Certificateless Key-Insulated Encryption: Cryptographic Primitive for Achieving Key-Escrow Free and Key-Exposure Resilience

Libo He(✉), Chen Yuan, Hu Xiong, and Zhiguang Qin

School of Information and Software Engineering,
University of Electronic Science and Technology of China,
Chengdu 610054, Sichuan, China
libowqrs@gmail.com, yuanchenincn@gmail.com, {xionghu,qinzg}@uestc.edu.cn

Abstract. Certificateless encryption (CLE) alleviates the heavy certificate management in traditional public key encryption and the key escrow problem in the ID-based encryption simultaneously. Current CLE schemes assumed that the user's secret key is absolutely secure. Unfortunately, this assumption is too strong in case the CLE is deployed in the hostile setting and the leakage of a secret key is inevitable. In this paper, we present a new concept called a certificateless key-insulated encryption scheme (CL-KIE). We argue that this is an important cryptographic primitive that can be used to achieve key-escrow free and key-exposure resilience. We also present an efficient CL-KIE scheme based on bilinear pairing. After that, the security of our scheme is proved under the Bilinear Diffie-Hellman assumption in the random oracle model.

Keywords: Bilinear pairing · Certificateless cryptography · Key-insulated

1 Introduction

1.1 Motivation and Related Work

In a traditional public key cryptosystem, every user owns a pair of a public key which will be published and publicly accessible and a private key which will be preserved by the user himself. In 1978, Rivest et al. [14], who first publicly published the RSA algorithm whose security is relied on practical difficulty of factoring the product of two large prime number. It is the first practical Public Key Encryption in nowadays. ElGama algorithm is another widely used public key cryptography which is based on the Diffie-Hellman key exchange. It was described by ElGamal [15] in 1985. The public key cryptosystem needs Public Key Infrastructure(PKI) to offer the authentication and validation for the public key. But PKI will encounter a lot of challenges on efficiency and scalability for its complicated structure. In 1984, the Identity-based Encryption has been proposed by Shamir [1]. By this approach, the private key generated in Key

Y. Wang et al. (Eds.): BigCom 2016, LNCS 9784, pp. 387–395, 2016.
DOI: 10.1007/978-3-319-42553-5_33

Generation Center(KGC) could be arbitrary characters related to users identity. So the certificate will not be necessary but the key escrow problem arises that the malicious authority can impersonate any users to get the corresponding private key. In 2001, Boneh and Franklin [16] proposed an identity-based encryption system based on Weil pairing over elliptic curves and finite fields. Based on the rapid calculation of the bilinear pairing, the identity-based encryption becomes a research hotspot since then. To solve the problem of key escrow in Identity-based Encryption and avoid the use of certificates to guarantee the authenticity of public keys in Public Key Encryption, the Certificateless Public Key Encryption(CL-PKE) has been introduced by Al-Riyami and Paterson [2] in 2003. In CL-PKE, the private key is separated into two parts: one partial private key is still generated in KGC, and the secret key is selected by the user himself. The malicious KGC only can get the partial private key, hence, the Certificateless Public Key Encryption solves the problems of key escrow. Since then, several other relevant certificateless schemes [4–9] have been developed.

The leakage of a private key is the devastating disaster for the public key cryptosystem since it means all security guarantees are lost. To avoid key exposure, Dodis et al. proposed the notion of key-insulated security in 2002 [3]. In their approach, the private key is composed of two parts: one part is generated by the master key and the other is created by the helper key from a physically-secure device. The lifetime of the private key is divided into N time periods and the private key is updated in every time period with the help of the helper key. Meanwhile, the public key is maintained during the whole key updating. By this approach, even the adversary who steals the private key in the present time period can not get the private key in the former or later period. It solves the problem of leakage of private key successfully to some extent.

