A Survey of Certificateless Encryption Schemes and Security Models

Alexander W. Dent

Information Security Group, Royal Holloway,
Egham Hill, Egham, Surrey, U.K.
a.dent@rhul.ac.uk

Abstract. In this paper we survey the different security models that have been proposed for certificateless encryption, comment on their practical significance, and propose a consistent, new nomenclature for these models. We also survey all known certificateless encryption schemes and point out that there are no known certificateless encryption schemes that achieve the highest levels of security without using the random oracle methodology. Lastly, we discuss the difficulties in proving the security of a certificateless scheme in the standard model, and propose possible ways of finding a solution to this problem.

1 Introduction

In 1978, Rivest, Shamir and Adleman [23] proposed the first public-key encryption scheme. This scheme was a concrete realisation of a seemingly paradoxical conjecture of Diffie and Hellman [15]: that it was possible for an entity (the sender) to securely send another entity (the receiver) a message without these two entities having a pre-existing shared secret key, without any online contact between them, and even without the receiver knowing that they were about to receive a message. This functionality is achieved by generating a pair of keys instead of just one: a public key that is widely distributed for encryption, and a related private key that is kept secret and used for decryption.

History, though, has shown that public key encryption has significant practical problems. In particular, the sender has to be sure that the public key that they have is the correct public key for the receiver. Hence, we require public key infrastructures — a series of trusted third parties that can be relied upon to check a receiver's identity and vouch for the connection between that identity and a particular public key. Public key management is the most costly, cumbersome and inefficient part of any framework that makes use of public key cryptography.

One way of avoiding the tiresome need for a public key infrastructure is to use an identity-based encryption scheme [25]. The most a public key infrastructure can do is to confirm a link between a digital identity and a public key, where a digital identity is some bitstring that uniquely identifies the user in some context. An identity-based encryption scheme removes the need for a public key infrastructure by setting an entity's public key to be equal to their digital

identity. Of course, in such a situation, an entity cannot be expected to compute their own private key; hence, there must exist a trusted third party who initially sets up the system and uses their secret knowledge of the system to compute private keys for other entities.

Identity-based encryption seems to remove the need for a public key infrastructure, replacing it with the need for a key generation centre that computes a user's private key for them. This is more efficient, but has a significant disadvantage too. The fact that the trusted third party computes the private decryption keys for a particular entity means that that trusted third party can decrypt all messages sent to that entity. Thus, an honest-but-curious key generation centre can read the messages of every user in the system. There also significant practical problems associated with identity-based encryption, including the problem of handling key revocation.

In 2003, Al-Riyami and Paterson proposed a new type of encryption scheme that avoids the drawbacks of both traditional public-key encryption and identity-based encryption [2]. They termed this new type of encryption *certificateless public-key encryption* (CL-PKE) because their encryption scheme did not require a public key infrastructure. Roughly speaking, their idea was to combine the functionality of a public key scheme with that of an identity based scheme. Hence, to encrypt a message, a sender requires both the receiver's identity and a public key value produced by the receiver. Similarly, to decrypt a ciphertext, a receiver requires the partial private key corresponding to their identity (which is given to them by a key generation centre) and the private key corresponding to the distributed public key.

This paper surveys and extends the known results about certificateless encryption schemes. In particular, we will propose a new nomenclature that can be used to clarify the contradictory definitions that are currently prominent in the area, survey the known certificateless encryption schemes, and discuss the issues relating to the difficulty of proving the security of a certificateless encryption scheme in the standard model.

2 Security Models For CL-PKE

This section will examine the various security models proposed for a certificateless public-key encryption scheme. This has been an area of some controversy. As we shall see, the original security definitions proposed by Al-Riyami and Paterson [1, 2] are very strong, and have been criticised for not realistically reflecting an attacker's capabilities. Furthermore, we suggest that Al-Riyami and Paterson's security definitions are not consistent in their strength, i.e. that a certificateless scheme is held to a higher standard of security with regard to Type I attackers, than the standard to which it is held with regard to Type II attackers. We then survey the most common relaxations of Al-Riyami and Paterson's security notions and propose a new nomenclature.

2.1 CL-PKE Schemes

The definition of a certificateless public-key encryption scheme was introduced by Al-Riyami and Paterson [1, 2]. A certificateless public-key encryption scheme is defined by seven probabilistic, polynomial-time algorithms:

- `Setup`: This algorithm takes as input a security parameter 1^k and returns the master private key msk and the master public key mpk. This algorithm is run by a KGC in order to initially set up a certificateless system.
- `Extract-Partial-Private-Key`: This algorithm takes as input the master public key mpk, the master private key msk, and identifier $ID \in \{0,1\}^*$. It outputs a partial private private key d_{ID}. This algorithm is run by a KGC once for each user, and the corresponding partial private key is distributed to that user in a suitably secure manner.
- `Set-Secret-Value`: This algorithm takes as input the master public key mpk and an entity's identifier ID as input, and outputs a secret value $x_{ID} \in \mathcal{S}$ for that identity. This algorithm is run once by the user. Note that the secret value space $\mathcal{S}$ is somehow defined by the master public key mpk and an entity's identity ID.
- `Set-Private-Key`: This algorithm takes as input the master public key mpk, an entity's partial private key d_{ID} and an entity's secret value $x_{ID} \in \mathcal{S}$. It outputs the full private key sk_{ID} for that user. This algorithm is run once by the user.
- `Set-Public-Key`: This algorithm takes as input the master public key mpk and an entity's secret value $x_{ID} \in \mathcal{S}$. It output a public key $pk_{ID} \in \mathcal{PK}$ for that user. This algorithm is run once by the user and the resulting public key is widely and freely distributed. The public-key space $\mathcal{PK}$ for a particular user is defined by the master public key mpk and the user's identity ID.
- `Encrypt`: This algorithm takes as input the master public key mpk, a user's identity ID, a user's public key $pk_{ID} \in \mathcal{PK}$ and a message $m \in \mathcal{M}$. It outputs either a ciphertext $C \in \mathcal{C}$ or the error symbol $\perp$. Note that the message space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$ are somehow defined by a combination of the master public key mpk, the user's public key pk_{ID} and the user's identity ID.
- `Decrypt`: This algorithm takes as input the master public key mpk, a user's private key sk_{ID} and a ciphertext $C \in \mathcal{C}$. It returns either a message $m \in \mathcal{M}$ or the error symbol $\perp$.

A certificateless public-key encryption scheme allows anybody to encrypt a message for a particular receiver using publicly available information (in exactly the same way as a traditional public-key encryption scheme or an identity-based encryption scheme). However, unlike a traditional public-key encryption scheme, no certificates are needed. This is because an attacker who publishes a false public key pk_{ID} for an identity will still not be able to decrypt messages encrypted under that public key, because the key generation centre will not release the partial public key d_{ID} to the attacker and so the attacker will not be able to compute the full private key.

This functionality is also given by identity-based cryptography, but identity-based encryption schemes have the disadvantage that the key generation centre can always decrypt messages. A certificateless public-key encryption scheme does not have this disadvantage. An honest-but-curious KGC can always compute an identity's partial private key d_{ID}, but since they will not know the secret value x_{ID} associated with that entity's public key pk_{ID}, they will not be able to form the full private key either. Of course, a malicious KGC that masquerades as a user by publishing a false public key can still break the security of the system; however, it is unclear how to prevent this threat and we will not consider it any further.

An equivalent method for constructing public keys

We may replace the `Set-Secret-Value` and `Set-Public-Key` algorithms with a single `Set-Public-Key` algorithm that works as follows:

- `Set-User-Keys`: This algorithm takes as input the master public key mpk and an entity's identifier ID as input, and outputs a secret value x_{ID} and a public key pk_{ID} for that identity. This algorithm is run once by the user.

