SCIENCE CHINA
Information Sciences

• Supplementary File •

An efficient pairing-free certificateless signature scheme for resource-limited systems

Liangliang Wang[1,3], Kefei Chen[2,3*], Yu Long[4] & Huige Wang[4]

[1]*College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai* 200090*, China;*
[2]*School of Science, Hangzhou Normal University, Hangzhou* 310036*, China;*
[3]*Science and Technology on Communication Security Laboratory, Chengdu* 610041*, China;*
[4]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China*

Appendix A Introduction

With the development of computation and communication, resource-limited systems have been quite widespread, such as wireless sensor networks (WSNs) [1,2], PDAs, cell phones and smartcards. However, these systems cannot execute complex applications very well due to the limitations of computing resource, storage space and communication bandwidth. On account of openness, security problem has gradually become a bottleneck of the further development and applications of these systems. As one of the most significant security measures, public key cryptography (PKC) [3] is playing an important role in resource-limited systems. In addition, confidentiality, integrity, availability, controllability and non-repudiation of the messages can be achieved when messages are transmitted over a public channel. Therefore, how to construct lightweight public key cryptography schemes appropriate for resource-limited systems is worth studying. For example, Li et al. [4] did similar work once.

In traditional PKC, every user has a key pair, in which a private key and a public key are involved. The private key is kept secret, while the public key is published. When compared with symmetric key cryptography, it is not necessary to take the problem of the private key transmission in PKC into account. However, the authenticity of the public key is a key problem for PKC. In such a system, users' public keys are authenticated by corresponding certificates that are actually signatures on these public keys. These certificates are generated by a trusted-by-all third party called certificate authority (CA) and managed by a significant mechanism called public key infrastructure (PKI). Note that the certificates management requires high computing resource and huge storage resource, which is considered to be a expensive process. In order to solve this problem, the concept of identity-based public key cryptography (ID-PKC) was introduced by Shamir [5], in which users' public keys are usually set by means of unique information concerning users' identities, such as identity card number, social security number and email address. In addition, certificates are no longer required to validate the authenticity of public keys as in traditional PKC. A trusted third party called key generation center (KGC) is responsible for the private keys generation for all users. Supposing that a user wants to obtain a private key, the user's identity is sent to the KGC first and then the KGC employs the master secret key to generate a corresponding private key for the user. Unfortunately, this type of cryptography also has a shortcoming, that is the key escrow problem. In other words, all users' private keys are generated by the KGC, therefore it can control all users' private keys, so as to decrypt ciphertexts sent to them and forge valid signatures authenticated by their public keys on the condition that the KGC becomes malicious.

In order to avoid the shortcoming above, the concept of certificateless public key cryptography (CL-PKC) was introduced by Al-Riyami and Paterson [6], in which a user's private key consists of a partial private key that is generated by a semi-trusted third party KGC and a secret value that is held by the user. In this case, the KGC cannot obtain user's full private key and the key escrow problem in ID-PKC is eliminated in CL-PKC. Unlike ID-PKC, a user's public key in CL-PKC requires to be computed from the user's private key. Therefore, this kind of cryptography is not actually an ID-based cryptography. However, the public key does not need to be authenticated. Due to the features of no public key certificates and semi-trusted KGC of the system, there are two types of adversaries defined in [6] for CL-PKC: Type I adversary who serves as an external third party with ability of replace users' public keys and Type II adversary serves as a malicious KGC with ability of possess the master secret key of the KGC.

* Corresponding author (email: kfchen@hznu.edu.cn)

Appendix A.1 Related work

Pairing-based CL-PKS schemes. This type of CL-PKS schemes are mainly designed with heavy computation of bilinear pairings. In Asiacrypt 2003, the concept of CL-PKC was first introduced by Al-Riyami and Paterson [6] and the first CL-PKS scheme was proposed. Thereafter, an attack was proposed by Huang et al. [7] to indicate that the CL-PKS scheme proposed in [6] is insecure against Type I adversary. What's more, they fixed this problem by means of proposing a new scheme accompanying with a formal security proof. In the same year, a CL-PKS scheme was proposed by Li et al. [8] without providing a formal security proof. Moreover, another efficient CL-PKS scheme was proposed by Gorantla and Saxena [9], which was also found to be insecure against Type I adversary by Cao et al. [10]. In 2006, Zhang et al. [11] proposed an efficient CL-PKS scheme that was based on bilinear pairings and demonstrated its security in the ROM. Yap et al. [12] proposed an efficient pairing-based CL-PKS scheme, which was respectively demonstrated to be insecure against Type I adversary by Park [13] and Zhang and Mao [14]. In 2007, Huang et al. [15] visited CL-PKS again and potential adversaries were divided into three kinds: Normal adversary, Strong adversary and Super adversary. Besides, they visited security models of CL-PKS again and proposed two specific schemes with different security levels. One scheme is proved to be secure against Normal Type I and Super Type II adversaries. The other one is proved to be secure against Super Type I and Type II adversaries. In 2004, Yum and Lee [16] proposed a generic construction of CL-PKS. However, Hu et al. [17] demonstrated that the generic construction is insecure against the key replacement attack. And then, Hu et al. improved the scheme and demonstrated its security in a simplified security model. In 2009, Du and Wen [18] proposed an efficient pairing-based certificateless short signature scheme that was proved to be secure in the ROM. In 2012, an efficient strongly secure short CL-PKS scheme was proposed by Tso et al. [19] and a security proof was given under the security model defined in [15]. Besides pairing-based CL-PKS, pairing-based certificateless hybrid signcryption was introduced by Li et al. [20] and two pairing-based certificateless authentication protocols are proposed by Xiong et al. [21, 22]. According to these observations, it can be found that the majority of pairing-based CL-PKS schemes are insecure against Type I adversary and they are not appropriate for resource-limited systems very well because of heavy computation of bilinear pairings. Therefore, it is very attractive for researchers to design provably-secure pairing-free CL-PKS schemes.

