International Journal of Security and Its Applications
Vol.9, No.1 (2015), pp.147-154
http://dx.doi.org/10.14257/ijsia.2015.9.1.16

# Comment on New Construction of Efficient Certificateless Aggregate Signatures

Yulei Zhang and Caifen Wang

*College of Computer Science and Engineering, Northwest Normal University*
*Lanzhou, 730070, P.R. China*
*zhangyl@nwnu.edu.cn*
*wangcf@nwnu.edu.cn (corresponding author)*

***Abstract***

*Aggregate signature can combine n signatures on n messages from n users into single signature, and the verifier should be convinced by the aggregate signature that n users indeed sign n messages. Since aggregate signature can greatly reduce the length of total signature and the cost of verification, it is widely used in environments with low bandwidth communication, low storage and low computability. Recently, Liu et al. [H Liu, S Wang, M Liang and Y Chen, "New Construction of Efficient Certificateless Aggregate Signatures", International Journal of Security and Its Applications Vol.8, No.1 (2014), pp. 411-422] proposed an efficient certificateless aggregate signature scheme which is proven existentially unforgeable against adaptive chosen-message attacks. Unfortunately, Liu et al.'s new certificateless signature scheme is insecure. In this paper, giving concrete and simple attacks, we demonstrate that type II adversary key generation center can make ordinary-passive attack and malicious-active attack to forge legal certificateless signatures and certificateless aggregate signatures on any messages. Furthermore, we analyze possible reasons why key generation center succeeds in ordinary-passive attack and malicious-active attack.*

***Keywords:*** *Certificateless cryptography; Aggregate signature; key generation center attack*

## 1. Introduction

The aggregate signature can aggregate *n* signatures on *n* messages from *n* distinct users to a single signature. The notion of aggregate signature was introduced by Boneh *et al.* [1] firstly in Eurocrypt 2003. The aggregate signature can convince the verifier that *n* users indeed sign *n* original messages. Aggregate signature is useful to reduce bandwidth and storage. Hence, in some resource constrained environment such as Vehicular Ad hoc Networks [2] and Body Area Networks [3], it can be applied to reduce the cost of computation and communication.

In Asiacrypt 2003, Al-Riyami and Paterson [4] proposed the notion of certificateless public key cryptography and certificateless signature, which can simplify management problems of complicated certificate in traditional public key cryptography and overcome inherent problem of escrow key in identity-based public key cryptography. In the certificateless cryptography environment, to satisfy the above application, Castro and Dahab [5] presented the concept of certificateless aggregate signature (CLAS) and Gong *et al.*, [6] redefined the security model of certificateless aggregate signature. Later, Zhang *et al.*, [7] proposed a new certificateless aggregate signature scheme and perfected Gong's certificateless aggregate signature security model, but the efficiency of their schemes [6-7] is lower. Afterwards, researchers proposed several schemes with less pairings or fixed signature's length [8-11]. Recently, Liu *et al.*, [12] proposed another new construction of efficient certificateless aggregate signature scheme which only spent 3 pairings and is proven existentially unforgeable against adaptive chosen-message attacks. Unfortunately,

ISSN: 1738-9976 IJSIA

Liu *et al.'s* new certificateless aggregate signature scheme is insecure since type II adversary can forge signatures.

In this paper, we comment on Liu *et al.'s* new certificateless aggregate signature scheme. Giving concrete and simple attacks, we demonstrate that Liu et al.'s scheme is insecure. Since type II adversary key generation center (KGC) can make ordinary-passive attack and malicious-active attack, and impersonate any user to forge legal certificateless signature and certificateless aggregate signature on any messages. Furthermore, we analyze possible reasons why KGC can make ordinary-passive attack and malicious-active attack. We show that $r_iP_T$ and $x_iS$ are fixed in Liu *et al.*'s original scheme although KGC does not know user's secrete value $x_i$ and random value $r_i$, which are embedded in the $P_i=x_iP$ and $R_i=r_iP$. If KGC gets $r_iP_T$ and $x_iS$, he can make arbitrary attacks.

