Advances in Engineering Research, volume 113
International Forum on Mechanical, Control and Automation (IFMCA 2016)

A New Efficient Leakage-Free Certificateless Signature

Zhuo Gao[1, a] , Liang Hu[1,b] and Hongtu Li[1,c]

[1]College of Computer Science and Technology, Jilin University, Changchun, 130012, China

[a]gaozhuo03@163.com , [b]hul@jlu.edu.cn ,[c]liht@jlu.edu.cn

Keywords: certificateless signature, ESL attack, computational Diffie-Hellman problem

Abstract. Digital signature, a cryptographic algorithm, provides integrity, authentication and non repudiation messages.In the literature, several certificateless signature schemes have been designed. However, most of them are vulnerable to the ephemeral secret leakage (ESL) attack.We propose a new secure certificateless signature scheme which can resist ESL attacks.Moreover, we prove the security of two types of adversaries in certificateless signature schemes.The proposed scheme provides unforgeability based on the hardness assumption of CDH problem.

Introduction

In 1984,Shamir first introduced the identity-based public key cryptography[1]. This algorithm uses a string generated by the user's identity information to represent his public key. Thus it can simplify the management of certificates.However, all of the user's private key are generated by the key generation center PKG, which leads to the problem of key escrow. In 2003, the certificateless public key cryptography proposed by Al-Riyami and Paterson solved this problem [2]. In this system, the user's private key is composed of two parts:a partial private key generated by KGC and the secret value selected by user.

Several certificateless signature schemes(CLS) [3-5] have been proposed,since the certificateless public key cryptography appeared. However, most of the schemes are insecure and can be attack by a public key replacement. Later, some more secure certificateless signature scheme have been proposed, which can resist the public key replacement attack to some extent [6-9]. These schemes need to generate a ephemeral secret to generate signatures. Ephemeral secrets (i.e. random values) are chosen to generate signatures, called probabilistic signatures in the sense that a signer can issue distinct signatures for the same message[11]. Note that the ephemeral secret keys x or y are not the only session-specific secret information used by the parties—they also use secret random coins in the signature generation. We observe that if the adversary reveals these random coins, it can break the security of the protocol.[15]。

If the ephemeral secret keys are compromised, an adversary can reveal the private key of the signer from the corresponding signature, termed ephemeral secret leakage (ESL) attacks[11].This attack is possible and widely studied recently in [11-14]. Since the sender must rely on internal/external source of random number generator that may be controlled by anadversary[12].Note that,the schemes above are vulnerable to the ESL attack.Note that,if the attack gets the ephemeral secret,he can compute the user's secret key and forgery use it. Therefore, the scheme above can not resist the ESL attack.In this paper we propose a new certificateless signature scheme which is able to resist ESL attack.

Preliminaries

Bilinear Maps. Let G_1 be an additive group with prime order q ,P is the generator of G_1. Let G_2 be a multiplicative group with the same order.An admissible map e :$G_1 \times G_1 \rightarrow G_2$ is called a bilinear map if it satisfies the following properties[16]:

(1)Bilinearity: For any P,Q,R$\in G_1$, we have e(P+Q,R)=e(P,R)e(Q,R) and e(P,Q+R)= e(P,Q)e(P,R). In particular, for any a,b $\in Z_q^*$, $e(aP,bP)=e(P,P)^{ab}$ = e(P,abP)=e(abP,P)[9].

(2)Non-degeneracy: There exists P,Q$\in G_1$, such that e(P,Q)= 1.

(3)Computability: There is an efficient algorithm to compute e(P,Q) for all P,Q$\in G_1$.

Security Assumption. Discrete Logarithm (DL) Problem: Given a generator P of a cyclic group G with order q, and M$\in$G to find an integer a$\in Z_q^*$,such that M=aP.

Computational Diffie-Hellman(CDH) Problem: Given a generator P of a cyclic group G with order q, and given aP,bP for unknown a,b$\in Z_q^*$,to compute abP.

Framework of Leakage-Free Certificateless Signature Schemes.A CLS consists of six algorithms [9]. The description of each algorithm is as follows.

