Multimed Tools Appl (2015) 74:6519–6530
DOI 10.1007/s11042-014-2154-4

An efficient certificateless signature scheme without bilinear pairings

Kuo-Hui Yeh · Kuo-Yu Tsai · Chuan-Yen Fan

Received: 21 February 2014 / Revised: 2 June 2014 / Accepted: 9 June 2014 / Published online: 29 June 2014

Abstract During these years, the research field of certificateless signature (CLS) scheme is promptly investigated as the key escrow problem in identity-based cryptography can be solved via CLS concept. However, due to the bandwidth limitation of mobile communication and the resource-constraint property of handheld mobile devices, most CLS schemes cannot fulfill the requirement of computation efficiency for mobile communication architecture. Hence, the design of lightweight CLS protocol refined from traditional cryptosystem technologies for existing mobile communication environment becomes one of the most important research trends. In this paper, we demonstrate a novel CLS scheme which is immune against bilinear pairings. Without the heavy computation of bilinear pairings, our proposed scheme is efficient and practical for mobile communication. Meanwhile, the proposed CLS scheme possesses strong security density owing to the adoption of point addition of elliptic curve cryptography. A formal security analysis is presented to guarantee the security robustness of our CLS protocol under the hardness of breaking elliptic curve discrete logarithm problem.

Keywords Certificateless · Digital signature · Bilinear pairings · Cryptanalysis

1 Introduction

In traditional public key cryptography, signature schemes allow a singer to sign a message with his/her private key to guarantee non-repudiation property (and more). However, each signature activity must accompany with corresponding certificates to complete. To solve the certificate management problem, Shamir [11] introduced the concept of identity-based cryptosystem, where every user does not have an explicit public key as before. The public key is replaced by his/her publicly available identity information, which can uniquely identify him/her and can be

K.-H. Yeh
Department of Information Management, National Dong Hwa University, Hualien 974, Taiwan

K.-Y. Tsai (✉)
Department of Management Information Systems, Hwa Hsia Institute of Technology,
New Taipei City 235, Taiwan
e-mail: KuoYu.Nicklas.Tsai@gmail.com

C.-Y. Fan
CyberTrust Technology Institute, CTTI, Institute for Information Industry, New Taipei City 10622, Taiwan

undeniably associated with him/her. The corresponding private key is computed from a one-way trapdoor function of privileged information known only to the system authority, such as key generation center (KGC). Compared to certificate-based cryptosystem, identity-based cryptosystem does not require extra effort and information for users to validate the authenticity of public keys.

Based on the idea of self-certified cryptosystem, Al-Riyami and Paterson [1] proposed an approach in 2003, namely certificateless public key cryptography (CL-PKC). In this approach, KGC generates partial private key, each user then generates his/her private key and public key using user's secret value and partial private key. This concept was to oppose to KGC having access to each user's private key in identity-based approach, and was the absence of digital certificates and the important management overhead. However, CL-PKC approach is insecure against to type I adversary [8]. In 2004, Yum and Lee [15] proposed another CLS scheme. Nevertheless, Hu et al. [6] pointed out that Yum and Lee's CLS protocol cannot resist to type I adversary. Later, Li et al. [10] and Gorantla et al. [3] presented CLS schemes using bilinear pairings, respectively. Unfortunately, these schemes require heavy operation of bilinear pairing on signature verification. As a result, the development of CLS scheme without bilinear pairings is promptly proposed and investigated in recent years.

In 2011, He et al. [5] demonstrated a CLS scheme which does not adopt the technique of bilinear pairings. Without the heavy computation cost from bilinear pairings, the efficiency of He et al.'s CLS scheme is better than previous CLS protocols. Later, a variant of such CLS concept is adopted in the authors' another study involved with authenticated key agreement [4]. In 2012, however, Tian and Huang [12] and Tsai et al. [13] both presented that He et al.'s CLS scheme is vulnerable to a type II adversary who is able to access the master secret key of KGC. Recently, Gong and Li [2] proposed a CLS scheme without bilinear pairings. The authors claimed that their proposed scheme is secure against the super adversary. Nevertheless, the security claim is not solid. Yeh et al.'s [14] demonstrated that Gong and Li's CLS scheme cannot fulfill the claimed security robustness, i.e. resistance to the super adversary. Based on these observations, we can know that most of existing CLS schemes still have room for security enhancement.