Since then, key-insulated security has attracted much attentions and a lot of primitives for encryption [10–13] have been described. However, none of the prior key-insulated encryptions is constructed on CL-PKE. Current CL-PKE schemes assumed that the user's secret key is absolutely secure. Unfortunately, this assumption is too strong in case the CL-PKE is deployed in the hostile setting and the leakage of a secret key is inevitable. To alleviate this problem, we construct a new scheme which integrates CL-PKE and key-insulated notion. So this new scheme will not only prevent attacks from the malicious KGC but also avoid the leakage of the private key.

1.2 Contribution

In this paper, we present a new concept called a certificateless key-insulated encryption scheme (CL-KIE). We argue that this is an important cryptographic primitive that can be used to achieve key-escrow free and key-exposure resilience. We also present an efficient CL-KIE scheme based on bilinear pairing. After that, the security of our scheme is proved under the Bilinear Diffie-Hellman assumption in the random oracle model.

2 Formal Definition and Security Model

The proposed scheme is based on the bilinear pairing over the elliptic curve and finite field. The related security assumption is built on the Bilinear Diffie-Hellman problem. In this section, we formalize the definition of our new scheme CL-KIE and give a security model for the CL-KIE scheme.

2.1 Definition of CL-KIE

We formalize the CL-KIE (Certificateless Key Insulated Encryption) scheme, which consists of the following algorithms:

- **Setup:** The algorithm is given a security parameter k regarded as the security parameter, and returns *params* (system parameters), a `master-key` and a `helper-key`. The system parameters include a description of a finite message space $\mathcal{M}$, a description of a finite ciphertext space $\mathcal{C}$ and a randomness space $\mathcal{R}$.
- **SecretValExtract:** The algorithm takes as input *params* and an identity string ID_A and returns a random $x_A \in Z_q$ as the secret value associated with the entity A.
- **PartialKeyExtract:** The algorithm takes as input *params*, `master-key`, and an identity string $ID_A \in \{0,1\}^*$, and returns a partial private key D_A associated to ID_A.
- **HelperKeyUpdate:** The algorithm takes as input *params*, a time period i, `helper-key`, an identity string ID_A, and returns the helper key $HK_{A,i}$ at a time period i.
- **PrivateKeyUpdate:** The algorithm takes as input *params*, a time period i, the helper key $HK_{A,i}$, an identity string ID_A, the partial private key D_A and the secret value x_A, and outputs the private key $S_{A,i}$ at a time period i.
- **PublicKeyExtract:** The algorithm takes as input *params*, the secret value x_A and an identity string ID_A, and outputs a public key P_A of the entity A.
- **Encrypt:** The algorithm takes as input a time period i, *params*, ID_A, P_A and $M \in \mathcal{M}$. It returns a ciphertext $C \in \mathcal{C}$.
- **Decrypt:** The algorithm takes as input a time period i, *params*, $S_{A,i}$ and a ciphertext C. It returns the corresponding plaintext $M \in \mathcal{M}$.

2.2 Security Model

In this subsection, we give the the security model defined in Indistinguishability of Encryption Against Adaptive Chosen Ciphertext Attacker (IND-CCA2) game which is conducted between a challenger $\mathcal{S}$ and an adversary $\mathcal{A}$. In our scheme, we define two kind adversaries *TypeI* adversary ($\mathcal{A}_I$) and *TypeII* adversary ($\mathcal{A}_{II}$): $\mathcal{A}_I$ represents an external attacker, who can not access the *master-key* and *helper-key*. We allow $\mathcal{A}_I$ can replace the public key for any entity with a value of its choice since the lack of authentication for the public key in our scheme; $\mathcal{A}_{II}$ represents the malicious KGC, who can access the *master-key*.

We prohibit $\mathcal{A}_{II}$ from replacing the public key. First, we give a list of oracles that a general adversary in our scheme may carry out, then we define the IND-CCA2 game of the CL-KIE scheme for two kinds of adversaries respectively.

