International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:7, No:11, 2013

Efficient Aggregate Signature Algorithm and Its Application in MANET

Daxing Wang, Jikai Teng

***Abstract*—**An aggregate signature scheme can aggregate n signatures on n distinct messages from n distinct signers into a single signature. Thus, n verification equations can be reduced to one. So the aggregate signature adapts to Mobile Ad hoc Network (MANET). In this paper, we propose an efficient ID-based aggregate signature scheme with constant pairing computations. Compared with the existing ID-based aggregate signature scheme, this scheme greatly improves the efficiency of signature communication and verification. In addition, in this work, we apply our ID-based aggregate sig- nature to authenticated routing protocol to present a secure routing scheme. Our scheme not only provides sound authentication and a secure routing protocol in ad hoc networks, but also meets the nature of MANET.

***Keywords*—**Identity-based cryptography, Aggregate signature, Bilinear pairings, Authenticated routing scheme.

I. Introduction

IN the past decades, mobile communications have experienced an explosive growth. In particular, one area of mobile communication, the Mobile Ad hoc Networks (MANET) have attracted significant attention due to its multiple applications. MANET is a network consisting of mobile nodes which communicate with each other through wireless medium without any fixed infrastructure such as access points or base stations. Each node in ad hoc networks carries out networking functions such as packet forwarding, routing and network management, while only dedicated nodes like routers support networking functions in the wired network. Due to these characteristics, ad hoc network is especially exposed to security threats [1]. Therefore, security in ad hoc networks is an essential component for basic networking functions.

The Ad hoc On-demand Distance Vector routing protocol (AODV) [2] is one of the more popular routing algorithms for MANETs and mesh networks. Unfortunately, providing a secure and trustworthy version of AODV has been elusive. Secure routing in ad hoc networks has become an increasingly important topic, and many routing protocols have been proposed to secure ad hoc networks under different attack models. A brief summary of several notable works follows. Hu et al. proposed the Secure Efficient Ad hoc Distance vector routing protocol (SEAD) [3], which is based on the Destination Sequenced Distance Vector (DSDV) [4] routing protocol. An on-demand secure routing protocol, known as Ariadne [5], was also proposed to secure DSR. There are two secure routing protocols proposed to address the security vulnerabilities in AODV algorithms, SAODV [6] and ARAN [7]. Zapata et al. proposed Secure AODV (SAODV) to protect routing messages exchanged in AODV. Two mechanisms were incorporated into AODV to secure routing messages: digital signatures to authenticate non mutable fields and hash chains to secure mutable fields. Sanzgiri et al. designed another secure AODV algorithm, Authenticated Routing for Ad hoc Networks (ARAN). Similar to SAODV, each node has a certificate signed by a trusted certificate authority. ARAN achieves its security through the usage of signatures on a hop-by-hop basis.

The concept of aggregate signature was introduced by Boneh et al. [8]. Idea of the aggregate signature scheme is to combine n signatures on n different messages, signed by n (possibly different) signers, and to obtain a single aggregate signature which provides the same certainty as the n initial signatures.

An approach to the construction of IBS schemes is a generic transformation that converts any standard signature (SS) schemes into IBS schemes. This approach is to use a SS scheme and simply attach a certificate containing the public key of the signer to the signature. This certification-based approach is apparently folklore. Bellare et al. [9] formalized the idea by providing a generic and secure construction of IBS schemes from any secure SS scheme. Recently, Galindo et al. [10] proposed a generic construction of IBS schemes with additional properties by extending Bellare et al.'s construc- tion. Their results contain a generic construction of IBAS schemes from SS schemes which allow constant length aggregations [11], [12]. However, the length of its resulting IBAS is linear with respect to the number of signers n because it consists of the aggregate signature from base standard signatures together with additional n public keys. Also, the technique has few applications because there is only one SS scheme which is constant length aggregations, namely, BLS short signature schemes its AS scheme [13]-[15]. In that case, the converted IBAS scheme from Galindo et al. construction based on BLS scheme requires O(n) pairing computations. In practical situations where IBS provided by multiple signers for a long period of time are verified simultaneously, the verification cost and the flexibility would be preferable to the communication cost. We note that the pairing computation is the most time consuming in pairing based cryptosystems. Although there have been many works discussing the complexity of pairings and how to speed up the pairing computation, the computation

Daxing Wang is with the School of Mathematical Sciences, Chuzhou University, Anhui, China (e-mail: starleewipm@126.com).