ECC-based pairing-free CL-PKS schemes. This type of CL-PKS schemes are mainly designed with point scalar multiplication operations. In 2012, an efficient and provably-secure pairing-free CL-PKS scheme was proposed by He et al. [23]. Afterwards, Tian and Huang [24] and Tsai et al. [25] indicated that this scheme was insecure against Type II adversary, in which Type II adversary in [25] is a strong one who can not only possess the master secret key but also can replace the master public key of the KGC. In addition, Tsai et al. gave a modified scheme as well, however a formal security proof for their scheme was not provided. Gong and Li [26] pointed out that schemes proposed by He et al. [23] and Tsai et al. [25] merely demonstrated to be secure against Normal adversary [15]. Moreover, they proposed a real CL-PKS scheme and demonstrated that their scheme is secure against Super adversary [15]. However, Yeh et al. [27] demonstrated that the scheme proposed in [26] cannot satisfy the security requirements described in [26]. In their attack, Type I adversary's capabilities were enlarged to allow it to replace the KGC's public key. Besides, an efficient CL-PKS scheme using ECC was proposed by Islam et al. [28] and an efficient pairing-free certificateless designated verifier signature scheme was proposed by He et al. [29]. In 2014, an efficient pairing-free CL-PKS scheme was respectively proposed by Yeh et al. [30] and Liu et al. [31]. Based on further observations, it can be found that the majority of these ECC-based pairing-free schemes cannot achieve expected security levels.

General pairing-free CL-PKS schemes. This type of CL-PKS schemes are designed with finite field operations, including modular exponentiation operations, modular multiplication operations, modular inverse operations and addition operations. In 2009, a new DLP-based CL-PKS scheme without bilinear pairings was proposed by Harn et al. [34]. Furthermore, three kinds of adversaries were defined informally for CL-PKS and a loose security analysis was given. In 2012, an efficient RSA-based CL-PKS scheme without bilinear pairings was proposed by Zhang and Mao [35]. Unfortunately, He et al. [36] showed that this scheme was insecure against Type I adversary soon. Besides general pairing-free CL-PKS, a general pairing-free certificateless ring signature scheme was proposed by Qin et al. [32]. So far, general pairing-free CL-PKS schemes seem to be rare. It remains to be an open problem to construct a general pairing-free CL-PKS scheme, for which a formal security proof can be given under a formal adversary model.

Appendix B Security proof

Theorem 1. Under the hardness assumption of the DLP, the proposed scheme is able to achieve existentially unforgeable against adaptively chosen message attacks in the ROM.

Lemma 1. If a Type I adversary $\mathcal{A}_I$ is able to output a correct signature in Game I with probability ε after running in time t and issuing q_{par} queries to the partial private key extraction oracle, q_{pub} queries to the public key extraction oracle, q_{pubr} queries to the public key replacement oracle, q_{H_1} queries to the random oracle H_1, q_{H_2} queries to the random oracle H_2, q_{pri} queries to the private key extraction oracle and q_{sig} queries to the signing oracle, then there exists a algorithm $\mathcal{B}$ to solve the DLP with probability

$$\varepsilon' > (\varepsilon - \frac{1}{2^l}) \times (1 - \frac{1}{q_{par}})^{q_{par}} \times (\frac{1}{2^{|p|}})^{q_{pubr}} \times (1 - \frac{1}{q_{par}})^{q_{pri}} \times \frac{1}{q_{par}},$$

where $|p|$ is bit length in Z_p, within time

$$t < t' + (q_{pub} + 3q_{pubr} + 8q_{sig})t_e + (2q_{pub} + 3q_{pubr} + 6q_{sig})t_m,$$

where t_m denotes the the time of executing a modular multiplication operation and t_e denotes the the time of executing a modular exponentiation operation.

Proof. In this proof, our goal is to show there exists a algorithm $\mathcal{B}$ can solve the DLP with the aid of $\mathcal{A}_I$.