Setup: This algorithm is run by the KGC that accepts as input a security parameter k to generate a master-key *msk* and a list of system parameters *params*.

Partial-Private-Key-Extract: This algorithm is run by the KGC that takes as input a user's identity ID, params and *msk* to generate the user's partial private key D_{ID}.

Set-Secret-Value: This algorithm is run by a user that takes as input params and a user's identity ID to generate the user's secret value S_{ID}.

Set-Public-Key: This algorithm is run by a user that takes as input a params, a user's identity ID and secret value S_{ID} to generate the user's public key PK_{ID}.

Sign: This algorithm is run by a user that takes params, a message M, the user's identity ID, public key PK_{ID}, secret value S_{ID} to produce a signature σ on message M.

Verify: This algorithm is run by a verifier that takes as input a message M, a signature σ, params, a signer's identity ID and his public key PK_{ID} and to output true if the signature is valid, or ⊥ .

Security Model for Leakage-Free Certificateless Signature Schemes. There are two types of adversaries which named Type I adversary and Type II adversary in CL-PKC.

A type I adversary A_I does not have access to the master-key, but he has the ability to replace the public key of any entity with a value of his choice.

A type II adversary A_{II} has access to the master-key but cannot perform the public key replacement[16].

The security of a CLS scheme is modeled via the following two games between a challenger C and an adversary A_I or A_{II} .

Leakage-Free CLS Schemes

Our leakage-free CLS scheme consists of seven algorithms which are described as follows.

Setup: This algorithm generates the params and the master key of the system.

(1)Given a security parameter k, the KGC chooses a cyclic additive group G_1 with prime order q.P is the generator of G_1.KGC chooses a cyclic multiplicative group G_2 with the same order. and a bilinear map e : $G_1 \times G_1 \rightarrow G_2$.

(2)The KGC also chooses a random s$\in Z_q^*$ as the master-key and sets P_{pub} = sP.

(3)KGC chooses distinct hash functions H_1 : {0, 1} $\rightarrow G_1$, H_2 : {0, 1} $\rightarrow G_1$, f_1 : {0, 1} $\rightarrow Z_q^*$,

$f_2 : \{0, 1\} \rightarrow Z_q^*$, $f_3 : \{0, 1\} \rightarrow Z_q^*$.

The system parameter list is params=(G_1,G_2, e, P, P_{pub}, H_1,H_2,f_1,f_2,f_3). The master-key is s.

Partial-Private-Key-Extract: This algorithm takes params, master-key s and a user's identity $ID_i \in \{0, 1\}$ as input. It generates the partial private key for the user as follows:

(1)Computes $Q_{ID}=H_1(ID_i)$.

(2)Outputs the partial private key $D_{ID}=sQ_{ID}$.

It is easy to see that DID is valid on ID for the key pair (P_{pub}, s), and user can check its correctness by checking whether $e(D_{ID}, P) =e(Q_{ID}, P_{pub})$.

Set-Secret-Value: This algorithm takes as input params and a user's identity ID_i. It then selects a random $x \in Z_q^*$ and outputs x as the user's secret value.

Set-Public-Key: This algorithm accepts params, a user's identity ID_i and this user's secret value x as input.It produces the user's public key $PK_{ID}=xP$.

Sign: To sign a message M using the partial private key D_{ID} and the secret value x, the signer, whose identity is ID_i and the corresponding public key is PK_{ID}, performs the following steps.

(1)Choose a random $r \in Z_q^*$, compute $U = rP$.

(2)Compute T=f1(IDi,m,U,DID),set $R=U+TP$.

(3)Compute $V=h_1 \times D_{ID}+(T+r)\times HID+h_2 \times S_{ID} \times Q_{ID}$,where $h_1=f_2(ID_i,m,R,PK_{ID})$,$h_2=f_3(ID_i,m,R)$ and $HID=H_2(ID_i,m,R)$.

(4)Output $\sigma = (R, V)$ as the signature on M.

Verify: To verify a signature σ on a message M for an identity ID_i and public key PK_{ID}, the user performs the following steps.