In recent years, with the popularity of mobile communication, industries yearn for an efficient and robust signature scheme to support the enormous needs from on-line services for mobile commerce. Nevertheless, due to the bandwidth limitation of mobile communication architecture and the resource-constraint property of handheld mobile devices, most signature schemes cannot fulfill the requirement of computation efficiency for mobile communication environment. Hence, the design of lightweight and robust CLS protocols refined from traditional cryptosystem technologies for mobile communication becomes one of the most important research areas. Hence, in this paper we will introduce a secure and efficient CLS scheme without bilinear pairings to efficiently and effectively satisfy all the needs from existing mobile communication environment.

2 Preliminary

2.1 Elliptic curve

Let the notation E/F_p denote an elliptic curve E over a prime finite field F_p, defined by an equation $y^2=x^3+ax+b$, where $a_i,b_i\in F_p$ are constants such that $\Delta=4a^3+27b^2\neq0$. All points $P_i=(x_i, y_i)$ on E and the infinity point O forms a cyclic group G under the operation of point addition $R=P+Q$ defined according to a chord-and-tangent rule. In particular, we define $t{\cdot}P=P+P+\ldots+P$ (t times) as *scalar multiplication*. Note that P is a generator of G with order n.

2.2 The overview of certificateless signature scheme

According to the study [1], two types of CLS scheme, denoted as CLS and CLS*, exist. A normal CLS scheme consists of seven phases, i.e. *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key*, *Sign* and *Verify*. We briefly review each phase as follows.

- *Setup*: With the security parameter k, KGC generates a master secret key mk, a corresponding master public key P_{pub} and the public parameters *params*.
- *Partial-Private-Key-Extract*: With the master secret key mk, the public parameters *params* and the user i's identity ID_i, KGC generates a partial secret key D_i for the user i.
- *Set-Secret-Value*: The user i randomly selects a value $x_i \in Z_n^*$ as his/her secret.
- *Set-Private-Key*: With the public parameters *params*, the user i's partial private key D_i and his/her chosen secret value x_i, the user i generates a full private key. Note that in some studies, *Set-Private-Key* phase may be integrated with *Set-Secret-Value* phase.
- *Set-Public-Key*: With the public parameters *params* and the user i's secret value x_i,, the user i outputs his/her public key $PK_{i.}$
- *Sign*: With any target message m, this phase outputs a signature $\sigma_i=(R_i,T_i,\tau_i)$ on m.
- *Verify*: With the signature $\sigma_i=(R_i,T_i,\tau_i)$ of the message m, this phase returns 1 if $\sigma_i=(R_i, T_i,\tau_i)$ is valid. Otherwise, it returns 0.

Furthermore, the other kind of certificateless signature scheme CLS* also possesses seven phases: *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key*, *Sign* and *Verify*. The main difference between CLS and CLS* is in the procedure of *Partial-Private-Key-Extract* phase which additionally requires the user i's public key as an input.

3 Adversaries against certificateless signature scheme

In general, there exist two categories of adversaries against certificateless signature scheme, i.e. type I and type II Adversaries [1]. The type I adversary models an outside adversary who does not know the master secret key of KGC; however, the type I adversary is able to replace any entity's public key with specific values chosen by the adversary itself. The type II adversary models a malicious KGC who is allowed to access to the master secret key of KGC. Nevertheless, the type II adversary cannot replace the public keys of other entities. In addition, based on the security model defined by Huang et al. [7], type I and II adversaries against CLS schemes can further be classified into three categories: normal, strong and super levels. A normal-level type I (and II) adversary only has the ability to learn valid signatures. A strong-level type I (and II) adversary is able to replace a public key to forge a valid signature when the adversary possesses a corresponding secret value. A super-level type I (and II) adversary is able to learn valid signatures for a replaced public key without any submission. Here, we only present the definition of the super-level type I adversary j which will mainly be involved with the cryptanalysis of Gong-Li's CLS scheme [2]. The game is performed between a challenger C and a super-level type I adversary j for a CLS scheme as follows.

Initialization C runs the *Setup* algorithm and generates a master secret key mk, public system parameters *params*. Next, C keeps mk and gives *params* to the adversary j.

Queries The adversary j can adaptively issue the following oracle queries [2, 5], i.e. *ExtractPartialPrivateKey*(i), *ExtractSecretValue*(i), *RequestPublicKey*(i), *ReplacePublicKey*(i), and *Sign*(i, m), to C.

Output Eventually, the adversary j outputs (ID_t, m_t, σ_t). The adversary j wins the game if

(1) *ExtractPartialPrivateKey* (t) and *Sign*(t, m_t) queries have never been queried.
(2) $1 \leftarrow Verify(params, m_t, PK_t, P_{pub}, \sigma_t)$. Note that PK_t and P_{pub} may be replaced by the adversary j.

