

# On the Public Key Replacement and Universal Forgery Attacks of Short Certificateless Signature

Mingwu Zhang<sup>\*,a,1</sup>, Tsuyoshi Takagi<sup>a</sup>, Bo Yang<sup>b</sup>

<sup>a</sup>Graduate School of Mathematics, Kyushu University,  
744, Motoooka, Nishi-ku, Fukuoka, 819-0395, Japan

<sup>b</sup>College of Informatics, South China Agricultural University,  
P.O. Box 510642 Guangzhou, P.R.China

---

## Abstract

Certificateless cryptography eliminates the need of certificates in the PKI and solves the inherent key escrow problem in the ID-based cryptography. Recently, Du and Wen proposed a short certificateless signature scheme without MapToPoint hash function, and the signature size is short enough with only half of the DSA signature. In this paper, after the detailing the formal of certificateless signature scheme, we show that the Du and Wen's short certificateless signature scheme is insecure which is broken by a type-I adversary who has the ability in replacing users' public keys and accessing to the signing oracles, and it also cannot resist on the universal forgery attack for any third user.

*Key words:* Certificateless cryptographic, Short signature, Public key replacement attack, Existential forgery

---

## 1. Introduction

Certificateless cryptography[1, 2, 6], which eliminates the need of certificates of the Public Key Infrastructure(PKI) in public key cryptography settings and solves the inherent key escrow issue in the identity-based cryptography settings, was first proposed by Al-Riyami and Paterson[1]. In certificateless cryptography, user's private keys are generated not only by the Key Generation Center(KGC) but also by users themselves. i.e., KGC

---

\*Corresponding author

*Email addresses:* csmwzhang@gmail.com (Mingwu Zhang),  
takagi@math.kyushu-u.ac.jp (Tsuyoshi Takagi), byang@scau.edu.cn (Bo Yang)

only issues a partial private key to each user while the user independently generates his/her additional public/secret key pair. Consequently, the KGC is unable to obtain full secret key of any user, which can eliminate the full trustworthy and dependency on KGC in identity-based cryptosystems.

There are two adversaries defined for certificateless cryptosystems[1, 2]: Type-I adversary and Type-II adversary. Type-I Adversary models a dishonest user who can replace a user's public key with a false key of its choice, and Type-II adversary models a malicious KGC who can access the partial private key of a user. In[3], Yum and Lee proposed a generic construction of certificateless signature whose construction was built upon two primitives: a conventional digital signature scheme and an identity-based signature scheme. On the security of the generic construction, the generic construction is secure against KGC attack and key replacement attack if the signature scheme is existential unforgeable against chosen message attack (EUF-CMA) and the identity-based signature scheme is existential unforgeable against chosen message and identity attack (EUF-CMIA). However, Hu et al.[4] showed that the security requirements were insufficient to support the security claim in [3].

Short signature, is required for system/device with low bandwidth ability or small computation power, was first proposed successfully by Boneh, Lynn and Shacham[5] that the BLS signature is only half the size of DSA signature(320-bit) while its security is the same level of DSA's. It maybe deploy in devices such as PDA or cell phone for saving power and capability, and human requirements for asking a user's key in the signature (For instance, product registration systems often ask users to key in a signature provided on a barcode label or RFID identity).

Combined the features of certificateless signature(CLS) schemes and short signature schemes, short certificateless signatures(SCLS) schemes were constructed to restrict the malicious KGC behaviors[6, 7, 10, 11]. Huang et al.[6] revisited the security models of short CLS schemes and proposed two CLS schemes. They divided three kinds of adversaries against certificateless signatures according to their attack power into *normal adversary*, *strong adversary* and *super adversary* by their attack abilities, respectively. Combined with the known type-I adversary and type-II adversary, normal type-I adversary, strong type-I adversary can be obtained. In[7], Shim showed that the short CLS scheme in[6] is insecure against type-I adversary who can replace user's public keys and access to the signing oracle under the replaced public keys.

In[8], Boneh and Boyen proposed a short signature scheme that is strongly existentially unforgeable under an adaptive chosen message attack in the

standard security model under a new intractability assumption called Strong Diffie-Hellman(SDH), where its signature size is as short as DSA signature for comparable security. Later, Shao et al.[9] presented an attack way to forge a valid signature using public key altering and replacing model to break the scheme in[8]. Furthermore, they argued that the well-accepted notion of security for signature schemes, namely existential unforgeability against adaptive chosen-message attacks, is not adequate for the multi-user setting.

