Research Article

Certificateless Public Key Encryption Scheme with Hybrid Problems and Its Application to Internet of Things

Rui Guo,¹ Qiaoyan Wen,¹ Huixian Shi,² Zhengping Jin,¹ and Hua Zhang¹

¹ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

² Department of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710062, China

Correspondence should be addressed to Rui Guo; grbupt@gmail.com

Received 9 October 2013; Revised 31 December 2013; Accepted 19 January 2014; Published 12 March 2014

Academic Editor: Fuzhong Nian

Copyright © 2014 Rui Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Certificateless cryptography aims at combining the advantages of public key cryptography and identity based cryptography to avoid the certificate management and the key escrow problem. In this paper, we present a novel certificateless public key encryption scheme on the elliptic curve over the ring, whose security is based on the hardness assumption of Bilinear Diffie-Hellman problem and factoring the large number as in an RSA protocol. Moreover, since our scheme requires only one pairing operation in decryption, it is significantly more efficient than other related schemes. In addition, based on our encryption system, we also propose a protocol to protect the confidentiality and integrity of information in the scenario of Internet of Things with constrained resource nodes.

1. Introduction

In a traditional public key cryptography (PKC) scheme, public key certificates signed by a certificate authority (CA) are employed to ensure the authenticity of public keys. Thus, the PKC takes a huge effort to manage the certificates, including revocation, storage, distribution, and verification, which places a computational burden on the whole system. To simplify the complex certificate management process, Shamir proposed the concept of identity-based public key cryptography (ID-PKC) [1], where an entity is allowed to use his identity such as email and IP address as his public key. However, the private keys of users are totally generated by a trusted third party named private key generator (PKG), which enables the PKG to impersonate any user to recover his/her confidential messages.

In order to resolve the inherent key escrow problem while preserving the advantage of ID-PKC, Al-Riyami and Paterson [2] introduced a new paradigm called certificateless public key cryptography (CL-PKC), which does not require the management of certificates and resolve the key escrow problem. Specifically, in CL-PKC, there also exists a trusted third

party called key generation center (KGC) which supplies a partial private key for entity. Then, the entity takes advantage of this partial private key and a secret value picked by itself to generate the full private key. According to this method, KGC cannot obtain the user's private key to decrypt his/her ciphertext anymore, and it avoids the escrow problem in ID-PKC. Therefore, CL-PKC is considered as lying in between PKC and ID-PKC. However, it should be emphasized that so far certificateless public key encryption (CL-PKE) schemes have been constructed within the framework of identity-based encryption (IBE) schemes from Weil pairing proposed by Boneh and Franklin [3]. As a result, the CL-PKE schemes in the literature are always based on the bilinear pairings.

Recently, the pairings, such as Weil pairing and Tate pairing, have been of essential use in CL-PKE, because of the excellent properties of bilinearity and nondegeneracy [4–7]. The central idea is the construction of a mapping between two useful cryptographic groups that allow for new cryptographic schemes based on the reduction of one problem in one group to a different group and usually is easier problem than the other group. Therefore, the construction based on the bilinear pairing is more concise than the existing

methods of cryptographic scheme constructions and it owns the merits of security. In Cheng and Comley's scheme [4], they constructed a more efficient scheme and then extended it to an authenticated encryption. Shi and Li [5] proposed a CL-PKE scheme which was based on the Weil pairing. That scheme worked in a kind of parallel model and it was more efficient on computation. In [6], Dent et al. presented the first constructions for CL-PKE schemes that were provably secure against strong adversaries in the standard model. For raising efficiency, Sun and Li [7] proposed a short-ciphertext CCA2 secure certificateless encryption scheme under the standard bilinear Diffie-Hellman assumption. Due to short ciphertext and convincing security, their scheme had practical value.

The Internet of Things (IoT) [8, 9] is a kind of network, which can connect everything to the Internet by using of the RFID, Infrared Sensors, GPS, Laser Scanner, Sensor Node, and other sensor equipments, thus possessing the ability of identifying, locating, tracking, monitoring, managing, and other intelligent actions. In IoT, it requires sensor nodes to be employed in unprotected and even hostile environments, and therefore IoT brings lots of research challenges; one of the important issues is the security [10, 11].

In this paper, we employ use of the bilinear pairing to design a certificateless encryption scheme on the elliptic curves over the ring Z_n [12, 13], which overcomes the security defect of the Koyama et al.'s scheme [14] whose security is only based on the problem of factoring the large number as in RSA. Furthermore, we prove that our scheme is secure in the random oracle model, provided that the BDH assumption and the problem of factoring the large number are intractable. Comparing with the related CL-PKE schemes, our proposal offers better performances than others on the efficiency and security. At last, based on our CL-PKE scheme, we also present another protocol to protect the confidentiality and integrity of data transmitted between the gateway node and the sensor node in IoT.

The remainder of this paper is organized as follows. In Section 2, we introduce the preliminary concepts, including properties of the elliptic curve over the ring Z_n , the related computational problems, and the formal definition and security model of CL-PKE. In Section 3, we present our CL-PKE scheme and analyze the security and performance of it. Based on our CL-PKE, in Section 4, we propose a scheme to protect the confidentiality and integrity of information in the business of IoT. Finally, we conclude the paper in Section 5.

2. Preliminaries

In this section, we summarize the definitions and properties of elliptic curve over a ring $Z_n = Z/nZ$, bilinear pairing, and models of CL-PKE, where n is a RSA modulus.

2.1. Elliptic Curve over the Ring

Definition 1. Let F_q be a field and prime $q > 3$; all the points $(x, y) \in F_q \times F_q$ satisfying the equation $y^2 \equiv x^3 + ax + b \pmod{q}$ compose the elliptic curve over F_q together with the point at infinity denoted $\mathcal{O}$, where $a, b \in F_q$ are two

parameters satisfying $4a^3 + 27b^2 \not\equiv 0 \pmod{q}$; such a curve is denoted by $E_q(a, b)$.

The addition operation on the points of $E_q(a, b)$ can be defined as follows.

Let $P = (x_1, y_1)$, $Q = (x_2, y_2)$ be the points of $E_q(a, b)$; then $-P = (x_1, -y_1)$.