At the beginning, $\mathcal{B}$ is given a random challenge tuple (p, g, β) of DLP and it aims to output α such that $g^\alpha = \beta \bmod p$. The algorithm $\mathcal{B}$ initializes $\mathcal{A}_I$ with the system parameters $(p, q, g, P_{pub}, H_1, H_2)$. And then $\mathcal{B}$ acts as a challenger to respond $\mathcal{A}_I$'s oracle queries and simulates the oracles of our scheme as below.

Partial private key extraction queries: When $\mathcal{A}_I$ queries this oracle with an identity ID_i, $\mathcal{B}$ maintains a list L_{par} as (ID_i, PS_{ID_i}) to record queries and answers between $\mathcal{A}_I$ and $\mathcal{B}$. If $\mathcal{B}$ can find (ID_i, PS_{ID_i}) in the list L_{par}, $\mathcal{B}$ returns PS_{ID_i} to $\mathcal{A}_I$. Otherwise, $\mathcal{B}$ randomly selects $c \in [1, q_{par}]$.

1. If $i \neq c$, $\mathcal{B}$ selects random $PS_{ID_i} \in Z_q^*$, returns PS_{ID_i} to $\mathcal{A}_I$ and saves (ID_i, PS_{ID_i}) in the list L_{par}.
2. If $i = c$, $\mathcal{B}$ sets $ID_i = ID^*$ and outputs "failure" and halts.

Public key extraction queries: When $\mathcal{A}_I$ queries this oracle with an identity ID_i, the challenger $\mathcal{B}$ maintains a list L_{pub} as $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, s_{ID_i})$ to record queries and answers between $\mathcal{A}_I$ and $\mathcal{B}$. If the challenger $\mathcal{B}$ can find $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, s_{ID_i})$ in the list L_{pub}, $\mathcal{B}$ returns (PK_{1ID_i}, PK_{2ID_i}) to $\mathcal{A}_I$. Otherwise,

(1) If $ID_i = ID^*$, $\mathcal{B}$ randomly selects $PK_{2ID_i} \in Z_p^*$ and computes $e_i \in Z_q^*$, $\mathcal{B}$ sets $PK_{1ID_i} = PK_{2ID_i}\beta^{-1}{P_{pub}}^{-e_i} \bmod p$, returns (PK_{1ID_i}, PK_{2ID_i}) to $\mathcal{A}_I$ and saves $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, \perp)$ in the list L_{pub}.

(2) Otherwise, $\mathcal{B}$ recovers the corresponding (ID_i, PS_{ID_i}) from L_{par}. If $\mathcal{B}$ cannot find (ID_i, PS_{ID_i}) in the list L_{par}, $\mathcal{B}$ issues a partial private key extraction query on ID_i to get a new PS_{ID_i}. Then $\mathcal{B}$ randomly selects $e_i \in Z_q^*$ and $v_i \in Z_q^*$, computes $PK_{2ID_i} = g^{PS_{ID_i}}{P_{pub}}^{e_i} \bmod p$ and $PK_{1ID_i} = g^{v_i} \bmod p$. Thus $\mathcal{B}$ returns (PK_{1ID_i}, PK_{2ID_i}) to $\mathcal{A}_I$ and saves (ID_i, PS_{ID_i}) and $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, v_i)$ in the lists L_{par} and L_{pub}, respectively.

Public key replacement queries: When $\mathcal{A}_I$ queries this oracle with a tuple $(ID_i, \widetilde{PK_{1ID_i}}, \widetilde{PK_{2ID_i}})$, $\mathcal{B}$ recovers the corresponding (ID_i, PS_{ID_i}) from L_{par}. If $\mathcal{B}$ cannot find (ID_i, PS_{ID_i}) in the list L_{par}, $\mathcal{B}$ issues a partial private key extraction query on ID_i to get a new PS_{ID}. Then $\mathcal{B}$ checks if the equation $g^{PS_{ID}}{P_{pub}}^{H_1(ID, \widetilde{PK_{1ID_i}}, \widetilde{PK_{2ID_i}})} = \widetilde{PK_{2ID_i}} \bmod p$ holds. If the equation holds, $\mathcal{B}$ outputs "failure" and halts. Otherwise, $\mathcal{B}$ continues to perform the following two cases,

(1) If $\mathcal{B}$ can find $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, s_{ID_i})$ in the list L_{pub}, $\mathcal{B}$ sets $PK_{1ID_i} = \widetilde{PK_{1ID_i}}$ and $PK_{2ID_i} = \widetilde{PK_{2ID_i}}$, then $\mathcal{B}$ saves $(ID_i, \widetilde{PK_{1ID_i}}, \widetilde{PK_{2ID_i}}, \perp, \perp)$ in the list L_{pub}.

(2) Otherwise, $\mathcal{B}$ issues a public key extraction query on ID_i to obtain a new PK_{1ID_i} and a new PK_{2ID_i}. Then $\mathcal{B}$ sets $PK_{1ID_i} = \widetilde{PK_{1ID_i}}$, $PK_{2ID_i} = \widetilde{PK_{2ID_i}}$ and saves $(ID_i, \widetilde{PK_{1ID_i}}, \widetilde{PK_{2ID_i}}, \perp, \perp)$ in the list L_{pub}.

