ then set $(c, g, r) \leftarrow (0, 0, 0)$.
- If $\ell > 0$ then (c, g, r) is such that $(c-1)Q_{\mathsf{PK}}Q_{\mathsf{E}} + (g-1)Q_{\mathsf{E}} + r = \ell$ with $1 \le r \le Q_{\mathsf{E}}$, $1 \le g \le Q_{\mathsf{PK}}$ and $1 \le c \le Q_{\mathsf{ID}}$.
- Run $\mathcal{A}$ as follows.
  Answer an encryption query $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID})$ on user ID with index $1 \le i \le Q_{\mathsf{ID}}$ as follows.
  - If $i < c$ the left message m_0 is encrypted.
  - If $i > c$ the right message m_1 is encrypted.
  - If $i = c$ then
    - $j \leftarrow j + 1$
    - If $j < g$ then the left message m_0 is encrypted.
    - If $j > g$ then the right message m_1 is encrypted.
    - If $j = g$ then
      - $ctr \leftarrow ctr + 1$
      - If $ctr \le r$ then the left message m_0 is encrypted.
      - If $ctr > r$ then the right message m_1 is encrypted.

  Other oracle queries are answered truthfully.
- Return the bit d that $\mathcal{A}$ eventually outputs.

If we let P_ℓ denote the probability that the output is 1 in the ℓ-th experiment, we see that:

$$\mathbf{Adv}^{\mathsf{mCL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A}}(k) = P_0 - P_{Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}}.$$

We now describe an adversary $\mathcal{B}$ attacking the scheme in the single user $\mathsf{CL\text{-}CCA}$ security game.

- Pick $\ell \in \{1, \ldots, Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}\}$ uniformly at random.
- Set (c, r) such that $(c-1)Q_{\mathsf{PK}}Q_{\mathsf{E}} + (g-1)Q_{\mathsf{E}} + r = \ell$ with $1 \le r \le Q_{\mathsf{E}}$, $1 \le g \le Q_{\mathsf{PK}}$ and $1 \le c \le Q_{\mathsf{ID}}$.
- Generate $(\mathsf{SK}_{i,j}, \mathsf{PK}_{i,j})$ for $1 \le j \le Q_{\mathsf{PK}}$ and $1 \le i \le Q_{\mathsf{ID}}$ except $(i, j) = (c, g)$
- Run $\mathcal{A}$ as follows.
  Answer an encryption query $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID})$ on user ID with index $1 \le i \le Q_{\mathsf{ID}}$ as follows.
  - If $i < c$ return an encryption of m_0.
  - If $i > c$ return an encryption of m_1.
  - If $i = c$ then
    - $j \leftarrow j + 1$
    - If $j < g$ return an encryption of m_0.
    - If $j > g$ return an encryption of m_1.
    - If $j = g$ then
      - $ctr \leftarrow ctr + 1$
      - If $ctr < r$ return an encryption of m_0.
      - If $ctr = r$ call $\mathcal{B}$'s challenge oracle and return the result.
      - If $ctr > r$ return an encryption of m_1.

  Oracle queries are answered as described below.
- Return the bit d that $\mathcal{A}$ eventually outputs.

Algorithm $\mathcal{B}$ needs to simulate a ReqPK oracle which allows multiple public keys to be associated with a user. This is done as follows for type I.

- $\mathsf{ReqPK}(\mathsf{ID})$. Let i denote the index of ID and j be the request number on this identity. If $(i, j) \neq (c, g)$ then $\mathcal{B}$ returns $\mathsf{PK}_{i,j}$. For query (c, g), the provided request public key oracle is called to obtain $\mathsf{PK}_{c,g}$.

For type M, algorithm $\mathcal{B}$ uses the same procedure and also adds/replaces the provided D in the list.

