Malaysian Journal of Mathematical Sciences 9(S) June: 53-69 (2015)
Special Issue: The 4th International Cryptology and Information Security Conference 2014 (Cryptology 2014)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

Certificateless Public Key Encryption (CL-PKE) Scheme Using Extended Chebyshev Polynomial over the Finite Field $\mathbb{Z}_p$

[1]Mohammed Benasser Algehawi and [2*]Azman Samsudin

[1]Libyan Defense Ministry, Tripoli, Libya

[2]School of Computer Sciences, Universiti Sains Malaysia, 11800 USM Penang, Malaysia

E-mail: malgehawi@yahoo.com and azman@cs.usm.my

*Corresponding author

ABSTRACT

In this paper, we introduce a new alternative model for a Certificateless Public Key Encryption (CL-PKE) scheme. Our proposed CL-PKE scheme uses the Identity Based Encryption (IBE) scheme extended over the finite field $\mathbb{Z}_p$ to generate its encryption and decryption keys. The proposed system is applicable, secure, and reliable.

1. INTRODUCTION

Certificateless Public Key Encryption is a scheme in which there is no intervention by the Key Generation Center (KGC) during the generation of the encryption and decryption keys. The CL-PKE scheme was introduced by Al-Riyami and Paterson, 2003. The CL-PKE scheme uses the Identity Based Encryption (IBE) scheme introduced by Boneh and Franklin, 2003 for the generation of the users' encryption and decryption keys. The decryptor receives a partial-private-key from the KGC through a secure channel. Unlike the Private Key Generator (PKG) in the IBE scheme introduced by Boneh and Franklin, 2003, the KGC has no access to the users' secret information and the only role of the KGC is to generate the partial-private-key. The received partial-private-key will be used by the decryptor to generate its actual private-key. The security of the CL-PKE scheme against the

Indistinguishable Chosen Ciphertext Attack (IND-CCA2) has been proved (Al-Riyami and Paterson, 2003). The strength of the scheme is inherited from the bilinear Diffie-Hellman hard problem (BDHP), the hard problem that is used in the parameters generation of this scheme. The formal definition of the CL-PKE concept and its aspects are briefly explained in the following subsection based on the CL-PKE description (Al-Riyami and Paterson, 2003).

2. BASIC AND FULLY SECURE CL-PKE SCHEME BASED ON PAIRINGS

The execution of the basic CL-PKE scheme starts with the introduction of the security parameter k and the BDHP parameter generator $\mathcal{G}$ to the setup algorithm to generate the system parameters. The basic CL-PKE scheme is performed as follows:

A. Setup: This algorithm generates the master-key s and the public parameters; $g = \langle G_1, G_2, \hat{e}, n, P, P_0, H_1, H_2 \rangle$.

This algorithm runs as follows:

1. Generate two groups G_1 and G_2 of prime order q and an admissible map $\hat{e}: G_1 \times G_1 \to G_2$.
2. Choose a random generator $P \in G_1$.
3. Choose a random master-key $s \in \mathbb{Z}_q^*$ and calculate the public key $P_0 = sP$.
4. Select two hash functions $H_1: \{0,1\}^* \to G_1^*$ and $H_2: G_2 \to \{0,1\}^n$ for some bit-length n.
5. Choose the message space $\mathcal{M} = \{0,1\}^n$ and the ciphertext space $C = G_1 \times \{0,1\}^n$.

B. Partial-Private-Key-Extract: This algorithm generates the partial private key for party A as follows:

1. Takes as an input the master-secret key s and the identity of party A, $ID_A \in \{0,1\}^*$.
2. Map the identity ID_A to the group G_1^* such that $Q_A = H_1(ID_A) \in G_1^*$.
3. Finally, generate party A's partial private key, $D_A = sQ_A \in G_1^*$.

C. Set-Secret-Value: This algorithm outputs the randomly selected value $x_A \in \mathbb{Z}_q^*$ as the secret value of party A by taking the public parameters g and A's identity ID_A as the inputs.

D. Set-Private-Key: This algorithm generates party A's private key as follows:

1. Takes as input the public parameters g, A's partial private key D_A, and A's secret value x_A.
2. Generates A's private key $S_A \in G_1^*$, such that $S_A = x_A D_A = x_A s Q_A$.

E. Set-Public-Key: This algorithm generates A's public key as follows:

1. Takes as input the public parameters g and A's secret value x_A.
2. Generate A's public key $P_A = \langle X_A, Y_A \rangle$, such that $X_A = x_A P$ and $Y_A = x_A P_0 = x_A sP$.

F. Encryption: This algorithm encrypts the message M as follows:

1. First checks that $X_A, Y_A \in G_1^*$ and make sure that $\hat{e}(X_A, P_0) = \hat{e}(Y_A, P)$. If so then encrypt the message M, otherwise output $\perp$ and abort the encryption.
2. Set $Q_A = H_1(ID_A) \in G_1^*$.
3. Select a random value $r \in G_1^*$.
4. Compute the ciphertext $C = (U, V)$, where, $U = rP$ and $V = M \oplus H_2(\hat{e}(Q_A, Y_A)^r)$.