Tso, Yi and Huang[10] proposed a short certificateless signature scheme against realistic adversary model where an adversary is not allowed to get any valid signature under false public keys, which is as efficient as BLS short signature scheme in both communication and computation, but the security is in the realistic adversary model that the adversary can query a signing oracle to obtain valid signatures of original public keys but cannot obtain valid signatures of false public keys.

In the several CLS schemes[1, 4, 6], a special hash function called *MapToPoint* function which is used to map an identity information into a point on elliptic curve is required. However, the hash function is inefficient although there has been much discussion on the construction of such hash algorithm. Therefore, using general cryptographic hash function instead of the *MapToPoint* function can improve the efficiency of CLS schemes. In[11], Du and Wen proposed a certificateless short signature scheme without *MapToPoint* hash function. The signature size is approximate 160-bit which is comparable to the BLS short signature scheme.

In this paper, we show that Du and Wen's short certificateless signature scheme in[11] is insecure which cannot against two attacks scenarios: (1)it is broken by a type-I adversary who replaces user's public key and gets access to the signing oracle, and (2) it cannot resist on the universal forgery for any third user of his/her choice. That is, anyone can forge a valid certificateless signature if he got a previous signature and receiver's public key.

The rest of the paper is organized as follows: In Section 2, we recite the certificateless signature algorithm and model its security definitions and requirements. We review Du and Wen's CLS scheme in Section 3, and describe its attack deficiency and explain its reason in Section 4. Concluding remarks is drawn in Section 5.

## 2. Model of Certificateless Signature(CLS)

### 2.1. The CLS model

The CLS scheme is comprised of the following seven probabilistic polynomial time algorithms: *Setup*, *PartialKeyGen*, *UserKeyGen*, *UserPrivKey*, *UserPubKey*, *CL-Sign* and *CL-Veri*.

*Setup*: This algorithm is performed by KGC that accepts a security parameter  $k$  to generate a master-key and system parameters  $params$ .

*PartialKeyGen*: This algorithm is performed by KGC that takes a user's identity  $ID$ , a parameter list  $params$  and system master-key as inputs to produce the user's partial secret key  $D_{ID}$ .

*UserKeyGen*: This algorithm is run by a user that takes the user's identity  $ID$  as input, and outputs the user's secret value  $x_{ID}$ .

*UserPrivKey*: This algorithm takes  $params$ , a user's partial private key  $D_{ID}$  and his secret value  $x_{ID}$  as inputs, and outputs the full private key  $sk_{ID}$ .

*UserPubKey*: Take as inputs  $params$  and a user's secret value  $x_{ID}$  and/or his  $D_{ID}$ , this algorithm generates a public key  $P_{ID}$  for the user.

*CL-Sign*: This algorithm accepts a message  $m \in \mathcal{M}$ , the signer's identity  $ID$  together with corresponding public key  $P_{ID}$ , a parameter list  $params$  and the signing key  $sk_{ID}$  to generate a certificateless signature  $\sigma$  on message  $m$ .

*CL-Veri*: Take as inputs a message  $m$ , a signature  $\sigma$ , public list  $params$ , the signer's identity  $ID$  and corresponding public key  $P_{ID}$ , this algorithm outputs *true* if the signature is valid, or  $\perp$  otherwise.

The consistent of a CLS scheme satisfies:

For all  $k \in N$ ,  $m \in \{0, 1\}^*$ ,  $ID \in \{0, 1\}^*$ , if  $(s, params) \leftarrow Setup(1^k)$ ,  $D_{ID} \leftarrow PartialKeyGen(s, params, ID)$ ,  $x_{ID} \leftarrow UserKeyGen(params, ID)$ ,  $sk_{ID} \leftarrow UserPrivkey(params, D_{ID}, x_{ID})$ ,  $P_{ID} \leftarrow UserPubKey(params, ID, sk_{ID})$ ,  $\sigma \leftarrow CL-Sign(params, sk_{ID}, m)$ , it requires that:

$$CL-Veri(params, P_{ID}, ID, m, \sigma) = 1 \quad (1)$$

### 2.2. Security Requirements of CLS

The CLS scheme should be secure against existential forgery under adaptive-chosen-message attacks and adaptive-chosen-identity attacks. Informally, existential forgery means that the adversary attempts to forge a signature on identities and messages of his choice.