H_1 queries: When $\mathcal{A}_I$ queries this oracle with an input $(ID_i, R_{ID_i}, PP_{ID_i})$, here we always assume that $\mathcal{A}_I$ has made a public key extraction on ID_i to obtain R_{ID_i} and PP_{ID_i} which are actually PK_{1ID_i} and PK_{2ID_i}. Thereby, $\mathcal{B}$ recovers the corresponding $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, s_{ID_i})$ form the list L_{pub} and returns e_i to $\mathcal{A}_I$.

H_2 queries: When $\mathcal{A}_I$ queries this oracle with an input (ID_i, m_i, u_i), $\mathcal{B}$ maintains a list L_2 of the form (ID_i, m_i, u_i, h_i) to record queries and answers between $\mathcal{A}_I$ and $\mathcal{B}$. If $\mathcal{B}$ can find (ID_i, m_i, u_i, h_i) in the list L_2, $\mathcal{B}$ returns h_i to $\mathcal{A}_I$. Otherwise, $\mathcal{B}$ randomly selects $h_i \in Z_q^*$, returns h_i to $\mathcal{A}_I$ and saves (ID_i, m_i, u_i, h_i) in the list L_2.

Private key extraction queries: When $\mathcal{A}_I$ queries this oracle with an identity ID_i,

(1) If $ID_i = ID^*$, $\mathcal{B}$ outputs "failure" and halts.

(2) Otherwise, $\mathcal{B}$ recovers the corresponding (ID_i, PS_{ID_i}) and $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, s_{ID_i})$ from the lists L_{par} and L_{pub}. If $\mathcal{B}$ cannot find (ID_i, PS_{ID_i}) and $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, s_{ID_i})$ in the corresponding lists L_{par} and L_{pub}, $\mathcal{B}$ issues a partial private key extraction query and a public key extraction query on ID_i to get a new PS_{ID_i} and a new s_{ID_i}, then $\mathcal{B}$ returns $s_{ID_i} - PS_{ID_i}$ to $\mathcal{A}_I$ and saves (ID_i, PS_{ID_i}) and $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, s_{ID_i})$ in the lists L_{par} and L_{pub}, respectively.

Signing queries: When $\mathcal{A}_I$ queries this oracle with (ID_i, m_i), here we always assume that ID_i has been queried before.

(1) If $ID_i \neq ID^*$, $\mathcal{B}$ outputs a signature σ_i on message m_i by the private key returned to $\mathcal{A}_I$, then $\mathcal{B}$ returns σ_i to $\mathcal{A}_I$.

(2) Otherwise, $\mathcal{B}$ selects random $s_i, h_i \in Z_q^*$, recovers the corresponding $(ID_i, PK_{1ID_i}, PK_{2ID_i}, e_i, s_{ID_i})$ from the list L_{pub}. Then $\mathcal{B}$ compute $u_i = g^{s_i}{PK_{1ID_i}}^{h_i}{P_{pub}}^{e_i h_i}{PK_{2ID_i}}^{-h_i} \bmod p$. Thus the tuple (u_i, h_i, s_i) comprises a correct signature σ_i on m_i for the identity ID_i, $\mathcal{B}$ returns σ_i to $\mathcal{A}_I$ and saves (ID_i, m_i, u_i, h_i) in the list L_2. Note that $\mathcal{B}$ outputs "failure" and halts if the tuple (ID_i, m_i, u_i) has already been defined in L_2 and (PK_{1ID_i}, PK_{2ID_i}) may have been replaced.

Correctness: It is easy to check the correctness of the simulated signature as below.

$$\begin{aligned} g^{s_i}{PK_{1ID_i}}^{h_i}{P_{pub}}^{e_i h_i} &= g^{s_i}(PK_{2ID_i}\beta^{-1}{P_{pub}}^{-e_i})^{h_i}{P_{pub}}^{e_i h_i} \\ &= g^{s_i}{PK_{2ID_i}}^{h_i}\beta^{-h_i}{P_{pub}}^{-e_i h_i}{P_{pub}}^{e_i h_i} \\ &= g^{s_i}{PK_{2ID_i}}^{h_i}\beta^{-h_i} \\ &= g^{s_i}{PK_{1ID_i}}^{h_i}{P_{pub}}^{e_i h_i} \end{aligned}$$

$$= g^{s_i} {PK_{1ID_i}}^{h_i} {P_{pub}}^{e_i h_i} {PK_{2ID_i}}^{-h_i} {PK_{2ID_i}}^{h_i}$$
$$= u_i {PK_{2ID_i}}^{h_i} \bmod p$$

Forgery: Lastly, $\mathcal{A}_I$ stops to issue queries and outputs a correct signature $(\widehat{u}, \widehat{h}, \widehat{s})$ with respect to $(\widehat{m}, \widehat{ID})$. If $\widehat{ID} \neq ID^*$, $\mathcal{B}$ outputs "failure" and halts. Otherwise, by replays of $\mathcal{B}$ with the same random tape but different choices of the oracle H_2, according to the forking lemma [33], $\mathcal{B}$ can obtain another correct signature $(\widehat{u}, \widehat{h}', \widehat{s}')$. Both of the two correct signatures should satisfy the verification equation as follows.