G. Decryption: This algorithm decrypts the ciphertext. Upon receiving the ciphertext $C = (U, V)$, the ciphertext will be decrypted using A's private key S_A as follows:

$$\begin{aligned} M &= V \oplus H_2\big(\hat{e}(S_A, U)\big) \\ &= V \oplus H_2\big(\hat{e}(x_A s Q_A, rP)\big) \\ &= V \oplus H_2(\hat{e}(Q_A, x_A sP)^r) \\ &= V \oplus H_2(\hat{e}(Q_A, Y_A)^r) \\ &= M. \end{aligned}$$

The adaptation of the CL-PKE scheme that is fully secure against IND-CCA2 is obtained by incorporating the technique proposed by Fujisaki and Okamoto in 1999. After adding the two extra hash functions H_3 and H_4

as random oracles, this fully secure CL-PKE scheme will be as the following (Al-Riyami and Paterson, 2003):

The setup algorithm will be as in the basic CL-PKE scheme except that two extra random oracles, $H_3: \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$ and $H_4: \{0,1\}^n \rightarrow \{0,1\}^n$ will be added.

The new public parameters will be $g = \langle G_1, G_2, \hat{e}, n, P, P_0, H_1, H_2, H_3, H_4 \rangle$, the message space will be the same as in the basic scheme, and the ciphertext space will be defined as $C = G_1 \times \{0,1\}^{2n}$. The encryption and the decryption algorithms will be executed as follows:

A. Encryption:

1. First checks that $X_A, Y_A \in G_1^*$ such that $\hat{e}(X_A, P_0) = \hat{e}(Y_A, P)$. If the equivalence holds then encrypt the message otherwise output $\perp$ and abort the encryption.
2. Set $Q_A = H_1(ID_A) \in G_1^*$.
3. Select $\sigma \in \{0,1\}^n$.
4. Set $r = H_3(\sigma, M)$.
5. Send the ciphertext $C = [U, V, W]$ where $U = rP$, $V = H_2(\hat{e}(Q_A, Y_A)^r)$, and $W = M \oplus H_4(\sigma)$.

B. Decryption: Party B will decrypt C as follows:

1. Calculate $\sigma = V \oplus H_2\big(\hat{e}(U, S_A)\big)$.
2. Calculate $M = W \oplus H_4(\sigma)$.
3. Set $r = H_3(\sigma, M)$. If $U \neq rP$ then output $\perp$ and reject the ciphertext, otherwise accept the decrypted message, M.

In this paper, we will present a new CL-PKE scheme that uses the same method used by Al-Riyami and Paterson in 2003. The proposed CL-PKE uses the mechanism of the IBE scheme as introduced by Algehawi and Samsudin in 2010 for the generation of the users' encryption and decryption keys. The strength of the IBE scheme (Algehawi & Samsudin 2010) is based on Chebyshev map bilinearity and the discrete Chebyshev hard problem (DCP) which arises after extending the map over the finite field $\mathbb{Z}_p$. To demonstrate its validity for CL-PKE purposes, the characteristics of the Chebyshev polynomial extended over the finite field $\mathbb{Z}_p$ are explained in the following section.

3. CHEBYSHEV POLYNOMIAL

In Sections 3.1 and 3.2, the Chebyshev polynomial will be briefly explained, both in the real domain $\mathbb{R}$ and the extended finite field $\mathbb{Z}_p$. The pertinent definitions and properties of the Chebyshev polynomial extended over the finite field $\mathbb{Z}_p$ will be given. These properties strengthen the proposed scheme security.

3.1 Chebyshev Polynomial in the Real Domain

The Chebyshev polynomial in the real domain has some properties that make it usable for cryptography purposes. The Chebyshev polynomial in the real domain has been defined in many publications (Amig et al., 2008; Xiao et al., 2007; Yoon and Yoo, 2008). The definition of the n^{th} term T_n of the Chebyshev polynomial can be written as follows:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad (1)$$

where $n \in \mathbb{N}$, $n \geq 2$, $x \in \mathbb{R}$, and the initial terms are $T_0(x) = 1$ and $T_1(x) = x$.

The Chebyshev polynomial in the real domain has been proven to be weak for the purpose of cryptography (Bergamo et al., 2004; Han, 2008; Xiang et al., 2009). The small range of the real domain, which is [-1,1], results in weakening the DCP hard problem obtained from the one-way function of the Chebyshev polynomial.