Definition 1 A CLS scheme is existentially unforgeable against a super-level type I adversary, if for any polynomially bounded super-level Type I adversary j, $Succ_j$ is negligible, where $Succ_j$ is the success probability that j wins in the above game.

4 The proposed CLS scheme

In this section, we propose a new CLS scheme which is extended from Gong and Li's protocol [2]. In our proposed CLS scheme, we mainly strengthen the connection among values h_i, k_i and l_i with public values such as T_i, PK_i, R_i and P_{pub}. This design makes the adversary hard to eliminate the connection among the values T_i, $l_ik_iPK_i$, R_i and h_iP_{pub} in the signature $\sigma_i=(R_i,T_i,\tau_i)$ at each session. In general, our CLS scheme consists of six algorithms, i.e. *Setup*, *PartialPrivateKeyExtract*, *SetSecretValue*, *SetPublicKey*, *Sign* and *Verify.*

Setup Given k, KGC generates a group G of elliptic curve points with prime order n and determines a generator P of G. Then, KGC chooses the master key $mk=s\in Z_n^*$, and three secure hash functions $H_1:\{0,\ 1\}^*\times G\times G\rightarrow Z_q^*$, $H_2:\{0,\ 1\}^*\times G\times G\times G\times G\rightarrow Z_q^*$ and $H_3:\{0,\ 1\}^*\times\{0,\ 1\}^*\times G\times G\times G\times G\rightarrow Z_q^*$. Next, KGC creates the master public key $P_{pub}=s\cdot P$. Finally, KGC publishes $params=\{G,P,P_{pub},H_1,H_2,H_3\}$, and keeps mk secretly.

PartialPrivateKeyExtract Given $params$, mk, and user i 's identity ID_i, KGC generates a random number $r_i\in Z_n^*$, and calculates $R_i=r_i\cdot P$, $h_i=H_1(ID_i,R_i,P_{pub})$, and $s_i=r_i+h_is \bmod n$. Next, KGC returns the partial private key $D_i=(s_i,R_i)$ to the user. The validity of D_i is based on if $s_i\cdot P=R_i+h_i\cdot P_{pub}$ holds.

SetSecretValue Given $params$, the user i with identity ID_i picks a random number $x_i\in Z_n^*$ as his/her secret value.

SetPublicKey Given $params$ and x_i, the user i computes $PK_i=x_i\cdot P$ as his/her public key.

Sign Given $params$, D_i, x_i and a message m, the user i computes
$T_i=t_i\cdot P$ with a random number $t_i\in Z_n^*$ and

$$k_i = H_2\left(T_i, ID_i, PK_i, R_i, P_{pub}\right),$$
$$l_i = H_3\left(m, T_i, ID_i, PK_i, R_i, P_{pub}\right)$$

And

$$\tau_i = t_i + l_i(k_ix_i + s_i) \bmod n.$$

Now, $\sigma_i=(R_i,T_i,\tau_i)$ is the signature of the message m.

Verify Given *params*, ID_i, PK_i, m and σ_i=(R_i,T_i,τ_i), the verifier examines the validity of σ_i.

Compute

$$h_i = H_1(ID_i, R_i, P_{pub}) ,$$
$$k_i = H_2(T_i, ID_i, PK_i, R_i, P_{pub}),$$

and

$$l_i = H_3(m, T_i, ID_i, PK_i, R_i, P_{pub}).$$

Verify whether the equation $\tau_i{\cdot}P$=T_i+l_i($k_i{\cdot}PK_i$+R_i+$h_i{\cdot}P_{pub}$) holds.

Correctness:

$$\begin{aligned}\tau_i{\cdot}\ P &= [t_i + l_i(k_i x_i + s_i)] \cdot P \\ &= t_i{\cdot}P + l_i(k_i x_i{\cdot}P + r_i{\cdot}P + h_i s{\cdot}P) \\ &= T_i + l_i(k_i{\cdot}PK_i + R_i + h_i{\cdot}P_{pub})\end{aligned}$$

5 Security analysis

Based on the hardness of solving the elliptic curve discrete logarithm problem (ECDLP for short), we prove that our proposed certificateless signature scheme without bilinear pairings is existentially unforgeabe against a super-level type I adversary and the super-level type I adversary defined in Section 2.

Definition 2. Elliptic Curve Discrete Logarithm Problem (ECDLP) Given a group G of elliptic curve points with prime order n, a generator P of G, and a point $x{\cdot}P$, it is computational infeasible to derive x, where $x \in Z_n^*$.