Note that $\mathcal{B}$ has all the necessary information to enable him to answer $\mathsf{ExtractPSK}$ and $\mathsf{ExtractFSK}$ queries. Decryption queries (in both models) can also be answered perfectly. In type I, algorithm $\mathcal{B}$ uses the knowledge of secret values with the decryption oracle with fake label or using the decryption oracle with real label if dealing with $\mathsf{PK}_{c,g}$. In type M model, it either uses the decryption oracle provided to it, or the knowledge of SK and D, where the latter is provided by $\mathcal{A}$.

Note first that the restrictions on the partial private key, full private key and decryption oracles in the multi-user game include those in the single-user game. Hence the above simulation is consistent according to the rules of $\mathsf{ID\text{-}CCA}$ game.

Let us denote the bit hidden in the single-user encryption oracle by b. We have that

$$\Pr[d = 1|b = 1] = \sum_{i=1}^{Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}} \Pr[d = 1 \wedge \ell = i|b = 1],$$
$$\Pr[d = 1|b = 0] = \sum_{i=1}^{Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}} \Pr[d = 1 \wedge \ell = i|b = 0].$$

Note that conditioned on a fixed value of $\ell = i$, when $b = 0$, algorithm $\mathcal{A}$ is run by $\mathcal{B}$ according to the environment in the i-th experiment and when $b = 1$, it is run in the $(i-1)$-st experiment environment. The advantage of $\mathcal{B}$ in the single user game is the difference between the above probabilities. Noting the cancellation in the telescoping sum and given the fact that ℓ is chosen uniformly at random we obtain.

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{B}}(k) &= \Pr[d = 1|b = 1] - \Pr[d = 1|b = 0] \\
&= \Pr[d = 1 \wedge \ell = 1|b = 1] - \Pr[d = 1 \wedge \ell = Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}|b = 0] \\
&= 1/(Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}) \cdot (\Pr[d = 1|b = 1 \wedge \ell = 1] - \Pr[d = 1|b = 0 \wedge \ell = Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}]) \\
&= 1/(Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}) \cdot (P_0 - P_{Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}}) \\
&= 1/(Q_{\mathsf{ID}}Q_{\mathsf{PK}}Q_{\mathsf{E}}) \cdot \mathbf{Adv}^{\mathsf{mCL\text{-}CCA\text{-}x}}_{\mathsf{CLE},\mathcal{A}}(k).
\end{aligned}$$

□

D Type I* Security

Proof. We prove the lemma using a game hop as follows.

Game_0 : This game is identical to the $\mathsf{CL\text{-}CCA\text{-}I}$ game:

$$\mathbf{Adv}^{\mathsf{Game}_0}_{\mathsf{CLE},\mathcal{A}}(k) = \mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}I}}_{\mathsf{CLE},\mathcal{A}}(k).$$

Game_1 : In this game, the adversary is not allowed to trigger the following event.

– Event E: The partial private key for the challenge identity is extracted.

We show that:

$$\mathbf{Adv}^{\mathsf{Game}_0}_{\mathsf{CLE},\mathcal{A}}(k) - \mathbf{Adv}^{\mathsf{Game}_1}_{\mathsf{CLE},\mathcal{A}}(k) \leq \mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}M}}_{\mathsf{CLE},\mathcal{B}}(k).$$

The games are identical unless E happens. Conditioned on E, we build a type M adversary $\mathcal{B}$ to break the $\mathsf{CL\text{-}CCA\text{-}M}$ security of the scheme. This algorithm operates as follows.

– Algorithm $\mathcal{B}$ obtains I from its type M game environment. It generates a pair $(\mathsf{Msk}, \mathsf{Mpk})$ by running $\mathbb{G}_{\mathsf{CLE}}(I)$. It then passes (I, Mpk) to $\mathcal{A}$.