The rest of the paper is organized as follows. Section 2 describes background concepts of bilinear pairing, computational Diffe-Hellman problem and the model of certificateless aggregate signature. Section 3 reviews Liu *et al.'s* scheme. Section 4 analyzes scheme's security and shows two type attacks. Section 5 shows the possible reasons and flaws why KGC can realize those attacks. Finally, Section 6 concludes the paper.

## 2. Preliminaries

This section introduces basic concepts of bilinear pairing, computational Diffe-Hellman problem, and the formal model of certificateless aggregate signature.

### 2.1. Bilinear Pairing

Let $G_1$ be a cyclic additive group with prime order $p$, $G_2$ be a cyclic multiplicative group with the same order and $P$ be a generator of $G_1$. Let $e$: $G_1\times G_1\rightarrow G_2$ be a bilinear mapping with the following properties:

(1) The map $e$ is bilinear: for all $P, Q\in G_1$, $a, b\in Z_q^*$, $e(aP,bQ)=e(P,Q)^{ab}$.

(2) The map $e$ is non-degenerate: there exists $P\in G_1$, $Q\in G_1$, so that $e(P,Q)\neq 1$.

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

We say that $(G_1,G_2)$ are bilinear groups if there exists the bilinear mapping $e$:$G_1\times G_1\rightarrow G_2$ as above. Bilinear pairings can be constructed from the *Weil* pairing or the *Tate* pairing.

### 2.2. Computational Diffie–Hellman Problem

Given $(P, aP, bP)$, for unknown $a, b\in Z_q^*$, compute $abP$.

The success probability of any probabilistic polynomial-time algorithm $\boldsymbol{A}$ solving the Computational Diffie–Hellman problem in $G_1$ is defined to be $Succ_{A,G_1}^{CDH} = \Pr[(P, aP, bP) = abP]$.

### 2.3. Component of Certificateless Aggregate Signature Scheme

In this subsection, we review the definition of certificateless aggregation signature given in [6-7]. In general, certificateless aggregate signature scheme considers seven algorithms as following:

- *Setup*: It is a probabilistic polynomial algorithm which takes a security parameter $k$ as input and returns a master secret key $s$, master public key $P_{pub}$ and a list of system parameters *params*.
- *PartialKey-Gen*: It is a deterministic polynomial algorithm which takes user's identity $ID_i$, *params* and master key $s$ as input and returns the user's partial private key $D_i$.

 

- *UserKey-Gen*: It is a probabilistic polynomial algorithm which takes $ID_i$, *params*, $x_i$ and $D_i$ as input and returns the user's secret/public key pair $(SK_i=(x_i,D_i),PK_i)$ after user selects a random value $x_i$ as his secret value.
- *Partial-Sign*: It is a probabilistic polynomial algorithm which takes *params*, $ID_i$, $D_i$ and $x_i$ as input and returns a signature $\sigma_i$ on the message $m_i$.
- *Partial-Verify*: It is a deterministic polynomial algorithm which takes *params*, $ID_i$, the user's public key $PK_i$ and $\sigma_i$ on messages $m_i$ as input, then outputs either *accept* or *reject*.
- *Aggregate-Sign*: It is a deterministic polynomial algorithm which takes *n* user's identity $ID_i$, the corresponding public key $PK_i$ and signature $\sigma_i$ on message $m_i$ as input and returns an aggregate signature $\sigma$ on messages $\{m_1,\ldots,m_n\}$.
- *Aggregate-Verify*: It is deterministic polynomial algorithm which takes *params*, the identities set $\{ID_1,\ldots,ID_n\}$, the corresponding public key set $\{PK_1,\ldots,PK_n\}$ and an aggregate signature $\sigma$ on messages $\{m_1,\ldots,m_n\}$ as input, then outputs either *accept* or *reject*.

In above algorithms, certificateless signature scheme includes previous five algorithms and certificateless aggregation signature scheme includes all of the algorithms. In certificateless signature scheme and certificateless aggregation signature scheme, two types of adversaries are considered: Type I adversary $A_{\mathrm{I}}$ and Type II adversary $A_{\mathrm{II}}$. The adversary $A_{\mathrm{I}}$ represents a normal attacker. He cannot access to the master secret key *s* to get the partial private key $D_i$, but he has ability to replace any user's public key with a value of his choice. The adversary $A_{\mathrm{II}}$ represents a KGC. KGC has access to the master key and knows all partial private key $D_i$, but does not replace the target user's public key.

