A Light-Weight Certificate-Less Public Key Cryptography Scheme Based on ECC

Xuanxia Yao, Xiaoguang Han
School of Computer and Communication Engineering
University of Science and Technology Beijing (USTB)
Beijing, China, 100083
yaoxuanxia@163.com, guanguang1225@sohu.com

Xiaojiang Du
Department of Computer and Information Sciences
Temple University
Philadelphia, PA, 19122, USA
dxj@ieee.org

Abstract—**With the rapid development of mobile computing, more and more mobile devices, such as smart phones and tablets are able to access Internet. As these mobile devices are usually battery powered, energy efficiency is a very important issue. For most mobile applications, energy saving should be considered at the design stage. Of course, security application is no different. Public key cryptography plays an important role in network security, and it is still essential in mobile computing despite it needs high energy consumption. Considering Elliptic Curve Cryptography (ECC) is easy to perform in hardware and needs lower energy than other public key algorithms. We propose an ECC-based certificate-less public key cryptography scheme. The scheme is lightweight and can save energy for mobile devices. Firstly, it does not need certificate to prove the authenticity of a public key, which can save energy for certificate transmission. Secondly, it is constructed on the traditional ECC instead of bilinear pairing, which makes it lightweight and can save energy for computation. In addition, it avoids the key escrow issue, which makes it has higher security strength than traditional public key cryptography. These advantages make it very suitable for resources-constrained mobile devices.**

Keywords—elliptic curve cryptography; certificate-less public key cryptosystem; certificate-less public key encryption; certificate-less public key signature

I. Introduction

Public key cryptosystem is used in a wide range of security applications by providing digital signatures (or authentication) and convenient key exchanging services, where public key management is very important. In order to deal with the impersonation attack on the public key cryptography (PKC), it is obliged to authenticate public keys so as to make sure that the public keys are valid, authentic and indeed belong to the claimed users. A common method is to introduce an authority trusted by all users into the system [1]. The trusted authority can prevent the impersonation attack by three approaches, namely certificate-based public key cryptography (such as PKI), identity-based public key cryptography (ID-PKC), and certificate-less public key cryptography (CL-PKC) [2].

In certificate-based public key cryptosystem, the authenticity of a public key is guaranteed by a digital certificate, which binds a public key with the key's owner via a signature generated by the trusted certification authority (CA). In order to get an authentic and valid public key, one has to verify many certificates in a trust chain, which leads to large communication and computational overheads [3]. In addition, since an infrastructure has to be constructed to issue certificates, and certificate management is complicated, certificate-based public key cryptography is not suitable for resources-constrained mobile devices.

In identity-based public key cryptography, public keys are derived from users IDs (such as phone numbers, e-mail addresses and so on) and can surely be authenticated by themselves. It can guarantee the authenticity of public key and simplify the public key management. Certificates are no longer necessary [2]. However, since private keys are derived from users' public keys (or their identities) via a trusted authority (TA), the authority knows every user's private key and can decrypt users' cipher-texts or impersonate a user, which is referred to as the key-escrow problem. The key-escrow problem is undesirable for most applications.

Certificate-less public key cryptography is between the identity-based public key cryptography and the traditional certificate-based public key cryptography, which is constructed on the basis of ID-PKC. The authority is a Key Generation Center (KGC), which generates partial private key for a user according to its ID. Users generate their own key pairs. The private key is generated from its secret value associated with its public key and the partial private key. Since user identity is used with its public key, the public key can also be authenticated by itself and a certificate is unnecessary. Since the secret value is not computable from others, the authority cannot figure out the private key corresponding to a user's public key. In addition, CL-PKC eliminates the key escrow problem in ID-PKC [4]. These two appealing properties make it attractive and much work has been done on how to design a certificate-less public key cryptography scheme.

Unfortunately, the existing certificate-less public key cryptography schemes are not well suited for resources-constrained mobile devices because of their large computational overheads and/or communication overheads. Therefore, it is important to design a lightweight certificate-less public key cryptographic scheme for mobile computing.

The main contributions of this paper include two aspects: 1) we develop a lightweight certificate-less public key cryptographic scheme based on elliptic curve cryptography; and 2) we conduct a detailed performance analysis for the proposed scheme and compare it with existing schemes.

The rest of this paper is organized as follows. In section II, we summarize related work on certificate-less public key cryptography. Section III lists notations used in this paper and reviews related knowledge of ECC. Section IV presents the proposed lightweight certificate-less public key cryptography scheme. Section V analyzes the performance of the proposed scheme, and compares it with other schemes. Section VI concludes this paper.

II. Related Work

Certificate-less public key cryptography was firstly introduced by Al-Riyami and Paterson in 2003 [5]. CL-PKC can not only authenticate public keys without any certificates but also eliminate the key escrow problem in ID-PKC. Due to the aforementioned reasons, CL-PKC has been a very hot research topic in the field of cryptology and information security. At present, Many CL-PKC schemes have been proposed.

Most of the existing CL-PKC schemes are based on bilinear pairing [5-9], which means expensive computational costs. For example, the time needed for one bilinear pairing operation is twice of that for one modular exponential operation and 20 times of that for one point multiplication [10]. In addition, the bilinear pairing based CL-PKC schemes have a lot of system parameters, which cause large communication overheads.