– It answers various oracle queries using the equivalent oracles provided to it in type M game, with the following exceptions:
  - ExtractPSK(ID) queries are answered using the knowledge of Msk.
  - ReplacePK(ID, PK, SK) queries (which by event E will be on identities other than ID^*) are answered by maintaining a list L which mimics the Fake list and contains tuples $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}, \mathsf{S})$. This list is initialized, for any identity ID submitted to the ReqPK oracle, by obtaining the PK from the equivalent ReqPK(ID, D) oracle, with D computed using Msk, and storing $(\mathsf{ID}, \mathsf{PK}, \perp, \mathsf{D}, \perp)$ on L. Algorithm $\mathcal{B}$ simply replaces the entry on L which has ID with the provided (PK, SK) and recomputes the S component using SK and D.
  - Decrypt queries on $\mathsf{L} = \mathsf{fake}$ and $\mathsf{ID} \neq \mathsf{ID}^*$ are answered using the list L. If ID does not appear on the list, it is added as described above. On $\mathsf{ID} = \mathsf{ID}^*$, algorithm $\mathcal{B}$ uses the label Real as no public key replacement queries are allowed by event E.
– When a $\mathcal{A}$ outputs a message pair and an identity, algorithm $\mathcal{B}$ also outputs this in its own game environment. Note that since we have conditioned on event E the public key of the challenge identity is never replaced and hence this query is allowed in type M game environment.
– Various oracles queries in the second stage are answered as in the first stage.
– When $\mathcal{A}$ outputs a bit b, algorithm $\mathcal{B}$ also outputs this as its own guess.

It is easily checked that conditioned on event E, the above simulation is perfect for the Game_0 and Game_1 environments. Moreover, the hidden bit in the type M game matches that in Game_1.

Now observe that the rules of Game_1 is identical to those in the type I* experiment. Therefore:

$$\mathbf{Adv}^{\mathsf{Game}_1}_{\mathsf{CLE},\mathcal{A}}(k) = \mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}I}^\star}_{\mathsf{CLE},\mathcal{A}}(k).$$

Putting the above together we obtain the desired result. □

E The CLE-2-CE Transformation

Proof. Let $\mathcal{A}$ be an adversary against the CLE scheme in the CL-CCA-I* model.

Game_0 : This game is identical to the CL-CCA-I* game:

$$\mathbf{Adv}^{\mathsf{Game}_0}_{\mathsf{CLE},\mathcal{A}}(k) = \mathbf{Adv}^{\mathsf{CL\text{-}CCA\text{-}I}^\star}_{\mathsf{CLE},\mathcal{A}}(k).$$

Game_1 : This game chooses an index ℓ uniformly at random from $\{1, \ldots, Q_{\mathsf{ID}}\}$. The following restriction is introduced in this game.

– $\mathcal{A}_1$ is not allowed to output an identity ID^* whose index is not ℓ.

A straightforward argument shows that:

$$\mathbf{Adv}^{\mathsf{Game}_1}_{\mathsf{CLE},\mathcal{A}}(k) = (1/Q_{\mathsf{ID}}) \cdot \mathbf{Adv}^{\mathsf{Game}_0}_{\mathsf{CLE},\mathcal{A}}(k).$$

We now construct an adversary $\mathcal{B}$ against the associated certified scheme in the private channels mCE-$\mathcal{F}$-CCA-I$^-$ model as follows.

– Algorithm $\mathcal{B}$ first chooses an index ℓ in $\{1, \ldots, Q_{\mathsf{ID}}\}$ uniformly at random.
– It receives (I, Mpk) from the certified game and passes it onto $\mathcal{A}$. It simulates $\mathcal{A}$'s oracles as follows.