- (a) If $P = \mathcal{O}$, then $-P = \mathcal{O}$, and $P + Q = Q$;
- (b) if $Q = -P$, then $P + Q = \mathcal{O}$;
- (c) if $Q \neq -P$, then $P + Q = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$, and

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{if } P \neq Q, \\ \frac{3x_1^2 + a}{2y_1}, & \text{if } P = Q. \end{cases} \quad (1)$$

From these definitions, we know the fact that the addition operation on the points of $E_q(a, b)$ can be made as an Abelian group, where $\mathcal{O}$ is the neutral element of $E_q(a, b)$.

Let $Z_n = Z/nZ$, where $n = pq$ is the product of two large primes as the RSA modulus. The Chinese Remainder Theorem says that there is an isomorphism of rings

$$Z_{pq} \simeq Z_p \oplus Z_q \quad (2)$$

given by

$$x \bmod pq \longleftrightarrow (x \bmod p, x \bmod q). \quad (3)$$

This yields a bijection between elements in Z_{pq} and pairs of elements, one in Z_p and the other in Z_q . Thus, the following theorem can be given.

Theorem 2. Let p and q be odd integers with $\gcd(p, q) = 1$; let E be an elliptic curve defined over Z_{pq} . Then, there is a group isomorphism

$$E(Z_{pq}) \simeq E(Z_p) \oplus E(Z_q). \quad (4)$$

Therefore, there is an isomorphism mapping

$$\begin{aligned} \varphi : E(Z_{pq}) &\longrightarrow E(Z_p) \oplus E(Z_q), \\ P_{pq} &\longmapsto [P_p, P_q], \end{aligned} \quad (5)$$

where $P_{pq} \in E(Z_{pq})$, $P_p \in E(Z_p)$ and $P_q \in E(Z_q)$.

Let $Q_{pq} \in E(Z_{pq})$, $Q_p \in E(Z_p)$ and $Q_q \in E(Z_q)$; we can get

$$\begin{aligned} \varphi(P_{pq}) &= [P_p, P_q], \\ \varphi(Q_{pq}) &= [Q_p, Q_q], \\ \varphi(P_{pq} + Q_{pq}) &= \varphi(P_{pq}) + \varphi(Q_{pq}) = [P_p, P_q] + [Q_p, Q_q] \\ &= [P_p + Q_p, P_q + Q_q]. \end{aligned} \quad (6)$$

Because φ is an isomorphism mapping, there is an inverse mapping of it, denoted ψ . Thus,

$$\psi(\varphi(P_{pq} + Q_{pq})) = P_{pq} + Q_{pq} = \psi([P_p + Q_p, P_q + Q_q]), \quad (7)$$

which means that the addition law on $E(Z_{pq})$ can be defined as an isomorphism mapping of the addition law in $E(Z_p)$ and $E(Z_q)$, with the convention that the infinity point $\mathcal{O}$ of $E(Z_{pq})$ can be presented by $[\mathcal{O}_p, \mathcal{O}_q]$, where $\mathcal{O}_p$ and $\mathcal{O}_q$ are the points at infinity on $E(Z_p)$ and $E(Z_q)$, respectively. Then, $E(Z_{pq})$ is an Abelian group under this definition of point addition.

For more details concerning the addition law on elliptic curves over the ring, see [12, 13, 15]. Besides, there is an important theorem in constructing our CL-PKE over the ring as shown below.

Theorem 3. Let $E_n(a, b)$ be an elliptic curve such that $\gcd(4a^3 + 27b^2, n) = 1$, $n = pq$, and let N be $\text{lcm}(\#E_p(a, b), \#E_q(a, b))$. Then, for any $P \in E_n(a, b)$ and any integer k ,

$$(k \cdot N + 1) \cdot P = P \text{ over } E_n(a, b). \quad (8)$$

2.2. Bilinear Pairings. Let G_1 be a cyclic additive group generated by an elliptic curve point P , whose order is N ; let G_2 be a multiplicative group of the same order. Assume that bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ with the following properties.

- (1) *Bilinearity*: for all $X, Y, Z \in G_1$, $e(X, Y + Z) = e(X, Y) \cdot e(X, Z)$ and $e(X + Y, Z) = e(X, Z) \cdot e(Y, Z)$. Consequently, $\forall a, b \in \mathbb{Z}_q$, we have $e(aX, bY) = e(X, Y)^{ab} = e(abX, Y)$, and so forth.
- (2) *Nondegeneracy*: $e(P, P) \neq 1_{G_2}$, where 1_{G_2} denotes the identity element of the group G_2 .
- (3) *Computability*: there exists an efficient algorithm to compute $e(X, Y)$ for any $X, Y \in G_1$.

We also consider the following computational problem in $\langle G_1, G_2, e \rangle$ as above which will form the basis of security for our CL-PKE scheme.

Definition 4. The Bilinear Diffie-Hellman Problem (BDHP) is given $\langle P, aP, bP, cP \rangle$ with uniformly random choices of $a, b, c \in \mathbb{Z}_N^*$; compute $e(P, P)^{abc} \in G_2$. An algorithm $\mathcal{A}$ has the advantage ϵ' in solving the BDHP in $\langle G_1, G_2, e \rangle$ if $\Pr[\mathcal{A}(\langle P, aP, bP, cP \rangle) = e(P, P)^{abc}] = \epsilon'$.

The BDHP is said to be ϵ' -intractable if there is no algorithm that $\mathcal{A}$ solves this problem with ϵ' .

2.3. Definition of CL-PKE. According to [2], a CL-PKE scheme consists of seven probabilistic, polynomial time (PPT) algorithms: *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key*, *Encrypt*, and *Decrypt*.

Setup. On input a security parameter 1^k , this algorithm returns the system parameters $param$ and the master key msk .

The system $param$ includes the plaintext space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$. After this algorithm is over, the KGC publishes $param$ and keeps the msk secret.

Partial-Private-Key-Extract. On input $param$, msk , and an identity ID for the entity, KGC executes this algorithm and returns the partial private key D_{ID} to entity via a confidential and authentic channel.

Set-Secret-Value. On input $param$ and an identity ID, entity executes this algorithm and returns entity's secret value x_{ID} .

Set-Private-Key. On input $param$, entity's partial private key D_{ID} , and secret value x_{ID} , this algorithm returns the entity's full private key SK_{ID} . Note that this algorithm is executed by the entity itself.