Jikai Teng is with Institute of Information Engineering, State Key Laboratory of Information Security, Chinese Academy of Sciences, zhongguancun, Beijing, China (e-mail: jikai@is.iscas.ac.cn).

of the pairing still remains time consuming. Thus, to construct a practically usable scheme, the number of pairing computations should be minimized [16], [17]. In this paper, we propose an IBS scheme which allows an IBAS scheme with constant pairing computations. Our IBAS scheme requires neither an extra communication round nor a certain synchronization for aggregating randomness, while it does not achieve compactness.

We note that the pairing computation is the most time consuming in pairing based cryptosystems. Although there have been many works discussing the complexity of pairings and how to speed up the pairing computation, the computation of the pairing still remains time consuming. Thus, to construct a practically usable scheme, the number of pairing computations should be minimized. In this paper, we propose an IBS scheme which allows an IBAS scheme with constant pairing computations. Our IBAS scheme requires neither an extra communication round nor a certain synchronization for aggregating randomness, while it does not achieve compactness. The rest of the paper is organized as follows. In Section II, we provide the preliminaries about aggregate signature. In Section III, we propose a new IBAS scheme and compare with existing ones. After that, we present a security authenticated routing protocol in Section IV. Concluding remarks are given in Section V.

II. Preliminaries

A. Definitions and Computational Assumptions

Let G_1 be a cyclic additive group of order q, G_2 be a cyclic multiplicative group of order q, a map $e: G_1 \to G_2$ is said to be bilinear if it satisfies the following properties:

(1) Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and for all $a, b \in Z$.

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

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

We call such a bilinear map as admissible bilinear map. The Weil pairing and Tate pairing associate with super-singular elliptic curve can be modified to create such bilinear map.

The Computation Diffie-Hellman Problem (CDHP) is to compute abP for given $P, aP, bP \in G_1$. The Bilinear Diffie-Hellman Problem (BDHP) is to compute $e(P, P)^{abc}$ for given $P, aP, bP, cP \in G_1$ for any $a, b, c \in Z_p$.

B. Components of IBAS Schemes

An IBAS scheme

$$IBAS = (setup, Extract, Sign, Agg, AVerify) \quad (1)$$

based on the IBS scheme

$$IBS = (setup, Extract, Sign, Verify) \quad (2)$$

is specified by five polynomial time algorithms with the following functionality:

Setup. The randomized parameter generation algorithm Setup takes input 1^k, where $k \in Z$ is the security parameter and outputs some publicly known system parameters.

Extract. The randomized private key extraction algorithm Extract takes input a user identity ID and a master secret msk, and outputs a private key

$$S_{ID} \leftarrow Extract(msk, m). \quad (3)$$

Sign. The randomized signing algorithm Sign takes input a private key S_{ID} corresponding to ID and a piece of message $m \in \{0,1\}^*$, and outputs a signature

$$\sigma \leftarrow Sign(S_{ID}, m). \quad (4)$$

Verify. The randomized verification algorithm Verify takes input an identity ID, a message $m \in \{0,1\}^*$, and outputs True if

$$\text{Verify}(m, ID, \sigma) = 1, \quad (5)$$

or False otherwise.

Agg. The aggregate signature generation algorithm Agg based on the Sign algorithm takes input a sequence of sig- natures $\{\sigma_i\}_{i=1}^n$ on $\{m_i\}_{i=1}^n$ for $\{ID_i\}_{i=1}^n$ and outputs an aggregate signature

$$\sigma \leftarrow Agg(\sigma_1, \cdots, \sigma_n). \quad (6)$$

AVerify. The aggregation verification algorithm AVerify takes input a sequence of identities $(ID_i, \cdots, ID_n)$, messages $(m_1, \cdots, m_n)$ and an aggregate signature σ and outputs True if

$$\text{AVerify} \quad (m_1, \cdots, m_n, ID_1, \cdots, ID_n) = 1 \quad (7)$$

or False otherwise.

III. New Efficient ID-Based Aggregate Signature

A. Proposed ID-Based Aggregate Signature: IBAS

Now, we propose a new IBS scheme which allows constructing an efficient IBAS scheme.