To improve the efficiency, researchers have proposed a few CL-PKC schemes without using bilinear pairing(e.g., [3, 11-17]). For instance, Yum *et al.* [12], as well as Hu *et al.* [13] presented certificate-less public key signature schemes by combining traditional public key signature and identity based signature, which avoid bilinear pairing operation. Harn *et al.* [15] constructed a no pairing certificate-less public key signature scheme based on ElGamal signature. Baek *et al.* [11] also proposed a certificate-less public key encryption without pairing, which is based on the Schnorr's signature. And Lai *et al.* [18] presented a pairing-free certificate-less public key encryption scheme based on RSA. Although these schemes do not use pairings, they are still not efficient enough for resources-constrained mobile devices because they use modular exponential operations.

Since scalar multiplication on elliptic curve runs much faster than modular exponential operation, researchers have been designing no-pairing public key cryptography schemes based on ECC. In 2010, Ge *et al.* [16] presented a no-pairing public key signature scheme based on ECC. This scheme needs 7 point multiplication operations in the verification algorithm, so it is still not efficient enough. He *et al.* [17] proposed an efficient and pairing-free certificate-less signature scheme based on ECC, however, it cannot defend attacks from KGC. In 2012, the Internet Engineering Task Force (IETF) released RFC 6507, which describes an elliptic curve-based certificate-less signatures for identity based encryption (ECCSI) [3]. In this scheme, the pair of public key and private key is constructed by the common Key Management Service (KMS), which acts as a root of trust for both communication parties. Since the private key is generated by KMS, the key escrow problem still exists. Strictly speaking, it is not a real certificate-less public key signature scheme. In 2013, Wang *et al.* [14] presented a certificate-less signature scheme without pairings based on the Elliptic Curve Digital Signature Algorithm (ECDSA) [21] and Schnorr's signature [19], which needs 1 scalar multiplication in signature algorithm and 7 scalar multiplications in verification algorithm, so it is more efficient than other certificate-less public key signature schemes. However, most of the above schemes need more than one complicated mapping function, which makes them less efficient.

ECC algorithm can run efficiently in mobile devices [20-23]. Based on this fact, we present a no-pairing certificate-less public key cryptography scheme based on ECC in this paper. Our scheme employs the idea of the elliptic curve integrated encryption schemes to construct a certificate-less public key encryption algorithm, and the certificate-less public key signature algorithm is based on ECDSA [21] and ECIES [24]. The encryption algorithm needs 2 scalar multiplications and the decryption algorithm only needs 1 scalar multiplication. The number of scalar multiplications for signature and verification is 1 and 3, respectively, which is better than Wang's scheme [14]. Our scheme is based on ECC but it has the two appealing features of CL-PKC. Both encryption and signature can be realized by the scalar multiplication. Furthermore, it only needs one simple hash function, while similar schemes need more than one complicated mapping function. These properties give our scheme distinct advantages over the existing schemes.

III. Preliminaries

Preliminaries for the proposed certificate-less public key cryptographic scheme include three aspects. Subsection III-A lists notations used in this paper. Subsection III-B reviews elliptic curve cryptography, and subsection III-C discusses certificate-less public key cryptography.

A. Natations

The notations or parameters used in the rest of the paper are listed in Table I.

TABLE I. Notations

Notations	Meaning
l	The security parameter of the certificate-less public key system, which is the size in bits of the private key for the elliptic curve cryptographic algorithm to be run.
q	A large prime.
F_q	A finite field with q elements $\{0, q\text{-}1\}$.
E	An elliptic curve over the finite field F_q, who should have a subgroup with a large prime order.
$\#E$	The number of points on one elliptic curve, which is called the curve order.
G	A base point on the elliptic curve E,
n	A large prime, which is defined to be the order of a subgroup on E.

Notations	Meaning
l	The security parameter of the certificate-less public key system, which is the size in bits of the private key for the elliptic curve cryptographic algorithm to be run.
$\boldsymbol{O}$	The zero element of an elliptic curve
H	A cryptographic hash function, who can map arbitrary strings to strings of l bits.
$HMAC$	A MAC generation function, the length of MAC is l.
s	The master private key of the certificate-less public key system
P_{pub}	The master public key of the certificate-less public key system
$params$	The public parameters of the certificate-less public key system
a	An element of the finite field F_q, which is used as a factor of an elliptic curve.
b	An element of the finite field F_q, which is used as a factor of an elliptic curve.
h	The cofactor of the elliptic curve, which is computed by $h=\#E/n$.

B. Elliptic Curve Cryptography

ECC was introduced by Victor Miller and Neal Koblitz in 1985, whose security depends on the Elliptic Curve Discrete Logarithm problem (ECDLP). An ECC scheme can be defined by a set of parameters (q, a, b, G, n, h) and applied to digital signatures, encryption and key exchange. ECC has two features, which make it very suitable for resources-constrained environments.

- It needs much smaller key size than other public key cryptographic schemes (such as RSA) for the same level of security.
- Its' scalar multiplication is much faster than modular exponential operation and is easy to be implemented in hardware.

In this paper, we employ ECC to construct a certificate-less public key cryptography, which can be used for both encryption and signature. The encryption algorithm is based on the Elliptic Curve Diffie-Hellman (ECDH) and Elliptic Curve Integrated Encryption Scheme (ECIES). The signature is based on the Elliptic Curve Digital Signature Algorithm (ECDSA). The related preliminaries of ECC is reviewed as follows.