(1)Compute $h_1=f_2(ID_i,m,R,PK_{ID})$,$h_2=f_3(ID_i,m,R)$ and $HID=H_2(ID_i,m,R)$.

(2)Verify $e(V,P)=e(QID,h_1 \times P_{pub}+h_2 \times PK_{ID})e(HID,R)$ .If the equation holds, output true. Otherwise, output $\perp$.

Security Proof

Theorem 1. In the random oracle model,the proposed CLS scheme is existential unforgeable against the A_I adversary assuming the CDH problem is intractable.

Proof. Let C be a CDH attacker who receives a random instance (P,aP,bP) of the CDH problem in G_1 .Let A_I be a super type I adversary that breaks the proposed signature. We show how C can use A_I to solve the CDH problem, that is to compute abP.

First C sets P_{pub}=aP,where P_{pub} is the public key of the KGC.Then C selects params={G_1,G_2,e,q,P,P_{pub},H_1,H_2,f_1,f_2,f_3} and sends it to A_I.

H_1 Queries:Suppose A_I can make at most qH_1 times H_1 queries,C chooses $j \in [1,qH_1]$.C maintains an initially empty list H_1^{list} of tuples (ID_j,QID_j,α_j). On receiving a new query $H_1(ID_i)$, C responds as follows:

(1)If ID_i has appeared on the H_1^{list} , then C responds with $H_1(ID_i)=Q_{ID} \in G_1$.

(2)If ID_i has not appeared on the list, then C selects $\alpha \in Z_q^*$ at random.

a.If $i=j$,Set $H_1(ID_i)=Q_{ID} = \alpha bP$.Return Q_{ID} as answer.

b.Otherwise,set $H_1(ID_i)=Q_{ID} = \alpha P$.

Adds (ID_i,Q_{ID},α) to H_1^{list} and return Q_{ID} as respond.

Partial-Private-Key Queries:C keeps an initially empty list K^{list} of tuples (ID_j,PK_{IDj},S_{IDj},D_{IDj}). When A_I issues a query on ID_i, C responds as follows:

(1)If ID_i has appeared on the K^{list} ,then C responds with D_{ID}.

(2)If ID_i has not appeared on the list,then C finds for H_1^{list} the tuple (ID_i,QID_i,α_i).

a.If there is a tuple (ID_i,QID_i,α_i) on the H1list and ID_i=ID_j,abort.

b.If $ID_i \neq ID_j$,set $D_{ID}=\alpha_i P_{pub}=\alpha_i \times aP$,$S_{ID}=\perp$ and $PK_{ID}=\perp$.

c.Otherwise,set $D_{ID}=\alpha_i P_{pub}=\alpha_i \times aP$,$S_{ID}=\perp$ and $PK_{ID}=\perp$.

Adds (ID_i,PK_{ID},S_{ID},D_{ID}) to K^{list} and return D_{ID} as respond.

H_2 Queries: C keeps an initially empty list H_2^{list} of tuples (ID_j,m_j,U_j,h_2). Whenever A_I issues a query (ID_i,m_i,U_i) to H_2, the same answer from the list H_2 will be given if the request has been asked before. If the query (ID_i,m_i,U_i) is new, C selects a random $\beta \in Z_q^*$ and set $h_2=\beta P$.Adds (ID_i,m_i,U_i,h_2) to H_2^{list} and returns h_2 as answer.

f_1 Queries: C keeps an initially empty list f_1^{list} of tuples (ID_j,m_j,U_j,D_{IDj},f_1). Whenever A_I issues a query (ID_i,m_i,U_i,D_{IDi}) to f_1, the same answer from the list f_1^{list} will be given if the request has been asked before. If the query (ID_i,m_i,U_i,D_{IDi}) is new, C selects a random $t \in Z_q^*$ and set f1=t.Adds (ID_i,m_i,U_i,D_{IDi},f_1) to f_1^{list} and returns f_1 as answer.