Setup. Given a security parameter $k \in Z$, the algorithm works as follows:

1. Run the parameter generator on input k to generate a prime q, two groups G_1, G_2 of order q, a generator P in G_1 and an admissible pairing $e: G_1 \times G_2 \to G_2$.
2. Pick a random $s \in Z_q^*$ and set $P_{pub} = sP$.
3. Choose cryptographic hash functions

$$H_1: \{0,1\}^* \to G_1 \text{ and } H_2: \{0,1\}^* \to Z_q \quad (8)$$

The system parameters is

$$Params = (q, G_1, G_2, e, P, P_{pub}, H_1, H_2) \quad (9)$$

Extract. For a given string $ID \in \{0,1\}^*$,

1. We compute

$$Q_{ID} = H_1(ID) \in G_1 \quad (10)$$

2. Set the private key S_{ID} to be $s \cdot Q_{ID}$, where s is a master secret key.

Sign. Given a private key S_{ID} and a message $M \in \{0,1\}^*$,

1. Choose $r \in_R Z_q^*$ and compute

$$U = r \cdot P \in G_1 \quad (11)$$

2. Compute

$$h = H_2(ID, M, U) \in Z_q \quad (12)$$

and

$$V = S_{ID} + h \cdot r \cdot P_{pub} \in G_1 \quad (13)$$

The signature on m is

$$\sigma = (U, V) \quad (14)$$

Verify. Give a signature $\sigma = (U, V)$ of m for an identity ID, we perform the following algorithm.

1. Compute

$$Q_{ID} = H_1(ID) \in G_1 \quad (15)$$

and

$$h = H_2(ID, M, U) \in Z_q \quad (16)$$

3. Verify the following equation

$$e(V, P) = e(Q_{ID} + h \cdot U, P_{pub}) \quad (17)$$

holds or not. If it holds, accept the signature.

By bilinearity of the pairing e, the consistency of the scheme is easy to verify:

$$e(V, P) = e(S_{ID} + h \cdot r \cdot P_{pub}, P) = e(Q_{ID} + h \cdot U, P_{pub}) \quad (18)$$

Agg. Let $A = \{A_1, \cdots, A_n\}$ be the set of users. For an aggregating subset of users $S \subseteq A$, assign to each user an index i, ranging from $1 to k = |S|$.

1. Each user $A_i \in S$ computes (U_i, V_i) on a message $M_i \in \{0,1\}^*$.

2. Compute

$$V = \sum_{i=1}^{k} V_i \quad (19)$$

and output

$$\sigma = (U_1, \cdots, U_k, V) \quad (20)$$

as an aggregate signature.

AVerify. Given an aggregate signature $\sigma = (U_1, \cdots, U_k, V)$ as above,

1. Compute

$$Q_i = H_1(ID_i) \quad (21)$$

and

$$h_i = H_2(ID_i, m_i, U_i), \ i = 1, \cdots, k. \quad (22)$$

4. Verify the following equation

$$e(V, P) = e(\sum_{i=1}^{k} (Q_i + h_i \cdot U_i), P_{pub}) \quad (23)$$

holds or not. If it holds, accept the aggregate signature, or reject otherwise.

B. Implementation and Comparison

In this section, we report on the implementation of our SAR scheme and analysis of its performance. Then, we propose a method to improve the performance of our scheme in a typical applications scenario. We used the bilinear pairing based cryptography (PBC) to implement our SAS scheme. The key size of elliptic curve systems should be at least 160 bits and the key size of discrete logarithm systems should be at least 1024 bits. For 80-bit security, we therefore selected the Miyaji-Nakabayashi-Takano (MNT) curve with embedding degree 6 since this embedding degree is close to the optimal value, i.e., 1024/160=6.4 for this level of security. In the MNT curve with embedding degree 6, the group size of G should be at least 171 bits and the group size of G_1 should be at least 1024 bits since the security of the G_1 group is related to the security of the discrete logarithm. Therefore, we used a 175-bit MNT curve that is generated by the MNT parameter generation program in the PBC library.

We implemented and measured the performance of our SAS scheme on a notebook computer with a Pentium Dual-Core E6500 2.93 GHz CPU. The PBC library on the test machine can compute a pairing operation in 13.0 ms, an exponentiation operation of G_1 and G_2 in 1.55 ms and 18.3 ms respectively. We assume that there are 100 users who participate in the sequential aggregate signature system (indexed 1 to 100). At first, the setup algorithm takes about 0.159 seconds to generate the public parameters since it requires three exponentiations in G_1 and five exponentiations in G_2. The key generation algorithm for each user takes about 0.185 seconds since it requires six exponentiations in G_2 and one pairing. The

aggregate signing algorithm mainly consists of verifying the previous aggregate signature and adding its own signature into the aggregate signature. The time to generate an aggregate signature is proportional to the index number of the user who participates in the aggregate signing algorithm. Furthermore, this algorithm spends nearly 98 percent of its time on verifying the previous aggregate signature since it should compute $4l+14$ numbers of exponentiation in G_2 where l is the number of previous signers. For example, if a user's index is 50 in the aggregate signing algorithm, then the algorithm verifies the previous aggregate signature in 2.421 seconds, and adds its signature into the aggregate signature in 0.065 seconds.