Set-Public-Key. On input $param$ and entity's secret value x_{ID} , this algorithm returns the public key PK_{ID} to the entity. This algorithm is also executed by the entity itself.

Encrypt. Running by a sender. On input message $M \in \mathcal{M}$, public key PK_{ID} , and identity ID of an entity, this algorithm returns a ciphertext $C \in \mathcal{C}$.

Decrypt. Running this determinate algorithm by a receiver. On input $param$, $C \in \mathcal{C}$, and a private key SK_{ID} , this algorithm returns a message $M \in \mathcal{M}$, which is either a plaintext message or a "Reject" message.

2.4. Security Model for CL-PKE. In CL-PKE, there are two types of adversary with different capabilities, Type I and Type II adversaries [2]. A difference between these two attackers is that $\mathcal{A}_I$ does not have access to the master key of KGC, while $\mathcal{A}_{II}$ does have. Specifically, the adversary $\mathcal{A}_I$ in Type I represents a normal third party attacker against the CL-PKE scheme; that is, $\mathcal{A}_I$ is not allowed to access to the master key but $\mathcal{A}_I$ may request public keys and replace public keys with values of its choice. The adversary $\mathcal{A}_{II}$ represents a malicious KGC who generates partial private keys of users. The adversary $\mathcal{A}_{II}$ is allowed to have access to the master key but not replace a public key.

Definition 5. A CL-PKE scheme is IND-CCA secure if neither polynomially bounded adversary $\mathcal{A}$ of Type I nor Type II has a nonnegligible advantage against the challenger in the following game.

Setup. The challenger $\mathcal{CH}$ takes a security parameter 1_k as inputs and runs the Setup algorithm; then it sends the resulting system parameters $param$ to $\mathcal{A}$. If $\mathcal{A}$ is of Type I, $\mathcal{CH}$ keeps the master secret key msk to itself. Otherwise, it returns msk to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ is given access to the following oracles.

- (1) *Partial-Key-Extract-Oracle*: upon receiving a partial key query for a user's identity ID, $\mathcal{CH}$ computes D_{ID} and returns it to $\mathcal{A}$. (Note that it is only useful to Type I adversary.)

- (2) Private-Key-Request-Oracle: upon receiving a private key query for a user's identity ID, $\mathcal{CH}$ computes SK_{ID} and returns it to $\mathcal{A}$. It outputs $\perp$ (denotes failure) if the user's public key has been replaced (in the case of Type I adversary).
- (3) Public-Key-Request-Oracle: upon receiving a public key query for a user's identity ID, $\mathcal{CH}$ computes PK_{ID} and returns it to $\mathcal{A}$.
- (4) Public-Key-Replace-Oracle: for identity ID and a valid public key, $\mathcal{A}$ replaces the associated user's public key with the new one of its choice (this is only for Type I adversary). The new value will be recorded and used by $\mathcal{CH}$ in the coming computations or responses to the adversary's queries.
- (5) Decryption-Oracle: On input a ciphertext and an identity, it returns the correct decryption of ciphertext, which is encrypted under the private key corresponding to the current value of the public key associated with an identity of the user, even if the corresponding public key for the user ID has been replaced.

Challenge Phase. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs and submits two messages (M_0, M_1) , together with a challenge identity ID^* of the uncorrupted secret key. Note that $\mathcal{A}$ is not allowed to know the private key of ID^* in anyway. The challenger $\mathcal{CH}$ picks a random bit $\beta \in \{0, 1\}$ and computes C^* , which is the encryption of M_β under the current public key PK_{ID^*} for ID^* . If the output of the encryption is $\perp$, $\mathcal{A}$ immediately loses the game. Otherwise, C^* is delivered to $\mathcal{A}$.

Phase 2. Now $\mathcal{A}$ issues a second sequence of queries as in Phase 1. A decryption query on the challenge ciphertext C^* for the combination of ID^* and PK_{ID^*} is not allowed.

Guess. Finally, $\mathcal{A}$ outputs its guess β' for β . The adversary wins the game if $\beta' = \beta$ and the advantage of $\mathcal{A}$ in this game is defined to be $\text{Adv}(\mathcal{A}) = |\Pr(\beta' = \beta) - 1/2|$. The adversary $\mathcal{A}_i$ ($i = \text{I, II}$) breaks an IND-CCA secure CL-PKE scheme with $(q_H, q_{\text{par}}, q_{\text{pub}}, q_{\text{prv}}, q_D, \epsilon)$ if and only if the guessing advantage of $\mathcal{A}_i$ that makes q_H times the random oracle $H(\cdot)$, q_{par} times Partial-Key-Extract-Oracle, q_{pub} times Public-Key-Request-Oracle, q_{prv} times Private-Key-Request-Oracle, and q_D times Decryption-Oracle queries is greater than ϵ . The scheme is said to be $(q_H, q_{\text{par}}, q_{\text{pub}}, q_{\text{prv}}, q_D, \epsilon)$ -IND-CCA secure if there is no attacker $\mathcal{A}_i$ that breaks IND-CCA secure scheme with $(q_H, q_{\text{par}}, q_{\text{pub}}, q_{\text{prv}}, q_D, \epsilon)$.

3. Our CL-PKE Scheme

In this section, we propose a CL-PKE scheme based on the bilinear pairing over the ring and evaluate its performance.

3.1. Construction. The proposed CL-PKE scheme consists of the following seven PPT algorithms.

Setup. Let G_1, G_2 be bilinear groups of order N with an arbitrary generator $P \in G_1$, and $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear pairing, where G_1 is a elliptic curve $E_N(a, b)$. The KGC selects an msk $s \in \mathbb{Z}_N^*$ at random and computes $X = sP$ as the master public key. Then, it chooses two collision resistant hash functions $H_1 : \{0, 1\}^\omega \rightarrow G_1^*, H_2 : G_2 \rightarrow \{0, 1\}^*$, where ω denotes the bit-length of identity. The system parameters are $param = \{G_1, G_2, e, P, X, H_1, H_2\}$ and the master secret key is $msk = s$.

Partial-Private-Key-Extract. On input an entity's identity $ID \in \{0, 1\}^\omega$, this algorithm computes $Q_{ID} = H_1(ID) \in G_1^*$ and sends the partial private key $D_{ID} = s \cdot Q_{ID} \in G_1^*$ to entity via a secure channel.