f_2 Queries: C keeps an initially empty list f_2^{list} of tuples (ID_j,m_j,R_j,PK_{IDj},f_2). Whenever A_I issues a query (ID_i,m_i,R_i,PK_{IDi}) to f_2, the same answer from the list f_2^{list} will be given if the request has been asked before. If the query (ID_i,m_i,R_i,PK_{IDi}) is new, C selects a random $u \in Z_q^*$ and set f_2=u.Adds ((ID_i,m_i,R_i,PK_{IDi},f_2) to f_2^{list} and returns f_2 as answer.

f_3 Queries: C keeps an initially empty list f_3^{list} of tuples (ID_j,m_j,R_j,f_3). Whenever A_I issues a query (ID_i,m_i,R_i) to f_3, the same answer from the list f_3^{list} will be given if the request has been asked before. If the query (ID_i,m_i,R_i) is new, C selects a random $v \in Z_q^*$ and set f_3=v.Adds (ID_i,m_i,R_i,f_3) to f_3^{list} and returns f_3 as answer.

Public-Key Queries: When A_I issues a query on ID_i, C responds as follows:

(1)If ID_i has appeared on the K^{list} and $PK_{IDi} \neq \perp$,then C responds with PK_{IDi}.

(2)If $PK_{ID}=\perp$,select $x' \in Z_q^*$ in random.Set S_{ID}=x' and PK_{ID}=x'P.Then C responds with PK_{ID}.

(3)Otherwise,select $x \in Z_q^*$ in random.Set $D_{ID}=\perp$,S_{ID}=x and PK_{ID}=xP.Adds (ID_i,PK_{ID},S_{ID},D_{ID}) to K^{list} and return PK_{ID} as respond.

Public-Key-Replacement Queries:A_I can choose a new public key PK_{ID}' for the user whose identity is ID_i.C first finds the list K^{list}. If ID_i has appeared on the list and $PK_{ID} \neq \perp$,then set $D_{ID}=\perp$,$S_{ID}=\perp$ and PK_{ID}=PK_{ID}'.Otherwise,make Public-Key Queries,then update PK_{ID}=PK_{ID}'. Adds (ID_i,PK_{ID},S_{ID},D_{ID}) to K^{list}.

Secret-Value Queries: Suppose the query is on ID_i.C find the list K^{list}.If IDi has appeared on the list and $S_{ID}=\perp$,it means that the public key of ID_i has been replaced.C returns $\perp$ .Else,C returns S_{ID}.If ID_i has not appeared on the list,C makes Public-Key Queries on ID_i and returns the corresponding S_{ID} as answer.

Sign Queries: Note that at any time during the simulation, equipped with those partial private keys for any $ID_i \neq ID_j$, A_I is able to generate signatures on any message. If ID_i=ID_j,A_I issues a query (mi,PK_{IDi}) where m_i means a message and PK_{IDi} means a current public key chosen by A_I to the signature whose private key is associated with ID_j. On receiving this, C creates a signature as follows:

(1)Select $k,u,v \in Z_q^*$ in random.Set U_i=P-kP and R_i=kP.

(2)Compute $V_i=\alpha_i b(uP_{pub}+vPK_{IDi})+\beta P$.Output the signature σ_i=(R_i,V_i).

Forgery: Finally, A_I outputs a signature (M ,σ = (R ,V),ID ,PK_{ID}^*) which means (R ,V) is a valid signature on message M for identity ID and public key PK_{ID}^*. If $ID^* \neq ID_j$, C aborts. By forking lemma [17], C replays A_I with the same random tape but different choice of the hash function h_1 for another choice of f_2.Now A_I outputs another forged signature (M ,σ =(R ,V'),ID ,PK_{ID}^*). Both of the two signatures are valid,so they should satisfy the following equations.

$$e(V^*,P)=e(Q_{ID}^*,h_1^*\times P_{pub}+h_2^*\times PK_{ID}^*)e(HID^*,R^*) \quad (1)$$

$$e(V;,P)=e(Q_{ID}^*,h_1'\times P_{pub}+h_2^*\times PK_{ID}^*)e(HID^*,R^*) \quad (2)$$

Where $P_{pub}=aP$,$Q_{ID}^*=\alpha\times bP$,$HID=\beta^*P$ and $h_1^*\neq h_1'$.Hence we have $V^*=\alpha\times b\times h_1^*\times aP+\alpha\times b\times h_2^*\times PK_{ID}^*+\beta^*R^*$ and $V'=\alpha\times b\times h_1'\times aP+\alpha\times h_2^*\times PK_{ID}^*+\beta R^*$.C can compute $abP=(V^*-V')/(\alpha\times(h_1^*-h_1'))$.So C can successfully obtain the solution of the CDH problem.