### 2.4. Adversaries Model of Certificateless Aggregate Signature Scheme

We define game I for $A_{\mathrm{I}}$ and game II for $A_{\mathrm{II}}$.

**Game I**: Let *B* be the game challenger and *k* be a security parameter. Challenger *B* executes algorithm *Setup* and generates the system parameters *params* and a master secret key *s*. Challenger *B* holds *s* and sends *params* to $A_{\mathrm{I}}$. During the simulating between *B* and $A_{\mathrm{I}}$, the following oracles can be accessed.

① *Hash-Query*: $A_{\mathrm{I}}$ can access the rights of all of hash query in the certificateless aggregation signature scheme and achieve the corresponding hash's values.

② *PartialKey-Query*: *B* executes the algorithm *PartialKey-Gen* to generate the partial private key $D_i$ and returns it to $A_{\mathrm{I}}$ when $A_{\mathrm{I}}$ queries the partial private key of target identity $ID_i$.

③ *PublicKey-Query*: *B* executes the algorithm *UserKey-Gen* to generate the public key $PK_i$ and returns it to $A_{\mathrm{I}}$ when $A_{\mathrm{I}}$ queries the public key of target identity $ID_i$.

④ *SecretValue-Query*: *B* executes the algorithm *UserKey-Gen* to generate the secrete value $x_i$ and returns it to $A_{\mathrm{I}}$ when $A_{\mathrm{I}}$ queries the secrete values of target identity $ID_i$. If user's public key has been replaced $A_{\mathrm{I}}$ returns the value $\perp$.

⑤ *PublicKeyReplace-Query*: $A_{\mathrm{I}}$ is able to replace the public key $PK_i'$ with his selected public key $P_i$.

⑥ *PartSign-Query:* $A_{\mathrm{I}}$ can get access to the signature $\sigma_i$ of public key $PK_i$ and the corresponding target identity $ID_i$.

Based on the above queries, $A_{\mathrm{I}}$ outputs the aggregate signature $\sigma^*$ on messages $M^* = \{m_1^*,\ldots, m_n^*\}$ under the identities $ID^* = \{ID_1^*,\ldots, ID_n^*\}$ and the corresponding public keys $P^* = \{PK_1^*,\ldots,PK_n^*\}$.

We say that $A_{\mathrm{I}}$ wins Game I, if the following conditions are satisfied.

- $\sigma^*$ is a valid signature and can be verified by algorithm *Aggregate-Verify*.

- At least one identity $ID_i^*$ has not been submitted to oracle *PartialKey-Query*. Without loss of generality, we let $i$=1.
- The oracle *PartSign-Query* has never been queried with $(ID_1^*, P_1^*, m_1^*)$ by $A_{\mathrm{I}}$.

**Definition 1**. A certificateless aggregate signature scheme is said to be Type-I secure and is proven existentially unforgeable against adaptive chosen-message and chosen-identity attacks, if there is no probabilistic polynomial-time adversary $A_{\mathrm{I}}$ which wins Game I with non-negligible advantage.

**Game II**: Let $B$ be the game challenger. Challenger $B$ executes algorithm *Setup* and generates the system parameters *params* and a master secret key $s$. Challenger $B$ sends $s$ and *params* to $A_{\mathrm{II}}$. During the simulating between $B$ and $A_{\mathrm{II}}$, the following oracles can be accessed.

① *Hash-Query*, *PublicKey-Query*, *SecretValue-Query* and *PartSign-Query* are the same as queries of Game I.

② Since $A_{\mathrm{II}}$ can get the master secret key to compute the partial private key, this game does take no account of oracle *PartialKey-Query*. Meanwhile, $A_{\mathrm{II}}$ is forbidden to execute the *PublicKeyReplace-Query*.