3.2 Chebyshev Polynomial Extended Over the Finite Fields $\mathbb{Z}_p$

The extension of the Chebyshev polynomial over the finite field $\mathbb{Z}_p$ has been discussed in several places in the literature (Algehawi and Samsudin, 2010; Bi and Wang, 2009; Maze, 2003; Wang et al., 2008; Wang and Zhao, 2010). This extension does not alter the bilinear property of the Chebyshev polynomial, but it does strengthen the DCP hard problem. This extension and its details, including the one way function of the extended Chebyshev polynomial, have been explained extensively (Algehawi and Samsudin, 2010; Bi and Wang, 2009; Maze, 2003; Wang et al., 2008; Wang and Zhao, 2010). The DCP of the extended Chebyshev polynomial has been proven to be as hard as the Discrete Logarithmic Problem (DLP), such that can be used safely for cryptographic purposes (Bi and Wang, 2009; Wang et al., 2008; Maze, 2003). The extended Chebyshev polynomial equation is defined as follows:

$$T_n(x) = \left(2xT_{n-1}(x) - T_{n-2}(x)\right) (mod\ p), \tag{2}$$

where the initial terms are $T_0(x) = 1\ (mod\ p)$ and $T_1(x) = x\ (mod\ p)$.

The semi group property (bilinearity) of the extended Chebyshev polynomial can be derived as follows:

$$\begin{aligned} T_r\left(T_s(x)\right) mod\ p &= T_r(T_s(x) mod\ p))\ mod\ p \\ &= T_{rs}(x)\ mod\ p \\ &= T_s(T_r(x)(mod\ p))\ mod\ p \\ &= T_{rs}(x)\ mod\ p \\ &= T_s\left(T_r(x)\right)\ mod\ p, \end{aligned} \tag{3}$$

where $r, s, x \in \mathbb{Z}_p^*, and\ r, s \geq 2$.

From Eq. 2, the representation of the secret information n can be written as a product of primes, $n = S_1^{k_1} \times S_2^{k_2} \times \cdots \times S_m^{k_m}$, where $S_1, S_2, \cdots, S_m$ are prime numbers and $k_1, k_2, \cdots, k_m$, $m \in \mathbb{Z}^+$. Based on this expression, Eq. 2 can be represented as the following:

$$\begin{aligned} T_n(x) &= T_{S_1^{k_1} \times S_2^{k_2} \times \cdots \times S_m^{k_m}}(x)(mod\ p) \\ &= T_{S_1^{k_1}}\left(T_{S_2^{k_2}}\left(\cdots T_{S_m^{k_m}}(x)\cdots\right)\right)(mod\ p). \end{aligned} \tag{4}$$

If $T_n(x)$ and x are known, to find n, one has to compute $T_r(x)$ for all $r = S_1^{k_1} \times S_2^{k_2} \times \cdots \times S_l^{k_l}$, $l \in \mathbb{Z}^+$ and find the r for which $T_n(x) = T_r(x)$ process is infeasible for large n.

4. IBE SCHEME USING THE CHEBYSHEV POLYNOMIAL EXTENDED OVER FINITE FIELD $\mathbb{Z}_p$

The IBE scheme using the Chebyshev polynomial extended over finite field $\mathbb{Z}_p$ consists of four main algorithms, Setup, Extraction, Encryption, and Decryption. The basic IBE scheme is explained below:

Based on Eq. 4, the basic IBE scheme is executed as follows (Algehawi and Samsudin, 2010):

A. Setup (by PKG):

1. Choose a large prime number p, a large (≥ 2) secret number $s \in \mathbb{Z}_p^*$ as the secret key, and a global parameter $G \in \mathbb{Z}_p^*$.
2. Use Eq. 4 to calculate P_{Pub} as follows:
$$P_{Pub} = T_s(G)\ (mod\ p), \quad (5)$$
where $\{p, G, P_{Pub}\}$ are the public parameters of the PKG.
3. Publish the public parameters $(p, G, P_{Pub}, H_1, H_2)$ where $H_1: \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2: \mathbb{Z}_p^* \rightarrow \{0,1\}^n$.

B. Extraction (by PKG): The PKG extracts the shared key, K_{shared}, for party B based on Eq. 4 as follows:
$$K_{shared} = T_s\left(T_{B_{Pub}}(A_{Pub})\ mod\ p\right)\ mod\ p, \quad (6)$$
where $B_{Pub} = H_1(ID_B)$ is party's B identity.

C. Encryption (by Party *A*):

1. Choose a large integer $A_{Pri} \in \mathbb{Z}_p^*$ where $A_{Pri} \geq 2$ as its private key.
2. Calculate the public key, A_{Pub}, using Eq. 4 as follows:
$$U = A_{Pub} = T_{A_{Pri}}(G)\ (mod\ p). \quad (7)$$
3. Calculate the shared secret key, K_{shared}, as follows:
$$K_{shared} = T_{A_{Pri}}\left(T_{B_{Pub}}(P_{Pub})\ (mod\ p)\right)\ (mod\ p), \quad (8)$$
where $B_{Pub} = H_1(ID_B)$ is Party B's identity.
4. Encrypt message M using the shared secret key, K_{shared}, and produce the ciphertext component $V = M \oplus H_2(K_{shared})$. Send the ciphertext $C = (U, V)$ to party B.