This is functionally equivalent formulation to the formulation given by Al-Riyami and Paterson. It is easy to see that a scheme present in the original way can also be presented in this new form: the new `Set-User-Keys` algorithm is defined to be the algorithm that executes the original `Set-Secret-Value` algorithm and `Set-Public-Key` algorithm.

A little bit of thought is required to show that a certificateless scheme presented in this new formulation can also be presented in the old formulation. Suppose that the `Set-User-Keys` algorithm takes as input $p(k)$ random bits. We define the `Set-Secret-Value` algorithm to be the algorithm that outputs a set of $p(k)$ random bits, x'_{ID}. The `Set-Public-Key` algorithm is defined to be the algorithm takes the random bits x'_{ID} as input, runs `Set-User-Keys` to determine a public/private key pair, and outputs the public key. Similarly, `Set-Private-Key` is defined to be the algorithm that runs `Set-User-Keys` using the random bits x'_{ID} to determine the 'proper' secret value x_{ID} and then uses this to create a private key in the normal way.

The Baek, Safavi-Naini and Susilo formulation

The only significant departure from the Al-Riyami and Paterson formulation of a certficateless encryption scheme was proposed by Baek, Safavi-Naini and Susilo [4]. In their model, a public key can only be computed after a partial private key has been obtained[1]. In other words, they make a slight change to the `Set-Public-Key` algorithm:

[1] Technically, the Baek *et al.* model proposed that the KGC return a partial public key (used to help compute the full public key) and a partial private key (used to help compute the full private key). However, the security model they provided is equivalent to the one given here.

– `Set-Public-Key`: This algorithm takes as input the master public key mpk, an entity's partial private key d_{ID} and secret value x_{ID}. It outputs a public key pk_{ID} for that user. This algorithm is run once by the user and the resulting public key is widely distributed.

This slight change (along with the corresponding changes to the security models) allows Baek, Safavi-Naini and Susilo to prove the security of a certificateless encryption scheme based on the CDH problem alone (i.e. without requiring elliptic curve pairings for either the scheme or the security proof). This is the only known scheme that achieves full security and does not require a pairing. The only slight drawback of this formulation is that it does not allow messages to be encrypted "into the future". Under the Al-Riyami and Paterson formulation, an entity may publish a public key pk_{ID} without necessarily knowing the partial private key, and therefore they may receive messages that they cannot decrypt until the KGC releases the partial private key to them. However, the Baek, Safavi-Naini and Susilo formulation requires that the entity has obtained the partial private key before they can release their full public key; hence, an entity that releases a public key must necessarily have enough information to compute the full private key.

Since the Al-Riyami and Paterson formulation is more prevalent, we will continue to use it in this paper; however, we expect all of our results to apply to the Baek, Safavi-Naini and Susilo formulation too.

2.2 The General Security Model

The security of a certificateless encryption scheme is expressed by two (very similar) games. In this section we will describe a basic framework. In both cases, an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is trying to break the IND-CCA2 security, the formal model describing confidentiality. The game runs as follows:

1. The challenger generates a master key pair $(mpk, msk) = \texttt{Setup}(1^k)$.
2. The attacker executes $\mathcal{A}_1$ on mpk and (possibly) some extra information aux. During its execution $\mathcal{A}_1$ may have access to certain oracles (described subsequently). $\mathcal{A}_1$ terminates by outputting an identity ID^*, two messages of equal length (m_0, m_1), and some state information $state$.
3. The challenger randomly chooses a bit $b \in \{0, 1\}$ and computes the challenge ciphertext $C^* = \texttt{Encrypt}(mpk, ID^*, pk_{ID^*}, m_b)$ using the value of pk_{ID^*} currently associated with the identity ID^*. If the public key pk_{ID^*} does not exist, then the challenger computes a public key pk_{ID^*} for ID^* by running the `Set-Secret-Value` and `Set-Public-Key` algorithms.
4. The attacker executes $\mathcal{A}_2$ on the input $(C^*, state)$. During its execution $\mathcal{A}_2$ may have access to certain oracles (described subsequently). $\mathcal{A}_2$ terminates by outputting a guess b' for b.

The attacker wins the game if $b = b'$ and its advantage is defined to be:

$$|Pr[b = b'] - 1/2| \tag{1}$$

It now remains to define the oracles that the attacker may have access to:

- **Request Public Key**: The attacker supplies an identity ID and the challenger responds with the public key pk_{ID} for ID. If the identity ID has no associated public key, then the challenger generates a public key for ID by running `Set-Public-Key` and `Set-Secret-Value` (if necessary).
- **Replace Public Key**: This oracle models the attacker's ability to convince a legitimate user to use an invalid public key. This can happen because public keys are no longer verified by a trusted third party, and a user may be given a false public key by an attacker and believe it to be correct. The attacker supplies an identity ID and a valid public key value $pk_{ID} \in \mathcal{PK}$, and the challenger replaces the current public key value with the value pk_{ID}.
- **Extract Partial Private Key**: The attacker supplies an identity ID and the challenger responds with the partial private key d_{ID}. If the identity has no partial private key, then the challenger generates a partial private key by running `Extract-Partial-Private-Key` on ID using msk.
- **Extract Private Key**: The attacker supplies an identity ID and the challenger responds with the private key sk_{ID}. If the identity has no associated private key, then the challenger generates a private key using `Set-Private-Key` (after running `Set-Secret-Value` and `Extract-Partial-Private-Key` if necessary).

An attacker may also have access to one or more different types of decryption oracle:

- **Strong Decrypt**: The attacker supplies an identity ID and a ciphertext C, and the challenger responds with the decryption of C under the private key sk_{ID}. Note that if the attacker has replaced the public key for ID, then this oracle should return the correct decryption of C using the private key that is associated with the current public key pk_{ID} (or $\perp$ if no such private key exists).
- **Weak Decrypt**: The attacker supplies an identity ID, a secret value $x_{ID} \in \mathcal{S}$, and a ciphertext C. The challenger computes that full private key sk_{ID} for the identity from the (correct) partial private key d_{ID} and the supplied secret value x_{ID}; then returns the decryption of C under this private key sk_{ID}. If either process fails, then the oracle returns $\perp$. Note that this functionality can be achieved by a strong decryption oracle.
- **Decrypt**: The attacker supplies an identity ID and a ciphertext C, and the challenger responds with the decryption of C under the original private key sk_{ID} for ID. Note that this functionality can be achieved by a strong decryption oracle.

A certificateless scheme is proven secure by showing that any attacker attempting to break the scheme only has a negligible chance of success.

Definition 1 (Negligible Function). *A function $f : \mathbb{N} \to \mathbb{R}$ is said to be negligible if, for all polynomial p, there exists an integer $N(p)$ such that $|f(x)| \leq 1/|p(x)|$ for all $x \geq N(p)$.*

2.3 Type I Attackers

The Type I security model is designed to protect against a third party attacker (i.e. anyone except the legitimate receiver or the KGC) who is trying to gain some information about a message from its encryption. There has been some debate about how to precisely formulate this notion and we survey the main attempts in this section. We also comment on their correctness, and provide a new and consistent nomenclature for the different notions.

Strong Type I Security

The original definition proposed by Al-Riyami and Paterson is as follows.

Definition 2. *A certificateless encryption scheme is Strong Type I secure if every probabilistic, polynomial-time attacker $\mathcal{A}^I = (\mathcal{A}_1^I, \mathcal{A}_2^I)$ has negligible advantage in winning the IND-CCA2 game subject to the following oracle constraints:*

- *$\mathcal{A}^I$ cannot extract the private key for the challenge identity ID^* at any time,*
- *$\mathcal{A}^I$ cannot extract the private key of any identity for which it has replaced the public key,*
- *$\mathcal{A}^I$ cannot extract the partial private key of ID^* if $\mathcal{A}^I$ replaced the public key pk_{ID^*} before the challenge was issued,*
- *$\mathcal{A}_2^I$ cannot query the strong decrypt oracle on the challenge ciphertext C^* for the identity ID^* unless the public key pk_{ID^*} used to create the challenge ciphertext has been replaced,*
- *$\mathcal{A}^I$ cannot query the weak decrypt or decrypt oracles (although this functionality can be given by the strong decrypt oracle).*

In this model, the attacker is given no extra information, i.e. aux is the empty bit-string.