1) Elliptic Curve Discrete Logarithm Problem

Given a elliptic curve E=(q, a, b, G, n, h) over a finite field F_q, choosing a secret number s from [1, n-1], we can compute $Q=s\times G$ easily. However, it is very difficult to compute s via Q and G, which is referred to as the ECDLP.

2) Elliptic Curve Diffie-Hellman

Elliptic Curve Diffie-Hellman is a Diffie-Hellman key exchange scheme based on elliptic curves, which can help two parties that communicate over an insecure channel to generate a shared secret. The shared secret can be used (as a key) to provide data confidentiality and integrity. Furthermore, ECDH can achieve the same security level with much smaller key size than the original Diffie-Hellman scheme.

Assume that Alice and Bob use the same ECC system (q, a, b, G, n, h) to generate their key pairs (S_A, P_A) and (S_B, P_B) respectively. According to the ECDH scheme, a shared secret $K_{A,B}$ can be generated by (1).

$$K_{A,B}= S_A\times P_B= S_B\times P_A= S_A\times S_B\times G \qquad (1)$$

3) Elliptic Curve Integrated Encryption Scheme

ECIES is an encryption scheme based on ECC. As its name indicates, ECIES is an integrated encryption scheme, which can provide both data confidentiality and integrity. It uses ECDH to generate a shared secret, from which the encryption key and the integrity (or Message Authentication Code, MAC) key are derived. The confidentiality is guaranteed by a symmetric cryptographic algorithm with the encryption key and the integrity is guaranteed by a MAC generation function with the integrity key.

To encrypt a message m for Bob, Alice should do the following 7 steps:

① Select a random integer r from [1, n-1].
② Compute $R = r\times G$.
③ Compute $K = r\times P_B = (K_X, K_Y)$.
④ Check whether $K = \boldsymbol{O}$ or not, if yes, go to Step ①.
⑤ Compute $K_{ENC}\|K_{MAC}= KDF(K_X)$, here, KDF is a key derivation function.
⑥ Compute $c=ENC (K_{ENC}|, m)$ and $e =HMAC (K_{MAC}, c)$.
⑦ Send "$R\|c\|e$" to Bob.

To decrypt the cipher-text "$R\|c\|e$", Bob should do 5 steps:

① Check whether R is on the elliptic curve, if not, output error symbol "$\perp$" and stop.
② Compute $K = S_B\times R = (K_X, K_Y)$ and check whether $K =O$, if yes, output "$\perp$" and stop.
③ Compute $K_{ENC}\|K_{MAC} = KDF(K_X)$.
④ Verify whether $e = HMAC(K_{MAC}, c)$, if not, output "$\perp$" and stop.
⑤ Compute $m = DEC(K_{ENC}|, c)$.

4) Elliptic Curve Digital Signature Algorithm

ECDSA is a variant of the Digital Signature Algorithm (DSA) on elliptic curve. When Alice wants to sends a message m and the signature to Bob, she should perform 5 steps.

① Choose a random integer k from [1, n-1].
② Compute $k\times G = (K_x, K_y)$.
③ Compute $r = K_x \bmod n$. If $r = 0$, go to step ①.
④ Compute $e = H(m)$.
⑤ Compute $s= k^{-1}\{e+S_A\times r\} \bmod n$. If s=0, go to step ①.

Alice's signature for the message m is (r, s).

In order to verify the message m and the signature (r, s), Bob should do the following 7 steps:

① Check whether r and s are in [1, n-1] or not, if not, output "$\perp$" and stop.
② Compute $e = H (m)$.
③ Compute $w = s^{-1} \bmod n$.
④ Compute $u_1 = e\times w \bmod n$ and $u_2 = r\times w \bmod n$.
⑤ Compute $(x_1, y_1) = u_1\times G+ u_2\times P_A$.
⑥ Compute $v = x_1 \bmod n$.
⑦ If $v = r$, accept the signature, otherwise, output "$\perp$".

C. Certificate-Less Public Key Cryptography

A certificate-less public key cryptographic scheme can be formally described by the following 7 algorithms.

1) Setup

The setup algorithm is run by the Key Generation Center (KGC) . It is used to set up a certificate-less public key system, which can be represented by the system public parameters *params*. All the system parameters should be known by all participants or interested parties. The system's master private key *mpk* and the master public key *msk* are generated by the security parameter *l*. The setup algorithm can be denoted as (*params*, *msk*, *mpk*) =Setup(1^l).

2) Partial Key Extraction

The partial key extraction algorithm is also run by KGC, which takes the system parameters *params*, master key and a user *ID* as input and outputs a partial private key d_{ID} and a partial public key R_{ID}. The partial key can be represented by (d_{ID}, R_{ID}) and should be sent to the user through a secure channel. The partial key extraction algorithm can be denoted by (d_{ID}, R_{ID}) = PartialKeyExtract(*params*, *msterkey*, *ID*).

3) Key Generation

The key generation algorithm is run by the user itself, and there are three tasks. The first one is to choose a secret value z_{ID} randomly for a user *ID* according to the system public parameters *params*. The second one is to generate the private key S_{ID} for a user *ID* from *params*, d_{ID} and z_{ID} and *ID*. The third one is to construct the public key P_{ID} according to *params*, R_{ID}, and z_{ID}. The key generation algorithm can be denoted by (S_{ID}, P_{ID})=Keygeneration (*params*, *partialkey*, z_{ID}, *ID*).