Set-Secret-Value. On input $param$ and an identity ID, entity picks a secret value $a \in \mathbb{Z}_N^*$. Return $x_{ID} = a$.

Set-Private-Key. On input $param$, ID and x_{ID} , entity obtains the private key SK_{ID} by computing $SK_{ID} = a \cdot D_{ID} = asH_1(ID) \in G_1^*$.

Set-Public-Key. On input $param$ and ID, this algorithm returns $PK_{ID} = x_{ID}Q_{ID}$ as the public key.

Encrypt. To encrypt $M \in \{0, 1\}^*$, entity selects a random value $r \in \mathbb{Z}_N^*$ and $\sigma \in \{0, 1\}^*$, computes $C = (c_1, c_2)$ such that

$$c_1 = rP, \quad c_2 = (M \parallel \sigma) \oplus H_2(e(PK_{ID}, rX)), \quad (9)$$

outputs the ciphertext $C = (c_1, c_2)$.

Decrypt. To decrypt ciphertext $C = (c_1, c_2)$ for the entity with identity ID and private key SK_{ID} , compute

$$M \parallel \sigma = c_2 \oplus H_2(e(SK_{ID}, c_1)), \quad (10)$$

and return M as plaintext.

Notice that if (c_1, c_2) is the encryption of M with public key PK_{ID} , we have

$$\begin{aligned} & c_2 \oplus H_2(e(SK_{ID}, c_1)) \\ &= (M \parallel \sigma) \oplus H_2(e(PK_{ID}, rX)) \oplus H_2(e(asQ_{ID}, rP)) \\ &= (M \parallel \sigma) \oplus H_2(e(Q_{ID}, P)^{ars}) \oplus H_2(e(Q_{ID}, P)^{ars}) \\ &= M \parallel \sigma. \end{aligned} \quad (11)$$

3.2. Security Analysis. In this section, we will show that the scheme described in the previous is secure in the random oracle model.

Theorem 6. Given that H_1 and H_2 are two collision resistant hash functions. The proposed CL-PKE scheme based on the ring $\mathbb{Z}_N$ is IND-CCA secure in the random oracle model assuming that the BDHP is intractable.

In the procession of attacking this system by an adversary $\mathcal{A}$, it chooses two messages M_0, M_1 and is given the challenge

ciphertext C^* for one of these two messages M_β by the challenger $\mathcal{CH}$ firstly. Then, $\mathcal{A}$ may make decryption queries but not ask for the decryption of C^* . If $\mathcal{A}$'s guess β' is equal to β , it wins the game.

In order to prove Theorem 6, we prove two lemmas firstly to show that our CL-PKE scheme is secure against Type I and Type II attacker whose behavior is as described in Definition 5.

Lemma 7. *The CL-PKE scheme is $(q_{H_1}, q_{H_2}, q_{par}, q_{pub}, q_{prv}, q_D, \epsilon)$ -IND-CCA secure against Type I attacker $\mathcal{A}$ in the random oracle assuming that the BDH problem is ϵ' -intractable, where $\epsilon' > (1/q_{H_2})(2\epsilon/e(q_{prv} + q_{par} + 1) - q_{H_1}/2^\omega - q_D q_{H_1}/2^\omega - q_D/N)$.*

Proof. In this lemma, Type I models an “outside” adversary, which can replace the public key of arbitrary identities but cannot corrupt the master secret key.

Let $\mathcal{A}_1$ be a Type I IND-CCA adversary against our scheme. Suppose that $\mathcal{A}_1$ has the advantage ϵ and makes q_{H_i} queries to random oracle H_i ($i = 1, 2$) and q_D decryption queries. We show how to construct an algorithm $\mathcal{B}$ to solve the BDH problem with the instance of (P, aP, bP, cP) by interacting with $\mathcal{A}_1$.

At the beginning, $\mathcal{B}$ simulates the algorithm *Setup* for $\mathcal{A}_1$ by supplying it with $param = \{G_1, G_2, e, P, X, H_1, H_2\}$, where H_1, H_2 is random oracles that will be controlled by $\mathcal{B}$. $\mathcal{B}$ chooses an index I uniformly at random with $1 \leq I \leq q_{H_1}$.

The adversary $\mathcal{A}_1$ may make queries of the random oracles H_i ($i = 1, 2$) at any time during its attack. $\mathcal{B}$ responds as follows.

H_1 Queries. $\mathcal{B}$ maintains a list of tuples $\langle ID_i, Q_i, t_i \rangle$ in H_1 List L_1 . On receiving a query ID_i to H_1 , $\mathcal{B}$ responds as follows.

- (1) If ID_i already appears on the list L_1 in a tuple $\langle ID_i, Q_i, t_i \rangle$, $\mathcal{B}$ responds Q_i as an answer.
- (2) Otherwise, if $i \neq I$, choose $t_i \in Z_N^*$ at random and compute $Q_i = t_i P$, add $\langle ID_i, Q_i, t_i \rangle$ to L_1 , and return Q_i as an answer.
- (3) If $i = I$, add $\langle ID_i, Q_i = cP, * \rangle$ to L_1 and return $Q_i = cP$ as an answer (where $*$ denotes the arbitrary value).

H_2 Queries. $\mathcal{B}$ maintains a list of tuples $\langle ID_i, e_i, R_i \rangle$ in H_2 List L_2 . On receiving a query $\langle ID_i, e_i \rangle$ to H_2 , $\mathcal{B}$ responds as follows.

- (1) If ID_i already appears on the list L_2 in a tuple $\langle ID_i, e_i, R_i \rangle$, $\mathcal{B}$ responds R_i as an answer.
- (2) Otherwise, pick $R_i \in \{0, 1\}^*$ at random, add $\langle ID_i, e_i, R_i \rangle$ to L_2 and return R_i as an answer.

Phase 1. After receiving $param$ from $\mathcal{B}$, $\mathcal{A}_1$ issues a sequence of polynomially bounded number of requests, each request being either a *Partial-Private-Key-Extraction*, a *Private-Key-Extraction*, a *Request-Public-Key*, a *Replace-Public-Key*, or *Decryption-Queries* for an entity. We assume that $\mathcal{A}_1$ always makes the appropriate H_1 queries on the identity ID before making one of these requests and never makes a decryption

query on a ciphertext obtained from the encryption oracle. $\mathcal{B}$ replies to these requests as follows.