Theorem 1 The proposed certificateless signature scheme without bilinear pairings can achieve existential unforgeability against a super-level type I adversary in the random oracle model, assuming the hardness of solving the ECDLP.

Proof Let α_1 be a polynomial-time algorithm that breaks the certificateless signature scheme with non-negligible advantage ε_1, and H_1,H_2,H_3 are three random oracles. The goal is to the algorithm α_1 for building a polynomial-time algorithm β that solves the ECDLP. That is, given a random instance (P, $x{\cdot}P$), the goal is to derive x.

In the **initialization phase**, β runs the *Setup* algorithm and generates a master secret key mk=$s \in Z_n^*$, public system parameters $params$=$\{G,P,P_{pub}\}$. Next, β keeps mk and gives $params$ to α_1.

In the **Query phase**, α_1 can adaptively issue the following oracle queries to β [2, 5], and each query is unique.

H_1: For responding to α_1's queries, β maintains a list $list_{H_1}$ storing (ID_i,R_i,P_{pub},h_i). Upon receiving an H_1 query for some (ID_i,R_i,P_{pub}) from α_1, β checks the $list_{H_1}$ and returns h_i to α_1. Detailed steps are as follows.

If (ID_i,R_i,P_{pub},h_i) exists in $list_{H_1}$, β directly returns h_i to α_1 and terminates the step.

Randomly choose an $h_i \in Z_n^*$.

Add (ID_i,R_i,P_{pub},h_i) into $list_{H_1}$.

Return h_i to α_1.

H_2: β maintains a list $list_{H_2}$ storing $(T_i,ID_i,PK_i,R_i,P_{pub},k_i)$ for responding to α_1's queries,. Upon receiving an H_2 query for some $(T_i,ID_i,PK_i,R_i,P_{pub})$ from α_1, β checks the $list_{H_2}$ and returns h_{2_i} to α_1. Detailed steps are as follows.

If $(T_i,ID_i,PK_i,R_i,P_{pub},k_i)$ exists in $list_{H_2}$, β directly returns k_i to α_1 and terminates the step.

Randomly choose an $k_i \in Z_n^*$.

Add $(T_i,ID_i,PK_i,R_i,P_{pub},k_i)$ into $list_{H_1}$.

Return k_i to α_1.

H_3: For responding to α_1's queries, β maintains a list $list_{H_3}$ storing $(m,T_i,ID_i,PK_i,R_i,P_{pub},l_i)$. Upon receiving an H_3 query for some $(m,T_i,ID_i,PK_i,R_i,P_{pub})$ from α_1, β checks the $list_{H_3}$ and returns h_i to α_1. Detailed steps are as follows.

If $(m,T_i,ID_i,PK_i,R_i,P_{pub},l_i)$ exists in $list_{H_3}$, β directly returns l_i to α_1 and terminates the step.

Randomly choose an $l_i \in Z_n^*$.

Add $(m,T_i,ID_i,PK_i,R_i,P_{pub},l_i)$ into $list_{H_1}$.

Return l_i to α_1.

ExtractPartialPrivateKey(i) Upon receiving such a query for some identity ID_i from α_1, β performs the following steps. Note that ID_i cannot be the target identity ID_i*.

Randomly choose two numbers $a_i,b_i \in Z_q$*.

Set $s_i=a_i$, $h_i=b_i$, and $R_i=a_i \cdot P - b_i \cdot P_{pub}$.

Return (ID_i,s_i,R_i) to α_1.

RequestPublicKey(i) Upon receiving such a query for some identity ID_i from α_1, β performs the following steps.

If $ID_i \neq ID_i$*, β fist simulates the *ExtractPartialPrivateKey* query for ID_i, where ID_i* is the target identity. Then, β randomly chooses a number $x_i \in Z_q$* and computes $PK_i=x_i \cdot P$. Finally, β returns PK_i to α_1.

If ID_i is the target identity, β randomly chooses three numbers $a_i,b_i,x_i \in Z_q$*, computes $R_i=a_i \cdot P$ and $PK_i=x_i \cdot P$, sets $h_i=b_i$. After that, β returns PK_i to α_1.

ExtractSecretValue(i) Upon receiving such a query for some identity ID_i from α_1, β simulates the *RequestPublicKey* query for the identity ID_i and returns x_i as a response.

ReplacePublicKey(i) Upon receiving such a query for some identity (ID_i,PK_i') from α_1, β performs the following steps.