This model gives as much power as possible to the attacker. Its only restrictions are those necessary to prevent attacks that would trivially allow the attacker to win, and to prevent the attacker from asking for the private key of a user with a replaced public key. This latter restriction was only made because it was considered too difficult to achieve a notion of security that doesn't have this restriction.

It should be noted that the model expects the challenger to be able to correctly respond to decryption queries made on identities for which the attacker has replaced the public key. This is a very strong notion of security and it is unclear whether it represents a realistic attack scenario. In general, decryption oracles are provided to an attacker to model the fact that the attacker may be able to gain some information from a legitimate receiver about the decryptions of some ciphertexts (for example, by bribing the legitimate receiver to give up the message or by deducing whether a ciphertext is a valid encryption of a particular message by observing the legitimate receiver's behaviour after receiving the ciphertext). This situation cannot happen if we replace a public key: when we replace a public key, we are duping a *sender* into encrypting a message using

a false public key that the receiver has not published. Under no circumstances will the receiver then attempt to decrypt that ciphertext using the private key corresponding to that replaced public key. Hence, providing a decryption oracle that will accurately decrypt ciphertexts encrypted under the replaced public key gives the attacker more power than it would have in practice.

This represents an interesting philosophical question in the construction of security models: do we give the attacker as much power as is possible (perhaps subject to the restriction that we must still be able to construct secure certificateless encryption schemes)? Or should the model only try to reflect a realistic attacker's abilities? The former approach leads to strong security models, and potentially more complex schemes. The latter approach may lead to more efficient schemes, but a scheme's security can only be guaranteed if an attacker's abilities have been correctly modelled.

Weak Type Ia Security

Several authors have judged Al-Riyami and Paterson's Type I security model to be too strong and proposed weaker versions. We will consider each of the major alternatives in turn. The strongest of these definitions, which we will term Weak Type Ia security, has been put forward by Bentahar *et al.* [9].

Definition 3. *A certificateless encryption scheme is Weak Type Ia secure if every probabilistic, polynomial-time attacker $\mathcal{A}^I$ has negligible advantage in winning the IND-CCA2 game subject to the following oracle constraints:*

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

In this model, the attacker is given no extra information, i.e. aux is the empty bit-string.

It should be noted that the original notion of Weak Type Ia security [9] did not give the attacker to the ability to request decryptions using the original private key value *after* the public key had been replaced. We make this small change to make the model more realistic. It does not affect the results presented by Bentahar *et al.*

The Weak Type Ia model seems to most realistically reflect the potential abilities of an attacker. The attacker can replace public keys with arbitrary values of its choice, thus allowing for senders to be duped, but the attacker can still ask a legitimate receiver to decrypt any ciphertext with his original private key value (using the decrypt oracle). Furthermore, the attacker may be able to dupe a legitimate receiver into changing his public key and secret value to that

of the attacker's choice (using a combination of the replace public key oracle and the weak decrypt oracle), and so obtain decryptions using private keys formed from arbitrary secret values.

Weak Type Ib Security

A weakening of this model gives Weak Type Ib security [30]:

Definition 4. *A certificateless encryption scheme is Weak Type Ib secure if every probabilistic, polynomial-time attacker $\mathcal{A}^I$ has negligible advantage in winning the IND-CCA2 game subject to the following oracle constraints:*

- *$\mathcal{A}^I$ cannot extract the private key for the challenge identity ID^* at any time,*
- *$\mathcal{A}^I$ cannot extract the private key of any identity for which it has replaced the public key,*
- *$\mathcal{A}^I$ cannot extract the partial private key of ID^* if $\mathcal{A}^I$ replaced the public key pk_{ID^*} before the challenge was issued,*
- *$\mathcal{A}^I$ cannot query the strong decrypt or weak decrypt oracles,*
- *$\mathcal{A}_2^I$ cannot query the decrypt oracle on the challenge ciphertext C^* and the identity ID^*.*

In this model, the attacker is given no extra information, i.e. aux is the empty bit-string.

In this model, the attacker can replace public keys (i.e. dupe senders) and can ask for decryptions of ciphertexts using the original private key values (using the decrypt oracle), but cannot dupe a recipient into decrypting messages using a secret value chosen by the attacker. This reflects security in a situation where users generate their public key values correctly (i.e. by using the `Set-Secret-Value` and `Set-Public-Key` algorithms) and never change their public key values once they are set.

Another interpretation of the difference between Weak Type Ia and Weak Type Ib security is based on the implementation of a certificateless scheme in a black-box device. The Weak Type Ib security model guarantees security in the case when the black box is tamper-proof in its generation of the secret value; the Weak Type Ia security model guarantees that the scheme is secure when the black-box can be be forced to re-generate its secret key value and may possible be influenced in the way that this occurs (for example, by side-channel attacks).

Weak Type Ic Security

Lastly, mostly for comparison with Type II attackers, we present a final weak notion of security. This model of security was briefly considered in an early version of a paper by Baek and Wang [5]. This notion of security can be achieved by a public-key encryption scheme alone.

Definition 5. *A certificateless encryption scheme is Weak Type Ic secure if every probabilistic, polynomial-time attacker $\mathcal{A}^I$ has negligible advantage in winning the IND-CCA2 game subject to the following oracle constraints:*

Table 1. A Summary of the Oracle Access Provided to a Type I Attacker

	Request Public Key	Replace Public Key	Extract Partial Private Key	Strong Decrypt	Weak Decrypt	Decrypt
Strong Type I	✓	✓	✓	✓		
Weak Type Ia	✓	✓	✓		✓	✓
Weak Type Ib	✓	✓	✓			✓
Weak Type Ic	✓		✓			✓

- $\mathcal{A}^I$ *cannot replace any public keys at any time,*
- $\mathcal{A}^I$ *cannot extract the private key for the challenge identity* ID^* *at any time,*
- $\mathcal{A}^I$ *cannot query the strong decrypt or weak decrypt oracles,*
- $\mathcal{A}_2^I$ *cannot decrypt the challenge ciphertext* C^* *for the identity* ID^**.*

In this model, the attacker is given no extra information, i.e. aux is the empty bit-string.

The different types of oracle access that these models give to an attacker is summarised in Table 1. It is simple to deduce the following relationships between the different notions of Type I security:

$$\text{Strong Type I} \Rightarrow \text{Weak Type Ia} \Rightarrow \text{Weak Type Ib} \Rightarrow \text{Weak Type Ic}$$

where $A \Rightarrow B$ if any scheme that is A secure must necessarily be B secure.

The partial private key of the challenge identity

There are some further variations on these security models. Several schemes are proven secure in a weakened model in which a Type I attacker is not allowed to query the partial private key extraction oracle on the challenge identity ID^*. We denote these models with an asterisk; for example, the Weak Type Ib* model is exactly the same as the Weak Type Ib security model except that the attacker is not allowed to query the partial private key extraction oracle on the challenge identity.

This security model also has a natural interpretation. It assumes that the attacker is unable to get hold of an identity's partial private keys except in specialised cases (for example, for the attacker's own identity or where an entity has been completed corrupted). This can be achieved by a system in which the KGC delivers the partial private key through some confidential channel. This is not an unreasonable assumption, given that the security models assume that an entity receives its partial private key through an integrity protected channel. Of course, this means that the KGC must be especially vigilant in verifying the identities of users before issuing partial private keys.