4) Encryption

The encryption algorithm is run by the sender, which takes *params*, the receiver's *ID*, the receiver's public key P_{ID}, and a message *m* to be encrypted as input, and outputs a ciphertext *c*. That is, *c*=Encryption(*params*, P_{ID}, *ID*, *m*).

5) Decryption

The decryption algorithm is run by the receiver, which takes *params*, the receiver's *ID*, the receiver's private key S_{ID}, and the ciphertext *c* as input, and outputs the corresponding plaintext *m* or an error symbol "$\perp$". That is *m*=Decryption(*params*, S_{ID}, *ID*, *c*).

6) Signature

The signature algorithm is run by the message sender, which takes *params*, the signatory's *ID*, the signatory's private key S_{ID} and message *m* as input, and outputs its signature *sig* on the message *m*. That is *sig*=Signature(*params*, *ID*, S_{ID}, *m*).

7) Verification

The verification algorithm is run by the message receiver, which takes *params*, the master public key *mpk*, the signatory's *ID*, the signatory's public key P_{ID}, message *m* and its signature *sig* as input, outputs the verification result of valid or invalid. That is, Verification(*params*, *mpk*, *ID*, P_{ID}, *m*, *sig*).

IV. Our Lightweight Certificate-Less Public Key Cryptographic Scheme

The proposed scheme consists of 7 components, which are presented as follows.

A. System Setup

In setup stage, according to the security parameter *l*, an elliptic curve *E* over the finite field *F_q* is defined by the set of parameters (*q*, *a*, *b*, *G*, *n*, *h*). The master public and private key pair are generated by KGC.

KGC selects *s* randomly from the interval [1,*n*-1] as the system's master private key, which is also called master key and should be kept secret by KGC.

KGC computes $P_{\mathrm{pub}}=s\times G$, which is the system's master public key.

The system's public parameter *params* is { *F_q*, *E*/ *F_q*, *G*, P_{pub}, *H*}.

B. Extract Partial Private Key

For user *A* with identity ID_A, KGC does the following 4 steps to extract partial public and private keys for it.

① Choose r_A randomly from interval [1, *n*-1].

② Compute d_A using (2).

$$d_A=(s+r_A\times H(ID_A)) \bmod n \tag{2}$$

③ If d_A=0, go back to Step ①.

④ Compute $R_A=r_A\times G$.

(d_A, R_A) is the partial key corresponding to user *A*'s identity ID_A. KGC sends (d_A, R_A) to user *A* via a secure channel.

C. Key Generation

To generate a public and private key pair, user *A* should do the following 7 steps after receiving its partial key.

① Choose an integer z_A randomly from interval [1, *n*-1].

② Compute *A*'s private key corresponding to ID_A by (3).

$$S_A=(d_A+z_A\times H(ID_A)) \bmod n \tag{3}$$

③ If S_A=0, go to step ①, otherwise keep S_A secretly.

④ Compute Z_A by (4).

$$Z_A=z_A\times G \tag{4}$$

⑤ Compute $X_A=R_A+Z_A$

⑥ If $X_A=\boldsymbol{O}$, go back to step ① to re-choose a new z_A.

⑦ X_A is *A*'s executive public key corresponding to its identity ID_A and will be released to participants. $P_A=S_A\times G$ is *A*'s actual public key based on ECC, which can be calculated by (5).

$$\begin{aligned} P_A &= S_A\times G=(d_A+z_A\times H(ID_A))\times G=d_A\times G+z_A\times H(ID_A)\times G \\ &=(s+r_A\times H(ID_A))\times G+H(ID_A)\times Z_A \\ &=s\times G+r_A\times H(ID_A)\times G+H(ID_A)\times Z_A \\ &=P_{\mathrm{pub}}+H(ID_A)\times(R_A+Z_A) \\ &=P_{\mathrm{pub}}+H(ID_A)\times X_A \end{aligned} \tag{5}$$

The relationship between a user *A*'s private key and its executive public key can also be seen in (5).

In addition, (5) shows that the public key P_A is not only bonded with its identity but also constructed on the system's master public key P_{pub}, which means no certificate needed to prove the binding relationship between a public key and the identity. In other words, the public key can prove its authenticity by itself.

D. Encryption

A user's public and private key pair can be used to encrypt and decrypt data. Consider that user A wants to send a message m encrypted by user B's public key, it should run the following 7 steps:

① Randomly select k from interval $[1, n\text{-}1]$.
② Compute $R = k\times G$.
③ Compute $K=k\times(P_{\text{pub}}+ H(ID_B)\times X_B)=(K\_x, K\_y)$.
④ If $K = \boldsymbol{O}$, go back to step ①.
⑤ Compute $K_{ENC}=H(K\_x)$ and $K_{MAC}=H(K\_y)$.
⑥ Compute $c_1=ENC\ (K_{ENC}, m)$ and $c_2 = HMAC(K_{MAC}, m)$. Here, ENC is a symmetric encryption scheme such as AES.
⑦ Send "$K\|\ c_1\|\ c_2$" to user B.

E. Decryption

After receiving "$K\|c_1\|\ c_2$", user B can decrypt c using its private key S_B. The decryption algorithm is described as follows:

① Compute $K'=S_B\times K= S_B\times k\times G=(K'\_x, K'\_y)$.
② Check whether $K'= \boldsymbol{O}$ or not, if yes, the cipher-text should be rejected, output "$\perp$" and go to step ⑦.
③ Compute $K'_{ENC}=H(K'\_x)$ and $K'_{MAC}=H(K'\_y)$.
④ Compute $m'=DEC(K'_{ENC}, c_1)$.
⑤ Compute $c_2'=HMAC(K'_{MAC}, m')$.
⑥ If $c_2= c_2'$, output m', else output "$\perp$".
⑦ Stop.