Based on the above queries, $A_{\mathrm{II}}$ outputs the aggregate signature $\sigma^*$ on messages $M^* = \{m_1^*,..., m_n^*\}$ under the identities $ID^* = \{ID_1^*,..., ID_n^*\}$ and the corresponding public keys $P^* = \{P_1^*,..., P_n^*\}$.

We say that $A_{\mathrm{II}}$ wins Game II, if the following conditions are satisfied.

- $\sigma^*$ is a valid signature and can be verified by algorithm *Aggregate-Verify*.
- At least one identity $ID_i^*$ has not been submitted to oracle *SecretValue-Query*. Without loss of generality, we let $i$=1.
- The oracle *PartSign-Query* has never been queried with $(ID_1^*, P_1^*, m_1^*)$ by $A_{\mathrm{II}}$

**Definition 2**. A certificateless aggregate signature scheme is said to be Type-II secure and is proven existentially unforgeable against adaptive chosen-message and chosen-identity attacks, if there is no probabilistic polynomial-time adversary $A_{\mathrm{II}}$ which wins Game II with non-negligible advantage.

More details of the security of certificateless aggregate signature scheme can be found in [6-7].

## 3. Review of Liu *et al.'s* Certificateless Aggregate Signature Scheme

In this section, we briefly review Liu *et al.'s* certificateless aggregate signature scheme [12]. The scheme consists of following algorithms: *Setup*, *PartialKey-Gen*, *UserKey-Gen*, *Partial-Sign*, *Aggregate-Sign and Aggregate-Verify*. The detail of algorithms is described as follows.

*Setup*: Given a security parameter $l$, the algorithm works as follows by KGC.

(1) Generates a cyclic additive group $G_1$ and a cyclic multiplicative group $G_2$ with prime order $q$, different generators $P$ and $S$ in $G_1$ and define bilinear pairing $e$: $G_1 \times G_1 \rightarrow G_2$.

(2) Selects a master key $\lambda \in Z_q^*$ and sets $P_T = \lambda P$.

(3) Chooses cryptographic hash functions $H_0: \{0,1\}^* \rightarrow G_1$ and $H_1, H_2: \{0,1\}^* \rightarrow Z_q^*$. The master key is $\lambda$. The system parameters are $\{G_1, G_2, e, q, P, S, P_T, H_0, H_1, H_2\}$.

*Partia-PrivateKey-Gen*: Given a user's identity $ID_i \in \{0,1\}^*$, KGC first computes $Q_i = H_0(ID_i)$, then he sets this user's partial key $D_i = \lambda Q_i$ and transmits it to user secretly.

*UserKey-Gen*: The user $ID_i$ selects a secret value $x_i \in Z_q^*$ as his secret value and computes $P_i = x_i P$ as his public key.

*Partial-Sign*: Given user's secret keys $(x_i, D_i)$, identity $ID_i$, the corresponding public key $P_i$ and a message $m_i \in \{0,1\}^*$, user performs the following steps:

 

(1) Chooses $r_i \in Z_q^*$ and computes $R_i = r_i P$.

(2) Computes $w_i=H_1(m_i,ID_i,P_i,R_i)$ and $t_i=H_2(m_i,ID_i,P_i,R_i)$.

(3) Computes $V_i = D_i + w_i x_i S + t_i r_i P_T$, outputs $(R_i,U_i)$ as the signature on $m_i$.

*Aggregate-Sign*: An aggregator aggregates a collection of individual signatures. For an aggregating set of $n$ users with identities $\{ID_1,\ldots,ID_n\}$, the corresponding public keys $\{P_1,\ldots,P_n\}$, and message-signature pairs$\{(m_1,R_1,U_1),\ldots,(m_n,R_n,U_n)\}$, the aggregator computes $U = \sum_{i=1}^{n} U_i$ and outputs $(R_1,\ldots,R_n,U)$ as the aggregate signature.

*Aggregate-Verify*: The verifier verifies the value of the aggregate signature $(R_1, \ldots R_n, U)$ to determine whether the aggregate signature signed by $n$ users, and verifier performs the following steps:

(1) Computes $w_i=H_1(m_i,ID_i,P_i,R_i)$ , $t_i=H_2(m_i,ID_i,P_i,R_i)$ and $Q_i=H_0(ID_i)$ for all $i \in [1, n]$.

