4-2009

RSA-based certificateless public key encryption

Junzuo LAI
Shanghai Jiaotong University

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Shengli LIU
Shanghai Jiaotong University

Weidong KOU
Xi Dian University

DOI: https://doi.org/10.1007/978-3-642-00843-6_3

RSA-Based Certificateless Public Key Encryption

Junzuo Lai[1], Robert H. Deng[2], Shengli Liu[1], and Weidong Kou[3]

[1] Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200030, China
{laijunzuo,slliu}@sjtu.edu.cn
[2] School of Information Systems,
Singapore Management University, Singapore 178902
robertdeng@smu.edu.sg
[3] School of Computer Science and Technology
Xi Dian University, Xi'an 710071, China
kou_weidong@yahoo.com.cn

Abstract. Certificateless Public Key Cryptography was first introduced by Al-Riyami and Paterson in order to eliminate the inherent key-escrow problem of Identity-Based Cryptography. In this paper, we present a new practical construction of certificateless public key encryption scheme without paring. Our scheme is, in the random oracle model, provably secure under the assumption that the RSA problem is intractable.

Keywords: Certificateless public key encryption, RSA.

1 Introduction

In order to solve the key escrow problem that is inherent in identity-based cryptography (IBC) [20], while at the same time, eliminate the use of certificates in the traditional public key cryptography (PKC), Al-Riyami and Paterson [1] introduced the concept of certificateless public key cryptography (CL-PKC). Different from IBC, a user's public key in CL-PKC is no longer an arbitrary string; instead, the public key is generated by the user based the user's secret information as well as a partial private key obtained from a trusted authority called Key Generation Center (KGC). As such, public keys in CL-PKC do not need to be explicitly certified. Note here that the KGC does not know the user's private keys since they contain secret information generated by the users themselves, thereby removing the escrow problem in IBC.

Since the introduction of CL-PKC [1], many concrete constructions of certificateless public key encryption (CL-PKE) schemes have been proposed. The schemes in [3,6,21,22] were proven secure in the random oracle model [4] while the schemes in [14] and [18] are secure without the random oracles.

F. Bao, H. Li, and G. Wang (Eds.): ISPEC 2009, LNCS 5451, pp. 24–34, 2009.

There were also efforts to construct generic CL-PKC schemes. The first generic CL-PKE scheme was proposed in [23] and was later shown in [16] to be insecure under the model of [1]. In [5], the authors extended the concept of key encapsulation mechanism to IBE and CL-PKE, and built generic constructions of identity-based key encapsulation mechanism and certificateless public key encapsulation mechanism.

One notable feature in the research of CL-PKE has been the development of a number of alternative security models that are substantially weaker than the original model of [1]. These different models are summarized by Dent [7]. Moreover, Dent et al. [8] presents a generic construction as well as a concrete construction for certificateless encryption schemes that are provably secure against strong adversaries in the standard model.

Au et al. [2] pointed out the weakness of the previous security models and analyzed some previous schemes under an enhanced malicious KGC model. They showed that the CL-PKE scheme in [16] is secure against malicious KGC attacks under random oracle assumption. Hwang and Liu [13] proposed a new CL-PKE scheme which is secure against malicious KGC attacks. Its security is proven in the standard model. In addition, Huang and Wong [12] proposed a generic construction of certificateless encryption which is proven secure against malicious KGC attacks in the standard model.

Other Related Work. Gentry [10] introduced a different but related concept named certificate based encryption (CBE). This approach is closer to the context of a traditional PKI model as it involves a certification authority (CA) providing an implicit certification service for clients' public keys. Liu et al. proposed the first self-generated-certificate public key encryption (SGC-PKE) scheme in [14], which defends the DoD attack that exists in CL-PKE. Lai and Kou [15] proposed a SGC-PKE scheme without using pairing.

Contribution. In spite of the recent advances in implementation technique, the paring computation is still considered as expensive compared with the "standard" operations such as modular exponentiations in finite fields. Baek et al. [3] proposed the first CL-PKE scheme without pairing, which was related to the early works on the self-certified keys [11,19].

In this paper, inspired by the identity-based key agreement protocol proposed by Okamoto and Tanaka [17] and whose security relies on the RSA problem, we present a new CL-PKE scheme without paring. Due to the extensive deployment of RSA, our scheme is better off in compatibility with the existing cryptosystems. In addition, in [3], the Type I adversary is not allowed to replace the challenge identity's public key, which is the main attacking means of Type I adversary. Compared with the scheme in [3], our scheme does not have this limitation.

Organization. The rest of the paper is organized as follow. We give some related definitions in Section 2. The model of CL-PKE is also reviewed in this section. The proposed CL-PKE scheme and its security analysis is presented in Section 3. Finally concluding remarks are given in Section 4.

2 Preliminaries

2.1 Computational Problems

Definition 1. *The RSA problem is, given a randomly generated RSA modulus n, an exponent e and a random z, to find* $y \in \mathbb{Z}_n^*$ *such that* $y^e = z$.