Proof of the decryption is given below:

$$\begin{aligned}K &=k\times(P_{\text{pub}}+ H(ID_B)\times X_B)\\ &= k\times(s\times G + H(ID_B) \times (R_B+ Z_B))\\ &= k\times(s\times G +H(ID_B) \times (r_B+ z_B)) \times G)\\ &= k\times(s\times G +(r_B\times H(ID_B) + z_B\times H(ID_B)) \times G)\\ K' &=s_B\times K= s_B\times k\times G\\ &= (d_B+ z_B\times H(ID_B)) \times k\times G\\ &=(s+ r_B\times H(ID_A) + z_B\times H(ID_A)) \times k\times G\\ &=k\times(s\times G +(r_B\times H(ID_B) + z_B\times H(ID_B)) \times G)\\ &= K\end{aligned}$$

As show above, both the encryption key and the integrity key can be reconstructed by the receiver. Then the integrity of message m can be verified.

The above encryption and decryption algorithms are similar to the elliptic curve integrated encryption scheme (ECIES) or the ECIES-based schemes. However, since the executive public key is not identical to the traditional public key in ECC, the process of generating encryption key and integrity key is different from the traditional ECIES.

F. Signature

The public and private key pair generated in the subsection IV-C can also be used for signature and verification. We construct our signature and verification process based on the basic idea of the ECDSA signature algorithm.

The process of user A generating a signature on message m is illustrated as following 7 steps.

① Choose k randomly from interval $[1, n\text{-}1]$.
② Compute $R=k\times G =(r\_x,r\_y)$. R is a point in the sub-group of elliptic curve E and r_x is the X-axis value of point R.
③ Compute s_1 by (6), which is the first part of the signature.

$$s_1= r\_x \bmod n \tag{6}$$

④ Check whether (7) is true, if yes, go back to step ①.

$$H(m\|ID_A) + S_A \times s_1 = 0 \bmod n \tag{7}$$

⑤ Compute s_2 by (8), which is the 2th part of the signature.

$$s_2= k\times(H(m\|ID_A)^{-1}+S_A\times s_1) \bmod n \tag{8}$$

⑥ The signature s consists of s_1 and s_2, that is, $s=(s_1,s_2)$.
⑦ User A sends message "$ID_A\|X_A\|m\|\ s_1\|s_2$" to user B.

G. Verification

After receiving the above message, the verifier B conducts the following 8 steps to verify the signature.

① Check whether both s_1 and s_2 are in the interval $[1,n\text{-}1]$, if not, the signature should be rejected, go to step ⑧.
② Compute hash value HM according to (9).

$$HM=H\ (m\|ID_A) \tag{9}$$

③ Calculate $v1$ by (10).

$$v_1=(HM \times s_2) \bmod n \tag{10}$$

④ Calculate $v2$ by (11).

$$v_2=(\ s_1 \times s_2) \bmod n \tag{11}$$

⑤ Calculates P_A according to (12).

$$P_A =P_{\text{pub}}+H(ID_A)\times X_A \tag{12}$$

⑥ Calculates R' by (13).

$$R'= v_1\times G+ v_2\times P_{\text{A}}=(r'\_x, r'\_y) \tag{13}$$

In (13), r'_x is the X-axis value of R'.

⑦ Check whether (14) is true, if yes, the signature passes the verification; otherwise, the signature is invalid.

$$r'\_x \bmod q= s_1 \tag{14}$$

⑧ Stop.

The verification process depends on (15).

$$\begin{aligned}R' &= v_1\times G + v_2\times(P_{\text{pub}}+H(ID_A)\times X_A))\\ &= v_1\times G + v_2\times(P_{\text{pub}}+H(ID_A)\times X_A))\\ &= v_1\times G +v_2\times(s\times G+ H(ID_A)\times(R_A+ Z_A))\\ &= v_1\times G+ v_2\times(s\times G+ H(ID_A)\times(r_A+ z_A) \times G)\\ &= v_1\times G+ v_2\times(s+ r_A\times H(ID_A)+z_A\times H(ID_A)) \times G\\ &= v_1\times G+ v_2\times(d_A+z_A\times H(ID_A)) \times G\\ &= v_1\times G+ v_2\times s_A\times G\\ &= (H(m\|ID_A) \times s_2+s_1\times s_2\times s_A)\times G\\ &= (H(m\|ID_A) + s_1\times s_A) \times s_2\times G\\ &= (H(m\|ID_A) + s_1\times s_A) \times k\times(H(m\|ID_A)^{-1}+ s_A\times s_1) \times G\\ &= k\times G\\ &=(r\_x,r\_y).\end{aligned} \tag{15}$$

V. Performance Analysis

To evaluate the proposed certificate-less public key cryptography scheme objectively, we analyze its performances from both security and overheads.

A. Security Analysis

Security has two meanings; they are the security of the key and the security of the algorithms. Security of the key refers to the public key's security, because the private key is generated and kept secretly by the user itself.