We can improve the performance of the aggregate verification algorithm by preprocessing the exponentiations in G_2. To use the preprocessing method, users should keep the public keys of the previous users. If the set of users who participate in the aggregate signature system is not changed or changed a little (as in the routing and the certification cases), then users can preprocess the public keys of previous users after running the first aggregate signing algorithm. Additionally, we can preprocess the public parameters and precompute elements for verification in an offline mode. If the preprocessing method is used, then the time to verify an aggregate signature is reduced to 30 percent of the original time to verify.

Here, we will compare our scheme with schemes in [11]-[13] in terms of the computational efficiency (i.e., number of expensive cryptographic operations such as exponentiations or bilinear maps). The detailed comparison result is given in Table I. We use P and SM as abbreviations for pairing computation and scalar multiplications respectively. We show that the number of extra bits required to support our scheme is limited by a small constant, hence the solution is pretty feasible in resource-constrained environments such as MANET, as well as in other ad hoc settings.

TABLE I
COMPARISON OF SCHEMES

IBAS Scheme	Signature length	Sign	Averify
[11]	$(k+1)\|G_1\|$	2 *SM*	$(2k+1)P$
[12]	$(k+1)\|G_1\|$	3 *SM*	$(k+1)P+kSM$
[13]	$(k+1)\|G_1\|$	1 *SM*	$(k+1)P+kSM$
Our scheme	$(k+1)\|G_1\|$	2 *SM*	$2P+kSM$

IV. SECURITY ROUTING SCHEME

In this subsection, we present a security routing scheme with on-demand routing protocol which consists of three phase: Initialization phase, route discovery phase and route maintenance phase. The security of it is based on the ID-based aggregate signature presented above.

A. Initialization Phase

Initialization phase is performed only once prior to the formation of the Ad hoc network. In this phase, off-line server sets up system parameters and distributes each node's private key securely.

B. Route Discovery Phase

Route discovery makes a node discover dynamically a route to any other node. Route discovery has three stages: the initiator node broadcasts a route discovery packet called RDP, the intermediate nodes process the RDP message, and the target node receiving the RDP message returns a route reply message called REP to the initiator node. By verifying the aggregate signature, the target node can authenticate each intermediate node on a path and check the integrity of the message. The main advantage is that it requires less com- munication cost. Moreover, it needs no certificate chain. A route request message contains six fields:

<RDP, IPA, IPX, seq, nodelist, aggsign>.

The RDP is a packet type identifier, IPA and IPX are the node A and X's IP address respectively. The seq is incremented whenever node A issues a new RDP, the nodelist is a list of intermediate nodes on the route between initiator and target node X, and the aggsign is an aggregate signature integrated by node A and intermediate nodes. When any node receives an RDP, it processes the message according to the following steps:

Step 1. If the RDP message from node A has received recently, namely the pair (IPA, seq) for the RDP is found in this node's received request list, then discard the message and do not process it further.

Step 2. Otherwise, if this node is not the target of the RDP, then add this node's identity to the nodelist and generate its own signature on the following fields:

<RDP, IPA, IPX, seq, nodelist>,

and aggregate its signature into the aggregate signature, then rebroadcast the message.

Step 3. Otherwise, if this node is the target of RDP, then verify the aggregate signature in the RDP.

(1) If the aggregate signature is valid, then return a REP message to node A;

(2) Otherwise, discard the message and do not process it further.

A route reply message contains the following fields:

<REP, IPX, seq, nodelist, sign>.