Definition 2. *The Computational Diffie-Hellman (CDH) problem in* $\mathbb{Z}_n^*$ *is, given p,q,n (where* $p = 2p'+1, q = 2q'+1, n = pq$ *with* p', q' *being two equal-length large primes),* $g \in \mathbb{Z}_n^*$ *of order* $p'q'$, g^a *and* g^b *for uniformly chosen* $a, b \in \mathbb{Z}_n^*$, *to compute* g^{ab}.

2.2 Certificateless Public Key Encryption

A generic CL-PKE is a tuple of algorithm described as follows [3]:

- Setup: Takes as input a security parameter κ and outputs a common parameter *params* and a master secret *msk*.
- PartialKeyExtract: Takes as input *params*, *msk* and an identity ID. It outputs a partial private key d_{ID}.
- SetSecretValue: Takes as input *params* and an identity ID. It outputs a secret value s_{ID}.
- SetPrivateKey: Takes as input *params*, d_{ID} and s_{ID}. It outputs a private key $\mathsf{SK}_{\mathsf{ID}}$.
- SetPublicKey: Takes as input *params*, d_{ID} and s_{ID}. It outputs a public key $\mathsf{PK}_{\mathsf{ID}}$.
- Enc: Takes as input *params*, a message m, a receiver's identity ID and $\mathsf{PK}_{\mathsf{ID}}$. It outputs a ciphertext c.
- Dec: Takes as input *params*, $\mathsf{SK}_{\mathsf{ID}}$ and a ciphertext c. It outputs a message m or the failure symbol $\perp$.

We insist that CL-PKE satisfies the obvious correctness requirement that decapsulation "undoes" encapsulation.

Note that, the above model of CL-PKE is slightly weaker than the original one given in [1] as a user must authenticate herself to the KGC in order to obtain a partial private key to create a public key, while the original CL-PKE model does not require a user to contact the KGC to setup her public keys. However, as argued in [3], this modified model preserves the unique property of CL-PKE that no certificates are required in order to guarantee the authenticity of public keys, which is the main motivation of CL-PKE.

Security Model. There are two types of adversaries [1]. Type I adversary models an "outsider" adversary, who does not have the KGC's master secret key but it can replace public keys of arbitrary identities with other public keys of its own choices. It can also obtain partial and full secret keys of arbitrary identities. Type II adversary models an "honest-but-curious" KGC, who knows the master secret key (hence it can compute partial secret key by itself). It is still allowed to obtain full secret key for arbitrary identities but is not allowed to replace public keys at any time.

Security in CL-PKE is defined using the following game between an attack algorithm $\mathcal{A}$ and a challenger.

Setup. The challenger runs the Setup algorithm and gives $\mathcal{A}$ the resulting system parameter *params*. If $\mathcal{A}$ is of Type I, the challenger keeps the master secret key *msk* to itself; otherwise, it gives *msk* to $\mathcal{A}$.

Query phase 1 The adversary $\mathcal{A}$ adaptively issues the following queries:

- **Public-Key-Request** query: On input an identity ID, the challenger runs $\mathsf{SetPublicKey}(params, d_{\mathsf{ID}}, s_{\mathsf{ID}})$, where the partial private key d_{ID} and the secret value s_{ID} of the identity ID are obtained from $\mathsf{PartialKeyExtract}$ and $\mathsf{SetSecretValue}$, respectively, and forwards the result to the adversary.
- **Partial-Key-Extract** query: On input an identity ID, the challenger runs $\mathsf{PartialKeyExtract}(params, msk, \mathsf{ID})$ and returns the result to $\mathcal{A}$. Note that it is only useful to Type I adversary.
- **Private-Key-Request** query: On input an identity ID, the challenger runs $\mathsf{SetPrivateKey}(params, d_{\mathsf{ID}}, s_{\mathsf{ID}})$, where the partial private key d_{ID} and the secret value s_{ID} of the identity ID are obtained from $\mathsf{PartialKeyExtract}$ and $\mathsf{SetSecretValue}$, respectively, and forwards the result to the adversary. It outputs $\perp$ if the uesr's public key has been replaced in the case of Type I adversary.
- **Public-Key-Replace** query (for Type I adversary only): On input an identity and a valid public key, it replaces the associated user's public key with the new one.
- **Dec** query: On input a ciphertext and an identity, returns the decrypted message using the private key corresponding to the current value of the public key associated with the identity of the user.

Challenge query: After making a polynomial number of queries, $\mathcal{A}$ outputs two messages m_0, m_1 and an identity ID^*. The challenger picks a random bit $\beta \in \{0, 1\}$, sets $c^* = \mathsf{Enc}(parmas, m_\beta, \mathsf{ID}^*, \mathsf{PK}_{\mathsf{ID}^*})$ and sends c^* to $\mathcal{A}$.