Simulate the *RequestPublicKey* query for the identity ID_i.

Set $PK_i=PK_i'$.

Sign(i, m) Upon receiving such a query for (i, m) from α_1, β performs the following steps.

β simulates the *ExtractPartialPrivateKey* query to obtain (ID_i,s_i,R_i).

β randomly chooses a number $x_i \in Z_q$* and computes $PK_i=x_i \cdot P$.

β chooses two random numbers $a_i,b_i \in Z_q$* sets $\tau_i=a_i$, $l_i=b_i$, and computes $T_i=\tau_i \cdot P - l_i(k_i \cdot PK_i + s_i \cdot P)$, where $s_i \cdot P = R_i + h_i \cdot P_{pub}$. After that, β returns $\sigma_i=(R_i,T_i,\tau_i)$ to α_1.

At the final phase, α_1 successfully outputs $\sigma_i^*=(R_i^*,T_i^*,\tau_i^*)$ for the target ID_i* with non-negligible advantage ε_1. Hence, the algorithm β can solve the ECDLP with the at least advantage $\frac{1}{q_{H_1}}\left(1-\frac{1}{q_{H_1}}\right)^{q_n}\varepsilon_1$, where q_{H_1} denotes the maximum number of queries to H_1, and q_n denotes the maximum number of *ExtractPartialPrivateKey* queries. That contradicts the hardness of solving the ECDLP.

Theorem 2 The proposed certificateless signature scheme without bilinear pairings is can achieve existential unforgeability against a super-level type II adversary in the random oracle model, assuming the hardness of solving the ECDLP.

Proof Let α_2 be a polynomial-time algorithm that breaks the certificateless signature scheme with non-negligible advantage ε_2, and H_1,H_2,H_3 are three random oracles. The goal is to the algorithm α_2 for building a polynomial-time algorithm β that solves the ECDLP. That is, given a random instance (P, $x\cdot P$), the goal is to derive x.

In the **initialization phase** and the **Query phase**, α_2 and β performs the same tasks as described in Theorem 1. At the final phase, α_2 successfully outputs $\sigma_i^*=(R_i^*,T_i^*,\tau_i^*)$ for the target ID_i* with non-negligible advantage ε_1. Hence, the algorithm β can solve the ECDLP with the at least advantage $\frac{1}{q_{H_1}}\left(1-\frac{1}{q_{H_1}}\right)^{q_m}\varepsilon_1$, where q_{H_1} denotes the maximum number of queries to H_1, and q_m denotes the maximum number of *ExtractSecretValue* queries. That also contradicts the hardness of solving the ECDLP.

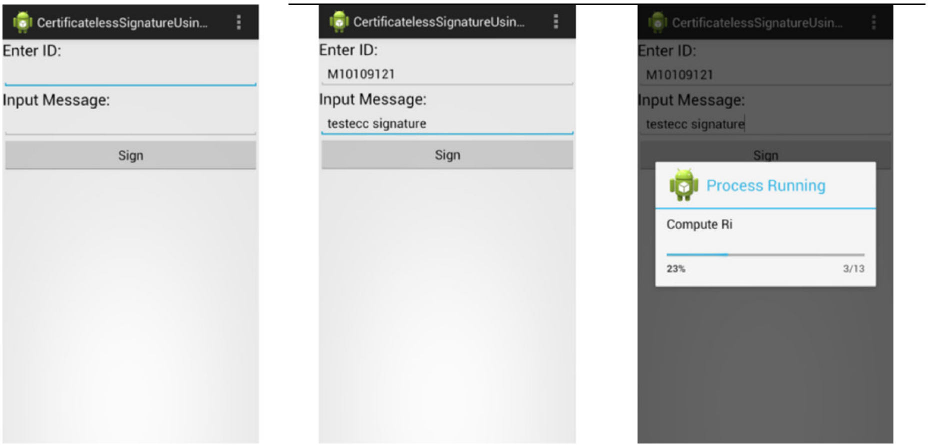
6 Prototype implementation

To evaluate the practicability and feasibility of the proposed CLS scheme, we implement our scheme on a resource-constrained mobile device embedded with one of the most popular linux-based operating systems: Android. In this section, we introduce the environment setup followed by the implementation results of the proposed CLS scheme. The basic implementation environment is shown in Table 1, where an HTC ONE X, JDK and Java Elliptic Curve Cryptography (JECC) [9] are adopted in the prototype system implementation. Next, we will report the implementation results.