The REP is a packet type identifier and IPX, seq, nodelist fields are set to the corresponding values from the RDP message. The seq is incremented whenever the target node issues a new RDP and the sign is a signature of node X. To describe this procedure given in Fig. 1 in details, we take an example that the initiator Node A attempts to discover a route to the target node X. Let node A's next hop be Node B, Node B's next hop be Node C, and Node C's next hop be the target node X.

$$A \rightarrow Sign_A[RDP, IP_A, IP_X, seq(A)]$$
$$Aggsign = Sign_A[RDP, IP_A, IP_X, seq(A)]$$
$$B \rightarrow Sign_B[RDP, IP_A, IP_X, seq(B)]$$
$$Aggsign = Aggsign + Sign_A[RDP, IP_A, IP_X, seq(B)]$$
$$C \rightarrow Sign_C[RDP, IP_A, IP_X, seq(B,C)]$$
$$Aggsign = Aggsign + Sign_C[RDP, IP_A, IP_X, seq(B,C)]$$
$$X: Verify\ the\ aggregate\ signature$$
$$Aggsign \stackrel{?}{=} (Sign_A + Sign_B + Sign_C)$$
$$X \rightarrow A: Sign_X[REP, IP_A, IP_X, seq(B,C)]$$
$$Sign_X = Sign_X[REP, IP_A, IP_X, seq(B,C)]$$

Fig. 1 The procedure of REP

C. Route Maintenance Phase

If, for example, node B discovers that the link to node C is broken, it sends an error message (ERR) towards the source of the route. The ERR message has the following format:

$Sign_B$< ERR, IPA, IPX, seq, nodelist >

Since it is in general difficult to distinguish malicious ERR message from correct ERR messages, especially in very volatile networks, it may be useful to maintain a count of the number of ERR messages that each node generates. If a node generates an abnormally high number of ERR messages (compared with other nodes), it is likely that this node is malicious (since ERR are signed and it can be verified that such a node actually generated those messages). Hence such a node must be avoided during routing.

D. Performance Analysis

We used NS2 to study the performance efficiency of SAR and SAODV when there is no attacker. We assume that all nodes were loosely time synchronized with each other with fixed synchronization errors. In the simulations, nodes moved following the random waypoint mobility model with a maximum speed 20m/s. The simulation space was a rectangular region with a size of 1500m×300m with 50 nodes. The maximum end to end network delay was 0.1s. The communication range for each node was set to 250m. There were a total 15 pairs of communicating nodes, with each source sending out constant bit rate (CBR) traffic with packet sizes of 64 bytes at a rate of 4 packets/second. The link bandwidth was set to 1Mbps. The hash size, MAC size and key size were set to 80 bits, while the signature size was set to 1024 bits. The TESLA time interval was set to 1s, and the synchronization error was set to 0.1s. The time to generate a signature was set to 10ms and the time to verify a signature was set to 1ms. We omitted the time needed to compute hashes in the simulations.

Each node had a unique hash which corresponded to its identity. Further, we compared the performance of SAR with SAODV under the same network topology and simulation parameters. In order to maximize the advantages of SAODV, we performed the simulations for both cache-enabled and cache-disabled versions of SAODV. We evaluated the performance of SAR by comparing the following metrics to those of AODV and SAODV: Packet delivery ratio, routing overhead (in terms of number of packets), routing overhead (in terms of number of bytes), and packet delivery delay.

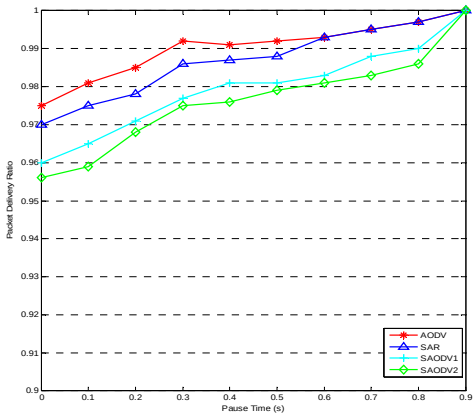

Fig. 2 Performance comparison: Packet Delivery Ratio

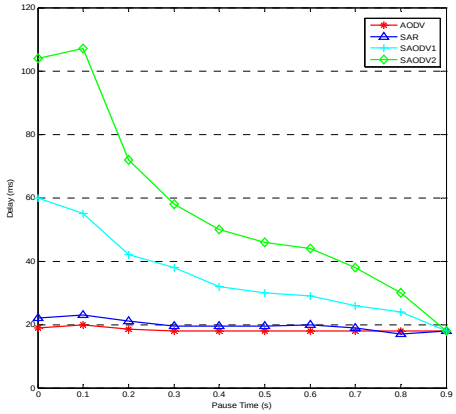

Fig. 3 Performance comparison: Packet Delivery Delay

The simulation results are shown in Figs. 2 and 3, which is based on an average over 60 runs of different movement files for each pause time. The 95% confidence intervals for the metrics are plotted as error bars. The packet delivery ratio of SAR degrades at most 1% for all pause times, which illustrates that SAR performs better than both versions of SAODV. The packet delivery ratio for SAODV with cache-enabled (SAODV1) is better than that for SAODV without cache (SAODV2).The routing overhead in terms of the number of packets for SAR is at the same level as AODV. For SAR, the routing overhead is about twice the amount of baseline AODV. We also observed that the amount of bytes needed for routing overhead was roughly 4 times larger for SAODV than SAR. For SAR, the average packet delivery delay is slightly increased due to the increased communication overhead, while the increase in delay for SAODV is roughly 3 times with cache enabled and 5 times without cache support. Overall, SAR outperforms both SAODV versions in all of the performance metrics that we examined.