We define the two types of attacker against a certificateless cryptosystem as follows:

1. **KGC Attacker.** The attacker who knows only the partial key but not the additional secret key of the user is not able to do any cryptographic operation as the user. We name this attacker as malicious-but-passive KGC attacker such that even though the KGC is malicious, we actually assume that the KGC is passive, in the sense that the KGC would not actively replace the user public key or corrupt the user secret key. For example, the malicious KGC may passively eavesdrop the signature sent to a user and try to perform a forgery using his knowledge of the user partial key.
2. **Key Replacement Attacker(KPA).** A third party who can replace the user's public/secret key pair but does not know the user's partial key issued by the KGC cannot do any cryptographic operation as the user either.

Along with the attackers model above, there are two types of security models in a CLS scheme, Type-I security and Type-II security, along with two types of adversaries,  $\mathcal{A}_I$  and  $\mathcal{A}_{II}$ , respectively.

We first define the following oracles that can be accessed by the adversaries according to the game specifications. For simulating oracles, we assume that the game simulator keeps a history of "query-answer" lists while adversaries  $\mathcal{A}$  interact with challenger  $\mathcal{C}$ .

- (1) *PartialKeyGen-Oracle*( $\mathcal{O}_{PKE}$ ): When  $\mathcal{A}$  requests the partial private key for a user with identity  $ID$ ,  $\mathcal{C}$  responds the user's secret key  $D_{ID}$  by running *PartialKeyGen* algorithm.
- (2) *UserKeyGen-Oracle*( $\mathcal{O}_{USK}$ ): When  $\mathcal{A}$  requests the secret key for a user with identity  $ID$ ,  $\mathcal{C}$  responds the user's full secret key  $x_{ID}$  by running *UserKeyGen* algorithm.
- (3) *UserPubKey-Oracle*( $\mathcal{O}_{UPK}$ ): When  $\mathcal{A}$  requests the public key of a user with identity  $ID$ ,  $\mathcal{C}$  answers the corresponding public key  $P_{ID}$ .
- (4) *PublicKeyReplacement-Oracle*( $\mathcal{O}_{PKR}$ ): This query is to replace the public key  $P_{ID}$  for an identity  $ID$  with a new value  $P'_{ID}$ . On receiving such a query,  $\mathcal{C}$  updates the public key to the new value  $P'_{ID}$ .
- (5) *CL-Sign-Oracle*( $\mathcal{O}_{CLS}$ ): When  $\mathcal{A}$  requests a signature on a message  $m$  for a user with identity  $ID$ ,  $\mathcal{C}$  responds a valid signature  $\sigma$  for  $m$  by running *CL-Sign* algorithm. If the user's public key has been replaced by  $\mathcal{A}$ , then  $\mathcal{C}$  cannot find  $sk_{ID}$  and thus the signing oracle's answer may be incorrect. In such case, we assume that  $\mathcal{A}$  additionally submits

the secret value  $x'_{ID}$  corresponding to the replaced public key to the signing oracle.

**Type-I adversary.** Type-I adversary  $\mathcal{A}_I$  simulates attacks when the adversary compromises the user secret key or replaces the user public key. Informally,  $\mathcal{A}_I$  represents a third party who may compromise the target user's private key or replace his public key, but  $\mathcal{A}_I$  does not get access to the user's partial key nor the master key of the KGC.

**Type-II adversary.** Type-II adversary  $\mathcal{A}_{II}$ , called malicious-but-passive KGC, simulates attacks when the adversary knows system master key, but  $\mathcal{A}_{II}$  can no longer get access to private key nor replace public key. Moreover, for that adversary obtains the master key,  $\mathcal{A}_{II}$  can generate the partial key of any user. Informally,  $\mathcal{A}_{II}$  models an eavesdropping KGC or a colluder of the KGC, who knows master key and can derive the value of any user's partial key, but cannot obtain user's private key or replace public key.

**Type-I game.** The type-I game is performed between a challenger  $\mathcal{C}$  and the Type-I forger  $\mathcal{A}_I$  for a CLS signature scheme as follows:

- Initialization.  $\mathcal{C}$  runs *Setup* algorithm to generate the master key and public parameters to forger  $\mathcal{A}_I$ . Note that  $\mathcal{A}_I$  does not know the master key.
- Queries. Forger  $\mathcal{A}_I$  may require the  $\mathcal{O}_{PKE}, \mathcal{O}_{USK}, \mathcal{O}_{UPK}, \mathcal{O}_{PKR}, \mathcal{O}_{CLS}$  queries to  $\mathcal{C}$  by an adaptive manner.
- Signature forgery. Finally,  $\mathcal{A}_I$  outputs a signature  $\sigma^*$  for signer  $ID^*$  on message  $m^*$ , and wins the game if
  - (1)  $\mathcal{A}_I$  has never asked  $\mathcal{O}_{PKE}$  or  $\mathcal{O}_{USK}$  of the user  $ID^*$ ,
  - (2)  $ID^*$  can not be an identity for which performs  $\mathcal{O}_{PKR}$  to replace public key and  $\mathcal{O}_{PKE}$  to extract the partial private key,
  - (3)  $\sigma^*$  has never been queried by the  $\mathcal{O}_{CLS}$  oracle, and
  - (4)  $\text{CL-Veri}(params, \sigma^*, ID^*, m^*) \neq \perp$ .

**Type-II game.** The type-II game is modeled between the challenger  $\mathcal{C}$  and a malicious KGC adversary  $\mathcal{A}_{II}$  for CLS scheme as follows:

- Initialization.  $\mathcal{C}$  runs the *Setup* algorithm and sends *params* and master-key to the adversary  $\mathcal{A}_{II}$ . Note that adversary  $\mathcal{A}_{II}$  knows the master key and can obtain anyone's partial private key, so he need not perform  $\mathcal{O}_{PKE}$  oracle.

- Queries.  $\mathcal{A}_{II}$  may perform the  $\mathcal{O}_{USK}$ ,  $\mathcal{O}_{UPK}$ , and  $\mathcal{O}_{CLS}$  queries by an adaptive manner.
- Forger. Adversary  $\mathcal{A}_{II}$  outputs a tuple  $\sigma^*$  and win the game if
  - (1)  $\text{CL-Veri}(params, \sigma^*, ID^*, m^*) \neq \perp$ ,
  - (2)  $\mathcal{A}_{II}$  has never asked  $\mathcal{O}_{USK}$  oracle of the users  $ID^*$ , and
  - (3)  $\sigma^*$  has never been queried by the  $\mathcal{O}_{CLS}$  oracles.

### 3. Review of the CLS scheme by Du and Wen

In[11], Du and Wen presented a short CLS scheme with pairings in the random oracle model under the hardness assumption of k-CAA and Inv-CDH problem, which uses general cryptographic hash functions instead of MapToPoint functions. The scheme is listed as follows:

- *Setup*. Given a security parameter  $k$ , the KGC chooses two groups  $\mathcal{G}_1$  and  $\mathcal{G}_2$  of same prime order  $q > 2^k$  and a modified Tate pairing map  $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ .  $P$  is a generator of groups  $\mathcal{G}_1$  and  $g = \hat{e}(P, P) \in \mathcal{G}_2$ . KGC selects two hash functions  $H_1 : \{0, 1\} \rightarrow Z_q$ ,  $H_2 : \{0, 1\} \times \mathcal{G}_1 \rightarrow Z_q$ , and picks a random number  $s \in Z_q$  as system master key and computes its public key  $P_{pub} = sP \in \mathcal{G}_1$ . Afterwards, KGC publishes the system parameter list  $params = \{\mathcal{G}_1, \mathcal{G}_2, \hat{e}, q, P, g, P_{pub}, H_1, H_2\}$  and keeps master key  $s$ .
- *PartialKeyGen*. Given an identity  $ID \in \{0, 1\}^*$ , KGC computes  $Q_{ID} = H_1(ID)$ , and sets  $D_{ID} = \frac{1}{s+Q_{ID}}P$  as user's partial private key.  
 After the user with identity  $ID$  received the  $D_{ID}$ , he can verify the correctness of  $D_{ID}$  by checking the equation:  $\hat{e}(D_{ID}, Q_{ID}P) = g$ . For convenience, here it defines  $T = P_{pub} + Q_{ID}P$ .
- *UserKeyGen*. The user with identity  $ID$  picks randomly  $x_{ID} \in Z_q^*$  and sets  $x_{ID}$  as his secret value.
- *UserPrivKey*. The user outputs a pair  $(D_{ID}, x_{ID})$  as his private key  $sk_{ID}$ .
- *UserPubKey*. A user with secret value  $x_{ID}$  computes and publish his public key  $P_{ID} = x_{ID}(P_{pub} + Q_{ID}P) = x_{ID}T$ .