$$g^{\widehat{s}} {PK_{1ID^*}}^{\widehat{h}} {P_{pub}}^{e\widehat{h}} = \widehat{u} {PK_{2ID^*}}^{\widehat{h}} \text{ and } g^{\widehat{s}'} {PK_{1ID^*}}^{\widehat{h}'} {P_{pub}}^{e\widehat{h}'} = \widehat{u} {PK_{2ID^*}}^{\widehat{h}'},$$

Thereby, $\mathcal{B}$ can conclude the following relation

$$\begin{aligned} g^{\widehat{s}-\widehat{s}'} &= \beta^{\widehat{h}-\widehat{h}'} \\ &\Updownarrow \\ \log_g \beta &= \frac{\widehat{s}-\widehat{s}'}{\widehat{h}-\widehat{h}'} \end{aligned}$$

obviously, $\mathcal{B}$ can solve the given DLP instance by presenting the solution as $\frac{\widehat{s}-\widehat{s}'}{\widehat{h}-\widehat{h}'}$.

Probability analysis: Now let's analyze the success probability of $\mathcal{B}$ in this game. Since H_2 is considered as a random oracle, $\mathcal{A}_I$ may generate a correct signature with respect to $(\widehat{m}, \widehat{ID})$ without issuing the $H_2(\widehat{ID}, \widehat{m}, \widehat{u})$ query is at most $1/2^l$. In the partial private key extraction simulation, the probability of $\mathcal{B}$ does not halts is $(1 - 1/q_{par})^{q_{par}}$. In the public key replacement simulation, the probability of $\mathcal{B}$ does not halts is $(1/2^{|p|})^{q_{pubr}}$. In the private key extraction simulation, the probability of $\mathcal{B}$ does not halts is $(1 - 1/q_{par})^{q_{pri}}$. In the DLP computation, the probability of $\mathcal{B}$ does not halts is $1/q_{par}$. Therefore, the success probability of $\mathcal{B}$ in this game should be at least

$$(\varepsilon - \frac{1}{2^l}) \times (1 - \frac{1}{q_{par}})^{q_{par}} \times (\frac{1}{2^{|p|}})^{q_{pubr}} \times (1 - \frac{1}{q_{par}})^{q_{pri}} \times \frac{1}{q_{par}},$$

The running time of $\mathcal{B}$ should be at most

$$t + (q_{pub} + 3q_{pubr} + 8q_{sig})t_e + (2q_{pub} + 3q_{pubr} + 6q_{sig})t_m.$$

Lemma 2. If a Type II adversary $\mathcal{A}_{II}$ is able to output a correct signature in Game II with probability ε after running in time t and issuing q_{pub} queries to the public key extraction oracle, q_{par} queries to the partial private key extraction oracle, q_{H_1} queries to the random oracle H_1, q_{H_2} queries to the random oracle H_2, q_{pri} queries to the private key extraction oracle and q_{sig} queries to the signing oracle, then there exists a algorithm $\mathcal{B}$ to solve the DLP with probability

$$\varepsilon' > (\varepsilon - \frac{1}{2^l}) \times (1 - \frac{1}{q_{pub}})^{q_{pri}} \times \frac{1}{q_{pub}},$$

within time

$$t' < t + (3q_{pub} + 8q_{sig})t_e + (3q_{pub} + 6q_{sig})t_m,$$

where t_m denotes the the time of executing a modular multiplication operation and t_e denotes the the time of executing a modular exponentiation operation.

Proof. The idea of this proof is similar to the proof of lemma 1, in this proof, our goal is to show there exists a algorithm $\mathcal{B}$ can solve the DLP with the aid of $\mathcal{A}_{II}$. The details of the proof of lemma 2 are given as follows.

At the beginning, $\mathcal{B}$ is given a random challenge tuple (p, g, β) of DLP and it aims to output α such that $g^\alpha = \beta \bmod p$. The algorithm $\mathcal{B}$ initializes $\mathcal{A}_{II}$ with the system parameters $(p, q, g, P_{pub}, H_1, H_2)$ and the master secret key x. And then $\mathcal{B}$ acts as a challenger to respond $\mathcal{A}_{II}$'s oracle queries and simulates the oracles of our scheme as below.

Public key extraction queries: When $\mathcal{A}_{II}$ queries this oracle with an identity ID_i, the challenger $\mathcal{B}$ maintains a list L_{pub} as $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, s_{ID_i})$ to record queries and answers between $\mathcal{A}_{II}$ and $\mathcal{B}$. If $\mathcal{B}$ can find $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, s_{ID_i})$ in the list L_{pub}, $\mathcal{B}$ returns (PK_{1ID_i}, PK_{2ID_i}) to $\mathcal{A}_{II}$. Otherwise, $\mathcal{B}$ randomly selects $c \in [1, q_{pub}]$.