Query phase 2 $\mathcal{A}$ makes a new sequence of queries.

Guess $\mathcal{A}$ outputs a bit β'. It wins the game if $\beta' = \beta$ under the following conditions:

- At any time, ID^* has not been submitted to the **Private-Key-Request** query.
- $(c^*, \mathsf{ID}^*, \mathsf{PK}_{\mathsf{ID}^*})$ have not been submitted to the **Dec** query.
- If it is Type I adversary, ID^* cannot be equal to an identity for which both the public key has been replaced and the partial private key has been extracted.

We define $\mathcal{A}$'s advantage in attacking the certificateless public key encryption $\mathsf{CL\text{-}PKE}$ as

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CL\text{-}PKE}} = |\mathsf{Pr}[\beta = \beta'] - \frac{1}{2}|.$$

Definition 3. *We say that a certificateless public key encryption CL-PKE is* $(t, q_{pub}, q_{par}, q_{prv}, q_d, \epsilon)$*-IND-CCA secure against Type I (resp. Type II) adversary* $\mathcal{A}_I$ *(resp.* $\mathcal{A}_{II}$*), if for all* t*-time algorithms* $\mathcal{A}_I$ *(resp.* $\mathcal{A}_{II}$*) making at most* q_{pub} ***Public-Key-Request*** *queries,* q_{par} ***Partial-Key-Extract*** *queries,* q_{prv} ***Private-Key-Request*** *queries and* q_d ***Dec*** *queries, have advantage at most* ϵ *in winning the above game.*

IND-CPA security is defined similarly, but with the restriction that the adversary cannot make **Dec** queries.

Definition 4. *We say that a certificateless public key encryption CL-PKE is* $(t, q_{pub}, q_{par}, q_{prv}, \epsilon)$*-IND-CPA secure, if it is* $(t, q_{pub}, q_{par}, q_{prv}, 0, \epsilon)$*-IND-CCA secure.*

3 Our Scheme

Our CL-PKE scheme is inspired by the RSA-based key agreement protocol [17] introduced by Okamoto and Tanaka. We first present our scheme and then show that it is IND-CPA secure. However, it is easy to turn our IND-CPA secure CL-PKE scheme into an IND-CCA secure CL-PKE scheme using the technique proposed by Fujisaki and Okamoto [9], as did in [3].

Setup(κ) Given a security parameter κ, a RSA group $< n, p, q, e, d, g >$ is generated, where p', q' are κ-bit prime numbers, $p = 2p' + 1, q = 2q' + 1, n = pq, e < \phi(n), \gcd(e, \phi(n)) = 1, ed \equiv 1 (\mathrm{mod}\ \phi(n))$, and ϕ denotes the Euler totient function. Chooses two cryptographic hash functions $H : \{0,1\}^* \rightarrow \mathbb{Z}_n^*, H_2 : \mathbb{Z}_n^* \rightarrow \{0,1\}^l$, where l is the length of the plaintext message. The master secret key is defined as $msk = d$. The common parameter is $params = (n, e, H, H_2)$.

PartialKeyExtract($params, msk, \mathsf{ID}$) Given $params$, $msk = d$ and an identity $\mathsf{ID} \in \{0,1\}^*$, outputs the partial private key

$$d_{\mathsf{ID}} = H(\mathsf{ID})^d.$$

SetSecretValue($params, \mathsf{ID}$) Given $params$ and an identity ID, randomly chooses $x_{\mathsf{ID}} \in \mathbb{Z}_n^*$ and outputs

$$s_{\mathsf{ID}} = x_{\mathsf{ID}}.$$

SetPrivateKey($params, d_{\mathsf{ID}}, s_{\mathsf{ID}}$) Given $params$, the partial private key d_{ID} and the secret value $s_{\mathsf{ID}} = x_{\mathsf{ID}}$ of an identity ID, outputs

$$\mathsf{SK}_{\mathsf{ID}} = x_{\mathsf{ID}}.$$

SetPublicKey($params, d_{\mathsf{ID}}, s_{\mathsf{ID}}$) Given $params$, the partial private key $d_{\mathsf{ID}} = H(\mathsf{ID})^d$ and the secret value $s_{\mathsf{ID}} = x_{\mathsf{ID}}$ of an identity ID, outputs

$$\mathsf{PK}_{\mathsf{ID}} = H(\mathsf{ID})^{d + x_{\mathsf{ID}}}.$$

$\mathsf{Enc}(params, m, \mathsf{ID}, \mathsf{PK}_{\mathsf{ID}})$ Given *params*, a message m and the public key $\mathsf{PK}_{\mathsf{ID}}$ of an identity ID, randomly chooses $r \in \mathbb{Z}_n^*$ and computes

$$c_1 = H(\mathsf{ID})^{er}, c_2 = H_2(\mathsf{PK}_{\mathsf{ID}}{}^{er} H(\mathsf{ID})^{-r}) \oplus m,$$

then outputs $c = (c_1, c_2)$.