Theorem 2. In the random oracle model,the proposed CLS scheme is existential unforgeable against the A_{II} adversary assuming the CDH problem is intractable.

Proof. Let C be a CDH attacker who receives a random instance (P,aP,bP) of the CDH problem in G_1 .Let A_{II} be a super type II adversary that breaks the proposed signature. We show how C can use A_{II} to solve the CDH problem, that is to compute abP.

First C selects a random $s\in Z_q^*$ as the master-key and sets $P_{pub}=sP$,where Ppub is the public key of the KGC.Then C selects params={$G_1,G_2,e,q,P,P_{pub},H_1,H_2,f_1,f_2,f_3$}.And initializes A_{II} with the params and the masker-key s.

H_1 Queries:Suppose A_{II} can make at most qH_1 times H_1 queries,C chooses $j\in[1,qH_1]$.C maintains an initially empty list H_1^{list} of tuples (ID_j,Q_{IDj},α_j). On receiving a new query $H_1(ID_i)$, C responds as follows:

(1)If ID_i has appeared on the H_1^{list} , then C responds with $H1(IDi)=Q_{ID}\in G_1$.

(2)If ID_i has not appeared on the list, then C selects $\alpha\in Z_q^*$ at random.Set $Q_{ID}=H_1(ID_i)=\alpha aP$.Adds ($ID,Q_{ID},\alpha$) to H_1^{list} and return Q_{ID} as respond.

Partial-Private-Key Queries:C keeps an initially empty list K^{list} of tuples (ID_j,PK_{IDj},S_{IDj}). When A_{II} issues a query on ID_i, C responds as follows:

(1)If ID_i has appeared on the K^{list},then C responds with D_{ID}.

(2)If ID_i has not appeared on the list,then C finds for H_1^{list} the tuple (ID_i,Q_{ID},α_i).

a.If there is a tuple (ID_i,Q_{ID},α_i)on the H_1^{list},then set $D_{ID}=sQ_{ID}=\alpha_i\times aP_{pub}$.

b.Otherwise,set $DID=sQ_{ID}=\alpha_i\times aP_{pub}$,$S_{ID}=\perp$ and $PK_{ID}=\perp$.

Adds ($ID_i,PK_{ID},S_{ID},D_{ID}$) to K^{list} and return D_{ID} as respond.

H_2 Queries: C keeps an initially empty list H_2^{list} of tuples (ID_j,m_j,U_j,h_2). Whenever A_{II} issues a query (ID_i,m_i,U_i) to H_2, the same answer from the list H_2 will be given if the request has been asked before. If the query (ID_i,m_i,U_i) is new, C selects a random $\beta\in Z_q^*$ and set $h_2=\beta P$.Adds ((ID_i,m_i,U_i,h_2) to H_2^{list} and returns h_2 as answer.

f_1 Queries: C keeps an initially empty list f_1^{list} of tuples (ID_j,m_j,U_j,DID_j,f_1). Whenever A_{II} issues a query (ID_i,m_i,U_i,DID_i) to f_1, the same answer from the list f_1^{list} will be given if the request has been asked before. If the query (ID_i,m_i,U_i,DID_i) is new, C selects a random $t\in Z_q^*$ and set $f_1=t$.Adds ((ID_i,m_i,U_i,DID_i,f_1) to f_1^{list} and returns f_1 as answer.