The list of oracles that a general adversary in CL-KIE may carry out:

– **Partial-Private-Key-Queries(PPK-Queries):** If necessary, $\mathcal{A}$ makes **PPK-Queries** on the identity ID_A, $\mathcal{S}$ returns the partial private key D_A associated with ID_A to $\mathcal{A}$.
– **Helper-Key-Queries(HK-Queries):** $\mathcal{A}$ makes **HK-Queries** on identity ID_A at a time period i, $\mathcal{S}$ returns the helper key $HK_{A,i}$ to $\mathcal{A}$.
– **Secret-Value-Queries(SV-Queries):** If necessary, $\mathcal{A}$ makes **SV-Queries** on the identity ID_A, $\mathcal{S}$ returns the secret value x_A associated with ID_A to $\mathcal{A}$.
– **Public-Key-Queries(PK-Queries):** $\mathcal{A}$ makes **PK-Queries** on the identity ID_A, $\mathcal{S}$ returns the helper key P_A to $\mathcal{A}$.
– **Public-Key-Replace(PK-Replace):** If necessary, $\mathcal{A}$ can repeatedly make **PK-Replace** to set the public key P_A for any value of its choice.
– **Decryption-Queries(Dec-Queries):** $\mathcal{A}$ makes **Dec-Queries** for a ciphertext C on identity ID_A at a time period i. If the recovered redundancy in M is valid, $\mathcal{S}$ returns the associated plaintext M to $\mathcal{A}$.

The IND-CCA2 game for the CL-KIE scheme can be defined between two different Adversaries ($\mathcal{A}_I$ and $\mathcal{A}_{II}$) and the challenger $\mathcal{S}$ as follows:

– **Chosen Ciphertext Security for CL-KIE on $\mathcal{A}_I$**
  - **Setup:** The challenger $\mathcal{S}$ takes as input a security parameter k and execute the **Setup** algorithm. It returns *params* expect *master-key* and *helper-key* to $\mathcal{A}_I$.
  - **Phase 1:** $\mathcal{A}_I$ can access a sequence of oracles: **PPK-Queries**, **HK-Queries**, **SV-Queries**, **PK-Replace**, **Dec-Queries**. These queries may be requested adaptively, and restricted by the rule of adversary behavior.
  - **Challenge:** $\mathcal{A}_I$ outputs two equal-length plaintext $M_0^*, M_1^* \in M$, associated with the challenge identity ID_A^* and a time period i^*. The challenger $\mathcal{S}$ picks a random number $b \in \{0,1\}$, and generates C^* in relation to (i^*, M_b^*, ID^*). C^* is delivered to $\mathcal{A}_I$ as a target challenge.
  - **Phase 2:** $\mathcal{A}_I$ continues to access a sequence of oracles as in Phase 1, and $\int$ responds these queries as in Phase 1.
  - **Guess:** At the end, $\mathcal{A}_I$ outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b = b'$. We define $\mathcal{A}_I's$ advantage in this game to be $Adv(\mathcal{A}_{\mathcal{I}}) = 2(Pr[b = b'] - \frac{1}{2})$.

  There are a few restrictions on the $\mathcal{A}_I$ as follows:
  - $\mathcal{A}_I$ is not allowed to extract the private key on ID_A^*.
  - If the public key has been replaced, $\mathcal{A}_I$ is not allowed to request **PPK-Queries** and **HK-Queries** simultaneously.
  - $\mathcal{A}_I$ is not allowed to do the following concurrently: to replace the public key on ID_A^* in Phase 1 and request the **PPK-Queries** and **HK-Queries** on ID_A^* simultaneously at any moment.
  - In Phase 2, $\mathcal{A}_I$ is not allowed to request **Dec-Queries** on ID_A^*.