Cheng and Comley Oracles

Cheng and Comley [11] have also proposed a variation on the above security models. In the Cheng and Comley variation, the attacker is not given access to an Extract Private Key oracle, but is instead given access to a Extract Secret Value oracle:

– **Extract Secret Value**: The attacker supplies an identity ID and the challenger responds with the secret value x_{ID} associated with that entity. If the identity has no associated secret value, then the challenger generates one by running `Set-Secret-Value`.

The attacker may not query both the Extract Partial Private Key oracle and the Extract Secret Value oracle on the challenge identity. Furthermore, the attacker may not query the Extract Secret Value oracle on any identity for which it has replaced the public key. We denote a security model in which the attacker has access to an Extract Secret Value oracle using a dagger; for example, the Weak Type Ib† model is exactly the same as the Weak Type Ib security model except that the attacker has access to an Extract Secret Value oracle instead of an Extract Private Key oracle.

This change gives rise to slightly more powerful security models. The attacker can simulate an Extract Private Key oracle by making queries to both the Extract Partial Private Key and Extract Secret Value oracles, and then assembling the full private key itself. However, the attacker now has the ability to find out the secret value associated with a public key (and, in particular, the public key associated with the challenge identity). This is not an ability that the attacker is guaranteed to have in the normal security models.

It can be argued that allowing the attacker access to an Extract Secret Value oracle does not reflect reality. In a normal mode of use, a secret value will be used to generate a public/private key pair for an entity and then deleted. In such a scenario, it does not seem likely that an attacker will be able to extract the secret value at any point. However, it might be possible for an attacker to persuade a receiver to give up some information about his secret value or to obtain some information about a secret value as it is generated (for example, using some form of side-channel analysis). Hence, whenever possible, it is prudent to prove the security of a CL-PKE in a model that allows access to an Extract Secret Value oracle, simply to provide a 'margin of error' for the security model.

Extract secret value oracles should definitely be included in any model that attempts to model the situation where certificateless encryption is used to encrypt messages 'into the future'. In such a situation, we have to protect against a curious receiver (who knows his own secret value) who wishes to read a message before being issued his partial private key.

2.4 Type II Attackers

Weak Type II Security

The second security definition states that an honest-but-curious key generation centre should not be able to break the confidentiality of the scheme. Hence, we allow the attacker to have access to master private key by setting $aux = msk$. This means that we do not have to give the attacker explicit access to an extract partial private key oracle, as they are able to compute these value for themselves. The most important point about Type II security is that the

KGC can trivially break the scheme if it is allowed to replace the public key for the challenge identity *before* the challenge is issued. Al-Riyami and Paterson [2] chose to prevent this from occurring by forbidding the KGC from replacing any public keys at all, proposing the following model:

Definition 6. *A certificateless encryption scheme is Weak Type II secure if every probabilistic, polynomial-time attacker* $\mathcal{A}^{II} = (\mathcal{A}_1^{II}, \mathcal{A}_2^{II})$*, which is given the auxiliary information aux = msk, has negligible advantage in winning the IND-CCA2 game subject to the following oracle constraints:*

- $\mathcal{A}^{II}$ *cannot extract the private key for the challenge identity* ID^* *at any time,*
- $\mathcal{A}^{II}$ *cannot query the extract partial private key oracle at any time,*
- $\mathcal{A}^{II}$ *cannot replace public keys at any time,*
- $\mathcal{A}^{II}$ *cannot query the strong decrypt or weak decrypt oracles at any time,*
- $\mathcal{A}_2^{II}$ *cannot query the decrypt oracle on the challenge ciphertext* C^* *and the identity* ID^**.*

This roughly corresponds to the weakest notion of security proposed for Type I attackers, and it is easy to see that any scheme that is Weak Type II secure is necessarily Weak Type Ic secure. Furthermore, this notion of security can be achieved by a public key encryption scheme alone.

Strong Type II Security

By denying the KGC the ability to replace public keys or query more powerful decryption oracles, we might be denying it the ability to perform certain attacks that might occur in practice, and we are certainly not providing it with the huge level of pwer provided to a Strong Type I attacker. Hence, we should consider whether the KGC gains any advantages if we allow it to replace public keys (subject to the restriction that it cannot replace the public key of the challenge identity until after the challenge has been issued) or allow it access to more powerful decryption oracles.

Clearly, if we do not give the attacker access to a strong decryption oracle, then the ability to replace public keys is of no use to the attacker. This is because the challenger never gives a response based on a replaced public key value; hence, the attacker gains no advantage by replacing a public key. Similarly, the weak decryption oracle is of no use to an attacker because the attacker can always compute the full private key of a user given their identity *ID* and their secret value x_{ID}. Hence, all of the Weak Type II security models that we might propose (based on the Weak Type I security models) are equivalent.

However, if we follow the principle that we should give the attacker as much power as possible, then there is some merit in considering a Strong Type II security model. This gives an equivalent security level for the scheme against Type II attackers as is demanded for Type I attackers. It is unreasonable to require a scheme to meet the Strong Type I security level, without also requiring to meet the Strong Type II security level.

Definition 7. *A certificateless encryption scheme is Strong Type II secure if every probabilistic, polynomial-time attacker* $\mathcal{A}^{II} = (\mathcal{A}_1^{II}, \mathcal{A}_2^{II})$*, which is given the auxiliary information* $aux = msk$*, has negligible advantage in winning the IND-CCA2 game subject to the following oracle constraints:*

- $\mathcal{A}^{II}$ *cannot extract the private key for the challenge identity* ID^* *at any time,*
- $\mathcal{A}^{II}$ *cannot extract the private key of any identity for which it has replaced the public key,*
- $\mathcal{A}^{II}$ *cannot query the extract partial private key oracle at any time,*
- $\mathcal{A}_1^{II}$ *cannot output a challenge identity* ID^* *for which it has replaced the public key,*
- $\mathcal{A}_2^{II}$ *cannot query the strong decrypt oracle on the challenge ciphertext* C^* *for the identity* ID^* *unless the public key* pk_{ID^*} *used to create the challenge ciphertext has been replaced,*
- $\mathcal{A}^{II}$ *cannot query the weak decrypt or decrypt oracles (although this functionality can be given by the strong decrypt oracle).*

The different types of oracle access that these models give to an attacker is summarised in Table 2.

Table 2. A Summary of the Oracle Access Provided to a Type II Attacker

	Request Public Key	Replace Public Key	Strong Decrypt	Decrypt
Strong Type II	✓	✓	✓	
Weak Type II	✓			✓

3 Surveying Certificateless Encryption Schemes

There have been several attempts at producing secure certificateless encryption schemes, using various models of security, and we will briefly discuss these constructions in this section. The constructions are summarised in Table 3. We differentiate between concrete constructions, which give a full description of a specific certificateless scheme, and generic constructions, which explain how to construct a certificateless scheme from other primitives.

Al-Riyami 1/Yum–Lee 1 The Al-Riyami 1/Yum-Lee 1 construction [1, 30] is a generic construction of a certificateless encryption scheme from an IND-CCA2 secure public-key encryption scheme and an IND-CCA2 secure identity-based encryption scheme. It is a sequential construction, in which the full ciphertext is formed by first encrypting the message with the public-key encryption scheme, and then encrypting this ciphertext with the identity-based encryption scheme. Its security proof uses a weakened version of the Weak Type Ib model in which the attacker cannot query the partial key extraction oracle on the challenge identity ID^*. Libert and Quisquater [20] show that if we allow partial key extraction