- $\mathsf{ReqPK}(\mathsf{ID})$[4]: On input a new identity ID, if the index is ℓ, algorithm $\mathcal{B}$ invokes user ID to register honestly, obtains $(\mathsf{ID}, \mathsf{PK})$ and stores $(\mathsf{ID}, \mathsf{PK}, \perp, \perp, \perp)$ in two initially empty list Real and Fake. If the index is different from ℓ, it performs a corrupt run of the registration protocol on ID by generating a $(\mathsf{SK}, \mathsf{PK})$ pair and receiving D from the CA. It stores $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}, \mathsf{S})$, where S is computed using the $\mathbb{S}_{\mathsf{CLE}}$ algorithm, in the two lists. For identities which are not new (I.e. already queried to this oracle) the PK component is looked up and then returned.
- $\mathsf{ReplacePK}(\mathsf{ID}, \mathsf{PK}, \mathsf{SK})$ When $\mathcal{A}$ asks to replace the public key of an identity ID, irrespective of its index, $\mathcal{A}$ replaces the corresponding entry on Fake with the provided PK and SK.
- $\mathsf{ExtractPSK}(\mathsf{ID})$: In type $\mathsf{I}^\star$ model, these queries can be answered using the Real list for all identities whose index is not ℓ. According to the rules of this game, such a query is not allowed on the identity with index ℓ.
- $\mathsf{ExtractFSK}(\mathsf{ID})$: Algorithm $\mathcal{B}$ returns the component S stored in list Real. Note that this query can always be answered as the index of ID will not be ℓ.
- $\mathsf{Decrypt}(\mathsf{real}, \mathsf{c}, \mathsf{ID})$: These queries are answered in two ways:
  - The index of ID is not ℓ: Algorithm $\mathcal{B}$ uses the knowledge of S.
  - The index of ID is ℓ: Algorithm $\mathcal{B}$ calls the decryption oracle provided to it with $(f, \mathsf{c}, \mathsf{ID}, \mathsf{PK})$, where PK is the original public key in list Real and $f := \mathrm{Id}$.
- $\mathsf{Decrypt}(\mathsf{fake}, \mathsf{c}, \mathsf{ID})$: These queries are answered in two ways. We assume the public key of ID is replaced as otherwise $\mathcal{B}$ can use the label real.
  - The index of ID is not ℓ: Algorithm $\mathcal{B}$ uses the knowledge of S can be computed using knowledge of D and the SK which $\mathcal{A}$ provides to $\mathcal{B}$.
  - The index of ID is ℓ: Algorithm $\mathcal{B}$ obtains SK from $\mathcal{A}$ and calls the decryption oracle provided to it on $(f, \mathsf{c}, \mathsf{ID}, \mathsf{PK})$ where PK is obtained from the Fake list and $f := (\mathrm{Id}, \mathrm{Id}, f_3, \mathrm{Id})$ with $f_3 := (\mathsf{SK}', \mathsf{D}') \mapsto (\mathsf{SK}, \mathsf{D}')$.
- When $\mathcal{A}$ outputs $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}^*)$ to obtain a challenge ciphertext, algorithm $\mathcal{B}$ is assured that the index of ID^* is ℓ. It passes $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}^*, \mathsf{PK}^*)$ to its left-right oracle to get a challenge and relays the answer to $\mathcal{A}$. Here PK^* is the public key obtained from the list Fake.
  Note that, since the CA *and* ID^* are honest, such a query is allowed according to the rules of the weakened certified game: $\mathcal{B}$'s encryption oracle does allow queries on $(\mathsf{ID}, \mathsf{PK})$ which are not on $\mathsf{RegListPub}$ with ID honest.
- Various oracle queries in the second stage are answered as in the first stage. It is easy to see that the restrictions on the decryption oracle after receiving the challenge ciphertext in the CLE game is compatible with those in the CE game.
- When $\mathcal{A}$ outputs a bit b, algorithm $\mathcal{B}$ will also output this as his own guess.

It follows that:

$$\mathbf{Adv}^{\mathsf{mCE}\text{-}\mathcal{F}\text{-}\mathsf{CCA}\text{-}\mathsf{I}^-}_{\mathsf{CE},\mathcal{B}}(k) = \mathbf{Adv}^{\mathsf{Game}_1}_{\mathsf{CLE},\mathcal{A}}(k).$$

Now let $\mathcal{A}$ denote an adversary against the CLE scheme in the $\mathsf{CL}\text{-}\mathsf{CCA}\text{-}\mathsf{M}$ model. We define Game_0 and Game_1 as we did above and construct an adversary $\mathcal{B}$ against the associated certified scheme in the Game_1 environment.