$\mathsf{Dec}(params, \mathsf{SK}_{\mathsf{ID}}, c)$ Given *params*, the private key $\mathsf{SK}_{\mathsf{ID}}$ of an identity ID, and a ciphertext $c = (c_1, c_2)$, outputs

$$m = H_2(c_1^{\mathsf{SK}_{\mathsf{ID}}}) \oplus c_2.$$

It can be easily seen that the above decryption algorithm is consistent, i. e.,

$$\begin{aligned} \mathsf{PK}_{\mathsf{ID}}{}^{er} H(\mathsf{ID})^{-r} &= H(\mathsf{ID})^{(d+x_{\mathsf{ID}})er} H(\mathsf{ID})^{-r} \\ &= H(\mathsf{ID})^{r} H(\mathsf{ID})^{erx_{\mathsf{ID}}} H(\mathsf{ID})^{-r} \\ &= c_1^{x_{\mathsf{ID}}} = c_1^{\mathsf{SK}_{\mathsf{ID}}}. \end{aligned}$$

We now prove the security of the scheme by two theorems.

Theorem 1. *Assume the hash functions H, H_2 are random oracles and the RSA problem is (t, ϵ)-intractable. Then, the above CL-PKE scheme is $(t', q_{pub}, q_{par}, q_{prv}, \epsilon')$-IND-CPA secure against Type I adversary $\mathcal{A}_I$ for*

$$t > t' + t_{ex}(q_H + q_{pub}), \epsilon > \frac{2\epsilon'}{q_{H_2}\tau(q_{par} + q_{prv} + 1)},$$

where t_{ex} denotes the time for computing exponentiation in $\mathbb{Z}_n^*$, τ denotes the base of the natural logarithm and q_H (resp. q_{H_2}) denotes the number of H (resp. H_2) queries by the adversary.

Proof. Let $\mathcal{A}_I$ be a Type I adversary that $(t', q_{pub}, q_{par}, q_{prv}, \epsilon')$-breaks the IND-CPA security of the certificateless public key encryption scheme described above. We construct an algorithm $\mathcal{B}$, that solves the RSA problem, as follows. $\mathcal{B}$ is given an instance of the RSA problem, which consists of (n, e, z). $\mathcal{B}$'s goal is to find $y \in \mathbb{Z}_n^*$ such that $y^e = z$. It interacts with $\mathcal{A}_I$ as follows.

Setup $\mathcal{B}$ maintains three lists H-List,H_2-List and KeyList. Initially the lists are empty. The common parameter $params = (n, e)$ is sent to $\mathcal{A}_I$. The master secret key $msk = d$, where $ed \equiv 1 (\mathrm{mod}\ \phi(n))$, is unknown to $\mathcal{B}$.

Query phase 1 $\mathcal{A}_I$ adaptively issues H, H_2, **Public-Key-Request**, **Partial-Key-Extract**, **Private-Key-Request** and **Public-Key-Replace** queries. $\mathcal{B}$ answers them as follows:

- H query on ID: If a record $(\mathsf{ID}, h_{\mathsf{ID}}, f_{\mathsf{ID}}, coin)$ appears in the H-List, sends h_{ID} to $\mathcal{A}_I$; otherwise, $\mathcal{B}$ picks $coin \in \{0, 1\}$ at random such that $\mathsf{Pr}[coin = 0] = \rho$. ($\rho$ will be determined later.) Then, randomly chooses $f_{\mathsf{ID}} \in \mathbb{Z}_n^*$. Finally, the record $(\mathsf{ID}, h_{\mathsf{ID}} = z^{coin} \cdot f_{\mathsf{ID}}^e, f_{\mathsf{ID}}, coin)$ is added to the H-List and h_{ID} is sent to $\mathcal{A}_I$.

- H_2 query on ω: If a record (ω, k) appears in the H_2-List, sends k to $\mathcal{A}_I$; otherwise, picks $k \in \{0,1\}^l$ at random, adds the record (ω, k) to H_2-List and sends k to $\mathcal{A}_I$.
- **Public-Key-Request** query on ID: Randomly chooses $x_{\mathsf{ID}} \in \mathbb{Z}_n^*$ and searches H-List for a record $(\mathsf{ID}, h_{\mathsf{ID}}, f_{\mathsf{ID}}, coin)$. Then, adds the record $(\mathsf{ID}, \mathsf{PK}_{\mathsf{ID}} = f_{\mathsf{ID}} h_{\mathsf{ID}}^{x_{\mathsf{ID}}}, \mathsf{SK}_{\mathsf{ID}} = x_{\mathsf{ID}}, coin)$ to KeyList and sends $\mathsf{PK}_{\mathsf{ID}}$ to $\mathcal{A}_I$.
- **Partial-Key-Extract** query on ID: Searches H-List for a record $(\mathsf{ID}, h_{\mathsf{ID}}, f_{\mathsf{ID}}, coin)$. If $coin = 0$, sends f_{ID} to $\mathcal{A}_I$; otherwise, aborts and terminates.
- **Private-Key-Extract** query on ID: Searches KeyList for a record $(\mathsf{ID}, \mathsf{PK}_{\mathsf{ID}}, \mathsf{SK}_{\mathsf{ID}}, coin)$. If $coin = 0$, sends $\mathsf{SK}_{\mathsf{ID}}$ to $\mathcal{A}_I$; otherwise, aborts and terminates.
- **Public-Key-Replace** query on $(\mathsf{ID}, \mathsf{PK}'_{\mathsf{ID}})$: Replaces $\mathsf{PK}_{\mathsf{ID}}$ with $\mathsf{PK}'_{\mathsf{ID}}$.