Table 3. A Survey of Certificateless Encryption Schemes

Scheme	Reference	Style	Model	Type I Model	Type II Model	Broken?
Al-Riyami 1 / Yum–Lee 1	[1, 30]	Generic		Weak Type Ib*	Weak Type II	[19, 20]
Al-Riyami 2	[1]	Generic		No proof given	No proof given	Sec. 3
Al-Riyami 3	[1]	Generic		No proof given	No proof given	[20]
Al-Riyami–Paterson 1	[1, 2]	Concrete	ROM	Strong Type I	Weak Type II	
Al-Riyami–Paterson 2	[1, 3]	Concrete	ROM	Strong Type I	Weak Type II	[20, 32]
Cheng–Comley	[11]	Concrete	ROM	Weak Type Ib*†	Weak Type II	
Baek–Safavi-Naini–Susilo	[4]	Concrete	ROM	Strong Type I*	Weak Type II	
Bentahar–Farshim–Malone-Lee–Smart	[9]	Generic	ROM	Weak Type Ia	Weak Type II	
Libert–Quisquater 1	[20]	Generic	ROM	Strong Type I	Weak Type II	
Libert–Quisquater 2	[20]	Generic	ROM	Strong Type I	Weak Type II	
Libert–Quisquater 3	[20]	Generic	ROM	Strong Type I	Weak Type II	
Libert–Quisquater 4	[20]	Concrete	ROM	Strong Type I	Weak Type II	
Liu–Au	[21]	Concrete		Weak Type Ib	Weak Type II	
Shi–Li	[27]	Concrete	ROM	Strong Type I	Weak Type II	
Yum–Lee 2	[31]	Generic		Weak Type Ia*	Weak Type II	[19]
Zhang–Feng	[32]	Concrete		No proof given	No proof given	

* = The attacker is not allowed to query the partial private key extraction oracle on the challenge identity
† = The attacker has access to a secret value extraction oracle instead of a private key extraction oracle

oracle queries on the identity ID^*, then the scheme can be broken. Galindo, Morillo and Ràfols [19] demonstrated that similar techniques can be used to show that the scheme is not Weak Type II secure.

Al-Riyami 2 The second Al-Riyami scheme is the "opposite" of the Al-Riyami 1/Yum–Lee 1 scheme, i.e. one composes an identity-based encryption scheme and a public-key encryption scheme by first applying the identity-based scheme and then applying the public-key encryption scheme. Al-Riyami does not offer a security proof for this scheme. Libert and Quisquater [20] note that this scheme is not Strong Type I secure; however, the scheme has greater security problems. The scheme is actually insecure in the Weak Type Ib model. An attacker can break the scheme by initially replacing the public key with one for which the corresponding private key is known. After receiving the challenge ciphertext, the attacker decrypts the public-key portion of the ciphertext using the (known) private key. This leaves the portion of the ciphertext encrypted using the identity-based encryption scheme. The attacker re-encrypts this ciphertext using the original public key value and submits this new ciphertext to a decryption oracle, which returns the challenge message.

Al-Riyami 3 The third Al-Riyami construction uses an identity-based encryption scheme and a public-key encryption scheme in parallel. The identity-based encryption scheme and the public-key encryption scheme are used to encrypt shares of the message which (if both shares are recovered) can be used to reconstruct the message. Al-Riyami does not offer a security proof for this scheme, and the scheme has now been broken by Libert and Quisquater [20].

Al-Riyami–Paterson 1–2 These schemes are concrete and require an elliptic curve pairing. The second scheme was independently broken by Libert and Quisquater [20], and Zhang and Feng [32]. It can be repaired by applying the certificateless Fujisaki-Okamoto [20] or by the techniques of Zhang and Feng [32]. There are no known attacks against the first scheme.

Baek–Safavi-Naini–Susilo The Baek–Safavi-Naini–Susilo [4] construction is a concrete construction that uses the slightly altered Baek–Safavi-Naini–Susilo formulation of a CL-PKE scheme. It is proven secure in a slightly weakened version of the Strong Type I security model, in which attacker cannot query the partial key extraction oracle on the challenge identity. However, unlike the Yum–Lee 1 construction, it is not clear whether this scheme can be broken if an attacker is able to make partial key extraction queries on the challenge identity. Therefore, we consider its full security to be an open problem. It is the only known scheme which does not require an elliptic curve pairing to be implemented.

Bentahar *et al.* Bentahar *et al.* [9] have two major contributions to the theory of certificateless public-key encryption. The first is to propose a general model for a certificateless KEM–DEM encryption scheme, similar to that proposed for public-key encryption schemes by Cramer and Shoup [12]. This allows for greater flexibility and efficiency when constructing a CL-PKE. Unfortunately,

the nature of their construction is such that only Weak Type Ia security can be achieved, although this can be achieved in the standard model. This should be compared with the generic Fujisaki–Okamoto style construction given by Libert–Quisqauter, which provides Strong Type I security, but only in the random oracle model. The second contribution of Bentahar *et al.* is to provide a generic construction for a CL-PKE KEM from a weakly secure public-key encryption scheme and a weakly secure identity-based encryption scheme. This generic construction uses a form of parallel composition of the public-key and identity-based encryption schemes, and is proven secure in the random oracle model.

Cheng–Comley Cheng–Comley [11] present a concrete CL-PKE construction based on the use of pairings, and is very similar to the Zhang–Feng scheme [32]. The scheme is proven secure in a variant of the Weak Type Ib security model. In particular, Cheng–Comley do not allow the partial public key oracle to be queried on the challenge identity. Furthermore, the attacker does not get access to an oracle which returns an entity's full private key, but instead have access to an oracle which will return the secret value for a (non-replaced) public key. The Cheng–Comley scheme proves full security by applying the Fujisaki–Okamoto transform [18] to a weaker CL-PKE scheme. It is unclear whether this results in a secure encryption scheme or not. However, Libert and Quisquater [20] have proposed a Fujisaki–Okamoto-style transform that converts a weakly secure CL-PKE into a strongly secure CL-PKE, under which the Cheng and Comley scheme appears secure.

Libert–Quisquater 1–3 Libert–Quisquater [20] provide four CL-PKE constructions. The first three of these constructions are generic constructions for a CL-PKE from a public-key encryption scheme and an identity-based encryption scheme. Libert-Quisquater note that, even though the generic sequential and parallel composition constructions (Al-Riyami 1/Yum–Lee 1, Al-Riyami 2 and Al-Riyami 3) are insecure against active attackers, these schemes are secure against passive attackers. Libert-Quisquater obtain three strongly secure CL-PKE schemes by developing a certificateless analogue of the Fujisaki–Okamoto transform [17] and applying this transformation to the three weakly secure certificateless schemes.

Libert–Quisquater 4 The final scheme that Libert–Quisquater [20] propose is a concrete construction based on the Sakai-Kasahara [24] identity-based encryption scheme and an ElGamal-style public-key encryption scheme [16]. As such, it is similar to the Shi–Li CL-PKE scheme, although the Libert–Quisquater scheme is more efficient.

Liu–Au This Liu–Au [21] is based on the the Waters encryption scheme [29] for passive security and the Boneh–Katz conversion for CCA security [10]. The paper also introduces the notion of a *denial of decryption attack* in which an attacker tries to trick a user into encrypting a message using a false public key, i.e. one that has not been produced by a user in possession of an appropriate partial private key. This would solve a key problem in certificateless cryptography: how

to decide which public key to trust when one does not have a trust authority to vouch for the correct one. Obviously, these denial of decryption attacks can only be prevented in a situation where a user receives their partial private key before computing a public key, i.e. in the Baek–Safavi-Naini–Susilo formulation. The authors give one such scheme, but are only abale to prove it secure in a very weak model. Hence, we still consider this to be an open problem.

Shi–Li The Shi–Li scheme [27] is adapted from an earlier scheme by the same authors [26] and based on the Sakai-Kasahara [24] identity-based encryption scheme and an ElGamal-style public-key encryption scheme [16]. As such it is similar to, although slightly less efficient than, the Libert-Quisquater 4 CL-PKE scheme. The current version of the paper also claims to achieve full security by applying the Fujisaki-Okamoto transform to a weaker scheme. Libert and Quisquater [20] have shown the Fujisaki–Okamoto transform is insufficient to guarantee full security; however, by applying the improved Fujisaki–Okamoto–style transform proposed by Libert and Quisquater, full security can still be achieved.