- **Chosen Ciphertext Security for CL-KIE on $\mathcal{A}_{II}$**
  - **Setup:** The challenger $\int$ takes as input a security parameter k and execute the **Setup** algorithm. It returns *params* to $\mathcal{A}_{II}$.
  - **Phase 1:** $\mathcal{A}_{II}$ can access a sequence of oracles: **PPK-Queries**, **HK-Queries**, **Dec-Queries**. These queries may be requested adaptively, and restricted by the rule of adversary behavior.
  - **Challenge:** $\mathcal{A}_{II}$ outputs two-equal length plaintext $M_0^*, M_1^* \in M$, associated with the challenge identity ID_A^* and a time period i^*. The challenge $\mathcal{S}$ picks a random number $b \in \{0,1\}$, and generate C^* in relation to (i^*, M_b^*, ID^*). C^* is delivered to $\mathcal{A}_{II}$ as a target challenge.
  - **Phase 2:** $\mathcal{A}_{II}$ continues to access a sequence of oracles as in Phase 1, and $\mathcal{S}$ responds these queries as in Phase 1.
  - **Guess:** At the end, $\mathcal{A}_{II}$ outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b = b'$. We define $\mathcal{A}'_{II}s$ advantage in this game to be $Adv(\mathcal{A}_{\mathcal{II}}) = 2(Pr[b = b'] - \frac{1}{2})$.

  There are a few restrictions on the $\mathcal{A}_{II}$ as follows:
  - $\mathcal{A}_{II}$ is not allowed to replace the public key.
  - $\mathcal{A}_{II}$ cannot extract the private key on ID_A^* at any moment.
  - In Phase 2, $\mathcal{A}_I$ is not allowed to request **Dec-Queries** on ID_A^*.

3 KI-CLPKE Scheme

3.1 Bilinear Pairing

- **Bilinear Pairing**

  Let $\mathbb{G}_1$ denotes a cyclic additive group of order q for some large prime q, let $\mathbb{G}_2$ be a cyclic multiplicative group of the same order q, We can make use of a bilinear map:$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ above these two groups which must satisfy the following properties:
  - **Bilinearity**

    $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$

    $\hat{e}(P_1 + P_2, Q) = \hat{e}(P_1, Q)\hat{e}(P_2, Q)$

    $\hat{e}(P, Q_1 + Q_2) = \hat{e}(P, Q_1)\hat{e}(P, Q_2)$
  - **Non-Degeneracy**

    If P is the generator for $\mathbb{G}_1$, $\hat{e}(P, P)$ is the generator for $\mathbb{G}_2$.
  - **Computability**

    For $\forall P, Q \in \mathbb{G}_1$, $\hat{e}(P, Q)$ can be computed through a efficient algorithm in a polynomial-time.
- **Bilinear Diffie-Hellman(BDH) Problem**

  BDHP is for $a, b, c \in \mathbb{Z}_q$, given $P, aP, bP, cP \in \mathbb{G}_1$, to compute abc which satisfies $\hat{e}(P, Q)^{abc} \in \mathbb{G}_2$.

3.2 Construction

- **Setup:** We can randomly select a security parameter $k \in \mathbb{Z}^+$, the Setup algorithm works as follows:

Step1: Pick two groups $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, \times)$ of the same prime order q where $|q| = k$. Choose a generator P over $\mathbb{G}_1$ randomly, we can get a bilinear map $\hat{e} : G_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

Setp2: Choose a random $s \in \mathbb{Z}_q$ to compute $P_{pub} = sP$, the corresponding s can be regarded as the *master-key*: $M_{mk} = s$;
Choose a random $w \in \mathbb{Z}_q$ to compute $P_{hk} = wP$, the corresponding w can be regarded as the *helper-key*: $M_{hk} = w$.

Setp3: For some integer $n > 0$,we can select three cryptographic hash functions:
- $H_1 : \{0,1\}^n \rightarrow \mathbb{G}_1$
- $H_2 : \{0,1\}^n \times \mathbb{Z}^+ \rightarrow \mathbb{G}_1$
- $H_3 : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \{0,1\}^n$

The system parameters $params = (\mathbb{G}_1, \mathbb{G}_2, p, \hat{e}, n, P, P_{pub}, P_{hp}, H_1, H_2, H_3)$. The master key $M_{mk} = s$ and the master helper key $M_{hk} = w$.
The message space is $\mathcal{M} = \{0,1\}^n$, the ciphertext space is $\mathcal{C} = \{0,1\}^n \times \{0,1\}^n$, the randomness space is $\mathcal{R} = \{0,1\}^n$.