Challenge $\mathcal{A}_I$ submits two messages m_0, m_1 and an identity ID^* with the public key $\mathsf{PK}_{\mathsf{ID}^*}$. $\mathcal{B}$ searches H-List for a record $(\mathsf{ID}^*, h_{\mathsf{ID}^*}, f_{\mathsf{ID}^*}, coin)$. If $coin = 0$, it aborts and terminates; otherwise, $\mathcal{B}$ picks $r \in \mathbb{Z}_n^*$ at random. Let $r^* = d + r$, which is unknown to $\mathcal{B}$. Then $\mathcal{B}$ randomly chooses $c_2^* \in \{0,1\}^l$ and computes

$$c_1^* = H(\mathsf{ID}^*)^{er^*} = H(\mathsf{ID}^*)^{e(d+r)} = h_{\mathsf{ID}^*}^{1+er}.$$

Finally, it sends $c^* = (c_1^*, c_2^*)$ to $\mathcal{A}_I$.

Query phase 2 $\mathcal{A}_I$ makes a new sequence of queries, and $\mathcal{B}$ responds as in Query phase 1.

Guess: Finally, the adversary $\mathcal{A}_I$ outputs a bit β'. $\mathcal{B}$ picks a tuple (ω, k) from H_2-List at random and outputs $\frac{\mathsf{PK}_{\mathsf{ID}^*}^{1+er}}{\omega h_{\mathsf{ID}^*}^r f_{\mathsf{ID}^*}}$ as the solution to the RSA problem.

Probability Analysis: Let AskH_2^* denotes the event that $\mathsf{PK}_{\mathsf{ID}^*}^{er^*} H(\mathsf{ID}^*)^{-r^*}$ has been queried to H_2. Note that,

$$\begin{aligned}
\mathsf{PK}_{\mathsf{ID}^*}^{er^*} H(\mathsf{ID}^*)^{-r^*} &= \mathsf{PK}_{\mathsf{ID}^*}^{e(d+r)} H(\mathsf{ID}^*)^{-d-r} \\
&= \mathsf{PK}_{\mathsf{ID}^*}^{1+er} h_{\mathsf{ID}^*}^{-d} h_{\mathsf{ID}^*}^{-r} \\
&= \mathsf{PK}_{\mathsf{ID}^*}^{1+er} (z f_{\mathsf{ID}^*}^e)^{-d} h_{\mathsf{ID}^*}^{-r} \\
&= \mathsf{PK}_{\mathsf{ID}^*}^{1+er} z^{-d} f_{\mathsf{ID}^*}^{-1} h_{\mathsf{ID}^*}^{-r}.
\end{aligned}$$

If the event AskH_2^* happens, then $\mathcal{B}$ will be able to solve the RSA problem by choosing a tuple (ω, k) from the H_2-List and computing $\frac{\mathsf{PK}_{\mathsf{ID}^*}^{1+er}}{\omega h_{\mathsf{ID}^*}^r f_{\mathsf{ID}^*}}$ with the probability at least $\frac{1}{q_{H_2}}$, where q_{H_2} is the number of H_2 queries by the adversary. If the event AskH_2^* does not happen, $\mathcal{B}$'s simulations are perfect and are identically distributed as the real one from the construction.

We observe that the probability that $\mathcal{B}$ does not abort during the simulation is given by $\rho^{q_{par}+q_{prv}}(1-\rho)$ which is maximized at $\rho = 1 - 1/(q_{par} + q_{prv} + 1)$. Hence the probability that $\mathcal{B}$ does not abort is at most $\frac{1}{\tau(q_{par}+q_{prv}+1)}$, where τ denotes the base of the natural logarithm.