- *CL-Sign.* To sign a message  $m$ , a user with identity  $ID$  perform the following steps: (1)computes  $h = H_2(m, P_{ID})$ ; (2)computes  $\sigma = \frac{1}{r+h}P = \frac{1}{(r+h)(s+Q_{ID})}P$ ; (3)sets signature on message  $m$  as  $\sigma$ .
- *CL-Veri.* Anyone can verify the valid of the signature  $\sigma$  by:(1)computes  $h = H_2(m, P_{ID})$ ; (2)accepts the signature  $\sigma$  and returns *true* iff  $\hat{e}(\sigma, P_{ID} + hT) = g$ , otherwise returns  $\perp$  as invalid.

**Remark 1.** *The Du and Wen's short CLS scheme is significantly more efficient than previous known CLS schemes, and the size of signatures is approximate 160-bit under ECC cryptographic.*

#### 4. Cryptanalysis on Du and Wen's CLS scheme

In[11], the authors stated that the proposed CLS scheme is secure against a type-I adversary  $\mathcal{A}_I$  in the random oracle model, who does not have access to master key, but may replace public keys at will. They also claimed that the scheme is existential unforgeable against a type-II adversary  $\mathcal{A}_{II}$ , who does have access to master-key, but cannot replace public keys of users. However, their certificateless signature scheme is in fact neither insecure against a type-I adversary  $\mathcal{A}_I$ , nor against universal forgery attack on any signer and message. Precisely, not only did an adversary  $\mathcal{A}_I$  obtain a user's private key by replacing public attack, and then forge valid signatures on any messages for that signer without knowledge of the signer's partial private key, but also he could make a universal forgery for any message  $m$  and signer  $ID$  without any enough attack ability. The details of the attack are shown as following two scenarios.

##### 4.1. Public replacement attack

By this attack, Type-I adversary  $\mathcal{A}_I$  can obtain a user partial key by deploying the replacement of the public key of a signer. He does:

- picks  $x'_{ID} \in Z_q$ , and replaces user's public key with  $P'_{ID} = x'_{ID}P$  with  $\mathcal{O}_{UPK}$  request;
- requests a signature query  $\mathcal{O}_{CLS}$  for  $CL\text{-}Sign(m, ID, P'_{ID})$  and gets the signature  $\sigma'$ , where  $\sigma' = \frac{1}{x'_{ID}+h'}D_{ID}$ , where  $h' = H_2(m, P'_{ID})$ .

It is possible because the type-I adversary can get the access to signature oracle( $\mathcal{O}_{CLS}$ ) and user public keys replacement oracle( $\mathcal{O}_{PKR}$ ) of his choice in the security model of type-I game.



- Adversary  $\mathcal{A}_I$  computes the user's partial private key  $D_{ID}$  from  $\sigma'$  by:  

$$D_{ID} = (x'_{ID} + h)\sigma', \text{ where } x'_{ID} \text{ is selected by } \mathcal{A}_I.$$

Upon obtaining the user's partial private key, the adversary can produce the user's certificateless signatures on any message with respect to any public keys of his choice.

The weakness of the Du and Wen's short CLS scheme against the type-I adversary is due to the fact that the adapted signature scheme is deterministic, i.e., we can get the same signature  $\sigma$  if we make twice signature queries for the same message  $m$  and signer. At the same time, signature  $\sigma$  is linear and proportioned to user's partial key  $D_{ID}$ , so we can attain the partial key by public key replacing attack.

**Remark 2.** *Several certificateless public key schemes are vulnerable to replace public key attacks[12]. i.e., attacker  $\mathcal{A}$  can modify the public key  $\langle x_{ID}, P_{ID} \rangle = \langle x_{ID}P, x_{ID}P_{pub} \rangle$  used in Al-Riyami and Paterson's scheme[1] into  $\langle x_{ID}tP, x_{ID}tP_{pub} \rangle$ . Obviously, it satisfies the equality  $\hat{e}(x_{ID}, P_{pub}) = \hat{e}(P_{ID}, P)$ , then  $\mathcal{A}$  can forge a valid signature via an existential signature.*

#### 4.2. Universal existential unforgeable attack

Any forger can forge a new signature  $\sigma'$  successfully with a universal manner. Especially, forgery  $\mathcal{F}$  can perform a universal forgery for any user  $ID'$  on message  $m'$  by the following steps:

- requests a public key oracle  $\mathcal{O}_{UPK}$  query for  $P'_{ID}$  of identity  $ID'$ ;
- computes  $h = H_2(m', P'_{ID})$ ;
- computes  $Q'_{ID} = H_1(ID')$ , and  $V = (P'_{ID} + hT)^{-1}$  where  $T = P_{pub} + Q'_{ID}P$ ;
- sets the forged signature  $\sigma' = V$ .