1) The Security of Public Key

a) Public Key's Validity

The public key's validity is the perquisite to assure security of the scheme. The proposed scheme is constructed on ECC. Validity of the public key P_A can be verified by the following three conditions:

- X_A and P_A are all points on the elliptic curve E.
- $X_A \neq \boldsymbol{O}$ and $P_A \neq \boldsymbol{O}$.
- $n\times X_A = \boldsymbol{O}$ and $n\times P_A = \boldsymbol{O}$.

The reason is described in the explicit validation algorithm of a public key in ECDSA. Based on the three conditions, we prove the validity of the public key X_A and P_A as following steps:

① Since G is a base point on E, whose order n is a large prime number, the point set on the elliptic curve E generated by G, the point at infinity $\boldsymbol{O}$, and the point addition operation can form a cyclic group with order n.

② According to the closure nature of a cyclic group, for an integer i within the interval [1, n-1], $i\times G$ should still be in the group point set, which means $i\times G$ is a point on the elliptic curve E.

③ In the proposed scheme, both r_A and z_A are within the interval [1, n-1]. Therefore, R_A and Z_A are points on the elliptic curve E.

④ $X_A=R_A+Z_A=((r_A+z_A) \bmod n)\times G$. According to the cyclic nature of the elliptic curve group adopted in the proposed scheme (in fact, all the elliptic curve groups adopted in cryptograph are cyclic groups), X_A should also be a point on the elliptic curve E and in the group point set.

⑤ $P_A=S_A\times G$, $S_A=(d_A+ z_A\times H(ID_A)) \bmod n$, similar to step ④, P_A should also be a point on the elliptic curve E and in the group point set.

⑥ Prove that $X_A \neq \boldsymbol{O}$ and $P_A \neq \boldsymbol{O}$.

If $X_A = \boldsymbol{O}$, since $X_A=R_A+Z_A=(r_A+z_A)\times G$, we have $x_A= (r_A+z_A) \bmod n =0$, which can be expressed as in (16).

$$x_A = j\times n \tag{16}$$

Since x_A is in the interval [1,n-1], (16) means that x_A must have two factors x_1 and x_2 that satisfying $x_1\times x_2=n$, which conflicts with the fact that n is a prime. Therefore, there exist x_A such that $x_A \bmod n \neq 0$.

According to the definition of the cyclic elliptic group from the generator G with order n, for any $i\in[1,n\text{-}1]$, there should be $i\times G \neq \boldsymbol{O}$ and $n\times G = \boldsymbol{O}$. Since $(r_A+z_A) \bmod n\neq 0$, we have $X_A= (r_A+z_A)\times G \neq \boldsymbol{O}$.

⑦ Prove that $P_A \neq \boldsymbol{O}$, which is similar to step ⑥.

⑧ Prove that $n\times X_A = \boldsymbol{O}$ and $n\times P_A = \boldsymbol{O}$.

Similar to step ⑥,

$n\times X_A= n\times(r_A+z_A)\times G=(r_A+z_A)\times(n\times G)= (r_A+z_A)\times\boldsymbol{O} = \boldsymbol{O}$.

$n\times P_A= n\times s_A\times G= s_A\times(n\times G)= s_A\times\boldsymbol{O} = \boldsymbol{O}$.

Therefore, X_A is a valid executive public key corresponding to ID_A and P_A is a valid public key corresponding to ID_A. ID_A's private key is S_A.

b) Authenticity of the Public Key

The public key's authenticity means that a public key indeed belongs to the claimed owner. Similar to other certificate-less public key cryptography, the public key's authenticity of the proposed scheme is guaranteed by the following two facts:

- User's identity is used with its public key, which has been discussed in both the encryption algorithm and verification algorithm.
- The KGC's secret key s is used to generate a user's private key, which has been discussed in the key generation algorithm.

In addition, we can see from the key generation algorithm that both the user's ID and system's public key are integrated into the relationship between the private key and public key. This relationship bonds a public key with the user's ID. Even if a malicious attacker can successfully replace a victim's public key with its own public key, the partial key obtained by the victim from the authority is not known to the attacker, and the attacker still cannot generate a valid signature or decrypt the cipher text by the fake public key and the victim's ID.

2) Security of the Algorithm

The proposed certificate-less public key cryptography scheme is based on the original ECC scheme, but its key pair relationship is different from the original one. The key security has been shown above. Hence, the algorithm's security depends on ECC.

Since the encryption algorithm is based on ECIES and implemented by a symmetric cryptographic algorithm; the security of the encryption algorithm is determined by the symmetric cryptographic algorithm and the encryption key. We recommend AES as the symmetric cryptographic algorithm, which is secure at present.

In addition, the shared secret (used to derive the the encryption key and the integrity authentication key) is generated by ECDH, whose security was proved. Both the encryption key and the integrity authentication key are derived from the shared secret by a secure hash function. The sizes of the keys are specified by the system secure parameter l and can meet the secure requirements.

The signature and verification process are based on ECDSA. Since ECDSA is secure, our sign and verify algorithms are also secure.

B. Efficiency Analysis

Since energy consumption is proportion to the overheads, we analyze communication overhead and computational overhead of the proposed scheme overhead in this subsection.

1) Communication Overhead

Communication overhead mainly includes the energy consumption to transmit the system parameters, user's public key, encrypted message, signature, and so on, which is often represented by the length of the message to be transmitted.