Partial-Private-Key-Extraction. $\mathcal{B}$ maintains a Partial Private Key List of tuples $\langle ID_i, D_i \rangle$. On receiving a query ID_i , $\mathcal{B}$ responds as follows.

- (1) If $\langle ID_i, D_i \rangle$ exist in Partial Private Key List, return D_i as an answer.
- (2) Otherwise, pick i at random, so that $\Pr[i \neq I] = \delta$. (δ will be determined later.) If $i \neq I$, search the L_1 for a tuple $\langle ID_i, Q_i, t_i \rangle$, compute $D_i = t_i P$, add $\langle ID_i, D_i \rangle$ to the Partial Private Key List, and return D_i as an answer.
- (3) If $i = I$, return “Abort” and terminate.

Private-Key-Extraction. $\mathcal{B}$ maintains a Private Key List of tuples $\langle ID_i, x_i, D_i \rangle$. On receiving a query ID_i , $\mathcal{B}$ responds as follows.

- (1) If $\langle ID_i, x_i, D_i \rangle$ exist in Private Key List, return $\langle x_i, D_i \rangle$ as answer.
- (2) Otherwise, if $i \neq I$, run the simulation algorithm Request-Public-Key to get a tuple $\langle ID_i, x_i, PK_i \rangle$ and the simulation algorithm Private-Key-Extraction to get a tuple $\langle ID_i, D_i \rangle$, add $\langle ID_i, x_i, D_i \rangle$ to the Private Key List, and return $\langle x_i, D_i \rangle$ as an answer. (Note that if the corresponding public key has been replaced, such a private key query is not allowed.)
- (3) If $i = I$, return “Abort” and terminate.

Request-Public-Key. $\mathcal{B}$ maintains a Public Key List of tuples $\langle ID_i, x_i, PK_i \rangle$. On receiving a query ID_i , $\mathcal{B}$ responds as follows.

- (1) If $\langle ID_i, x_i, PK_i \rangle$ exist in Public Key List, return PK_i as an answer.
- (2) Otherwise, choose $x_i \in Z_N^*$, compute $PK_i = x_i H_1(ID_i)$, add $\langle ID_i, x_i, PK_i \rangle$ to the Public Key List, and return PK_i as an answer.

Replace-Public-Key. $\mathcal{A}_1$ may replace any public key with a new value of its choice and $\mathcal{B}$ records all the changes.

Decryption-Queries. On receiving a query $\langle ID_i, PK_i, C \rangle$, where $C = (c_1, c_2)$ and $PK_i = x_i H_1(ID_i)$. $\mathcal{B}$ responds as follows.

- (1) If $i \neq I$ and PK_i is the correct public key not a replaced one, $\mathcal{B}$ decrypts C by using the corresponding private key.
- (2) Otherwise, search L_2 for a tuple $\langle ID_i, e_i, R_i \rangle$. If such a tuple exists, $\mathcal{B}$ retrieves the related R_i to compute $M \parallel \sigma = c_2 \oplus R_i$ and returns M as an answer.
- (3) Otherwise, $\mathcal{B}$ picks $R_i \in \{0, 1\}^*$ at random, computes $M \parallel \sigma = c_2 \oplus R_i$ and returns M as an answer. Add $\langle ID_i, e_i, R_i \rangle$ to L_2 .

Challenge Phase. $\mathcal{A}_1$ then outputs two messages (M_0, M_1) and a challenge identity ID^* . On receiving a challenge query $\langle ID^*, (M_0, M_1) \rangle$:

- (1) If $ID^* \neq ID_I$, $\mathcal{B}$ aborts the game.
- (2) Otherwise, $\mathcal{B}$ selects $r^* \in Z_N^*$ and $\sigma \in \{0, 1\}^*$ randomly, computes $c_1^* = r^* P$ and $H_2(e(\text{SK}_{ID^*}, c_1^*)) = c_2^* \oplus (M_\beta \parallel \sigma)$ (note that $\mathcal{B}$ does not know " r^* "), returns $C^* = (c_1^*, c_2^*)$ as a target ciphertext.

Phase 2. $\mathcal{A}_1$ requests in the same ways as in *Phase 1*. Moreover, no private key extraction on ID^* is allowed and no decryption query can be made on the ciphertext C^* for the combination of identity ID^* and public key PK_{ID^*} that encrypted plaintext M_β .

Guess. $\mathcal{A}_1$ should make a guess β' for β . The adversary wins the game if $\beta' = \beta$.

Analysis. By AskH_2^* we denote the event that (ID^*, e_i^*) has been queried to H_2 . Also, by AskH_1^* we denote the event that ID^* has been queried to H_1 . If happens, $\mathcal{B}$ will be able to solve the BDH problem by choosing a tuple $\langle ID_i, e_i, R_i \rangle$ from L_2 and computing $H_2(e_i)$ with the probability at least $1/q_{H_2}$. Hence, we have $\epsilon' \geq (1/q_{H_2})\text{Pr}[\text{AskH}_2^*]$.

It is easy to notice that if $\mathcal{B}$ does not abort, the simulations of *Partial-Key-Extract*, *Private-Key-Request*, *Public-Key-Request*, and the simulated target ciphertext are identically distributed as the real one from the construction.

Now, we evaluate the simulation of the decryption oracle. If a public key PK_{ID} has not been replaced nor PK_{ID} has been produced by reselecting $x_i \in Z_N^*$, the simulation is perfect as $\mathcal{B}$ knowing the private key SK_{ID} corresponding to PK_{ID} . Otherwise, simulation error may occur while $\mathcal{B}$ is running the decryption oracle simulation specified above. Let DecErr be this event. Suppose that ID, PK_{ID}, C , where $C = (c_1, c_2)$ and $\text{PK}_{ID} = x_{ID}P$, have been issued as a valid decryption query. Even if C is valid, there is a possibility that C can be produced without querying (ID, e_i) to H_2 . Let Valid be an event that C is valid; let AskH_2 and AskH_1 be events that (ID, e_i) have been queried to H_2 and ID have been queried to H_1 , respectively. Since DecErr is an event that $\text{Valid} \mid \neg \text{AskH}_2$ happens after the entire simulation and q_D decryption oracle queries being performed, we have $\text{Pr}[\text{DecErr}] = q_D \text{Pr}[\text{Valid} \mid \neg \text{AskH}_2]$, where,

$$\begin{aligned}
 & \text{Pr}[\text{Valid} \mid \neg \text{AskH}_2] \\
 & \leq \text{Pr}[\text{Valid} \wedge \text{AskH}_1 \mid \neg \text{AskH}_2] \\
 & \quad + \text{Pr}[\text{Valid} \wedge \neg \text{AskH}_1 \mid \neg \text{AskH}_2] \\
 & \leq \text{Pr}[\text{AskH}_1 \mid \neg \text{AskH}_2] \\
 & \quad + \text{Pr}[\text{Valid} \mid \neg \text{AskH}_1 \wedge \neg \text{AskH}_2] \\
 & \leq \frac{q_{H_1}}{2^\omega} + \frac{1}{N}.
 \end{aligned} \tag{12}$$