Figure 1 shows the main input-pages of the proposed CLS scheme in which an ID_i = "M10109121" and a message m = "testecc signature" are adopted as the inputs for signature creation and verification (i.e., Fig. 1a and b). In Fig. 2, a simple simulation result for the signature creation and verification processes of our proposed CLS scheme is presented. Figure 2(a) shows the creation and verification processes are ongoing, and Fig. 2(b) soon demonstrates the verification is successfully done with a total computation time 2.199 s. Our prototype system further presents the computation cost of each parameter in Fig. 2(c) which concludes a result that the major overhead of our proposed CLS scheme is based on the computation of parameters P_{pub}, T_i, R_i and PK_i involving with the addition operation of Elliptic Curve Cryptography. Moreover, we can click each parameter to obtain a corresponding value, such as Fig. 3(a and b). Note that in ECC the x and y coordinates is computed, respectively, and this will result in two values for each parameter (e.g., Fig. 3a). After evaluating the feasibility of our proposed CLS scheme on Android-based mobile phone, we test the average overhead of the proposed CLS scheme. In Table 2, we obtained an average total computation time 1.916 s from 30 runs of our proposed CLS scheme. As mentioned before, the four

Table 1 Environment description

Smartphone	HTC ONE X: 1.5 GHz, Quad-core, RAM 1 GB, Android 4.1.1
Development environment	Android Studio 0.5.1
	JDK1.7.0_40, Android API 19
	Java Elliptic Curve Cryptography (jecc-alpha 1.1.tar.gz)



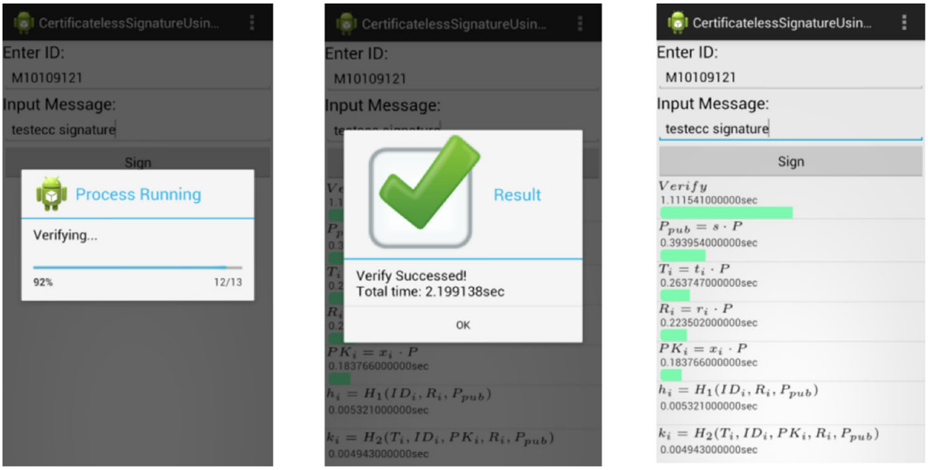
Enter ID: M10109121
Input Message: testecc signature

Compute Ri
23% 3/13

(a) Main Screen (b) Input ID_i and m (c) Computation progress

Fig. 1 The main pages of the implementation of our proposed CLS scheme, where ID_i=“M10109121” and a message m=“testecc signature” are the inputs for signature creation and verification

parameters PK_i, P_{pub}, R_i and T_i are essential overheads for total computation cost as the addition operation of ECC is exploited. The corresponding overhead percentage is 11.74 %, 10.69 %, 11.64 % and 11.87 %, respectively. In the future, these parameters could be the main re-design target when establishing a new and more efficient CLS scheme. With the above results, we believe that our implementation reflects the practicability and feasibility of our proposed CLS scheme for existing mobile communication environments involving with common handheld devices.