(1) If $i \neq c$, $\mathcal{B}$ first selects random $PS_{ID_i} \in Z_q^*$, $e_i \in Z_q^*$ and $v_i \in Z_q^*$, then $\mathcal{B}$ computes $PK_{2ID_i} = g^{PS_{ID_i}} {P_{pub}}^{e_i} \bmod p$ and $PK_{1ID_i} = g^{v_i} \bmod p$. $\mathcal{B}$ returns (PK_{1ID_i}, PK_{2ID_i}) to $\mathcal{A}_{II}$ and saves $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, v_i)$ in the list L_{pub}.

(2) If $i = c$, $\mathcal{B}$ sets $ID_i = ID^*$, and then $\mathcal{B}$ selects random $PS_{ID_i} \in Z_q^*$, $e_i \in Z_q^*$ and $v_i \in Z_q^*$, continues to compute $PK_{2ID_i} = g^{PS_{ID_i}} {P_{pub}}^{e_i} \bmod p$ and $PK_{1ID_i} = PK_{2ID_i} \beta^{-1} {P_{pub}}^{-e_i} \bmod p$, returns (PK_{1ID_i}, PK_{2ID_i}) to $\mathcal{A}_{II}$ and saves $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, v_i)$ in the list L_{pub}.

Remark: Note that the public key extraction oracle provides to $\mathcal{A}_{II}$ not only the public key extraction service, but also the partial private key extraction service. Although $\mathcal{A}_{II}$ possesses the master secret key x, but $\mathcal{A}_{II}$ cannot obtain the partial private key by itself. Because $\mathcal{A}_{II}$ cannot obtain the value of t_i from the equation $PK_{2ID_i} = g^{t_i} \bmod p$ or perform the replacement of the public key PK_{2ID_i} by selecting a new t_i. Therefore, $\mathcal{B}$ should also potentially provide the partial private

key extraction service to $\mathcal{A}_{II}$ in this game, while in some security proofs of CL-PKS schemes, it is not necessary for $\mathcal{B}$ to provide the partial private key extraction service to $\mathcal{A}_{II}$ because of the feasibility of deriving PS_{ID_i} by $\mathcal{A}_{II}$ itself.

H_1 queries: When $\mathcal{A}_{II}$ queries this oracle with an input $(ID_i, R_{ID_i}, PP_{ID_i})$, similarly we assume that $\mathcal{A}_{II}$ has made a public key extraction on ID_i to obtain R_{ID_i} and PP_{ID_i} which are actually PK_{1ID_i} and PK_{2ID_i}. Thereby, $\mathcal{B}$ recovers the corresponding $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, s_{ID_i})$ from the list L_{pub} and returns e_i to $\mathcal{A}_{II}$.

H_2 queries: When $\mathcal{A}_{II}$ queries this oracle with an input (ID_i, m_i, u_i), $\mathcal{B}$ maintains a list L_2 as (ID_i, m_i, u_i, h_i) to record queries and answers between $\mathcal{A}_{II}$ and $\mathcal{B}$. If $\mathcal{B}$ can find (ID_i, m_i, u_i, h_i) in the list L_2, $\mathcal{B}$ returns h_i to $\mathcal{A}_{II}$. Otherwise, $\mathcal{B}$ selects random $h_i \in Z_q^*$, returns h_i to $\mathcal{A}_{II}$ and saves (ID_i, m_i, u_i, h_i) in the list L_2.

Private key extraction queries: When $\mathcal{A}_{II}$ queries this oracle with an identity ID_i,

(1) If $ID_i = ID^*$, $\mathcal{B}$ outputs "failure" and halts.

(2) Otherwise, $\mathcal{B}$ recovers the corresponding $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, s_{ID_i})$ from the list L_{pub}. If $\mathcal{B}$ cannot find $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, s_{ID_i})$ in the list L_{pub}, $\mathcal{B}$ issues a public key extraction query on ID_i to get a new PS_{ID_i} and a new s_{ID_i}, then $\mathcal{B}$ returns $s_{ID_i} - PS_{ID_i}$ to $\mathcal{A}_{II}$ and saves $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, s_{ID_i})$ in the list L_{pub}.

Signing queries: When $\mathcal{A}_{II}$ queries this oracle with (ID_i, m_i), here we always assume that ID_i has been queried before.

(1) If $ID_i \neq ID^*$, $\mathcal{B}$ outputs a signature σ_i on message m_i by the private key returned to $\mathcal{A}_{II}$, then $\mathcal{B}$ returns σ_i to $\mathcal{A}_{II}$.