(2) Checks whether the equation $e(U, P) \stackrel{?}{=} e(\sum_{i=1}^{n} (Q_i + t_i R_i), P_T) e(S, \sum_{i=1}^{n} w_i P_i)$ holds or not. If it holds, accepts the signature else rejects it.

## 4. Security-Analysis of Liu *et al's* Scheme

Liu *et al*. [9] claimed their certificateless aggregate signature scheme is secure against Tpye II adversary $A_{II}$. Unfortunately, the scheme is insecure. $A_{II}$ can forge partial signature $(R_i,U_i)$ on message $m_i$ under the identity $ID_i$ and the corresponding public key $P_i$, he also can aggregate the individually imitative signature $(R_i,U_i)$ to generate the final aggregate signature$(R_1,\ldots,R_n,U)$ which is verified by *aggregate-verify* algorithm. In this section, we give concrete and simple attacks to show Liu et al.'s scheme is insecurity. The following will show how type II adversary KGC makes ordinary-passive attack and malicious-active attack.

- **KGC Attack I**

Since $\lambda$ is KGC's master secret key and the computing equation $r_i P_T = r_i \lambda P = \lambda R_i$ is correct, KGC can computer $\lambda R_i$ and get $r_i P_T$. Meanwhile, KGC knows users' partial secret key $D_i$, he can compute $x_i S = w_i^{-1}(V^* - D_i - t_i r_i P_T)$ after capturing signature $(R^*,V^*)$ on $m^*$, $w_i=H_1(m^*,ID_i,P_i,R^*)$ and $t_i=H_2(m^*,ID_i,P_i,R^*)$. Since KGC knows constant $x_i S$, he can forge user's certificateless signature on any messages and aggregate a collection of forged signatures to form forged certificateless aggregate signatures. This attack shows that KGC is an ordinary-passive attacker in Liu et al.'s scheme. The detail of attack is described as follows.

*Intercept partial-signature*. During a concrete application, the user $U_i$ whose identity is $ID_i$ with corresponding public key $P_i$ signs the message $m^*$ and computes the partial-signature $(R^*,V^*)$. Then, KGC intercepts this signature by some cryptanalysis methods.

*Compute Fixed Value*. KGC performs the following steps to compute fixed value $x_i S$.

(1) Computes $w_i=H_1(m^*,ID_i,P_i,R^*)$ and $t_i=H_2(m^*,ID_i,P_i,R^*)$, then easily computes $w_i^{-1} \in Z_q^*$.

(2) Computes $r^* P_T = r^* \lambda P = \lambda R^*$.

(3) Computes $x_i S = w_i^{-1} (V^* - D_i - t_i \lambda R^*)$, then gets fixed value $x_i S$.

*Forge Partial-Signature*. KGC performs the following steps to forge CLS signature $(R_i', V_i')$ on new message $m'$.

(1) Chooses $R_i' \in G_1$ randomly and computes $\lambda R_i'$.

(2) Computes $w_i=H_1(m',ID_i,P_i,R_i')$ and $t_i=H_2(m',ID_i,P_i,R_i')$.

(3) Computes $V_i' = D_i + t_i \lambda R_i' + w_i x_i S$ , outputs $(R_i', V_i')$ as the signature on $m'$.

The forged signature $(R_i', V_i')$ is valid because the following verification equation is correct.

$$e(V_i', P) = e(D_i + t_i \lambda R_i' + w_i x_i S, P) = e(D_i, P) e(t_i \lambda R_i', P) e(w_i x_i S, P)$$

$$= e(Q_i, P_T)e(t_i R_i^{'}, P_T)e(S, w_i P_i) = e(Q_i + t_i R_i^{'}, P_T)e(S, w_i P_i)$$

*Forge Aggregate-Signature*. Using above method, KGC can know $x_iS$ to all users ($i \in [1, n]$). So KGC can forge any user's partial signature on any messages and aggregate a collection of forged partial signatures.