Enter ID: M10109121
Input Message: testecc signature

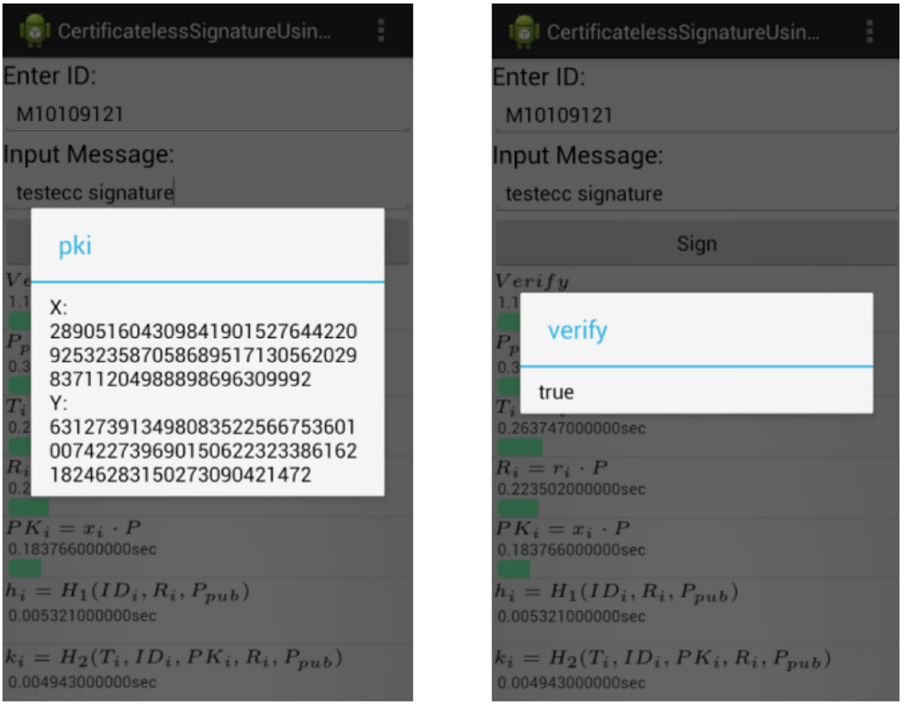
Verifying...
92% 12/13

Verify Successed!
Total time: 2.199138sec

$Verify$
1.111541000000sec
$P_{pub} = s \cdot P$
0.393954000000sec
$T_i = t_i \cdot P$
0.263747000000sec
$R_i = r_i \cdot P$
0.223502000000sec
$PK_i = x_i \cdot P$
0.183766000000sec
$h_i = H_1(ID_i, R_i, P_{pub})$
0.005321000000sec
$k_i = H_2(T_i, ID_i, PK_i, R_i, P_{pub})$
0.004943000000sec

(a) Verification process (b) Verification result (c) Parameter profile

Fig. 2 The snapshots of signature verification of the proposed CLS scheme, where the green bar below the each item in (**c**) represents the computation time for the parameter



Enter ID:
M10109121
Input Message:
testecc signature

pki

X:
289051604309841901527644220
925323587058689517130562029
8371120498889869630999 2

Y:
631273913498083522566753601
007422739690150622323386162
182462831502730904214 72

$PK_i = x_i \cdot P$
0.183766000000sec

$h_i = H_1(ID_i, R_i, P_{pub})$
0.005321000000sec

$k_i = H_2(T_i, ID_i, PK_i, R_i, P_{pub})$
0.004943000000sec

(a) Detail for parameter PK_i

Enter ID:
M10109121
Input Message:
testecc signature

Verify

verify

true

T_i
0.263747000000sec

$R_i = r_i \cdot P$
0.223502000000sec

$PK_i = x_i \cdot P$
0.183766000000sec

$h_i = H_1(ID_i, R_i, P_{pub})$
0.005321000000sec

$k_i = H_2(T_i, ID_i, PK_i, R_i, P_{pub})$
0.004943000000sec

(b) Detail for verification result

Fig. 3 The functionality for detail showing of each parameter

Table 2 The evaluation of computation cost of the proposed CLS scheme

Process	Sec	Percentage
h_i	0.001552267	0.081 %
ID_i	0.000340767	0.01778 %
k_i	0.002676633	0.13968 %
l_i	0.0033637	0.17554 %
m	0.0000779	0.00406 %
mk	0.000219867	0.01147 %
PK_i	0.224936767	11.7388 %
P_{pub}	0.204789233	10.6873 %
r_i	3.95667E-05	0.00206 %
R_i	0.2230888	11.6423 %
s_i	5.02667E-05	0.00262 %
τ	0.0000748	0.00390 %
t_i	3.99E-05	0.00208 %
T_i	0.227536567	11.8745 %
x_i	0.0000318	0.00165 %
Signature creation	0.893106466	46.60 %
Signature verification	1.023071467	53.40 %
Total computation time	1.916177933	100 %

7 Conclusion

In this paper, we have demonstrated a novel CLS scheme which is secure against super-level adversaries (type I and type II). Our proposed CLS scheme is superior to most CLS protocols [2, 3, 5, 6, 8, 10, 15] with either lower computation cost or better security robustness. That is, without the heavy computation of bilinear pairings, our proposed scheme is efficient to support the secure communication among mobile entities. Meanwhile, with the strong robustness property of elliptic curve cryptography, the proposed CLS protocol preserves high system security. We additionally implement a prototype system of our proposed CLS scheme and the corresponding results show the practicability and feasibility of the CLS scheme. In conclusion, we believe that our CLS scheme is more practical and suitable for securing existing mobile network environment.