(2) Otherwise, $\mathcal{B}$ selects random $s_i, h_i \in Z_q^*$, recovers $(ID_i, PK_{1ID_i}, PK_{2ID_i}, PS_{ID_i}, e_i, s_{ID_i})$ from the list L_{pub}. Then $\mathcal{B}$ compute $u_i = g^{s_i} {PK_{1ID_i}}^{h_i} {P_{pub}}^{e_i h_i} {PK_{2ID_i}}^{-h_i} \bmod p$. Thus the tuple (u_i, h_i, s_i) comprises a correct signature σ_i with respect to (m_i, ID_i), $\mathcal{B}$ returns σ_i to $\mathcal{A}_{II}$ and saves (ID_i, m_i, u_i, h_i) in the list L_2. Note that $\mathcal{B}$ outputs "failure" and halts if the tuple (ID_i, m_i, u_i) has already been defined in L_2. It is easy to check the correctness of the simulated signature as Lemma 1.

Forgery and Probability analysis: After all the queries, $\mathcal{B}$ can use the same technique as Lemma 1 to solve the given DLP instance by presenting the solution as $\frac{\hat{s}-\hat{s}'}{\hat{h}-\hat{h}'}$. We can also use the same analysis method as Lemma 1 to conclude the success probability of $\mathcal{B}$ in this game is at least

$$(\varepsilon - \frac{1}{2^l}) \times (1 - \frac{1}{q_{pub}})^{q_{pri}} \times \frac{1}{q_{pub}},$$

The running time of $\mathcal{B}$ is at most

$$t + (3q_{pub} + 8q_{sig})t_e + (3q_{pub} + 6q_{sig})t_m.$$

Appendix C Comparison with previous schemes

The efficiency and security levels of our general pairing-free CL-PKS scheme are analyzed in this section. In order to give fair comparison, we choose two known general pairing-free CL-PKS schemes [34,35] to compare with our scheme. Table C1 shows the comparison of different general pairing-free CL-PKS schemes in the aspects of computation cost and signature length. In the aspect of computation cost, we omit hash function operations, inverse operations, addition operations and comparison operations which are considered to be trivial, only consider modular exponentiation operations and modular multiplication operations. We let t_m, t_e denote the time of executing a modular multiplication operation and a modular exponentiation operation, respectively. In the aspect of signature length, we define $|x|$ is the bit length in Z_x. l is defined to be the security parameter as usually. $n = pq$ is a RSA modular number. From Table C1, we can see that Harn et al.'s scheme [34] requires a total of four modular multiplication operations and six modular exponentiation operations during the signing and verification processes; Zhang and Mao's scheme [35] requires a total of three modular multiplication operations and seven modular exponentiation operations during the signing and verification processes and our scheme only requires a total of four modular multiplication operations and five modular exponentiation operations during the signing and verification processes. As we known, a modular multiplication operation is far more effective than a modular exponentiation operation. Therefore, our scheme enjoys a lower computation cost. For the signature length, our scheme is shorter than Harn et al.'s and Zhang and Mao's schemes [34,35]. Table C2 shows the comparison of different general pairing-free CL-PKS schemes in the aspect of security levels. From Table C2, we can see that only our scheme can be proved secure against Type I and Type II adversaries at the same time. Harn et al.'s scheme [34] did not consider the security under the known Type I and Type II adversaries' attacks, while Zhang and Mao's scheme [35] is insecure against Type I adversary [36].

Table C1 Efficiency comparisons among different general pairing-free CL-PKS schemes

Scheme	Signing cost	Verification cost	Signature length
Harn et al.'s scheme	t_e+2t_m	5t_e+2t_m	$3\|p\| + \|p-1\|$
Zhang-Mao's scheme	3t_e+t_m	4t_e+2t_m	$\|n\| + 2l$
Our scheme	t_e+t_m	4t_e+3t_m	$\|p\| + 2\|q\|$

Table C2 Security levels comparison among different general pairing-free CL-PKS schemes

Scheme	Secure against Type I	Secure against Type II
Harn et al.'s scheme	No formal proof provided	No formal proof provided
Zhang-Mao's scheme	Insecure against Type I [36]	Secure against Type II
Our scheme	Scecure against Type I	Secure against Type II