- **SecretValExtract**$(params, ID_A)$: Given an identity ID_A and $params$, the algorithm outputs a random $x_A \in Z_q$ as the secret value for the entity A.
- **PartialKeyExtrat**$(params, M_{mk}, ID_A)$: Given an identity $ID_A \in \{0,1\}^*$ of the entity A, $params$ and M_{sk}, the algorithm computes $D_A = sH_1(ID_A)$.
- **HelperKeyUpdate**$(i, ID_A, M_{hk}, params)$: Given an identity string ID_A and a time period $i \in \{0, \ldots, n-1\}$, the helper generates a helper key $HK_{A,i}$ which can help the private key to be updated at the time period $i \in \{0, \ldots, n-1\}$:

$$HK_{A,i} = wH_2(ID_A, i)$$

- **PrivateKeyExtract**$(i, ID_A, HK_{A,i}, params, D_A, x_A)$: Given an identity ID_A, At a time period $i \in \{0, \ldots, n-1\}$, the private key is generated as:

$$\begin{aligned} S_{A,i} &= x_A H_1(ID_A) + D_A + HK_{A,i} \\ &= x_A H_1(ID_A) + sH_1(ID_A) + wH_2(ID_A, i) \end{aligned}$$

the value $S_{A,i-1}$ will be deleted subsequently.

- **PublicKeyExtract**$(params, x_A, ID_A)$: Given $params$ and x_A, the algorithm outputs $P_A = \langle X_A, Y_A \rangle = \langle x_A P, x_A sP \rangle$.
- **Encrypt**$(i, params, ID_A, P_A, M)$**:** At a time period $i \in \{0, \ldots, n-1\}$, to encrypt a plaintext $M \in \{0,1\}^n$, the algorithm does:
  1. Check whether the equality $\hat{e}(X_A, sP) = \hat{e}(Y_A, P)$ holds or not. If not, output $\perp$ and abort encryption.
  2. Select a random $r \in Z_q$, $U = rP$.
  3. Compute $\xi = \hat{e}(X_A, rH_1(ID_A))\hat{e}(P_{pub}, rH_1(ID_A))\hat{e}(P_{hk}, rH_2(ID_A, i))$.
  4. Output the ciphertext: $C = \langle i, U, M \oplus H_3(U, \xi) \rangle$.
- **Decrypt**$(i, params, S_{A,i}, C)$**:** Received the ciphertext $C = \langle i, U, V \rangle$ at the time period $i \in \{0, \ldots, n-1\}$, the algorithm performs the following steps:

1. Compute $\xi' = \hat{e}(U, S_{A,i})$.
2. Compute $M' = V \oplus H_3(U, \xi')$.
3. If the recovered redundancy in M is valid, then accept M' the plaintext.

4 Analysis

4.1 Security Proof

Theorem 1. *Let hash functions H_1, H_2, H_3 be random oracles. In IND-CCA2 game, the CL-KIE scheme against chosen ciphertext attacks for TypeI adversary is secure in the random oracle model, considering the BDH assumption.*