- Algorithm $\mathcal{B}$ first chooses an index ℓ in $\{1, \ldots, Q_{\mathsf{ID}}\}$ uniformly at random.
- It receives I from the certified game and passes it onto $\mathcal{A}$.
- When algorithm $\mathcal{A}$ returns a Mpk, algorithm $\mathcal{B}$ also returns to its game environment. It answers various oracles queries as follows.
- $\mathsf{ReqPK}(\mathsf{ID}, \mathsf{D})$: For this query, if the index of ID is not ℓ, algorithm $\mathcal{B}$ invokes user ID to register honestly. It uses the provided D to execute the registration protocol with the user. It obtains $(\mathsf{ID}, \mathsf{PK})$ and stores $(\mathsf{ID}, \mathsf{PK}, \perp, \mathsf{D}, \perp)$ in two initially empty lists Real and Fake. If the index is

[4] We omit D as it is only relevant in type M model. We also omit $\mathsf{cert} = \epsilon$ throughout the proof.

different from ℓ, it generates a pair $(\mathsf{SK}, \mathsf{PK})$ and stores $(\mathsf{ID}, \mathsf{PK}, \mathsf{SK}, \mathsf{D}, \mathsf{S})$ in the two lists, where S is computed using the $\mathbb{S}_{\mathsf{CLE}}$ algorithm. For non-new identities the component D is simply replaced.
- $\mathsf{ExtractFSK}(\mathsf{ID})$: Algorithm $\mathcal{B}$ returns the component S stored in list Real. Note that this query can always be answered, as the index of ID will not be ℓ.
- $\mathsf{Decrypt}(\mathsf{real}, \mathsf{c}, \mathsf{ID})$: These queries are answered in two ways:
  - The index of ID is not ℓ: Algorithm $\mathcal{B}$ uses the knowledge of S.
  - The index of ID is ℓ: Algorithm $\mathcal{B}$ uses the decryption oracle provided to it.

  Recall that fake decryption queries are non-existent in type M models due to lack of public key replacement queries.
- When $\mathcal{A}$ outputs $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}^*)$ to obtain a challenge ciphertext, algorithm $\mathcal{B}$ is assured that the index of ID^* is ℓ. It passes $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}^*, \mathsf{PK}^*)$ to its left-right oracle to obtain a ciphertext which it passes to $\mathcal{A}$. Here PK^* is the public key obtained from the list Fake. Note that $\mathcal{B}$ has corrupted the CA and ID^* is honest and appears on $\mathsf{RegListPub}$. Hence this query in the certified game environment is valid.
- Various oracle queries in the second stage are answered as in the first stage.
- When $\mathcal{A}$ outputs a bit b, algorithm $\mathcal{B}$ will also output this as his own guess.

Since the simulation above is perfect according to the rules of Game_1 and the hidden bit in two games are identical, we obtain:

$$\mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}M}}_{\mathsf{CE},\mathcal{B}}(k) = \mathbf{Adv}^{\mathsf{Game}_1}_{\mathsf{CLE},\mathcal{A}}(k).$$

Conversely suppose that there is an adversary $\mathcal{A}$ against the certified scheme in the $\mathsf{mCE}\text{-}\mathcal{F}\text{-}\mathsf{CCA}\text{-}\mathsf{I}^-$ model. We construct an adversary $\mathcal{B}$ against the underlying CLE scheme in the multi-user $\mathsf{mCL\text{-}CCA\text{-}I}^\star$ model as follows.