D. Decryption algorithm (by Party *B*):

1. Receive the shared key, K_{shared}, from the PKG through a secure channel.
2. Decrypt message M using the shared secret key, K_{shared}, as follows:
$$M = V \oplus H_2(K_{shared}).$$

To obtain the fully secure version of this scheme, another two hash functions, H_3 and H_4, will be added to the scheme as random oracles. The details of the fully secure version, its security proofs against IND-CCA2, and its practicality have been explained (Algehawi and Samsudin, 2010).

5. THE PROPOSED CL-PKE SCHEME

The new CL-PKE scheme using the Chebyshev polynomial extended over $\mathbb{Z}_p$ is discussed in this section. Section 5 explains the proposed basic CL-PKE scheme which uses the IBE concept introduced by Algehawi and Samsudin, 2010.

The basic version of the proposed CL-PKE scheme consists of seven algorithms:

A. Setup Algorithm:

1. Randomly choose a large prime p.
2. Choose a random master-key $s \in \mathbb{Z}_p^*$, where $s \geq 2$ and a global parameter $G \in \mathbb{Z}_p^*$.
3. Using Eq. 4, calculate the public key P_0 as follows:
$$P_0 = T_s(G)\ mod\ p. \tag{9}$$
4. Select two hash functions $H_1: \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and $H_2: \mathbb{Z}_p^* \rightarrow \{0,1\}^n$.
5. Choose the message space $M = \{0,1\}^n$, and ciphertext space $C = \mathbb{Z}_p^* \times \{0,1\}^n$, for some integer n.
6. Publish the public parameters, $g = \langle p, P_0, G, n, H_1, H_2 \rangle$.

B. Partial-Private-Key-Extract Algorithm: Generates the partial private key for party A as follows:

1. Takes as input, the identity of party A, $ID_A \in \mathbb{Z}_p^*$, and party's B public key B_{pub}.
2. Set $Q_A = H_1(ID_A) \in \mathbb{Z}_p^*$.
3. Use Eq. 4 to extract party A's partial private key as follows:
$$\begin{aligned} D_A &= T_s\left(T_{Q_A}\left(B_{pub}\right) mod\ p\right) mod\ p, \\ &= T_s\left(T_{Q_A}\left(T_{B_{pri}}(G)\right)\right)\ (mod\ p). \end{aligned} \tag{10}$$

C. Set-Secret-Value Algorithm: Select a random value $x_A \in \mathbb{Z}_p^*$ as the secret value for party A.

D. Set-Shared-Secret-Key Algorithm: Using Eq. 4, generate the shared secret key K_{shared} by taking as input the public parameters g, A's partial private key D_A, and A's secret value x_A as follows:

$$\begin{aligned} K_{shared} &= T_{x_A}(D_A)\ mod\ p, \\ &= T_{x_A}\left(T_s\left(T_{Q_A}\left(T_{B_{pri}}(G)\ mod\ p\right)\right)\right)\ mod\ p. \end{aligned} \tag{11}$$

E. Set-Public-Key Algorithm: Use Eq. 4 to generate A's public key P_A by using Eq. 4 as follows:

1. Take as input the public parameters g and A's secret value x_A.
2. Calculates the tuple, $P_A = \langle X_A, Y_A, Z_A \rangle$, where:

$$X_A = T_{x_A}(G)\ mod\ p. \tag{12}$$

$$\begin{aligned} Y_A &= T_{x_A}(U)\ mod\ p \\ &= T_{x_A}\left(T_{B_{Pri}}(G)\ mod\ p\right)\ mod\ p. \end{aligned} \tag{13}$$

$$\begin{aligned} Z_A &= T_{x_A}(P_0)\ mod\ p \\ &= T_{x_A}(T_s(G)\ mod\ p)\ mod\ p. \end{aligned} \tag{14}$$

F. Encryption Algorithm: The encryption is executed as follows:

1. First check that $X_A, Y_A, Z_A \in \mathbb{Z}_p^*$, and make sure that $T_{B_{Pri}}(X_A)\ mod\ p = Y_A$. If so, then encrypt the message; otherwise, output $\perp$, and abort the encryption.
2. Set $Q_A = H_1(ID_A) \in \mathbb{Z}_p^*$.
3. Choose a large integer $B_{Pri} \in \mathbb{Z}_p^*$ where $B_{Pri} \geq 2$ as the private key.
4. Generate the public key,

$$U = B_{Pub} = T_{B_{Pri}}(G)\ mod\ p. \tag{15}$$

5. Compute the ciphertext $C = (U, V)$, where, $V = M \oplus H_2(K_{shared})$ and the shared key K_{shared} is computed as follows:

$$\begin{aligned} K_{shared} &= T_{B_{Pri}}\left(T_{Q_A}(Z_A)\ mod\ p\right)\ mod\ p \\ &= T_{B_{Pri}}\left(T_{Q_A}\left(T_{x_A}(P_0)\ mod\ p\right)\ mod\ p\right)\ mod\ p \\ &= T_{B_{Pri}}\left(T_{Q_A}\left(T_{x_A}(T_s(G))\right)\right)(mod\ p). \end{aligned} \tag{16}$$

G. Decryption Algorithm: Upon receiving the ciphertext $C = (U, V)$, using the shared secret key K_{shared} generated earlier by Eq. 11, the ciphertext is decrypted as follows: $M = V \oplus H_2(K_{shared})$.