Now, the event $\mathsf{AskH}_2^* | \neg \mathsf{Abort}$ denoted by Good, where Abort denotes the event that $\mathcal{B}$ aborts during the simulation. If Good dose not happen, it is clear that

the adversary does not gain any advantage greater than $1/2$ to guess β. Namely, we have $\mathsf{Pr}[\beta' = \beta | \neg \mathsf{Good}] \leq 1/2$. Hence, by splitting $\mathsf{Pr}[\beta' = \beta]$, we obtain $|\mathsf{Pr}[\beta' = \beta] - \frac{1}{2}| \leq \frac{1}{2}\mathsf{Pr}[\mathsf{Good}]$. To sum up, we have $\epsilon > \frac{2\epsilon'}{q_{H_2}\tau(q_{par}+q_{prv}+1)}$.

Time Complexity. In the simulation, $\mathcal{B}$'s overhead is dominated by the exponentiation computation in response to $\mathcal{A}_I$'s H and **Public-Key-Request** queries. So, we have $t > t' + t_{ex}(q_H + q_{pub})$, where t_{ex} denotes the time for computing exponentiation in $\mathbb{Z}_n^*$.

This concludes the proof of Theorem 1.

Theorem 2. *Assume the hash functions H and H_2 are random oracles and the CDH problem is (t, ϵ)-intractable. Then, the above CL-PKE scheme is $(t', q_{pub}, q_{par}, q_{prv}, \epsilon')$IND-CPA secure against Type II adversary $\mathcal{A}_{II}$ for*

$$t > t' + t_{ex}(q_H + q_{pub}), \epsilon > \frac{2\epsilon'}{q_{H_2}\tau(q_{prv}+1)},$$

where t_{ex} denotes the time for computing exponentiation in $\mathbb{Z}_n^*$, τ denotes the base of the natural logarithm and q_H (resp. q_{H_2}) denotes the number of H (resp. H_2) queries by the adversary.

Proof. Let $\mathcal{A}_{II}$ be a Type II adversary that $(t', q_{pub}, q_{par}, q_{prv}, \epsilon')$-breaks the IND-CPA security of the CL-PKE scheme described above. We construct an algorithm $\mathcal{B}$, that solves the CDH problem, as follows. $\mathcal{B}$ is given an instance of the CDH problem, which consists of (n, p, q, g, g^a, g^b). $\mathcal{B}$'s goal is to compute g^{ab}. It interacts with $\mathcal{A}_{II}$ as follows.

Setup $\mathcal{B}$ maintains three lists H-List, H_2-List and KeyList. Initially the lists are empty. Then $\mathcal{B}$ picks $e < \phi(n), \gcd(e, \phi(n)) = 1$ at random and computes d such that $ed \equiv 1 (\mathrm{mod}\ \phi(n))$, where ϕ denotes the Euler totient function. (It can be computed by p, q.) Finally, it sends the common parameter $params = (n, e)$ and the master secret key $msk = d$ to $\mathcal{A}_{II}$.

Query phase 1 $\mathcal{A}_{II}$ adaptively issues H, H_2, **Public-Key-Request** and **Private-Key-Request** queries. $\mathcal{B}$ answers them in the following way:

- H query on ID: If a record $(\mathsf{ID}, h_{\mathsf{ID}}, t_{\mathsf{ID}})$ appears in the H-List, $\mathcal{B}$ sends h_{ID} to $\mathcal{A}_I$; otherwise, $\mathcal{B}$ randomly chooses t_{ID} such that $t_{\mathsf{ID}} < \phi(n)$, $\gcd(t_{\mathsf{ID}}, \phi(n)) = 1$, adds the record $(\mathsf{ID}, h_{\mathsf{ID}} = g^{t_{\mathsf{ID}}}, t_{\mathsf{ID}})$ to H-List and sends h_{ID} to $\mathcal{A}_{II}$.
- H_2 query on ω: If a record (ω, k) appears in the H_2-List, $\mathcal{B}$ sends k to $\mathcal{A}_I$: otherwise, $\mathcal{B}$ picks $k \in \{0,1\}^l$ at random, adds the record (ω, k) to H_2-List and sends k to $\mathcal{A}_{II}$.
- **Public-Key-Request** query on ID: $\mathcal{B}$ searches H-List for a record $(\mathsf{ID}, h_{\mathsf{ID}}, t_{\mathsf{ID}})$. Then, it picks $coin \in \{0,1\}$ at random such that $\mathsf{Pr}[coin = 0] = \rho$ (ρ will be determined later). Finally, it randomly chooses $x_{\mathsf{ID}} \in \mathbb{Z}_n^*$, adds the record $(\mathsf{ID}, \mathsf{PK}_{\mathsf{ID}} = h_{\mathsf{ID}}^{d+x_{\mathsf{ID}}} \cdot (g^a)^{t_{\mathsf{ID}} \cdot coin} = h_{\mathsf{ID}}^{d+a \cdot coin + x_{\mathsf{ID}}}, \mathsf{SK}_{\mathsf{ID}} = x_{\mathsf{ID}}, coin)$ to KeyList and sends $\mathsf{PK}_{\mathsf{ID}}$ to $\mathcal{A}_{II}$.