In our scheme, the system parameters *params* are $\{F\_q, E/F\_q, G, P_{\text{pub}}, H\}$, which can be denoted by $E(q, a, b, G, n, h)$, P_{pub} and H. Elliptic curve E needs at most $7l$ bits to represent the ECC's parameters, and $2l$ bits for the system public key P_{pub}. H can be the identifier of a secure hash function and the length of it is at most l bits. The total length of the system parameters is $10l$ bits. Since the system parameters for different schemes vary greatly, it is difficult to compute the length of them. In this paper, the length of the system parameters is not included in the total communication overhead.

The user's public key is a point on E, which needs $2l$ bits to describe.

The encrypted message consists of three parts. The first part is a point on E, which needs $2l$ binary bits to represent. The second part is the cipher-text of the data, and its length depends on the length of the message. The third part is the MAC value of the mesage, whose length is l bits. Hence, the overhead of the encrypted message is $(3l + |m|)$ bits, where $|m|$ is the length of message m.

A signature consists of two parts, both of which are integers in the interval $[1, n\text{-}1]$. Each of them needs at most l bits. Hence, a signature needs at most $2l$ bits.

The total communication overhead is $2l+3l+|m|+2l=(7l+|m|)$ bits, which is much smaller than that of the existing similar certificate-less public key cryptographic schemes. A detailed comparison of the communication overhead is given in Table II.

2) *Computational Overhead*

Since the proposed certificate-less public key cryptographic scheme is constructed by integrating the user's *ID* into ECC instead of identity based encryption, it has features of ID-PKC but gets rid of bilinear pairings. The operations in it include scalar multiplication, hash, HMAC, symmetric-cryptography-based encryption and other arithmetic operations. Since the overheads for hash, HMAC, symmetric cryptography algorithm and other arithmetic operations are much lower than that of scalar multiplication, we mainly include the number of scalar multiplications in computational overhead.

According to section IV, we can see that there are 2 scalar multiplications in the encryption algorithm, 1 scalar multiplication in the decryption algorithm, 1 scalar multiplication in the signature algorithm and 3 scalar multiplications in the verification algorithm. A comparison of the computational overhead between our scheme and existing schemes is given in Table II, which shows that our scheme has much less computational overhead.

3) *Comparison Analysis in Efficiency*

To evaluate the efficiency of the proposed scheme objectively, the communication overhead and computational overhead of our scheme and some popular or efficient certificate-less public key cryptography schemes are illustrated in Table II. For fair comparison, assumed that all the schemes are at the similar security strength (here, we use ECC with 160 bits key length as the reference security strength, that is l=160). In addition, the computation overhead is measured by the number of scalar multiplications. The computational overhead is calculated based on the approaches in [10], where one bilinear pairing is about 20 scalar multiplications, and one modular exponential operation is about 2 scalar multiplications. And the communication overhead is measured by binary bits.

Table II indicates that the proposed scheme provides both encrypt and sign algorithms and has much smaller communication and computational overheads than others. Novertheless, the security strength may be less than the schemes based on BDHP, GBDHP or other complicated assumptions.

TABLE II COMPARISON OF OVERHEADS

		Computational overhead				**Communication Overhead**			**Assumption**
		Encrypt	***Decrypt***	***Sign***	***Verify***	***Public Key Length***	***Cipher-text Length***	***Signature Length***	
[1]	CL-PKE1	22	22			$2l$	$3l+\|m\|$[a]		GBDHP[b]
	CL-PKE2	7	44			$2l$	$5l+\|m\|$		GBDHP
[5]	Basic CL-PKE	61	20			$4l$	$2l+\|m\|$		GBDHP
	Full CL-PKE	61	21			$4l$	$3l+\|m\|$		GBDHP
	CL-PKS			23	80	$4l$		$3l$	GBDHP
[6]		22	22			$2l$	$3l+\|m\|$		CDHP and BDHP
[7]				1	21	$2l$		$2l$	Inv-CDHP[c]
[8]				2	42	$2l$		$4l$	CDHP[d]
[14]				1	7	$2l$		$2l$	ECDLP[e]
[15]				4	6	$12.8l$		$25.6l$	DLP[f]
[17]				1	3	$4l$		$3l$	ECDLP
Our scheme		2	1	1	3	$2l$	$3l+\|m\|$	$2l$	ECDH[g] and ECDLP

a. $|m|$ is the length of message m

b. GBDHP is the abbreviation of general bilinear Diffie-Hellman problem

c. Inv-CDHP is the abbreviation of Inverse computational Diffie-Hellman problem

d. CDHP is the abbreviation of computational Diffie-Hellman problem

e. DLP is the abbreviation of discrete logarithm problem

f. ECDLP is the abbreviation of elliptic curve decisional discrete logarithm problem

g. ECDH is the abbreviation of elliptic curve Diffie-Hellma

VI. Conclusion

In order to deploy public-key-based security mechanisms in energy-constrained mobile devices, we presented a lightweight certificate-less public key cryptographic scheme, which uses the key management method in certificate-less public key encryption and signature schemes and can avoid the key escrow problem. In addition, the proposed scheme is very efficient owing to using ECC and simple hash operations instead of bilinear pairing on elliptic curve. The detailed analyses on security and overheads show that our scheme is secure and lightweight, which make it suitable for resources-constrained mobile devices with high security requirements.

Acknowledgment

This work was supported in part by Chinese National Scholarship Fund and Chinese National High Technology Research and Development Program 863 under Grant No. 2012AA121604, as well as the US National Science Foundation (NSF) under grant CNS-1065444.