Now, the event $(\text{AskH}_2^* \vee (\text{AskH}_1^* \mid \neg \text{AskH}_2^*) \vee \text{DecErr}) \mid \neg \text{Abort}$ is denoted by E , where Abort denotes an event that

$\mathcal{B}$ aborts during the simulation. The probability $\neg \text{Abort}$ that happens is given by $\delta^{q_{\text{prv}} + q_{\text{par}}} (1 - \delta)$ which is maximized at $\delta = 1 - 1/(q_{\text{prv}} + q_{\text{par}} + 1)$. Hence, we have $\text{Pr}[\neg \text{Abort}] \leq 1/e(q_{\text{prv}} + q_{\text{par}} + 1)$, where e denotes the base of the natural logarithm.

If E does not happen, it is clear that $\mathcal{A}_1$ does not gain any advantage greater than $1/2$ to guess β due to the randomness of the output of the random oracle H_2 . Namely, we have $\text{Pr}[\beta' = \beta \mid \neg E] \leq 1/2$.

By definition of ϵ , we have

$$\begin{aligned}
 \epsilon & < \left| \text{Pr}[\beta' = \beta] - \frac{1}{2} \right| \\
 & = \left| \text{Pr}[\beta' = \beta \mid \neg E] \text{Pr}[\neg E] \right. \\
 & \quad \left. + \text{Pr}[\beta' = \beta \mid E] \text{Pr}[E] - \frac{1}{2} \right| \\
 & \leq \left| \frac{1}{2} \text{Pr}[\neg E] + \text{Pr}[E] - \frac{1}{2} \right| \\
 & = \left| \frac{1}{2} (1 - \text{Pr}[E]) + \text{Pr}[E] - \frac{1}{2} \right| \\
 & = \frac{1}{2} \text{Pr}[E] \\
 & \leq \frac{1}{2 \text{Pr}[\neg \text{Abort}]} \\
 & \quad \times (\text{Pr}[\text{AskH}_2^*] + \text{Pr}[\text{AskH}_1^* \mid \neg \text{AskH}_2^*] + \text{Pr}[\text{DecErr}]) \\
 & \leq \frac{e(q_{\text{prv}} + q_{\text{par}} + 1)}{2} \left(q_{H_2} \epsilon' + \frac{q_{H_1}}{2^\omega} + \frac{q_D q_{H_1}}{2^\omega} + \frac{q_D}{N} \right).
 \end{aligned} \tag{13}$$

Consequently, we obtain $\epsilon' > (1/q_{H_2})(2\epsilon/e(q_{\text{prv}} + q_{\text{par}} + 1) - q_{H_1}/2^\omega - q_D q_{H_1}/2^\omega - q_D/N)$. $\square$

Lemma 8. The CL-PKE scheme is $(q_{H_1}, q_{H_2}, q_{\text{par}}, q_{\text{pub}}, q_{\text{prv}}, q_D, \epsilon)$ -IND-CCA secure against Type II attacker $\mathcal{A}$ in the random oracle assuming that the BDH problem is ϵ' -intractable, where $\epsilon' > (1/q_{H_2})(2\epsilon/e(q_{\text{prv}} + 1) - q_{H_1}/2^\omega - q_D q_{H_1}/2^\omega - q_D/N)$.

Proof. In this lemma, a Type II models an "insider" adversary, who has access to msk but cannot replace public keys of entities.

Let $\mathcal{A}_{\text{II}}$ be a Type II IND-CCA adversary against our scheme. Suppose that $\mathcal{A}_{\text{II}}$ has the advantage ϵ , makes q_{H_i} queries to random oracle H_i ($i = 1, 2$), and q_D decryption queries. We show how to construct an algorithm $\mathcal{B}$ to solve the BDH problem with the instance of (P, aP, bP, cP) by interacting with $\mathcal{A}_{\text{II}}$.

At the beginning, $\mathcal{B}$ simulates the algorithm Setup for $\mathcal{A}_{\text{II}}$ by supplying $\mathcal{A}_{\text{II}}$ with $\text{param} = \{G_1, G_2, e, P, X, H_1, H_2\}$, where H_1, H_2 are random oracles that will be controlled by $\mathcal{B}$. $\mathcal{B}$ chooses an index I uniformly at random with $1 \leq I \leq q_{H_1}$.

The adversary $\mathcal{A}_{II}$ may make queries of the random oracles H_i ($i = 1, 2$) at any time during its attack. $\mathcal{B}$ responds as follows.

H₁ Queries. $\mathcal{B}$ maintains a list of tuples $\langle ID_i, Q_i \rangle$ in H_1 List L_1 . On receiving a query ID_i to H_1 , $\mathcal{B}$ responds as follows.

- (1) If ID_i already appears on the list L_1 in a tuple $\langle ID_i, Q_i \rangle$, $\mathcal{B}$ responds Q_i as an answer.
- (2) Otherwise, if $i \neq I$, choose $Q_i \in G_1^*$ at random and add $\langle ID_i, Q_i \rangle$ to L_1 and return Q_i as an answer.

H₂ Queries. $\mathcal{B}$ maintains a list of tuples $\langle ID_i, e_i, R_i \rangle$ in H_2 List L_2 . On receiving a query $\langle ID_i, e_i \rangle$ to H_2 , $\mathcal{B}$ responds as follows.

- (1) If ID_i already appears on the list L_2 in a tuple $\langle ID_i, e_i, R_i \rangle$, $\mathcal{B}$ responds R_i as an answer.
- (2) Otherwise, pick $R_i \in \{0, 1\}^*$ at random, add $\langle ID_i, e_i, R_i \rangle$ to L_2 , and return R_i as an answer.