- Algorithm $\mathcal{B}$ obtains (I, Mpk) and passes it onto $\mathcal{A}$.
- When $\mathcal{A}$ invokes the user ID with label L. Then
  - If $\mathsf{L} = \mathsf{honest}$: Algorithm $\mathcal{B}$ calls ReqPK on ID to obtain PK and returns $(\mathsf{ID}, \mathsf{PK})$ to $\mathcal{A}$. Note that $\mathcal{B}$ can register multiple public keys for ID in the multi-user setting.
  - If $\mathsf{L} = \mathsf{corrupt}$: Algorithm $\mathcal{B}$ calls the partial private key extraction on ID to obtain D. This will enable $\mathcal{B}$ to simulate a run of the registration protocol with $\mathcal{A}$. When $(\mathsf{ID}, \mathsf{PK})$ is obtained.
- When $\mathcal{A}$ calls its left-right encryption oracle on $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}, \mathsf{PK})$, if the current public key of ID is PK, algorithm $\mathcal{B}$ calls its left-right oracle on $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID})$. If the current public key of ID is not PK, it first replaces the public key of ID with the provided PK and then performs the left-right encryption call. The resulting ciphertext is returned to $\mathcal{A}$.

  Note that in the weakened certified model, no encryption queries on $(\mathsf{ID}, \mathsf{PK})$ not registered and ID corrupted are allowed. Hence at this step, $\mathcal{B}$ never performs a left-right query involving an identity whose partial private key was (or will be) extracted. This is according to the rules of the type $\mathsf{I}^\star$ game.
- Suppose $\mathcal{A}$ places a decryption query on $(f, \mathsf{c}, \mathsf{ID}, \mathsf{PK})$. By the rules of the certified game, $(\mathsf{ID}, \mathsf{PK})$ must correspond to an honestly registered user.
  - Suppose $f = \mathrm{Id}$. Algorithm $\mathcal{A}$ simply its decryption oracle on $(\mathsf{fake}, \mathsf{c}, \mathsf{ID}, \mathsf{PK})$ and returns the result to $\mathcal{A}$.
  - Suppose a function $f \neq \mathrm{Id}$ with the description SK is provided. Algorithm $\mathcal{B}$ places the query $\mathsf{ReplacePK}(\mathsf{ID}, \mathsf{PK}, \mathsf{SK})$ first and then calls $\mathsf{Decrypt}$ on $(\mathsf{fake}, \mathsf{c}, \mathsf{ID}, \mathsf{PK})$. It returns the answer to $\mathcal{A}$.
- When $\mathcal{A}$ outputs a bit b, algorithm $\mathcal{B}$ also outputs this as his own guess.

This simulation shows that:

$$\mathbf{Adv}^{\mathsf{mCL\text{-}CCA\text{-}I}^\star}_{\mathsf{CLE},\mathcal{B}}(k) = \mathbf{Adv}^{\mathsf{mCE}\text{-}\mathcal{F}\text{-}\mathsf{CCA}\text{-}\mathsf{I}^-}_{\mathsf{CE},\mathcal{A}}(k).$$

Finally, suppose that there is an adversary $\mathcal{A}$ against the certified scheme in the mCE-CCA-M model. An adversary $\mathcal{B}$ against the underlying CLE scheme in the multi-user mCL-CCA-M model is constructed as follows.

- It obtains I and passes it onto $\mathcal{A}$.
- When $\mathcal{A}$ outputs Mpk, algorithm $\mathcal{B}$ also outputs this in its own game environment.
- When $\mathcal{A}$ invokes the user ID with label L = honest (note that no corrupt user registration queries are allowed), algorithm $\mathcal{B}$ sends ID to $\mathcal{A}$ to obtain a D. It then queries the public key extraction oracle on (ID, D). When it receives PK, it returns (ID, PK) to $\mathcal{A}$. Note that in the multi-user model, $\mathcal{B}$ is able to associate multiple public keys to users.
- When $\mathcal{A}$ calls its left-right encryption oracle on $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}, \mathsf{PK})$, we know that (ID, PK) corresponds to an honest registered user (and hence no public key replacement is need). Algorithm $\mathcal{B}$ calls its left-right oracle on $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ID}, \mathsf{PK})$ and returns the result to $\mathcal{A}$.
- Suppose $\mathcal{A}$ places a decryption query on $(\mathsf{c}, \mathsf{ID}, \mathsf{PK})$. By the rules of the certified game this (ID, PK) must correspond to an honestly registered user. Algorithm $\mathcal{A}$ calls its decryption oracle on $(\mathsf{real}, \mathsf{c}, \mathsf{ID}, \mathsf{PK})$ and returns the result to $\mathcal{A}$.
- When $\mathcal{A}$ outputs a bit b, algorithm $\mathcal{B}$ also outputs this as his own guess.