Yum–Lee 2 The second generic construction provided by Yum and Lee [31] is similar to Al-Riyami 1/Yum–Lee 1 in that it is the sequential composition of an encryption scheme controlled by the user and an encryption scheme controlled by the KGC. However, in the first construction, the user's encryption scheme is a public key encryption scheme and the KGC's encryption scheme is an identity-based encryption scheme. In this construction, both encryption schemes are identity-based. The authors claim to prove the security of the scheme in a weakened version of the Weak Type Ia security model, in which the attacker may not query the partial key extraction oracle on the challenge identity, and in the Weak Type II security model. Galindo, Morillo and Ràfols [19] show that it does not achieve security in the Weak Type II model. Similar techniques can be used to show that it does not achieve Type I security in any model in which the attacker is allowed to query the partial key extraction oracle on the challenge identity.

Zhang–Feng In the paper that demonstrates the insecurity of the Al-Riyami–Paterson 2 scheme, Zhang and Feng [32] propose a simple modification to the Al-Riyami–Paterson 2 scheme, which appears to restore the security of the scheme. However, no security proofs are given. The proposed scheme is very similar to the Cheng–Comley scheme [11].

4 On the Difficulty of Achieving Full Security in the Standard Model

A close examination of Table 3 shows that, while there have been CL-PKE schemes proven secure in strong security models using the random oracle methodology, and CL-PKE schemes that have been proven secure in weak models without the random oracle methodology, no-one has yet proven a scheme secure in a strong security model without the random oracle model. Indeed, it has been

suggested by several members of the cryptographic community that strong security proofs in the standard model cannot be achieved because "a Strong Type I decryption simulator is a Weak Type II attacker". In this section, we will investigate this claim, showing that certain proof techniques may not be used to prove the full security of a certificateless public-key encryption scheme in the standard model. After this we discuss possible solutions to the problem of proving full security.

An observational or black-box security proof is one in which an algorithm, called a *solver*, uses the attacker as a subroutine in solving a mathematical problem. The attacker is run as an independent subroutine, with no interference from the programme that is running it, i.e. as a black box. The solver answers all of the attacker's queries using a simulator, which must answer these queries well enough that the attacker cannot distinguish the simulator's responses from authentic ones. The solver computes its output based on observation of the queries that the attacker makes. This is a common format for many security proofs.

Consider the following specific type of black-box proof for the security of a certificateless scheme against a Strong Type I attacker. It should be noted that several assumptions are made as to the structure of the solver in the following proof. We shall discuss the correctness of these assumptions and the relevance of the result after the proof has been presented. We show the structure of the solver in Figure 1. The structure of the solver breaks down as follows:

1. The solver initially receives an instance of a problem that it needs to solve; this could be, for example, an instance of the CDH problem or a RSA problem. The solver executes a key generation algorithm which, based on the problem instance, generates a master public key mpk and possibly, depending on the nature of the solver, a master private key msk. It should be noted that the solver's key generation algorithm is likely to be vastly different to the "proper" key generation algorithm `Setup`! However, it is important that this key pair be indistinguishable from a proper key pair produced by the `Setup` algorithm. The key generation algorithm may also produce some extra information, based on the problem instance, which will be used to compute the challenge ciphertext.
2. The solver executes the attacker $\mathcal{A}_1^I$ on the input mpk. We note that the attacker is completely independent of the solver, i.e. no part of the solver has access to the program code of the attacker nor the attacker's random coins except for the subroutines that executes $\mathcal{A}_1^I$ and $\mathcal{A}_2^I$. Any oracle query made by the $\mathcal{A}_1^I$ should be answered by a simulator Sim_1. This simulator takes the master public key mpk and, if it has been generated, the master private key msk as input. The simulator must answer oracle queries in such a way that no attacker can distinguish (with non-negligible advantage) between oracle access to the simulator and oracle access to the appropriate certificateless algorithms.
3. The attacker $\mathcal{A}_1^I$ terminates by outputting an identity ID^*, two equal-length messages m_0 and m_1, and some state information $state(\mathcal{A}_1^I)$. At this point

the simulator Sim_1 also terminates and outputs some state information $state(Sim_1)$.

4. A challenge ciphertext C^* is then created by a ciphertext creation algorithm. This algorithm takes as input the triple (ID^*, m_0, m_1) output by the attacker $\mathcal{A}_1^I$, the state information $state(\mathcal{A}_1^I)$ output by the simulator Sim_1 and any extra information provided by the key generation algorithm. It outputs both the challenge ciphertext C^* and some state information *state*.
5. The solver then executes $\mathcal{A}_2^I$ on the inputs C^* and $state(\mathcal{A}_1^I)$. Again, this algorithm may make oracle queries. These queries are answered by a simulator Sim_2 which takes the state information *state* provided by the ciphertext creator as input. As before, we require that the simulator answers oracle queries in such a way that no attacker can distinguish (with non-negligible advantage) between oracle access to the simulator and oracle access to the appropriate correct algorithms.
6. The attacker $\mathcal{A}_2^I$ terminates by outputting a guess b' for the bit b. The simulator Sim_2 takes this as a final input and outputs its guess for the solution to the problem instance.

Theorem 1. *If there exists a black-box proof in the above form for the security of a certificateless encryption scheme against Strong Type I attackers, then that certificateless encryption scheme is insecure against Weak Type II attackers.*

Proof Suppose a black-box proof of this form exists for the security of a scheme against Strong Type I attackers, and consider a particular Strong Type I attacker $\mathcal{A}^I = (\mathcal{A}_1^I, \mathcal{A}_2^I)$, in which $\mathcal{A}_1^I$ works as follows:

1. Receive the public key *mpk* from the challenger.
2. Choose an identity *ID*, then randomly generate a secret value x_{ID} for that identity using `Set-Secret-Value` and a public key pk_{ID} for that identity using `Set-Public-Key`.
3. Replace the public key of the identity *ID* with the value pk_{ID}.
4. Choose two distinct messages m_0 and m_1 from the message space of the encryption scheme.
5. Select a bit $b \in \{0, 1\}$ uniformly at random.
6. Compute the encryption C of message m_b using the identity *ID* and the public keys pk_{ID} and *mpk*.
7. Submit C to the decryption oracle.
8. Receive the decryption of C from the decryption oracle.
9. Output (ID, m_0, m_1) and no state information.

The second part of the attacker, $\mathcal{A}_2^I$, is not important to our discussion. Since the simulator must answers oracle queries in a way that cannot be distinguished from the real oracle by any attacker, the simulator must respond to $\mathcal{A}_1^I$'s oracle query with the correct message m_b (as $\mathcal{A}_1^I$ "knows" what the simulator should return) with overwhelming probability. We use this fact to construct a Type II attacker.