Phase 1. After receiving *param* from $\mathcal{B}$, $\mathcal{A}_{II}$ issues a sequence of polynomially bounded number of requests, each request being either a *Private-Key-Extraction*, a *Request-Public-Key*, or *Decryption-Queries* for an entity. We assume that $\mathcal{A}_{II}$ always makes the appropriate H_1 queries on the identity ID before making one of these requests and never makes a decryption query on a ciphertext obtained from the encryption oracle. $\mathcal{B}$ replies to these requests as follows.

Private-Key-Extraction. $\mathcal{B}$ maintains a Private Key List of tuples $\langle ID_i, x_i, D_i \rangle$. On receiving a query ID_i , $\mathcal{B}$ responds as follows.

- (1) If $\langle ID_i, x_i, D_i \rangle$ exist in Private Key List, return $\langle x_i, D_i \rangle$ as an answer.
- (2) Otherwise, pick i at random, so that $\Pr[i \neq I] = \delta$. (δ is the same as it is in the proof of Lemma 7.) If $i \neq I$, run the simulation algorithm Request-Public-Key to get a tuple $\langle ID_i, x_i, PK_i \rangle$ and compute $D_i = sQ_i$, add $\langle ID_i, x_i, D_i \rangle$ to the Private Key List, and return $\langle x_i, D_i \rangle$ as an answer.
- (3) If $i = I$, return “Abort” and terminate.

Request-Public-Key. $\mathcal{B}$ maintains a Public Key List of tuples $\langle ID_i, x_i, PK_i \rangle$. On receiving a query ID_i , $\mathcal{B}$ responds as follows.

- (1) If $\langle ID_i, x_i, PK_i \rangle$ exist in Public Key List, return PK_i as an answer.
- (2) Otherwise, if $i \neq I$, choose $x_i \in Z_N^*$, compute $PK_i = x_i H_1(ID_i)$, add $\langle ID_i, x_i, PK_i \rangle$ to the Public Key List, and return PK_i as an answer.
- (3) If $i = I$, set $PK_i = aP$, add $\langle ID_i, *, PK_i = aP \rangle$ to the public key list, and return $PK_i = aP$ as an answer (where $*$ denotes the arbitrary value).

Decryption-Queries. On receiving a query $\langle ID_i, PK_i, C \rangle$, where $C = (c_1, c_2)$. $\mathcal{B}$ responds as follows.

- (1) If $i \neq I$, $\mathcal{B}$ decrypts C by using the private key $\langle x_i, D_i \rangle$.
- (2) Otherwise, search L_2 for a tuple $\langle ID_i, e_i, R_i \rangle$. If such a tuple exists, $\mathcal{B}$ retrieves the related R_i to compute $M \parallel \sigma = c_2 \oplus R_i$ and returns M as an answer.
- (3) Otherwise, $\mathcal{B}$ picks $R_i \in \{0, 1\}^*$ at random, computes $M \parallel \sigma = c_2 \oplus R_i$, and returns M as an answer. Add $\langle ID_i, e_i, R_i \rangle$ to L_2 .

Challenge Phase. $\mathcal{A}_{II}$ then outputs two messages (M_0, M_1) and a challenge identity ID^* . On receiving a challenge query $\langle ID^*, (M_0, M_1) \rangle$:

- (1) If $ID^* \neq ID_I$, $\mathcal{B}$ aborts the game.
- (2) Otherwise, $\mathcal{B}$ selects $r^* \in Z_N^*$ and $\sigma \in \{0, 1\}^*$ randomly, computes $c_1^* = r^* P$ and $H_2(e(\text{SK}_{ID^*}, c_1^*)) = c_2^* \oplus (M_\beta \parallel \sigma)$, returns $C^* = (c_1^*, c_2^*)$ as a target ciphertext.

Phase 2. $\mathcal{A}_{II}$ repeats the same methods it used in *Phase 1*. Moreover, no private key extraction on ID^* is allowed and no decryption query can be made on the ciphertext C^* for the combination of identity ID^* and public key PK_{ID^*} that encrypted plaintext M_β .

Guess. $\mathcal{A}_{II}$ should make a guess β' for β . The adversary wins the game if $\beta' = \beta$.

Analysis. This part is similar to *Analysis* in the proof of Lemma 7.

Consequently, we obtain $\epsilon' > (1/q_{H_2})(2\epsilon/e(q_{\text{prv}} + 1) - q_{H_1}/2^\omega - q_D q_{H_1}/2^\omega - q_D/N)$. $\square$

These two lemmas complete the proof of Theorem 6.

Furthermore, in our CL-PKE scheme, $G_1 = E_N(a, b)$ is an elliptic curve over the ring Z_N , where $N = pq$. According to Theorem 6, in order to run the algorithm on $E_N(a, b)$, such as addition and scalar multiplication, we should construct two elliptic curves $E_p(a, b)$ and $E_q(a, b)$ firstly and execute the corresponding operation on them, respectively. Then, we use the results of operation on $E_p(a, b)$ and $E_q(a, b)$ to present the operation on $E_N(a, b)$, which means that we should factoring N into p and q firstly. Therefore, the security of our CL-PKE scheme is also based on the intractability of factoring the large number.

3.3. Comparison to Related Schemes. In this section, we compare the proposed scheme with other related CL-PKE schemes on the computation complexity of encryption (Enc) and decryption (Dec), security level (Sec-Lev), and security assumption (Sec-Ass), where RSA in Table 1 represents the problem of factoring the large number. Without considering the addition of two points and hash function on the elliptic curves, all the schemes have three major operations, that is, Pairing (P), Scalar Multiplication (S), and Exponentiation (E).

TABLE 1: Comparison of the CL-PKE schemes.