Acknowledgments This work was partly supported by the Taiwan Information Security Center (TWISC) and National Science Council, Taiwan, under the Grants Numbers NSC 102-2218-E-259-004, NSC 102-2218-E-146-002 and NSC 102-2218-E-011-012.

References

1. Al-Riyami, Paterson K (2003) Certificateless public key cryptography. In Proceedings of ASIACRYPT 2003, Lecture Notes in Computer Science, vol. 2894, pp 452–473
2. Gong, Li P (2012) Further improvement of a certificateless signature scheme without pairing. Int J Commun Syst. doi:10.1002/dac.2457
3. Gorantla M, Saxena A (2005) An efficient certificateless signature scheme. In Proceedings of 2005 International Conference on Computational Intelligence and Security, pp 110–116
4. He D, Chen J, Hu J (2012) A pairing-free certificateless authenticated key agreement protocol. Int J Commun Syst 25:221–230
5. He D, Chen J, Zhang R (2012) An efficient and provably-secure certificateless signature scheme without bilinear pairings. Int J Commun Syst 25(11):1432–1442
6. Hu BC, Wong DS, Zhang Z, Deng X (2006) Key replacement attack against a generic construction of certificateless signature. In Proceedings of ACISP 2006, Lecture Notes in Computer Science, vol. 4058, pp 235–246
7. Huang X, Mu Y, Susilo W, Wong DS, Wu W (2007) Certificateless signature revisited. In Proceedings of ACISP 2007, Lecture Notes in Computer Science, vol. 4586, pp 308–322
8. Huang WS, Mu Y, Zhang F (2005) On the security of certificateless signature schemes from asiacrypt 2003. In Proceedings of CANS 2005, Lecture Notes in Computer Science, vol. 3810, pp 13–25
9. Java Elliptic Curve Cryptography project, http://jecc.sourceforge.net/
10. Li X, Chen K, Sun L (2005) Certificateless signature and proxy signature schemes from bilinear pairings. Lith Math J 45:76–83
11. Shamir (1985) Identity-based cryptosystems and signature schemes. In Proceedings of CRYPTO'84, Lecture Notes in Computer Science, vol. 196, pp 47–53
12. Tian M, Huang L (2012) Cryptanalysis of a certificateless signature scheme without pairings. Int J Commun Syst. doi:10.1002/dac.2310
13. Tsai J-L, Lo N-W, Wu T-C (2012) Weaknesses and improvements of an efficient certificateless signature scheme without using bilinear pairings. Int J Commun Syst. doi:10.1002/dac.2388
14. Yeh K-H, Tsai K-Y, Kuo R-Z, Wu T-C (2013) Robust certificateless signature scheme without bilinear pairings. In Proceeding of the 2013 International Conference IT Convergence and Security (ICITCS 2013), pp 1–4, 16–18
15. Yum D, Lee P (2004) Generic construction of certificateless signature. In Proceeding of the 9th Australasian Conference on Information Security and Privacy, pp 200–211

Kuo-Hui Yeh received his B.S. degree in Mathematics from the Fu Jen Catholic University, Taipei County, Taiwan, in 2000, and the M.S. and Ph.D. degrees in Information Management from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2005 and 2010, respectively. He is currently an assistant professor of Department of Information Management at the National Dong Hwa University, Hualien, Taiwan. His research interests include cloud computing, RFID applications and security, wireless network protocol, and anonymous authentication.

Kuo-Yu Tsai is an Assistant Professor at the Department of Management Information Systems, Hwa Hsia Institute of Technology, Taiwan. He received his Ph.D. Degree in the Department of Information Management from National Taiwan University of Science and Technology, Taiwan, in 2009. His recent research interests include information security, cryptography, network security, and cloud computing.

Chuan-Yen Fan majors in Information Management, and received his B.S. degree from Yuan Ze University, Taoyuan County, Taiwan, in 2012 and his M.S degree in 2014 from National Taiwan University of Science and Technology, Taipei, Taiwan. He currently focuses on the development of Android Application with Privacy Leakage Detection and he will serve as an engineer in Research and Development Substitute Services in CyberTrust Technology Institute, Institute for Information Industry, Taipei, Taiwan from 2014 to 2017.