For the bilinear map  $\hat{e}$  is non-degenerate and non-trivial, it easy sees that  $\hat{e}(\sigma', P'_{ID} + hT) = \hat{e}(P, P) = g$ . The weakness of this CLS scheme is due to the fact that: (1)The CL-Sign algorithm involved to user's secret key is unique, and signature verification equation is trivial; (2)The CL-Sign algorithm is deterministic and non-randomized. If the CLS is a certain combined operation of two deterministic standard signatures, one can query the partial private key  $D_{ID}$  of the user by removing the signature part involved to the user secret key  $x_{ID}$  from the  $\sigma$  using inverse operation[7]. For that both

the signature part involved to the user secret key on the same message and the partial private key of the user are unique for this CLS scheme.

To improve the security of CLS scheme, if the partial private key  $D_{ID}$  of a user is obtained the deterministic algorithm, the randomized signature should be adapted for the user secret key in CLS scheme. If randomized standard signature scheme be provided, the signature size will be increased. It is an interesting open issue to discuss the compact signature size and CL-Sign algorithms with an optimized balance.

## 5. Conclusion

Short certificateless signature is a useful cryptographic tool in the systems or devices with low bandwidth channel and/or low computation power, where it can prevent the malicious behavior from malicious-but-passive KGC. Recently, Du and Wen proposed an efficient CLS scheme with shorter signature size and higher computation efficiency without MapToPoint map function. In this paper we showed that the Du and Wen's CLS scheme is universally forgeable for any third party and cannot resist on the type-I adversary under replacing public keys attacks. This result shows that it is possible insecure if it combines a standard deterministic signature scheme into a certificateless ones.

## Acknowledgment

The authors grateful thank the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China under Grant 60773175 and 60973134, the Foundation of National Laboratory for Modern Communications under Grant 9140C1108020906, and the Support of JSPS Postdoctoral Research Project of Japan.

## References

- [1] S. Al-Riyami, K. Paterson, *Certificateless public key cryptography*, in: C.S. Laih (Ed.), ASIACRYPT 2003, LNCS 2894, Springer-Verlag, Taipei, 2003, pp.452-473.
- [2] X. Huang, W. Susilo, Y. Mu, F. Zhang, *On the security of a certificateless signature scheme*, in: Y.G. Desmedt et al. (Eds.), CANS 2005, LNCS 3810, Springer-Verlag, Brisbane, Australia, 2005, pp.13-25.

- [3] D. H. Yum and P. J. Lee. *Generic construction of certificateless signature*. In: 9th Australasian Conference Information Security and Privacy(ACISP'04), LNCS 3108, Springer-Verlag, 2004, pp.200-211.
- [4] B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng, *Key replacement attack against a generic construction of certificateless signature*. In: L. Batten and R. Safavi-Naini (Eds.): ACISP 2006, LNCS 4058, 2006, pp.235-246.
- [5] D. Boneh, B. Lynn, H. Shacham, *Short signatures from the Weil pairing*. ASIACRYPT'01, LNCS 2448, 2001, pp.514-532.
- [6] X. Huang, Y. Mu, W. Susilo, D. Wong, W. Wu, *Certificateless signature revisited*, in: J. Pieprzyk, H. Ghodosi, E. Dawson (Eds.), ACISP 2007, LNCS 4586, 2007, pp.308-322.
- [7] K. A. Shim, *Breaking the short certificateless signature scheme*. Information Sciences, 2009,179(3):303-306.
- [8] D. Boneh, X. Boyen, *Short signatures without random oracles and the SDH assumption in bilinear Groups*. Journal of Cryptology, 2008, 21(2): pp.149-177.
- [9] H. Shao, X. Zhang, F. Shao, *Cryptanalysis of short signature scheme without random oracles assumption*, 2009 International Conference on Computational Intelligence and Security(CIS'09), 2009, pp.414-417.
- [10] R. Tso, X. Yi, X. Huang, *Efficient and short certificateless signatures secure against realistic adversaries*. Journal of Supercomputing, April, 2010. DOI:10.1007/s11227-010-0427-x.
- [11] H. Du, Q. Wen, *Efficient and provably-secure certificateless short signature scheme from bilinear pairings*. Computer Standards & Interfaces, 2009,31(2):390-394.
- [12] C. Ma, and J. Ao, *Certificateless group oriented signature secure against key replacement attack*. International Journal of Network Security, 12,(1),2011:1-6.