Schemes	Enc	Dec	Sec-Lev	Sec-Ass
[2]	3P + 1S + 1E	1P + 1S	IND-CCA secure	GBDHP
[4]	1P + 2S + 1E	1P + 2S	IND-CCA secure	BDH
[5]	3S + 1E	1P + 3S	IND-CCA secure	K-BDHI
[6]	1P + 3S + 1E	4P	Strong type I/II secure	3-DDH
[7]	2P + 2S + 2E	2P + 1S	IND-CCA secure	BDH
Our scheme	1P + 2S	1P	IND-CCA secure	BDH/RSA

The essential operation of our proposed scheme is to compute a bilinear pairing. According to [16], the computation of a bilinear pairing becomes efficient. From Table 1, compared with the related works, our scheme needs only one pairing and two scalar multiplications in *Encrypt* and one pairing operation in *Decrypt*, which will consume less energy while preserving a higher security level. Therefore, the scheme proposed in this paper is more suitable to be applied in IoT with the characteristics of low cost, low power, multifunctional sensor nodes that are small in size and communicate wirelessly with each other in a short distance.

4. Application to Internet of Things

In this section, based on our proposed scheme, we present a protocol to protect the confidentiality and integrity of the transmitting information between the gateway node and the sensor node.

In IoT, Smart Car, as illustrated in Figure 1, the gateway nodes in the car is responsible for transmitting the information of driving, such as the speed and location that are collected by sensor nodes fixed in the car, to the platform so that it can be monitored in real time. Furthermore, the gateway node will deliver the data and signaling obtained from the platform to sensor nodes for updating and managing equipments. Consequently, the transmitted information between the gateway node and the sensor nodes should be protected in confidentiality and integrity from being eavesdropped and destroyed by an adversary and damaging the normal operation of a Smart Car as a result.

Considering the sensor node with identity $ID \in \{0, 1\}^w$ in the initial phase, the details of this encryption scheme are as follows.

The algorithms of *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, and *Set-Public-Key* are the same as that of the proposed CL-PKE scheme in Section 3.1.

Encrypt. To encrypt $M \in \{0, 1\}^*$, the gateway node of the Smart Car selects a random value $r \in Z_N^*$, computes $C = (c_1, c_2)$ such that

$$c_1 = rP, \quad c_2 = (M \parallel ID) \oplus H_2(e(PK_{ID}, rX)) \quad (14)$$

sends the ciphertext $C = (c_1, c_2)$ to the sensor node.

FIGURE 1: Transmitting the data and signaling to the sensor node in a Smart Car.

Decrypt. To decrypt ciphertext $C = (c_1, c_2)$, sensor node fixed in the car uses its private key SK_{ID} and computes

$$M' \parallel ID' = c_2 \oplus H_2(e(SK_{ID}, c_1)). \quad (15)$$

The sensor node checks whether the equation $ID' = ID$ holds. If the verification succeeds, return M' as the legal data and signaling. Otherwise, return "Reject".

According to the encryption scheme above, this encryption scheme can not only protect the confidentiality of the transmitted data and signaling between gateway node and sensor node in the Smart Car, but also the integrity. This is because the sensor node can determine the transmitted data that was distorted or destroyed by an adversary or the reasons of environment in the public wireless channel, provided that ID' is not equal to ID in the phase of *Decrypt*. Moreover, there is only one pairing operation in *Decrypt*, which satisfies the characteristics of low-cost, low-power, and low-computation of the sensor node in the Smart Car.

5. Conclusion

We have proposed a CL-PKE scheme on the elliptic curve over the ring Z_n and proved that the scheme is IND-CCA secure in the random oracle model, relative to the intractability of the BDHP and factoring the large number problem. A comparison in Table 1 concludes that the proposed scheme is advantageous over the existing related schemes on performance. Due to the appealing properties, based on the proposed one, we also present another CL-PKE scheme for Smart Car in the end, which can provide confidentiality and integrity.

Conflict of Interests

The authors declare that we have no conflict of interests.

Acknowledgments

This work is supported by NSFC (Grant nos. 61300181, 61272057, 61202434, 61170270, 61100203, 61121061), the Fundamental Research Funds for the Central Universities (Grant nos. 2012RC0612, 2011YB01, GK201402006), and China Postdoctoral Science Foundation (Grant no. 2013M530561).

References

- [1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, 1984.
- [2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '03)*, C. S. Lai, Ed., vol. 2894 of *Lecture Notes in Computer Science*, pp. 452–473, Springer, 2003.
- [3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.
- [4] Z. Cheng and R. Comley, "Efficient certificateless public key encryption," Tech. Rep. 2005/012, Cryptology ePrint Archive, 2005.
- [5] Y. Shi and J. Li, "Provable efficient certificateless public key encryption," Tech. Rep. 2005/287, Cryptology ePrint Archive, 2005.
- [6] A. W. Dent, B. Libert, and K. G. Paterson, "Certificateless encryption schemes strongly secure in the standard model," in *Proceedings of the International Workshop on Practice and Theory in Public-Key Cryptography (PKC '08)*, vol. 4939 of *Lecture Notes in Computer Science*, pp. 344–359, Springer, 2008.
- [7] Y. X. Sun and H. Li, "Short-ciphertext and BDH-based CCA2 secure certificateless encryption," *Science in China F*, vol. 53, no. 10, pp. 2005–2015, 2010.
- [8] International Telecommunication Union, *ITU Internet Reports 2005: the Internet of Things*, 2005.
- [9] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [10] R. H. Weber, "Internet of Things—new security and privacy challenges," *Computer Law and Security Review*, vol. 26, no. 1, pp. 23–30, 2010.
- [11] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of things," *Computer*, vol. 44, no. 9, Article ID 6017172, pp. 51–58, 2011.
- [12] A. Guillevis, "Comparing the pairing efficiency over composite-order and primeorder elliptic curves," in *Proceedings of the Applied Cryptography and Network Security (ACNS '13)*, vol. 7954 of *Lecture Notes in Computer Science*, pp. 357–372, Springer, 2013.
- [13] L. C. Washington, *Elliptic Curves: Number Theory and Cryptography*, Chapman and Hall/CRC, 2nd edition, 2008.
- [14] K. Koyama, U. Maurer, T. Okamoto, and S. Vanstone, "New public-key schemes based on elliptic curves over the ring Z_n ," in *Advances in Cryptology*, vol. 576 of *Lecture Notes in Computer Science*, pp. 252–266, Springer, 1992.
- [15] W. Bosma and H. W. Lenstra, "Complete systems of two addition laws for elliptic curves," *Journal of Number Theory*, vol. 53, no. 2, pp. 229–240, 1995.
- [16] X. Xiong, D. S. Wong, and X. Deng, "TinyPairing: a fast and lightweight pairing-based cryptographic library for wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, April 2010.