It follows that:

$$\mathbf{Adv}^{\mathsf{mCL\text{-}CCA\text{-}M}}_{\mathsf{CLE},\mathcal{B}}(k) = \mathbf{Adv}^{\mathsf{mCE\text{-}CCA\text{-}M}}_{\mathsf{CE},\mathcal{A}}(k).$$

□

F Extensions to Alternative CLE Definitions

Cheng and Comley [9] simplify the primitive definition of certificateless encryption by eliminating the full secret key extraction algorithm and integrating it into the decryption algorithm. In this primitive definition, the decryption algorithm takes a secret value SK and a partial private key D. The authors modify the security model to accommodate for these changes as follows.

- The full secret keys S are no longer stored in the Real and Fake lists as they no longer exists. The ExtractFSK oracle is also eliminated.
- To enable secret key extraction, a new oracle ExtractSK is added to both type I and type M models. This oracle on input an identity ID returns the SK component from the Real list. It is required that this oracle and the partial private key extraction oracle (in type I) are not called simultaneously on the challenge identity.

A CLE scheme according to this primitive definition can also be transformed into a certified encryption scheme via a CC-2-CE transformation. This transformation is identical to the one we have given for the Al-Riyami–Paterson setting modulo the obvious change that S's no longer exist. This transformation has similar security-preserving properties to those given for CLE-2-CE transformation defined in Section 4. In fact, this result is expected. As discussed in [11], the two formulations are equivalent in the sense that 1) a scheme in one setting can be recasted as a scheme in the other, and 2) type I and type M security are preserved under this operation. A small detail needs to be taken care of: we have used type $\mathsf{I}^\star$, rather than type I model, in our theorem. It can be checked that this model is also equivalent in the two settings.

Baek, Safavi-Naini and Susilo [3] define the certificateless encryption primitive in an alternative way. This new definition was designed to enable them to construct CLE schemes which do not rely on pairings. Furthermore, as noted in [11], with this primitive definition (but neither with the Cheng–Comley nor with the Al-Riyami–Paterson definitions) one can devise CLE schemes which can resist the so-called Denial-of-Decryption attacks [16].

Let us briefly mention how our results extend to this setting. The primitive definition is the same as the Cheng–Comley definition, except the following change: the user key-generation algorithm can depend on the partial private key D:

4. $\mathbb{U}_{\mathsf{CLE}}(\mathsf{ID}, \mathsf{D}, \mathsf{Mpk})$. A probabilistic algorithm for *user key-generation* which takes an identity, a partial private key and the master public key and outputs the full secret key SK and a public key PK.

The security models is modified accordingly.

- The SK on the Real and Fake lists now indicate the *full* private key of users.
- $\mathsf{ReplacePK}$ only accepts tuples of the form $(\mathsf{ID}, \mathsf{PK}, \perp)$ as the concept of secret *value* no longer exists.
- The $\mathsf{ExtractFSK}$ oracle now returns the full secret key SK of the users.
- Consistently with the above, the $\mathsf{Decrypt}$ oracle cannot be called with $\mathsf{L} = \mathsf{fake}$.

We may also define a $\mathsf{BSS\text{-}2\text{-}CE}$ transformation which converts any such CLE scheme to a certified encryption scheme. This transformation is similar to the one we have given above modulo the fact that the $\mathbb{U}_{\mathsf{CE}}$ algorithm runs the new user key-generation algorithm $\mathbb{U}_{\mathsf{CLE}}$. We may derive analogues results to those given in Theorem 1 for this transformation too. However, note that we no longer need to $\mathcal{F}$-extend the model as $\mathsf{Decrypt}$ queries with $\mathsf{L} = \mathsf{fake}$ are no longer possible. Therefore the security models in this and the certified setting are more closely related.