Consider a Weak Type II attacker $\mathcal{A}^{II} = (\mathcal{A}_1^{II}, \mathcal{A}_2^{II})$, in which $\mathcal{A}_1^{II}$ runs as follows:

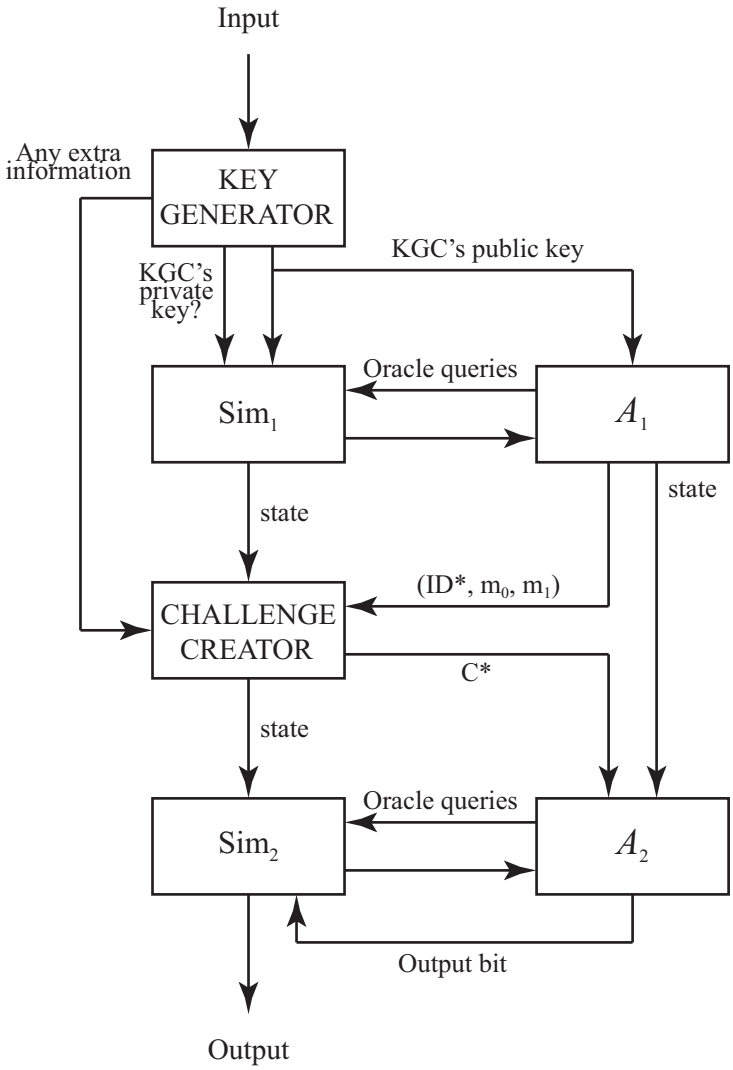

Fig. 1. The Structure of a Solver for a Black Box Proof of Security for a Strong Type I Attacker

1. Receive the public/private key pair (mpk, msk) from the challenger.
2. Chose an identity ID.
3. Choose two distinct messages m_0 and m_1 from the message space of the encryption scheme.
4. Output the triple (ID, m_0, m_1) and the state information (mpk, msk, ID, m_0, m_1).

After receiving this output, the challenger will then generate a random secret value x_{ID} for the identity ID using `Set-Secret-Value` and a public key pk_{ID} for that identity using `Set-Public-Key`. It will then choose a bit $b \in \{0, 1\}$ uniformly at random and compute the encryption C^* of m_b using the identity ID and the public keys pk_{ID} and mpk. $\mathcal{A}_2^{II}$ runs as follows:

1. Receive the challenge ciphertext C^* and the state information (mpk, msk, ID, m_0, m_1) from the challenger.
2. Request the public key pk_{ID} for the identity ID.
3. Execute Sim_1 (initialised using the master public key mpk and, if required, the master private key msk) to decrypt the ciphertext C^* using the identity ID and the public key pk_{ID}. The simulator will return a message m.
4. If $m = m_0$ output 0, otherwise output 1.

Since the key pair generated by the solver's key generation algorithm is indistinguishable from a key pair produced by the `Setup` algorithm, the Type II attacker will undertake the same steps as the Type I attacker up to the point in which it has received the response from the simulator Sim_1. Hence, the simulator will output the correct message $m = m_b$ with overwhelming probability. Thus, the Type II attacker will break the certificateless scheme with overwhelming probability. □

So we must ask the question: is it possible to prove that a certificateless scheme is secure in the standard model? Many people have chosen to interpret the relationship between Strong Type I simulators and Weak Type II attackers as meaning that it is impossible to construct a scheme that is provably secure against both Strong Type I and Strong Type II attackers. However, the existence of schemes [1–4, 20] which are provable secure against both Strong Type I and Weak Type II attackers in the random oracle model suggests that schemes can exist which are fully secure. We suggest that any attempt to prove the security of a certificateless scheme in the standard model must make use of one of the following avenues:

- **The Weak Type Ia model should be accepted**. A simple, although intellectually unsatisfying, solution would be to accept that the definition of a Type I attacker is too strong, and that the appropriate security definition should be the Weak Type Ia model. This would make the powers of the attackers in the Type I and Type II model consistent, and it doesn't seem to deprive the attacker of any avenue of attack that they are likely to have in practice.
- **The keys produced by the solver's key generation algorithm should not be indistinguishable from the `Setup` algorithms output**. In our

demonstration that a Type I simulator can be adapted to give a Type II attacker we insisted that the key pair produced by the solver's key generation algorithm must be indistinguishable from random. This allowed us to use the simulator Sim_1 with the keys generated by the `Setup` algorithm, even though the algorithm is designed to use the keys generated by the solver's key generation algorithm. However, in order for an observational proof to work, we only require that the Type I attacker be unable to distinguish the master public key from random. It might be possible to construct a security proof based on a solver whose key generation algorithm outputs public/private key pairs of a special form, but no attacker can distinguish public keys of this form from valid public keys output by the `Setup` algorithm.

- **The simulator Sim_1 should take extra information as input**. If a Type I simulator is to become a Type II attacker, then all the inputs to the simulator must be available in the Type II model (or, at least, simulated in such a way that the attacker cannot distinguish between the real input produced by the Type I solver's key generation algorithm and the input given to the simulator by the Type II attacker that is running it). However, if this input is unavailable, then the Type II attacker will not be able to execute Sim_1 correctly. This could mean that a standard model proof could be constructed based an assumption that explicitly provides extra information, such as the DDH assumption or the q-BDHI assumption.
- **The attacker should gain no advantage when given incorrect responses**. Often security proofs revolve around showing that responses given to an algorithm are indistinguishable from the responses given by a legitimate oracle. In many cases, though, the queries that might help an attacker decide whether it is interacting with a legitimate oracle or a simulator do not help the attacker break the scheme. Heuristically we can often see that even if the attacker does "realise" that it is interacting with an incorrect oracle (i.e. alter it's behaviour), their success probability is not significantly raised or lowered, and so they will still only be able to break the scheme if and only if they can solve the corresponding hard problem. However, no theoretical framework exists to examine the effects of improper oracle responses on an attacker's behaviour.
- **Game hopping techniques should be used**. In a game hopping proof, an attacker $\mathcal{A}$ is not used as a subroutine in a solver, but instead run as normal in the correct attack environment [8, 28]. This environment is slowly altered until it is shown that the attacker can have no significant advantage in wining. The proof shows that each time we change the environment, the attacker's advantage only increases a small amount. Therefore, if the attacker has no significant advantage in the final environment, then it could have had no significant advantage in the first (normal attack) environment. There is no reason why a game-hopping proof of security for a certificateless scheme against Type I attackers should imply the existence of a successful Type II attacker.
- **The attacker's black-box subroutine should be opened**. In every current security proof for an asymmetric encryption or encryption-like scheme,

the attacker is always considered to be a black box with its own source of random coins. However, when we execute the attacker subroutine within a solver, there is no reason why we should treat it as a black box. There is no reason, beyond the complexity involved in developing proofs, to stop a solver from using an attacker in a non-black-box way. This could involve choosing the attacker's coins in a way that is more advantageous; changing or resetting an attacker's state during execution; or rewinding and re-running an attacker if it makes unsuitable queries or to force it to find multiple, related solutions to a problem. This approach has been used to analyse signature schemes [22]) and zero-knowledge proofs [6].

- **Assumptions based on extractors should be used**. Observational security proofs involve running a solver to solve a given instance of a (hard) problem. This assumes that the instance of the underlying problem can be presented as input to the algorithm. However, a small number of security proofs have been reduced to problems of a different nature, which we call *extractor problems*. An extractor problem asks whether it is possible, for every attacker $\mathcal{A}$ that outputs problem instances, to find an extractor $\mathcal{A}^*$ that can solve these problem instances *given knowledge of the way they were generated*. The canonical example of an extractor assumption is the DHK assumption [7, 13, 14]. This is a very strong security assumption, but one that may allow the construction of security proofs in the Strong Type I and Strong Type II models.
- **Completely new proof techniques should be developed**. Currently, there are only two techniques that are commonly used for proving the security of any public key encryption algorithm: observational proofs and game-hopping proofs. This small number of techniques means that proofs are formulaic and the results often predictable. If the field of provable security is to move forward, then new proof techniques must be developed.