V. Conclusion

In this paper, we first proposed a new IBAS scheme with constant pairing computations. It achieves dramatic improvement in computational complexity for verification. In addition, based on the new aggregate signature, we design a secure routing protocol scheme, which could provide sound authentication in wireless ad hoc networks without certificate management problem. Meanwhile it reduces communication cost significantly.

Acknowledgments

This study is supported by the Natural Science Research Project of Education Office of Anhui Province (KJ2013B185), the Natural Science Research Project of Chuzhou University (2012kj001Z), and the National Natural Science Foundation (61303256).

We want to thank the members of our research group, who provided a lot of helpful advice for this paper.

References

[1] Ladislav Huraj, Vladimir Siladi, Jarmila Skrinarova, and Veronika Bojdova, Towards a VO Intersection Trust Model for Ad hoc Grid Environment: Design and Simulation Results, *IAENG International Journal of Computer Science*, 40:2, pp.53-61, 2013.

[2] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing, 2003. IETF RFC 3561.W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, pp. 123–135, 1993.

[3] Y. Hu, D. Johnson, and A. Perrig. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks*, pp.175-192, 2003.

[4] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In Proceedings of the SIGCOMM Conference on Communications Architectures, Protocols and Applications, pp. 234-244, 2003.

[5] Y. Hu, A. Perrig, and D. B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. In Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, pp. 12-23, New York, NY, USA, 2002. ACM Press.

[6] M.G. Zapata and N. Asokan. Securing ad hoc routing protocols. In Proceedings of ACM Workshop on Wireless Security (WiSe), pp. 1-10, 2002.

[7] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. A secure routing protocol for ad hoc networks. In Proceedings of the 10th IEEE International Conference on Network Protocols, pp. 78-87, 2002.

[8] D. Boneh, C. Gentry, B. Lynn, Aggregate and verifiably encrypted signatures from bilinear maps. Proceedings of Advances in Cryptology Eurocrypt'2003, Warsaw, pp.416–432.

[9] Bellare, M., Namprempre, C., Neven, G., Security proofs for identity based identification and signature schemes. In: Advances in Cryptology: Eurocrypt'04, LNCS 3027. Springer-Verlag, pp.268–286, 2004.

[10] Galindo, D., Herranz, J., Kiltz, E., On the generic construction of identity-based signatures with additional properties. In: Advances in Cryptology: Asiacrypt'06, LNCS 4284. Springer-Verlag, pp.179–193, 2006.

[11] Xu, J., Zhang, Z., Feng, D., ID-based aggregate signatures from bilinear pairings. In: CANS'06, LNCS 3810. Springer-Verlag, pp.110–119, 2006.

[12] Yoon, H.J., Cheon, J.H., Kim, Y., Batch verification with ID-based signatures. In: ICISC'08, LNCS 3506. Springer-Verlag, pp.233–248, 2008.

[13] Herranz, J., Deterministic identity-based signatures for partial aggregation. *The Computer Journal*, 49 (3), pp.322–330, 2011.

[14] Gentry C, Ramzan Z. Identity-based aggregate signatures , PKC 2006: 9th International Conference on Theory and Practice of Public Key Cryptography, LNCS. Berlin: Springer-Verlag, pp.257–273, 2006.

[15] Bellare M, Namprempre C, Neven G., Unrestricted aggregate sig- natures, Proceedings of ICALP 2007, LNCS 4596, Springer Verlag, pp.411-422, 2007.

[16] Yifei Zhang and Hongli Zhang, An Experience-Based Algorithm for Securing Network Coordinate Systems, *ICIC Express Letters, Part B: Applications*, vol.2, Issue 4, pp.995-1002, 2011.

[17] Li Yifan, Chen Huiyan, Application of Id-Based Aggregate Signature in MANETs, *Journal Of Electronics*, Vol.27 No.4, pp.516-521, 2012.