References

[1] Z. Cheng, L. Chen, L. Ling, and R. Comley, "General and efficient certificate-less public key encryption constructions", in Pairing'07 Proceedings of the First international conference on Pairing-Based Cryptography, Lecture Notes in Computer Science, vol. 4575, 2007, pp. 83-107.

[2] H. A. Housani, J. Baek and C. Y. Yeun, "Survey on certificate-less public key cryptography", in proceedings of 6th International Conference on Internet Technology and Secured Transactions, 11-14 December 2011, Abu Dhabi, United Arab Emirates. pp. 53-58.

[3] Elliptic curve-based certificate-less signatures for identity-based Encryption(ECCSI). RFC 6507. http://www.rfc-editor.org/info/rfc6507.

[4] A. W. Dent, "A brief Introduction to certificate-less encryption scheme and their infrastructures", 6th European Workshop, EuroPKI 2009, Pisa, Italy, September 10-11, 2009, Revised Selected Papers. Public Key Infrastructures, Services and Applications, Lecture Notes in Computer Science, vol. 6391, 2010, pp. 1-16.

[5] S. S. Al-Riyami, K. G. Paterson, "Certificate-less public key cryptography". In Laih CS, ed. Proc. of the ASIACRYPT 2003. LNCS, 2003, vol. 2894, pp. 452−473.

[6] Y. R. Lee, H. S. Lee, "An authenticated certificate-less public key encryption scheme", Trends in Mathematics Information Center for Mathematical Sciences vol. 8, no. 1, June, 2005, pp. 177-187.

[7] H. Du, Q. Wen, "Efficient and provably-secure certificate-less short signature scheme from bilinear pairings". Computer Standards and Interfaces, 2009, vol.31, no.2, pp. 390-394.

[8] L. Zhang, F. Zhang, "A new provably secure certificate-less signature scheme", in Proceeding of 2008 IEEE International Conference on Communications (ICC 2008). Beijing, China, 2008. pp. 1685-1689.

[9] K. Y. Choi, J. H. Park, J. Y. Hwang, "Efficient certificate-less signature schemes", in Proceedings of 5th International Conference Applied Cryptography and Network Security (ACNS 2007), Zhuhai, China, June 5-8, 2007, pp. 443-458.

[10] X. Cao, W. Kou, X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges". Information Sciences, 2010, vol. 180, no. 15, pp.2895-2903.

[11] J. Baek, R. Safavi-Naini, and W. Susilo, "Certificate-less public key encryption without pairing", in Proceedings of 8th International Conference on Information Security(ISC 2005), Singapore, September 20-23, 2005, Lecture Notes in Computer Science, vol. 3650, 2005, pp. 134-148.

[12] D. Yum, P. Lee, "Generic construction of certificate-less signature". in Proceedings of 9th Australasian Conference on Information Security and Privacy(ACISP 2004), Sydney, Australia, July 13-15, 2004. pp. 200-211.

[13] B. Hu, D. Wong, Z. Zhang, "Key replacement attack against a generic construction of certificate-less signature", in Proceedings of 11th Australasian Conference on Information Security and Privacy(ACISP 2006), Melbourne, Australia, 2006. pp. 235-246.

[14] Y. Wang, R. Zhang, "Strongly secure certificate-less signature scheme without pairings", Chinese Journal on Communications, vol.34, no.2, Feb 2013, pp 94-99,108.

[15] L. Harn, J. Ren, C. Lin. "Design of DL-based certificate-less digital signatures". Journal of Systems and Software, 2009, 82(5):789-793.

[16] A. Ge, Shaozhen Chen, "Strongly secure certificate-less signature scheme without pairings". Chinese Journal of Electronics & Information Technology, 2010, vol. 32, no.7, pp.1765-1768.

[17] D. He, J. Chen, R. Zhang. "An efficient and provably-secure certificate-less signature scheme without bilinear pairings". International Journal of Communication Systems, 2012, vol. 25, no. 11, pp.1432-1442.

[18] J. Lai, R. H. Deng, S. Liu, and W. Kou, "RSA-based certificate-less public key encryption", in Proceedings of 5th International Conference on Information Security Practice and Experience(ISPEC 2009), Xi'an, China, April 13-15, 2009. Lecture Notes in Computer Science, vol. 5451, 2009, pp. 24–34.

[19] C. P. Schnorr, "Efficient identification and signatures for smart cards", Advances in cryptology — CRYPTO' 89 Proceedings. Lecture Notes in Computer Science. Vol.435, 1990, pp. 239-252.

[20] H. Dong, B. Sheng, Q. Li, "Elliptic curve cryptography-based access control in sensor networks", Int. Journal of Security and Networks, vol.1, no. 3/4, 2006, pp.127-137.

[21] D. Johnson, A. Menezes, S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)", International Journal of Information Security, August 2001, vol 1, no. 1, pp. 36-63.

[22] D. McGrew, K. Igoe, M. Salter, "Fundamental elliptic curve cryptography algorithms", February 2011. Internet Engineering Task Force (IETF), Request for Comments: 6090.http://tools.ietf.org/html/draft-mcgrew-fundamental-ecc-04.

[23] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow,"Elliptic curve cryptography in practice". http://eprint.iacr.org/2013/734.pdf.

[24] V. G. Martínez, L. H. Encinas, and C. S. Ávila, "A survey of the elliptic curve integrated encryption scheme", Journal of Computer Science and Engineering, vol. 2, no. 2, AUGUST 2010, pp. 7-13.