f_2 Queries: C keeps an initially empty list f_2^{list} of tuples ($ID_j,m_j,R_j,PK_{IDj},f_2$). Whenever A_{II} issues a query (ID_i,m_i,R_i,PK_{IDi}) to f_2, the same answer from the list f_2^{list} will be given if the request has been asked before. If the query (ID_i,m_i,R_i,PK_{IDi}) is new, C selects a random $u\in Z_q^*$ and set $f_2=u$.Adds ($ID_j,m_j,R_j,PK_{IDj},f_2$) to f_2^{list} and returns f_2 as answer.

f_3 Queries: C keeps an initially empty list f_3^{list} of tuples (ID_j,m_j,R_j,f_3). Whenever A_{II} issues a query (ID_i,m_i,R_i) to f_3, the same answer from the list f_3^{list} will be given if the request has been asked before. If the query (ID_i,m_i,R_i) is new, C selects a random $v\in Z_q^*$ and set f_3=v.Adds (ID_i,m_i,R_i,f_3) to f_3^{list} and returns f_3 as answer.

Public-Key Queries: When A_{II} issues a query on ID_i, C responds as follows:

(1)If ID_i has appeared on the K^{list} and $PK_{ID}\neq\perp$,then C responds with PK_{ID}.

(2)If ID_i has not appeared on the list and $ID_i=ID_j$,select $w\in Z_q^*$ in random.Set S_{ID}=w and $PK_{ID}=w\times bP$.Adds (ID,PK_{ID},S_{ID}) to K^{list} and return PK_{ID} as respond.

(3)Otherwise,select $w\in Z_q^*$ in random.Set S_{ID}=w and PK_{ID}=wP.Adds (ID,PK_{ID},S_{ID}) to K^{list} and return PK_{ID} as respond.

Secret-Value Queries: Suppose the query is on IDi.C find the list K^{list}.If IDi has appeared on the list and $ID_i\neq ID_j$,C returns $\perp$ .Otherwise,C return S_{ID}.If IDi has not appeared on the list,C makes Public-Key Queries on ID_i and returns the corresponding S_{ID} as answer.

Sign Queries: Note that at any time during the simulation, equipped with those partial private keys for any $ID_i\neq ID_j$, A_{II} is able to generate signatures on any message. If $ID_i=ID_j$,A_{II} issues a query (m_i,PK_{IDi}) where mi means a message and PK_{IDi} means a current public key chosen by A_{II} to the signature whose private key is associated with ID_j. When C receives this, C creates a signature as follows:

(1)Select $k,u,v\in Z_q^*$ in random.Set Ui=P-kP and Ri=kP.

(2)Compute $V_i=\alpha_i b(uP_{pub}+vPK_{IDi})+\beta P$.Output the signature $\sigma_i=(R_i,V_i)$.

Forgery: Finally, A_{II} outputs a signature (M ,σ = (R ,V),ID ,PK_{ID}^*) which means (R ,V) is a valid signature on message M for identity ID and public key PKID*. If $ID^*\neq ID_j$, C aborts. By forking lemma [17], C replays A_{II} with the same random tape but different choice of the hash function h_2 for another choice of f_3.Now A_{II} outputs another forged signature (M ,σ'=(R ,V'),ID , PK_{ID}^*). Both of the two signatures are valid,so they should satisfy the following equations.

$$e(V^*,P)=e(Q_{ID}^*,h_1^*\times P_{pub}+h_2^*\times PK_{ID}^*)e(HID^*,R^*) \quad (3)$$

$$e(V',P)=e(Q_{ID}^*,h_1^*\times P_{pub}+h_2'\times PK_{ID}^*)e(HID^*,R^*) \quad (4)$$

Where $Q_{ID}^*=\alpha\times aP$,$PK_{ID}^*=w\times bP$,$HID=\beta^*P$ and $h_2^*\neq h_2'$.Hence we have $V^*=\alpha\times a\times h_1^*\times P_{pub}+\alpha\times a\times h_2^*\times w\times bP+\beta^*R^*$ and $V'=\alpha\times a\times h_1^*\times P_{pub}+\alpha\times a\times h_2'\times w\times bP+\beta^*R^*$.C can compute $abP=(V^*-V')/(\alpha\times w\times(h_2^*-h_2'))$.So C can successfully obtain the solution of the CDH problem.