5.1 Working Example

Table 1 shows a working example of the basic CL-PKE scheme. The results show that the communicating parties produced the same shared key $\mathrm{K_{shared}}$.

TABLE 1: Practical example of the proposed CL-PKE scheme (basic)

Algorithm	Step	Description	Generation method	Key value
Setup	A.1	Large prime p	Chosen	15485863
	A.2	Secret key $s \in \mathbb{Z}_p^*$, $s \geq 2$	Chosen	877
		Global parameter $G \in \mathbb{Z}_p^*$	Chosen	673
	A.3	Public key P_0	Eq. 9	14014563
Partial-Private-Key Extraction	B.2	Set the party A public value Q_A	$Q_A = H_1(ID_A)$	305
	B.3	Extracts the partial private key D_A	Eq. 10	3274740
Set-Secret-Value	C	Selected a random value $x_A \in \mathbb{Z}_q^*$	Chosen	859
Set-Shared-Secret-Key	D	Generates the shared secret key K_{shared}	Eq. 11	26560
Set-Public-Key P_A	E.2	Calculates X_A	Eq. 12	5787979
	E.2	Calculates Y_A	Eq. 13	9146288
	E.2	Calculates Z_A	Eq. 14	10741254
Encryption	F.1	checks that $X_A, Y_A, Z_A \in \mathbb{Z}_p^*$ and $T_{B_{Pri}}(X_A)\ mod\ p = Y_A$	Calculation	9146288 = Y_A
	F.2	Set the party A public value Q_A	$Q_A = H_1(ID_A)$	305
	F.3	Private key $B_{Pri} \in \mathbb{Z}_p^*,\ B_{Pri} \geq 2$	Chosen	587
	F.4	Public key B_{Pub}	Eq. 15	6118415
	F.5	Shared key K_{shared}	Eq. 16	26560
Decryption	G	Decrypt	No generation	N/A

5.2 Fully Secure CL-PKE Scheme

Because the proposed CL-PKE scheme is a one-way encryption scheme, its fully secure version can be obtained by applying the same transformations used by Al-Riyami and Paterson, 2003, and Boneh and Franklin, 2003.

These transformations have been proven to provide security strength against IND-CCA2, which is considered to be the highest security test for such a scheme. The fully secure version of the proposed CL-PKE scheme is similar to the basic version, except that in the execution of the Setup algorithm, two additional hash functions, H_3: $\{0,1\}^{\mathrm{n}} \times \{0,1\}^{\mathrm{n}} \rightarrow \mathbb{Z}_{\mathrm{p}}^{*}$ and H_4: $\{0,1\}^{\mathrm{n}} \times \{0,1\}^{\mathrm{n}}$ are used. With the additional hash functions, the encryption and decryption algorithms are as follows:

A. Encryption Algorithm:

1. First checks that $X_A, Y_A, Z_A \in \mathbb{Z}_p^*$ and make sure that $T_{B_{Pri}}(X_A)\ mod\ p = Y_A$. If so then encrypt the message otherwise output $\perp$ and abort the encryption.
2. Set $Q_A = H_1(ID_A)$.
3. Choose random value $\sigma \in \{0,1\}^n$.
4. Set party B's private key, $r = B_{Pri} = H_3(\sigma, M)$.
5. Compute the ciphertext $C = (U, V, W)$, where, $U = B_{Pub} = T_r(G)\ mod\ p$, $V = \sigma \oplus H_2(K_{shared})$, and $W = M \oplus H_4(\sigma)$. The shared key K_{shared} is generated by Eq. 16.

B. Decryption Algorithm:

1. Compute $\sigma = V \oplus H_2(K_{shared})$. The shared key is generated by Eq. 11.
2. Compute message, $M = W \oplus H_4(\sigma)$.
3. Set $r = B_{Pri} = H_3(\sigma, M)$, and test whether $U = T_r(G)\ mod\ p$. Accept the message M if equal, otherwise, reject the ciphertext.

5.3 Security Analysis

The security analysis of the proposed CL-PKE scheme is presented in two parts. The first part shows the strength of the shared secret key and its intractability against attacks. The second part shows the security analysis against the strongest security threat, the IND-CCA2. The analysis is performed by comparing the proposed CL-PKE scheme against the Al-Riyami and Paterson CL-PKE scheme (Al-Riyami and Paterson, 2003).