KGC computes $V^{**} = \sum_{i=1}^{n} V_i'$ and outputs $(R_1^{'},\ldots,R_n^{'},V^{**})$ as the forged aggregate signature. Obviously, for $i \in [1,n]$, $Q_i=H_0(ID_i)$, $w_i=H_1(m^{'},ID_i,P_i,R_i^{'})$ and $t_i=H_2(m^{'},ID_i,P_i,R_i^{'})$, the forged aggregate signature is valid since the following equation holds:

$$e(V^{**}, P) = e(\sum_{i=1}^{n}(t_i R_i' + Q_i), P_T)e(S, \sum_{i=1}^{n} w_i P_i)$$

- **KGC Attack II**

To certificateless signature and certificateless aggregate signature, KGC is the creator of signature system. The system parameters include generators and hash functions. So KGC can choose normal generator $P$ and particular generator $S=tP$ to compute a constant $x_iS=x_itP= tx_iP=tP_i$. Because KGC knows $D_i$, $x_iS$ and $r_iP_T=\lambda R_i$, so he can directly forge any user's certificateless signature on any messages but need not intercept signatures $(R^*,V^*)$ of users. This attack shows KGC is a malicious-active attacker. The detail of attack is described as follows.

*Generate Particular Generators and Parameters*. Selects $t \in Z_q^*$, and sets generators as $P$ and $S=tp$, then particular generator $S$ is created.

*Compute Fixed Value*. KGC computes fixed value $x_iS=x_itP= tx_iP=tP_i$.

*Forge Partial-Signature*. KGC performs the following steps to forge CLS signature$(R^*_i, V^*_i)$.

(1) Chooses $R^*_i \in G_1$ randomly and computes $\lambda R^*_i$

(2) Computes $w_i=H_1(m^*,ID_i,P_i,R^*_i)$ and $t_i=H_2(m^*,ID_i,P_i,R^*_i)$.

(3) Computes $V^*_i =D_i+t_i\lambda R^*_i+w_ix_iS$ directly, outputs $(R^*_i,V^*_i)$ as the signature on $m^*$. Obviously, signature$(R^*_i,V^*_i)$ is valid.

*Forge Aggregate-Signature*. This step is same as one of forging aggregate-signature in above KGC attack I, so we omit it. Then, KGC can easily forge aggregate signature, meanwhile, the forged aggregate signature is also valid.

## 5. Analysis of Possible Reasons

According to our analysis, the security of Liu et al's scheme depends on three secret information which are partial private key $D_i$, secrete values $x_i$ and random value $r_i$. In general, user does not know other user's keys $(D_i,x_i)$, KGC does not compute $x_i$ and $r_i$ from $P_i= x_iP$ and $R_i= r_iP$ otherwise Discrete Logarithm problem should be solved. But KGC can purposely select generators using his choice since KGC is creator of signature system. What's more, some important facts are ignored which $x_iS$ and $r_iP_T$ are fixed and $r_iP_T$ could be computed easily.

On the one hand, after KGC captured signature $(R^*,V^*)$ on $m^*$, he can compute $r_iP_T$ by computing equation $r_iP_T =r_i\lambda P=\lambda R^*$ and $x_iS$ by computing equation $x_iS =w_i^{-1}(V^*-D_i - t_ir_iP_T)$. So KGC can forge any user's partial signature on any messages.

On the other hand, KGC is the system's creator, he can select particular generator $S$ using his choice $S=tP$. Then, $x_iS$ is fixed because of computing equation $x_iS=x_itP= tx_iP=tP_i$. So, after KGC chooses $R_i \in G_1$ randomly, he also can forge any user's partial signature on any messages.

Although KGC does not know secret values $x_i$ and $r_i$, he still can compute $x_iS$ and $r_iP_T$ to forge any user's partial signature on any messages using $D_i$, $x_iS$ and $r_iP_T$. Then he combines the individually partial signature to form a valid certificateless aggregate signature which is verified by algorithm *Aggregate-Verify*.