Comparative Analysis

In this section, we have calculated and analyzed the security performance and efficiency of the proposed scheme. First, the proposed scheme can resist ESL attacks, however, the scheme[6-10] can not.Now, we describe the vulnerability of [6-10] for ESL attacks.

(1) In the efficient CLS phase of [6],the signature is σ(U,v,w).The signature generate as follows:select random $r_1,r_2\in Zp^*$. Compute $R = g^{r1}$,$R'=g^{r2}$,$v=H_2(M,R,R',P_i)\in Z_p^*$,$U=(x_i*v+r_1)D_i$,$w= x_iv +r_2$. The adversary A_I who knows $\{r_1,r_2,x_i\}$ can compute the partial secret key D_i as $D_i=U/(x_i*v+r1)$.The adversary A_{II} who knows $\{r_1,r_2,D_i\}$ can compute the secret key x_i as $x_i=(w-r_2)/v$.

(2) In the efficient CLS phase of [7],the signer generate the signature σ(h,V).The signature can generate as follows:compute SID=x*DID,Select random $r\in Z_q^*$ and set $R = g^r$. Compute $h=H_2(M||ID||R||PID)\in Z_q^*$, V=r*P+h*SID.The adversary A_I who knows $\{r,x_i\}$ can compute the secret key SID as SID=(V-r*P)/h.

(3) In the CLS phase of [8],he signature is σ(U,V).The signature generates as follows: compute

$Q_A = H_1(ID_A) \in G_1$.Select random $r \in Z_q^*$ and set $U = rQ_A \in G_1$.Set $h = H_2(m||U) \in Z_q^*$ and Compute $V =(r +h)S_A$.The adversary who knows r can compute the secret key S_A as $S_A=v/(r+h)$.

(4) In the CLS phase of [9],the signer generate the signature σ(U,V).The signature generates as follows:select random $r \in Z_q^*$. Compute $U = rP$,and $V = D_A +rH_2(m,ID_A,PK_A,U)+xH_3(m,ID_A,PK_A)$.The adversary A_I who knows {r,x} can compute the secret key D_A as $D_A=V-rH_2(m,ID_A,PK_A,U)-xH_3(m,ID_A,PK_A)$.

(5) In CLS phase of [10],the signature is σ(V,S,R).The signature can generate as follows:Select random $r \in Z_q^*$. Compute $S = D_{ID}/S_{ID}$, $R = (r-S_{ID})*P$ and $V = H_2(M,R,P_{ID})*r*P$.The adversary A_I who knows $\{r,S_{ID}\}$ can compute the secret key D_{ID} as $D_{ID}=S_{ID}*S$.

Then,we compare the efficiency of the proposed scheme with the schemes [6-10].For the performance comparison with respect to calculation cost,we considered the following notations.The performance comparison are showed in Table 1.

(i)T_e:The time of executing bilinear pairing operation.

(ii)T_{mul}:The time of executing multiplication operation on elliptic curve.

(iii)T_i:The time of executing modular inversion operation.

(iv)T_s:The time of executing exponentiation operation.

(v)T_h:The time of executing a map-to-point hash fuction.

(vi)G:The length of a point in G1.

(vii)Z:The length of a point in Zq*.

Table 1.The performance of the schemes.

Scheme	Sign	Verify	ESL attack	Signature Length
Scheme in [6]	$2*T_s+T_h+T_{mul}$	$T_e+2*T_{mul}+2*T_i+T_s+T_h$	Yes	G+2*Z
Scheme in [7]	$T_s+T_h+2*T_{mul}$	$2*T_e+T_{mul}+T_i+T_h$	Yes	2*G
Scheme in [8]	$2*T_h+2*T_{mul}$	$2*T_e+T_{mul}+2*T_h$	Yes	2*G
Scheme in [9]	$2*T_h+3*T_{mul}$	$4*T_e+3*T_h$	Yes	2*G
Scheme in [10]	$3*T_{mul}$	$2*T_e+2*T_{mul}+T_h$	Yes	3*G
The proposed scheme	$4*T_h+5*T_{mul}$	$3*T_e+2*T_{mul}+4*T_h$	No	2*G

The table shows that the bilinear operation of scheme [6] is the least, but the computational complexity is very high.Signature length and security performance which schemes [7-9] produces are similar. The scheme [10] makes small computational cost, but the signature length is long.Although our scheme produces some computational cost,however, it can resist ESL attacks and provable security in the random oracle model.