5 Conclusion

We draw several conclusions from this survey. First, we conclude that Al-Riyami and Paterson's security models are inconsistent in their strength. A scheme proven secure against Al-Riyami and Paterson's definitions is held to a higher standard against Type I attackers than Type II attackers. We provided the equivalent strong definition for Type II attackers.

We have also noted that the strong decryption oracles to which an attacker is given access in the Al-Riyami and Paterson model do not reflect reality. However, wherever possible it would be advantageous to prove a scheme's security in these model in order to give a 'margin of error' for the models. In cases where a scheme cannot be proven secure in the Strong Type I and Strong Type II models, it should be proven secure in the Weak Type Ia† and Weak Type II† models. These models most closely reflect possible real-world attacks against certificateless encryption schemes.

The problems associated with constructing a certificateless encryption scheme that is provably secure in the Strong Type I and Weak Type II models without

using the random oracle methodology were also considered. It was shown that the most obvious method for proving secure would not work, but that there are other methods that could conceivably be used to construct solutions. Of these, perhaps the most promising included the use of underlying assumptions that gave the Type I decryption simulator extra information (such as the DDH and q-BDHI problems) or the use of game-hopping proofs.

Lastly, it should be noted that certificateless encryption is not yet a viable cryptographic solution. There remain too many un-answered questions about how certificateless encryption would be used in practice. Almost all of these concern the infrastructure that needs to surround a certificateless encryption scheme. Most notably, certificateless encryption schemes cannot be considered viable unless the problems of key revocation and public-key distribution are solved. The key revocation problems for certificateless encryption mirror those of identity-based encryption in that there does not seem to be adequate way of revoking a user's right to decrypt messages or of informing users that a public key is no longer valid. The distribution problem is more unusual. Any solution to the problem of public-key distribution has to prevent a user being overwhelmed with 'false' public keys, and the problem of selecting a real public key from a false one. This problem has been partial solved by Liu and Au [21], but their security model is unconvincing and therefore we still regard this as an open problem.

Acknowledgements

The author is indebted to the provable security working group in ECRYPT's AZTEC lab for their help in discussing the topics of this paper. In particular, Caroline Kudla deserves credit for many interesting discussions at the start of this project. The author would also like to thank John Malone-Lee, Nigel Smart and Kenny Paterson for their comments on several versions of this paper, and to David Galindo for pointing out his work on the Yum–Lee constructions. The author gratefully acknowledge the financial support of the EPSRC.

References

1. S. Al-Riyami. *Cryptographic schemes based on elliptic curve pairings*. PhD thesis, Royal Holloway, University of London, 2004. Available from `http://www.isg.rhul.ac.uk/~kp/satthesis.pdf`.
2. S. S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In C. S. Laih, editor, *Advances in Cryptology – Asiacrypt 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 452–473. Springer-Verlag, 2003.
3. S. S. Al-Riyami and K. G. Paterson. CBE from CL-PKE: A generic construction and efficient schemes. In S. Vaudenay, editor, *Public Key Cryptography – PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 398–415. Springer-Verlag, 2005.
4. J. Baek, R. Safavi-Naini, and W. Susilo. Certificateless public key encryption without pairing. In J. Zhou and J. Lopez, editors, *Proceedings of the 8th International Conference on Information Security (ISC 2005)*, volume 3650 of *Lecture Notes in Computer Science*, pages 134–148. Springer-Verlag, 2005.

5. J. Baek and G. Wang. Repairing a security-mediated certificateless encryption scheme from PKC 2006. Available from `http://eprint.iacr.org/2006/159`, 2006.
6. B. Barak. How to go beyond the black-box simulation barrier. In *Proceedings of the 42nd Symposium on Foundations of Computer Science (FOCS 2001)*, pages 106–115. IEEE, 2001.
7. M. Bellare and A. Palacio. Towards plaintext-aware public-key encryption without random oracles. In P. J. Lee, editor, *Advances in Cryptology – Asiacrypt 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 48–62. Springer-Verlag, 2004.
8. M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *Advances in Cryptology – Eurocrypt 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer-Verlag, 2006.
9. K. Bentahar, P. Farshim, J. Malone-Lee, and N. P. Smart. Generic constructions of identity-based and certificateless KEMs. Available from `http://eprint.iacr.org/2005/058`, 2005.
10. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In A. Menezes, editor, *Topics in Cryptology - CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer-Verlag, 2005.
11. Z. Cheng and R. Comley. Efficient certificateless public key encryption. Available from `http://eprint.iacr.org/2005/012/`, 2005.
12. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2004.
13. I. B. Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum, editor, *Advances in Cryptology – Crypto '91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer-Verlag, 1991.
14. A. W. Dent. The Cramer-Shoup encryption scheme is plaintext aware in the standard model. In S. Vaudenay, editor, *Advances in Cryptology – Eurocrypt 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 289–307. Springer-Verlag, 2006.
15. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.
16. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.
17. E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimal cost. In H. Imai and Y. Zheng, editors, *Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68. Springer-Verlag, 1999.
18. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In M. Wiener, editor, *Advances in Cryptology – Crypto '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 535–554. Springer-Verlag, 1999.
19. D. Galindo, P. Morillo, and C. Ràfols. Breaking Yum and Lee generic constructions of certificate-less and certificate-based encryption schemes. In A. S. Atzeni and A. Lioy, editors, *Public Key Infrastructure: Third European PKI Workshop (EuroPKI 2006)*, volume 4043 of *Lecture Notes in Computer Science*, pages 81–91. Springer-Verlag, 2006.

20. B. Libert and J.-J. Quisquater. On constructing certificateless cryptosystems from identity based encryption. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, *Public Key Cryptography – PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 474–490. Springer-Verlag, 2006.
21. J. K. Liu and M. H. Au. Self-generated-certificate public key cryptosystem. Available from `http://eprint.iacr.org/2006/194`, 2006.
22. D. Pointcheval and J. Stern. Security proofs for signature schemes. In U. Maurer, editor, *Advance in Cryptology – Eurocrypt '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer-Verlag, 1996.
23. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.
24. R. Sakai and M. Kasahara. ID based cryptosystem with pairing on elliptic curve. Available from `http://eprint.iacr.org/2003/054`, 2003.
25. A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology – Crypto '84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.
26. Y. Shi and J. Li. Efficient certificateless public key encryption. Available from `http://eprint.iacr.org/2005/249/`, 2005.
27. Y. Shi and J. Li. Provable efficient certificateless public key encryption. Available from `http://eprint.iacr.org/2005/287/`, 2005.
28. V. Shoup. Sequences of games: A tool for taming complexity in security proofs. Available from `http://eprint.iacr.org/2004/332/`, 2004.
29. B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer-Verlag, 2005.
30. D. H. Yum and P. J. Lee. Generic construction of certificateless encryption. In Antonio Laganà *et al.*, editor, *Computational Science and Its Applications ICCSA 2004: Part I*, volume 3043 of *Lecture Notes in Computer Science*, pages 802–811. Springer-Verlag, 2004.
31. D. H. Yum and P. J. Lee. Identity-based cryptography in public key management. In S. K. Katsikas, S. Gritzalis, and J. Lopez, editors, *Public Key Infrastructure: First European PKI Workshop (EuroPKI 2004)*, volume 3093 of *Lecture Notes in Computer Science*, pages 71–84. Springer-Verlag, 2004.
32. Z. Zhang and D. Feng. On the security of a certificateless public-key encryption. Available from `http://eprint.iacr.org/2005/426`, 2005.