Proof. We first deal with the *TypeI* adversary A_I. For the first type adversary A_I adversary is an external attacker who can not get the *master-key* and *helper-key*, Given a BDH problem (P, aP, bP, cP), we can construct a challenger $\mathcal{S}$ to compute $\hat{e}(P,P)^{abc}$ by making use of A_I as an adversary. When games begin, $\mathcal{S}$ sets $P_{pub} = aP$ as an instance of BDH problem and simulates hash functions as random oracles. During the simulation, $\mathcal{S}$ needs to guess every bit in target plaintext M_1^* with a time period i^*. $\mathcal{S}$ will set $H_1(ID_A^*) = bP$, $H_2(ID_A^*, i^*) = (h^*,_{i^*} P)$, $V^* = H_3(U^*, \xi^*) = H_3(cP, \xi^*)$. In the challenge phase, $\mathcal{S}$ returns a simulated ciphertext $C^* = (i^*, U^*, V^*)$, which implies the parameter ξ^* is defined as:

$$\begin{aligned}\xi^* &= \hat{e}(X_A, rH_1(ID_A^*))\hat{e}(P_{pub}, rH_1(ID_A^*))\hat{e}(P_{hk}, rH_2(ID_A^*, i^*))\\ &= \hat{e}(x_A rP, bP)\hat{e}(bP, acP)\hat{e}(wP, r(h^*,_{i^*} P))\\ &= \hat{e}(P,P)^{abc}\hat{e}(aP, cP)^{x_A}\hat{e}(wP, (h^*,_{i^*})cP)\end{aligned}$$

Above all, $\mathcal{S}$ can get the solution to the BDH problem: $\hat{e}(P,P)^{abc} = \xi^*(\hat{e}(aP,cP)^{x_A}\ \hat{e}(wP,(h^*,_{i^*})cP))^{-1}$. So that,we can prove the security of the scheme for the *TypeI* adversary through this reduction.

Theorem 2. *Let hash functions H_1, H_2, H_3 be random oracles. In IND-CCA2 game, the CL-KIE scheme against chosen ciphertext attacks for TypeII adversary is secure in the random oracle model, considering the BDH assumption.*

Proof. We secondly deal with the *TypeII* adversary A_{II}. For the *TypeII* adversary is a malicious KGC attacker, Given a BDH problem (P, aP, bP, cP), we can construct a challenger $\mathcal{S}$ to compute $\hat{e}(P,P)^{a,b,c}$ by making use of $\mathcal{A}_{II}$ as an adversary. When games begin, $\mathcal{S}$ sets $X_A = aP$ as an instance of the BDH problem and simulates hash functions as random oracles. During the simulation, $\mathcal{S}$ needs to guess every bit in target plaintext M_2^* with a time period i^*. $\mathcal{S}$ will set $H_1(ID_A^*) = bP$, $H_2(ID_A^*, i^*) = (h^*,_{i^*} P)$, $V^* = H_3(U^*, \xi^*) = H_3(cP, \xi^*)$. In the challenge phase, $\mathcal{S}$ returns a simulated ciphertext $C^* = (i^*, U^*, V^*)$, which implies the parameter ξ^* is defined as:

$$\begin{aligned}\xi^* &= \hat{e}(X_A, rH_1(ID_A^*))\hat{e}(P_{pub}, rH_1(ID_A^*))\hat{e}(P_{hk}, rH_2(ID_A^*, i^*))\\ &= \hat{e}(aP, bcP)\hat{e}(bP, cP)^s\hat{e}(wP, r(h^*,_{i^*} P))\\ &= \hat{e}(P,P)^{abc}\hat{e}(bP, cP)^s\hat{e}(wP, (h^*,_{i^*})cP)\end{aligned}$$

Above all, $\mathcal{S}$ can get the solution to the BDH problem: $\hat{e}(P,P)^{abc} = \xi^*(\hat{e}(bP,cP)^s \ \hat{e}(wP,(h^*,_{i^*})cP))^{-1}$. So that,we can prove the security of the scheme for the $TypeII$ adversary through this reduction.

4.2 Performance Comparison

We compare the major computational cost of our scheme with certificateless public key cryptography proposed by Al-Riyami and Paterson [2] in Table 1. We assume both the two schemes are implemented on $\mid \mathbb{G}_1 \mid= 160$ bits, $\mid \mathbb{G}_2 \mid= 1024$ bits, $\mid p \mid= 160$ bits and hash value $= 160$ bits. We denote by M the point multiplication in $\mathbb{G}_1$, E the exponentiation in $\mathbb{G}_2$ and P the pairing computation. The other computations are trivial so we can omit them.