Conclusion

In this paper, we propose a new secure certificateless signature scheme. This new scheme can resist ESL attacks.Moreover, we prove the security of two types of adversaries in certificateless signature scheme.The proposed scheme provides unforgeability based on the hardness assumption of CDH problem.Therefore, this scheme can be used in many security applications.

Acknowledgements

This work was financially supported by the European Seventh Framework Program(FP7)

(GA-2011-295222) and the National Sci-Tech Support Plan of China (2014BAH02F03).

Reference

[1] Shamir A. Identity-Based Cryptosystems and Signature Schemes[J]. Lecture Notes in Computer Science, 1995, 21(2):47-53.
[2] S. Al-Riyami and K. Paterson.Certificateless public key cryptography.Lecture Notes in Computer Science, vol. 2894, pages 452-473, Springer-Verlag, 2003.
[3] Yulei Zhang,Saifen Wang,Yongjie Zhang,et al. A new efficient certificateless signature scheme based on [J]. Computer Engineering and Applications.2010, 46(14):84-87.In Chinese.
[4] Huang X, Susilo W, Mu Y, et al.On the Security of Certificateless Signature Schemes from Asiacrypt 2003[M]//International Conference on Cryptology and Network Security.2005:13-25.
[5] Yum D H, Lee P J. Generic Construction of Certificateless Signature[C]//Information Security and Privacy:, Australasian Conference,ACISP 2004,Sydney,Australia,July 13-15, 2004.
[6] Zhang L, Zhang F, Zhang F. New Efficient Certificateless Signature Scheme[M]// Emerging Directions in Embedded and Ubiquitous Computing. 2007:692-703.
[7] Yufeng Li,Yupei Liu,Zhangfang Zhu.Efficient certificateless signature scheme [J].Computer Engineering and Applications.2011, 47(10):23-26.In Chinese.
[8] Yap W S, Heng S H, Goi B M. An Efficient Certificateless Signature Scheme[J]. Lecture Notes in Computer Science, 2006, 4097:322-331.
[9] Zhang Z, Wong D S, Xu J, et al. Certificateless public-key signature: security model and efficient construction[C]// Applied Cryptography and Network Security, International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings. DBLP, 2006:293-308.
[10] Xu Z,Liu X,Zhang G,et al.A Certificateless Signature Scheme for Mobile Wireless Cyber-Physical Systems[J].International Journal of Computers Communications & Control,2008, III(4):489-494.
[11] Tseng Y M, Tsai T T, Huang S S. Leakage-Free ID-Based Signature[J]. The Computer Journal, 2015, 58(4):750-757.
[12] Islam S H, Li F. Leakage-Free and Provably Secure Certificateless Signcryption Scheme Using Bilinear Pairings[J]. The Computer Journal, 2015, 58(10):2636.
[13] Islam S H. A Provably Secure ID-Based Mutual Authentication and Key Agreement Scheme for Mobile Multi-Server Environment Without ESL Attack[J]. Wireless Personal Communications, 2014, 79(3):1975-1991.
[14] Islam S H, Biswas G P. An improved ID-based client authentication with key agreement scheme on ECC for mobile client-server environments[J]. Theoretical & Applied Informatics, 2012, 24(4):293-312.
[15] Lamacchia B, Lauter K, Mityagin A. Stronger Security of Authenticated Key Exchange[C]// International Conference on Provable Security. Springer-Verlag, 2006:1-16.
[16] Zhang L, Zhang F.A New Provably Secure Certificateless Signature Scheme[J]. 2008, 61(7):1685-1689.
[17] Pointcheval D, Stern J. Security Proofs for Signature Schemes[C]// International Conference on Theory and Application of Cryptographic Techniques. Springer-Verlag, 1996:387--398.