– **Private-Key-Extract** query on ID: $\mathcal{B}$ searches $\mathsf{KeyList}$ for a record $(\mathsf{ID}, \mathsf{PK}_{\mathsf{ID}}, \mathsf{SK}_{\mathsf{ID}}, coin)$. If $coin = 0$, it sends the $\mathsf{SK}_{\mathsf{ID}}$ to $\mathcal{A}_{II}$; otherwise, it aborts and terminates.

Challenge $\mathcal{A}_{II}$ submits two messages m_0, m_1 and an identity ID^* with the public key $\mathsf{PK}_{\mathsf{ID}^*}$. $\mathcal{B}$ searches $\mathsf{H\text{-}List}$ for a record $(\mathsf{ID}^*, h_{\mathsf{ID}^*}, t_{\mathsf{ID}^*})$ and $\mathsf{KeyList}$ for a record $(\mathsf{ID}^*, \mathsf{PK}_{\mathsf{ID}^*}, \mathsf{SK}_{\mathsf{ID}^*} = x_{\mathsf{ID}^*}, coin)$. If $coin = 0$, it aborts and terminates; otherwise, $\mathcal{B}$ randomly chooses $c_2^* \in \{0,1\}^l$. Let $r^* = b$, which is unknown to $\mathcal{B}$. Then $\mathcal{B}$ computes

$$c_1^* = (g^b)^{et_{\mathsf{ID}^*}} = (g^{t_{\mathsf{ID}^*}})^{er^*} = h_{\mathsf{ID}^*}^{er^*} = H(\mathsf{ID}^*)^{er^*}.$$

and sends $c^* = (c_1^*, c_2^*)$ to $\mathcal{A}_{II}$.

Query phase 2 $\mathcal{A}_{II}$ makes a new sequence of queries, and $\mathcal{B}$ responds as in Query phase 1.

Guess Finally, the adversary $\mathcal{A}_{II}$ outputs a bit β'. $\mathcal{B}$ picks a tuple (ω, k) from $\mathsf{H}_2\text{-}\mathsf{List}$ at random and outputs $\frac{\omega^{dt_{\mathsf{ID}^*}^{-1}}}{(g^b)^{x_{\mathsf{ID}^*}}}$ as the solution to the CDH problem. Note that, $\mathcal{B}$ knows p, q, so $t_{\mathsf{ID}^*}^{-1}$ can be computed.

Probability Analysis: Let AskH_2^* denotes the event that $\mathsf{PK}_{\mathsf{ID}^*}^{er^*} H(\mathsf{ID}^*)^{-r^*}$ has been queried to H_2. Note that,

$$\begin{aligned}
\mathsf{PK}_{\mathsf{ID}^*}^{er^*} H(\mathsf{ID}^*)^{-r^*} &= \mathsf{PK}_{\mathsf{ID}^*}^{eb} H(\mathsf{ID}^*)^{-b} \\
&= h_{\mathsf{ID}^*}^{eb(d+a+x_{\mathsf{ID}^*})} h_{\mathsf{ID}^*}^{-b} \\
&= h_{\mathsf{ID}^*}^{eb(a+x_{\mathsf{ID}^*})} \\
&= (g^{ab})^{et_{\mathsf{ID}^*}} (g^b)^{et_{\mathsf{ID}^*} x_{\mathsf{ID}^*}}.
\end{aligned}$$

If the event AskH_2^* happens, then $\mathcal{B}$ will be able to solve the CDH problem by choosing a tuple (ω, k) from the $\mathsf{H}_2\text{-}\mathsf{List}$ and computing $\frac{\omega^{dt_{\mathsf{ID}^*}^{-1}}}{(g^b)^{x_{\mathsf{ID}^*}}}$ with the probability at least $\frac{1}{q_{H_2}}$, where q_{H_2} is the number of H_2 queries by the adversary. If the event AskH_2^* does not happen, $\mathcal{B}$'s simulations are perfect and are identically distributed as the real one form the construction.

We observe that the probability that $\mathcal{B}$ does not abort during the simulation is given by $\rho^{q_{prv}}(1-\rho)$ which is maximized at $\rho = 1 - 1/(q_{prv}+1)$. Hence, the probability that $\mathcal{B}$ does not abort is at most $\frac{1}{\tau(q_{prv}+1)}$, where τ denotes the base of the natural logarithm.

Now, the event $\mathsf{AskH}_2^* | \neg\mathsf{Abort}$ denoted by Good, where Abort denotes the event that $\mathcal{B}$ aborts during the simulation. If Good dose not happen, it is clear that the adversary does not gain any advantage greater than $1/2$ to guess β. Namely, we have $\mathsf{Pr}[\beta' = \beta | \neg\mathsf{Good}] \leq 1/2$. Hence, by splitting $\mathsf{Pr}[\beta' = \beta]$, we obtain $|\mathsf{Pr}[\beta' = \beta] - \frac{1}{2}| \leq \frac{1}{2}\mathsf{Pr}[\mathsf{Good}]$. To sum up, we have $\epsilon > \frac{2\epsilon'}{q_{H_2}\tau(q_{prv}+1)}$.