 

## 6. Conclusions

Recently, Liu *et al.* proposed a new construction of efficient certificateless aggregate signature scheme. They claimed their scheme is provably secure in the random oracle model under the Computational Diffie–Hellman assumption. Unfortunately, Liu et al.'s scheme is insecure. In this paper, we analyze scheme's security and demonstrate two kinds of concrete attacks. Since type II adversary KGC can compute fixed value $r_iP_T$ and $x_iS$, he can make ordinary-passive attack and malicious-active attack, and impersonate any user to forge legal certificateless signature and certificateless aggregate signature on any messages. We will propose an improved scheme in the future to overcome the security weakness of Liu et al's scheme.

## Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grants 61163038, 61262056, 61262057, the Higher Educational Scientific Research Foundation of Gansu Province of China under Grant 2013A-014, the Young Teachers' Scientific Research Ability Promotion Program of Northwest Normal University under Grant NWNU-LKQN-12-32.

## References

[1] D. Boneh, C. Gentry, B. Lynn and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps", in: EUROCRYPT'2003, LNCS 3027, (**2003**), pp.416–432.
[2] Q. Wu, J. Domingo and U. Gonzalez, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications", IEEE Transactions on Vehicular Technology, vol. 59, no. 2, (**2010**), pp. 559-573.
[3] B. Latré, B. Braem, I. Moerman, C. Blondia and P. Demeester, "A Survey on wireless body area networks", Wireless Networks, vol. 7, no. 1, (**2011**), pp. 1-18.
[4] S. S. Al-Riyami and K. Paterson, "Certificateless Public Key Cryptography", in: ASIACRYPT'2003, LNCS 2894, (**2003**), pp. 452-473.
[5] R. Castro and R. Dahab, "Efficient certificateless signatures suitable for aggregation", Cryptology ePrint Archive, Available online: http://eprint.iacr.org/2007/454.
[6] Z. Gong, Y. Long, X. Hong and K. Chen, "Two certificateless aggregate signatures from bilinear maps", in: IEEE SNPD 2007, vol. 3, (**2007**), pp. 188-193.
[7] L. Zhang and F. Zhang, "Security model for certificateless aggregate signature schemes", in: International Conference on Computational Intelligence and Security (CIS'08), 2008, vol. 2, (**2008**), pp. 364-368.
[8] L. Zhang, B. Qin, Q. Wu and F. Zhang, "Efficient many-to-one authentication with certificateless aggregate signatures", Computer Networks, vol. 54, no. 14, (**2010**), pp. 2482-2491.
[9] H. Du, M. Huang and Q. Wen, "Efficient and Provably-Secure Certificateless Aggregate Signature Scheme", Acta Electronica Sinica, vol. 54, no. 14, (**2013**), pp. 2482-2491.
[10] H. Xiong, Z. Guan, Z. Chen and F. Li, "An Efficeent Certificateless Aggregate Signature with Const Pairing Computations", Information Sciences, vol. 219, no. 10, (**2013**), pp. 225-235.
[11] Y. Ming, X Zhao and Y. Wang, "Certificateless Aggregate Signature Scheme, Journal of University of Electronic Science and Technology of China, vol. 43, no. 2, (**2014**), pp. 188-193.
[12] H. Liu, S. Wang, M. Liang and Y. Chen, "New Construction of Efficient Certificateless Aggregate Signatures", International Journal of Security and Its Applications, vol. 8, no. 1, (**2014**), pp. 411-422.

## Authors

**Yulei Zhang,** he received the B.S and M.S. degree in Computer Software Theory from Northwest Normal University in 2001 and in 2010 respectively. From 2001, he works at Northwest Normal University. Now he is an assistant professor in computer science at Northwest Normal University and also a Ph.D. candidate at Northwest Normal University. His research includes information

security, cryptography and provable security. E-mail:zhangyl@nwnu.edu.cn.

**Caifen Wang,** he received the Ph.D. degree in cryptography from Xidian University in 2003. From 2003, she is a professor in computer science at Northwest Normal University. Her research interests include network security, cryptographic protocol and electronic commerce.

E-mail:wangcf@nwnu.edu.cn(corresponding author)

 