Table 1. Performance comparison

	CL-PKE	CL-KIE
PartialKeyExtract	M	$3M$
PubilicKeyExtract	$2M$	$2M$
Encrypt	$M+P+E$	$4M+3P$
Decrypt	P	P

From Table 1, we can see that in the PublicKeyExtract and Decrypt phase our scheme has the same computational cost as the CL-PKE scheme; However in the PrivateKeyExtract and Encrypt phase our scheme is less efficient on executed time compared with the CL-PKE scheme. Because the private key consisting of three parts in our scheme is more complicated than it in CL-PKE. The additional composition of the private key in our scheme can be updated with the time period changed, so our scheme provides the extra security capability that it can alleviate the problem for leakage of private key in hostile practical environment. This is a trade-off between efficiency and security capability.

5 Conclusion

In this paper, we firstly formalized the definition of a CL-KIE scheme based on the bilinear pairing and constructed the security model of the CL-KIE scheme for two different adversaries in IND-CCA2 game respectively. Then we gave the concrete construction of the CL-KIE scheme. After that, we proved the security of our scheme against the IND-CCA2 attacks in the random oracle under the BDH assumption. Finally, we compared the CL-KIE scheme with the CL-PKE scheme both on the security capacity and efficiency. Our scheme can achieve key-escrow free and key-exposure resilience in hostile practical environments.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant 61003230, Grant 61370026, Grant 61133016 and Grant 61202445.

References

1. Youngblood, C.: An Introduction to Identity-Based Cryptography. CSEP 590TU (2005)
2. 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)
3. Dodis, Y., Katz, J., Xu, S., Yung, M.: Key-insulated public key cryptosystems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, p. 65. Springer, Heidelberg (2002)
4. 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)
5. Dent, A.W., Libert, B., Paterson, K.G.: Certificateless encryption schemes strongly secure in the standard model. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 344–359. Springer, Heidelberg (2008)
6. Libert, B., Quisquater, J.-J.: On constructing certificateless cryptosystems from identity based encryption. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 474–490. Springer, Heidelberg (2006)
7. Liu, J.K., Au, M.H., Susilo, W.: Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS 2007), pp. 302–311. ACM, New York (2007)
8. Sun, Y., Li, H.: Short-ciphertext and BDH-based CCA2 secure certificateless encryption. Sci. China Inf. Sci. **53**(10), 2005–2015 (2010)
9. Yang, W., Zhang, F., Shen, L.: Efficient certificateless encryption withstanding attacks from malicious KGC without using random oracles. Secur. Commun. Netw. **7**(2), 445–454 (2014)
10. Bellare, M., Palacio, A.: Protecting against key exposure: strongly key-insulated encryption with optimal threshold. Appl. Algebra Eng. Commun. Comput. **16**(6), 379–396 (2006)
11. Hsu, C., Lin, H.: An identity-based key-insulated encryption with message linkages for peer-to-peer communication network. TIIS **7**(11), 2928–2940 (2013)
12. Hanaoka, Y., Hanaoka, G., Shikata, J., Imai, H.: Unconditionally secure key insulated cryptosystems: models, bounds and constructions. In: Deng, R.H., Qing, S., Bao, F., Zhou, J. (eds.) ICICS 2002. LNCS, vol. 2513, pp. 85–96. Springer, Heidelberg (2002)
13. Qiu, W., Zhou, Y., Zhu, B., Zheng, Y., Wen, M., Gong, Z.: Key-insulated encryption based key pre-distribution scheme for WSN. In: Park, J.H., Chen, H.-H., Atiquzzaman, M., Lee, C., Kim, T., Yeo, S.-S. (eds.) ISA 2009. LNCS, vol. 5576, pp. 200–209. Springer, Heidelberg (2009)
14. Rivestm, L.R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
15. El Gamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
16. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)