Time Complexity. In the simulation, $\mathcal{B}$'s overhead is dominated by the exponentiation computation in response to $\mathcal{A}_{II}$'s H and **Public-Key-Request**

query. So, we have $t > t' + t_{ex}(q_H + q_{pub})$, where t_{ex} denotes the time for computing exponentiation in $\mathbb{Z}_n^*$.

This concludes the proof of Theorem 2.

4 Conclusion

We have presented a new practical CL-PKE scheme that does not depend on the paring. We have proven that our scheme is, in the random oracle model, secure under the assumption that the RSA problem is intractable.

However, the model of our scheme is slightly weaker than the original model [1]. It is still an open problem to design a CL-PKE scheme without paring in the original model [1] that is IND-CCA secure, even relies on the random oracles.

Acknowledgement

This research is supported by the Office of Research, Singapore Management University.

References

1. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
2. Au, M., Chen, J., Liu, J., Mu, Y., Wong, D., Yang, G.: Malicious KGC attacks in certificateless cryptography. In: ASIACCS 2007, pp. 302–311. ACM Press, New York (2007)
3. Baek, J., Safavi-Naini, R., Susilo, W.: Certificateless public key encryption without pairing. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 134–148. Springer, Heidelberg (2005)
4. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: ACM CCS 1993, pp. 62–73. ACM Press, New York (1993)
5. Bentahar, K., Farshim, P., Malone-Lee, J.: Generic constructions of identity-based and certificateless KEMs. Cryptology ePrint Archive, Report 2005/058 (2005), http://eprint.iacr.org/2005/058
6. Cheng, Z., Comley, R.: Efficient certificateless public key encryption. Cryptology ePrint Archive, Report 2005/012 (2005), http://eprint.iacr.org/2005/012
7. Dent, A.W.: A survey of certificateless encryption schemes and security models. Cryptology ePrint Archive, Report 2006/211 (2006), http://eprint.iacr.org/2006/211
8. Dent, A., Libert, B., Paterson, K.: Certificateless encryption schemes strongly secure in the standard model. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 344–359. Springer, Heidelberg (2008)
9. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetirc and Symmetric Encryption Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)

10. Gentry, C.: Certificate-based encryption and the certificate revocation problem. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 272–293. Springer, Heidelberg (2003)
11. Girault, M.: Self-certified public keys. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 490–497. Springer, Heidelberg (1991)
12. Huang, Q., Wong, D.S.: Generic certificateless encryption in the standard model. In: Miyaji, A., Kikuchi, H., Rannenberg, K. (eds.) IWSEC 2007. LNCS, vol. 4752, pp. 278–291. Springer, Heidelberg (2007)
13. Hwang, Y.H., Liu, J.K., Chow, S.S.M.: Certificateless Public Key Encryption Secure against KGC Attacks in the Standard Model. Journal of Universal Computer Science, Special Issue on Cryptography in Computer System Security 14(3), 463–480 (2008)
14. Liu, J., Au, M., Susilo, W.: Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In: ASIACCS 2007, pp. 273–283. ACM Press, New York (2007)
15. Lai, J., Kou, W.: Self-Generated-Certificate Public Key Encryption Without Pairing. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 476–489. Springer, Heidelberg (2007)
16. Libert, B., Quisquater, J.: On constructing certificateless cryptosystems from identity based encryption. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 474–490. Springer, Heidelberg (2006)
17. Okamoto, E., Tanaka, K.: Key Distribution System Based on Identification Infromation. IEEE J. Selected Areas in Communications 7, 481–485 (1989)
18. Park, J.H., Choi, K.Y., Hwang, J.Y., Lee, D.H.: Certificateless Public Key Encryption in the Selective-ID Security Model (Without Random Oracles). In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 60–82. Springer, Heidelberg (2007)
19. Petersen, H., Horster, P.: Self-certified keys - concepts and applications. In: 3rd Int. Conference on Communications and Multimedia Security, pp. 102–116. Chapman and Hall, Boca Raton (1997)
20. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
21. Shi, Y., Li, J.: Provable efficient certificateless public key encryption. Cryptology ePrint Archive, Report 2005/287 (2005), `http://eprint.iacr.org/2005/287`
22. Sun, Y., Zhang, F., Baek, J.: Strongly Secure Certificateless Public Key Encryption without Pairing. In: Bao, F., Ling, S., Okamoto, T., Wang, H., Xing, C. (eds.) CANS 2007. LNCS, vol. 4856, pp. 194–208. Springer, Heidelberg (2007)
23. Yum, D.H., Lee, P.J.: Generic construction of certificateless encryption. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3043, pp. 802–811. Springer, Heidelberg (2004)