5.4 The Strength of the Shared Key

The proposed CL-PKE scheme inherits its security strength from the NP-hard Discrete Chebyshev Problem (DCP). Based on the NP-hard problem, we make the following claim:

Claim 1: Given all the public parameters $g = \langle p, P_0, G, n, H_1, H_2, H_3, H_4 \rangle$ and party A's public key $P_A = \langle X_A, Y_A, Z_A \rangle$, it is very easy for parties A and B to generate the shared secret key, K_{shared}. However, it is intractable for an adversary and the KGC to generate the shared secret key.

Proof.

1. **Party *A* (Decryptor):**

   Upon request, Party A will receive its partial private key D_A from the KGC through a secure channel and subsequently will generate its secret value $x_A \in \mathbb{Z}_q^*$. It is feasible to generate the shared secret key K_{shared} as shown by Eq. 11. Party A will then fuse its secret value x_A into the partial private key by using Eq. 4.

2. **Party *B* (Encryptor):**

   Party B has its private key B_{Pri}, Party A's public key Z_A and the sender's identity. It is feasible for Part B to generate the shared secret key K_{shared} as shown by Eq. 16. Party B will then fuse its private key B_{Pri} with party A's public key Z_A by using Eq. 4.

3. **The KGC:**

   Given all the public parameters $g = \langle p, P_0, G, n, H_1, H_2, H_3, H_4 \rangle$, party A's public key Z_A, and partial private key D_A, and party B's public key B_{Pub}, it is infeasible to generate the shared secret key K_{shared}, due to the DCP hard problem. To generate K_{shared}, the KGC needs the secret value of party A, x_A, and the private key of the party B, B_{Pri}. But both of the values x_A and B_{Pri} are already fused with the public values through the DCP hard problem. x_A is fused with Z_A as shown by Eq. 14, and B_{Pri} is fused with B_{Pub} as indicated by Eq. 17. Therefore, KGC will not be able to generate K_{shared}.

4. **The adversary:**

   The information the adversary can acquire is limited to the public values. These public values are the parameters g, party A's public key Y_A, and party B's public key B_{Pub}. Again, due to the fact that all of the secret and private values that needed for the generation of K_{shared} are fused with the public values by the DCP hard problem, it is impossible for the adversary to generate the shared secret key K_{shared}.

6. SECURITY AGAINST IND-CCA2

First, consider the comparison between the proposed CL-PKE schemes with Al-Riyami-Paterson CL-PKE scheme as shown in Table 2. IND-CCA is a type of strong security threat that is used to measure the security strength of ID-based schemes. Formally, the IND-CCA can be defined as the ability of an adversary $\mathcal{A}$ to successfully decrypt an intercepted ciphertext with probability $\Pr \geq 1/2$ given that, he has the ability to observe and intercept any ciphertext sent from the encryptor to the decryptor as well as the ability to choose the decryptions of any number of plaintexts associated with their public keys (Al-Riyami and Paterson, 2003; Bellare and Desai,1998; Dolev et al., 2000; Rackoff and Simon, 1991).

It has been proven that the security proof against IND-CCA for IBE scheme (Algehawi and Samsudin, 2010) is the same as the security proofs of the IBE scheme introduced by Boneh and Franklin, 2003, but each of them relies on a different hard problem. The security proof against IND-CCA for the CL-PKE scheme presented by Al-Riyami and Paterson is an extended version of the security proof of the Boneh-Franklin IBE scheme; both of them use IND-CCA as the measure against which to evaluate security strength. The proposed CL-PKE scheme and the CL-PKE scheme by Al-Riyami and Paterson follow the same steps in terms of their algorithms. Thus, the same extended version of the game played in Al-Riyami-Paterson CL-PKE scheme to prove its security strength against adversaries of types, $\mathcal{A}_{\mathrm{I}}$ and $\mathcal{A}_{\mathrm{II}}$ can also be played to prove the security strength of the proposed CL-PKE scheme. Therefore, the probability assumption for the proposed CL-PKE scheme can be devised from Theorem 1 (Al-Riyami and Paterson, 2003) as the following:

The new proposed CL-PKE scheme is IND-CCA secure against the two types of adversaries, $\mathcal{A}_{\mathrm{I}}$ and $\mathcal{A}_{\mathrm{II}}$, as explained by Al-Riyami and Paterson. If there is no polynomially bounded adversary $\mathcal{A}$ of either types I or II with a non-negligible advantage against the challenger in the games played, $\mathcal{A}$ advantage in this game is $\mathrm{Adv}(\mathcal{A}) = 2(\Pr[b = b'] - \frac{1}{2})$, where, $b, b' \in \{0,1\}$, and the adversary wins the game if $b = b'$. From the analysis, we summarize the following:

1. The security proof of Theorem 1 by Al-Riyami and Paterson is devised based on the layout and the steps of the algorithms composing the CL-PKE scheme, regardless of the mathematical

foundation which is used to relate the encryption and decryption keys.