References

1 I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer networks 38 (4) (2002) 393–422.
2 J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Computer networks 52 (12) (2008) 2292–2330.
3 W. Diffie, M. E. Hellman, New directions in cryptography, Information Theory, IEEE Transactions on 22 (6) (1976) 644–654.
4 F. Li, M. K. Khan, K. Alghathbar, T. Takagi, Identity-based online/offline signcryption for low power devices, Journal of Network and Computer Applications 35 (1) (2012) 340–347.
5 A. Shamir, Identity-based cryptosystems and signature schemes, in: Advances in cryptology, Springer, 1985, pp. 47–53.
6 S. S. Al-Riyami, K. G. Paterson, Certificateless public key cryptography, in: Advances in Cryptology-ASIACRYPT 2003, Springer, 2003, pp. 452–473.
7 X. Huang, W. Susilo, Y. Mu, F. Zhang, On the security of certificateless signature schemes from asiacrypt 2003, in: Cryptology and Network Security, Springer, 2005, pp. 13–25.
8 X.-x. Li, K.-f. Chen, L. Sun, Certificateless signature and proxy signature schemes from bilinear pairings, Lithuanian Mathematical Journal 45 (1) (2005) 76–83.
9 M. C. Gorantla, A. Saxena, An efficient certificateless signature scheme, in: Computational Intelligence and Security, Springer, 2005, pp. 110–116.
10 X. Cao, K. G. Paterson, W. Kou, An attack on a certificateless signature scheme., IACR Cryptology ePrint Archive 2006 (2006) 367.
11 Z. Zhang, D. S. Wong, J. Xu, D. Feng, Certificateless public-key signature: security model and efficient construction, in: Applied Cryptography and Network Security, Springer, 2006, pp. 293–308.
12 W.-S. Yap, S.-H. Heng, B.-M. Goi, An efficient certificateless signature scheme, in: Emerging Directions in Embedded and Ubiquitous Computing, Springer, 2006, pp. 322–331.
13 J. H. Park, An attack on the certificateless signature scheme from euc workshops 2006., IACR Cryptology ePrint Archive 2006 (2006) 442.
14 J. Zhang, J. Mao, Security analysis of two signature schemes and their improved schemes, in: Computational Science and Its Applications–ICCSA 2007, Springer, 2007, pp. 589–602.
15 X. Huang, Y. Mu, W. Susilo, D. S. Wong, W. Wu, Certificateless signature revisited, in: Information Security and Privacy, Springer, 2007, pp. 308–322.
16 D. H. Yum, P. J. Lee, Generic construction of certificateless signature, in: Information Security and Privacy, Springer, 2004, pp. 200–211.
17 B. C. Hu, D. S. Wong, Z. Zhang, X. Deng, Key replacement attack against a generic construction of certificateless signature, in: Information Security and Privacy, Springer, 2006, pp. 235–246.
18 H. Du, Q. Wen, Efficient and provably-secure certificateless short signature scheme from bilinear pairings, Computer Standards & Interfaces 31 (2) (2009) 390–394.
19 R. Tso, X. Huang, W. Susilo, Strongly secure certificateless short signatures, Journal of Systems and Software 85 (6) (2012) 1409–1417.
20 F. Li, M. Shirase, T. Takagi, Certificateless hybrid signcryption, Mathematical and Computer Modelling 57 (3) (2013) 324–343.
21 H. Xiong, Cost-effective scalable and anonymous certificateless remote authentication protocol, IEEE Transactions on Information Forensics and Security 9 (12) (2014) 2327–2339.
22 H. Xiong, Z. Qin, Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks, IEEE Transactions on Information Forensics and Security 10 (7) (2015) 1442–1455.
23 D. He, J. Chen, R. Zhang, An efficient and provably-secure certificateless signature scheme without bilinear pairings, International Journal of Communication Systems 25 (11) (2012) 1432–1442.
24 M. Tian, L. Huang, Cryptanalysis of a certificateless signature scheme without pairings, International Journal of Communication Systems 26 (11) (2013) 1375–1381.
25 J.-L. Tsai, N.-W. Lo, T.-C. Wu, Weaknesses and improvements of an efficient certificateless signature scheme without using bilinear pairings, International Journal of Communication Systems 27 (7) (2014) 1083–1090.
26 P. Gong, P. Li, Further improvement of a certificateless signature scheme without pairing, International Journal of Communication Systems 27 (10) (2014) 2083–2091.
27 K.-H. Yeh, K.-Y. Tsai, R.-Z. Kuo, T.-C. Wu, Robust certificateless signature scheme without bilinear pairings, in: IT Convergence and Security (ICITCS), 2013 International Conference on, IEEE, 2013, pp. 1–4.
28 S. H. Islam, G. Biswas, Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography, International Journal of Computer Mathematics 90 (11) (2013) 2244–2258.
29 D. He, J. Chen, An efficient certificateless designated verifier signature scheme., Int. Arab J. Inf. Technol. 10 (4) (2013) 389–396.

30 K.-H. Yeh, K.-Y. Tsai, C.-Y. Fan, An efficient certificateless signature scheme without bilinear pairings, Multimedia Tools and Applications (2014) 1–12.
31 W. Liu, Q. Xie, S. Wang, L. Han, B. Hu, Pairing-free certificateless signature with security proof, Journal of Computer Networks and Communications 2014.
32 Z. Qin, H. Xiong, G. Zhu, Z. Chen, Certificate-free ad hoc anonymous authentication, Information Sciences 268 (2014) 447–457.
33 D. Pointcheval, J. Stern, Security proofs for signature schemes, in: Advances in CryptologyEUROCRYPT?6, Springer, 1996, pp. 387–398.
34 L. Harn, J. Ren, C. Lin, Design of dl-based certificateless digital signatures, Journal of Systems and Software 82 (5) (2009) 789–793.
35 J. Zhang, J. Mao, An efficient rsa-based certificateless signature scheme, Journal of Systems and Software 85 (3) (2012) 638–642.
36 D. He, M. K. Khan, S. Wu, On the security of a rsa-based certificateless signature scheme., IJ Network Security 16 (1) (2014) 78–80.