2. By comparing our proposed CL-PKE scheme with Al-Riyami-Paterson CL-PKE scheme as shown by Table 2, we find that both schemes have the same algorithms with the same steps involved but that each of them relies on a different hard problem; that is, our proposed CL-PKE scheme is based on the DCP hard problem as, explained in Subsection 2.2, while the CL-PKE scheme relies on the BDHP hard problem.

3. The probability assumption of the security proof of the CL-PKE scheme is based on its application steps, which are the same as those of the proposed CL-PKE scheme. The difference between these two schemes is only in the underlying cryptography technique. This difference in the underlying technique does not affect the probability assumptions of the security proof used for the CL-PKE scheme; therefore, the same probability assumption can be used for the proposed CL-PKE scheme.

4. Finally, based on previous findings (Algehawi and Samsudin, 2010; Al-Riyami and Paterson, 2003; Boneh and Franklin, 2003; Maze, 2003), we can conclude that the new proposed CL-PKE scheme is IND-CCA secure against the two types of adversaries, $\mathcal{A}_I$ and $\mathcal{A}_{II}$ as defined by Al-Riyami and Paterson.

TABLE 2: Comparison between the proposed CL-PKE scheme and Al-Riyami-Paterson CL-PKE scheme

Algorithm	**Al-Riyami-Paterson CL-PKE**	**The proposed fully secure CL-PKE scheme**
Setup	1. Generate two groups G_1 and G_2 of prime order q and an admissible map $\hat{e}: G_1 \times G_1 \rightarrow G_2$. 2. Choose a random generator $P \in G_1$. 3. Choose a random master-key $s \in \mathbb{Z}_q^*$ and calculate the public key $P_0 = sP$. 4. Select the hash functions :$H_3: \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$, $H_4: \{0,1\}^n \rightarrow \{0,1\}^n$, $H_1: \{0,1\}^* \rightarrow G_1$, and $H_2: G_2 \rightarrow \{0,1\}^n$ for some bit-length n. 5. Choose the message space $\mathcal{M} = \{0,1\}^n$ and the ciphertext space $C = G_1 \times \{0,1\}^n$.	1. Randomly choose a large prime p. 2. Choose a random master-key $s \in \mathbb{Z}_p^*$, where $s \geq 2$ and a global parameter $G \in \mathbb{Z}_p^*$. 3. By using Eq. 4, calculate the public key P_0 as the following: $P_0 = T_s(G)\ mod\ p$. 4. Select two hash functions $H_1: \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2: \mathbb{Z}_p^* \rightarrow \{0,1\}^n$, $H_3: \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_p^*$, and $H_4: \{0,1\}^n \times \{0,1\}^n$ for some bit-length n. 5. Choose the message space $M = \{0,1\}^n$, and ciphertext space $C = \mathbb{Z}_p^* \times \{0,1\}^n$.

TABLE 2 (continued): Comparison between the proposed CL-PKE scheme and Al-Riyami-Paterson CL-PKE scheme

Algorithm	**Al-Riyami-Paterson CL-PKE**	**The proposed fully secure CL-PKE scheme**
Partial-Private-Key-Extract	Generate $D_A = sQ_A \in G_1^*$.	Generate $D_A = T_s\left(T_{Q_A}\left(T_{B_{pri}}(G)\right)\right)(mod\ p)$.
Set-Secret-Value	Selected a random value $x_A \in \mathbb{Z}_q^*$.	Selected a random value $x_A \in \mathbb{Z}_p^*$.
Set-private-Key	Generates A's private key $S_A \in G_1^*$, $S_A = x_A D_A = x_A s Q_A$.	Generate the shared secret key $K_{shared} \in \mathbb{Z}_p^*$, $K_{shared} = T_{x_A}\left(T_s\left(T_{Q_A}\left(T_{B_{pri}}(G)\right)\right)\right)\ (mod\ p)$.
Set-Public-Key	Generates the A's public key $P_A = \langle X_A, Y_A\rangle$, such that $X_A = x_A P$ and $Y_A = x_A P_0 = x_A sP$.	Generate the tuple, $P_A = \langle X_A, Y_A, Z_A\rangle$, such that: $X_A = T_{x_A}(G)\ mod\ p$. $Y_A = T_{x_A}(T_{B_{Pri}}(G))(\ mod\ p)$. $Z_A = T_{x_A}(T_s(G))(\ mod\ p)$.
Encryption	1. Checks that $X_A, Y_A \in G_1^*$ and $\hat{e}(X_A, P_0) = \hat{e}(Y_A, P)$. 2. Set $Q_A = H_1(ID_A) \in G_1^*$. 3. Chose $\sigma \in \{0,1\}^n$. 4. Set $r = H_3(\sigma, M)$. Set the ciphertext as $C = (U, V, W)$ where $U = rP$, $V = H_2(\hat{e}(Q_A, Y_A)^r)$, and $W = M \oplus H_4(\sigma)$.	1. Checks that $X_A, Y_A, Z_A \in \mathbb{Z}_p^*$ and that $T_{B_{Pri}}(X_A)\ mod\ p = Y_A$. 2. Set $Q_A = H_1(ID_A) \in \mathbb{Z}_p^*$. 3. Choose $\sigma \in \{0,1\}^n$. 4. Set $r = B_{Pri} = H_3(\sigma, M)$. Set the ciphertext as $C = (U, V, W)$, where, $U = T_r(G)\ mod\ p$, $V = \sigma \oplus H_2(K_{shared})$, and $W = M \oplus H_4(\sigma)$.
Decryption	1. Calculate $\sigma = V \oplus H_2(\hat{e}(U, S_A))$. 2. Calculate $M = W \oplus H_4(\sigma)$. 5. Set $r = H_3(\sigma, M)$. If $U \neq rP$ reject the ciphertext, otherwise accept the decrypted message, M.	1. Compute $\sigma = V \oplus H_2(K_{shared})$. 2. Compute message, $M = W \oplus H_4(\sigma)$. 5. Set $r = B_{Pri} = H_3(\sigma, M)$, if $U \neq T_r(G)\ mod\ p$ reject the ciphertext, otherwise accept the decrypted message, M.

7. CONCLUSION

In this paper, we have proposed a new CL-PKE scheme based on the Chebyshev polynomial extended over $\mathbb{Z}_p$. Our scheme is built to have the same properties as the well-known CL-PKE scheme of Al-Riyami and Paterson. The Discrete Chebyshev Problem (DCP) over finite field $\mathbb{Z}_p$ and the bilinearity property of the extended Chebyshev polynomial have been used to implement the CL-PKE scheme. Our scheme is well-tested and found to be secure, applicable and reliable.

REFERENCES

Algehawi, M. and Samsudin, A. (2010). A new Identity Based encryption scheme (IBE) Using Chebyshev Polynomial Extended over the Finite Field Z_p . *Phys. Lett. A*. **374**: 4670–4674.

Al-Riyami, S. and Paterson, G. (2003). Certificateless public key cryptography, in Lait C. S. (ed.): *Lecture Notes in Computer Science*. **2894**: 452-473.

Amig, J. M., Kocarev, L. and Szczepanski, J. (2007). Theory and practice of chaotic cryptography. *Phys. Lett. A.* **366**: 211–216.

Bellare, M., Desai, A., Pointcheval, D. and Rogaway, P. (1998). Relations among notions of security for public encryption schemes, in Krawczyk, H. (ed). *Lecture Notes in Computer Science.* **1462**: 26-45.

Bergamo, P., D'Arco, P., De Santis, A. and Kocarev, L. (2011). Security of public key cryptosystems based on Chebyshev polynomials, http://arxiv.org. Accessed 13 April 2011.

Bi, D. and Wang, D. (2009). A chaos public-key cryptosystem based on semi-group features. *Pro. 2nd Int. Conf. on Biomedical Engineering and Informatics*, Tianjin, China, October 2009, pp. 1-3.

Boneh, D. and Franklin, M. (2003). Identity-based encryption from weil pairing. *SIAM J. of Comp.* **32**: 586-615.

Dolev, D., Dwork, C. and Naor, M. (2000). Non-malleable cryptography. *SIAM J. of Comp.* **30**: 391-437.

Fujisaki, E. and Okamoto, T. (1999). Secure integration of asymmetric and symmetric encryption schemes, in Wiener, M. (ed.): *Lecture Notes in Computer Science*. **1666**: 537-554.

Han, S. (2008). Security of a key agreement protocol based on chaotic maps, *Chaos Solutions Fractals*.**38**: 764-768.

Maze, G. (2003). Algebric methods for constructing One-Way trapdoor functions, *University of Notre Dame*.

Rackoff, C. and Simon, D. (1991). Non-Interactive zero-knowledge proof of knowledge and chosen ciphertext attack, in Feigenbaum, J. (ed.): *Lecture Notes in Computer Science*. **576**: 433-444.

Wang, D., Yang, H., Yu, F. and Wang, X. (2008). A new key exchange scheme based on Chebyshev polynomials. *Proc. IEEE Congr. on Image and Signal Processing (CISP 08)*, Sanya, Hainan, China. pp. 124-127.

Wang, X. and Zhao, J. (2010). An improved key agreement protocol based on chaos. *Commun. Nonlinear. Sci. Numer. Simulat.***15**: 4052–4057.

Xiang, T., Wong, K. and Liao, X. (2009). On the security of a novel key agreement protocol based on chaotic maps. *Chaos Solutions Fractals*. **40**: 672-675.

Xiao, D., Liao, X. and Deng, S. (2007). A novel key agreement protocol based on chaotic maps. *Inf. Sci.* **177**: 1136–1142.

Yoon, E. and Yoo, K. (2008). A new key agreement protocol based on chaotic maps. P*roc. 2nd Int. Sympos. on Agent and Multi-Agent Systems: Technologies and Applications (KES-AMSTA 08),* Incheon, Korea, pp. 897–906.