Anonymous Secure Communication in Wireless Mobile Ad-Hoc Networks

Sk. Md. Mizanur Rahman[1], Atsuo Inomata[2], Takeshi Okamoto[1], Masahiro Mambo[1], and Eiji Okamoto[1]

[1] Graduate School of Systems and Information Engineering, University of Tsukuiba, Japan
[2] Japan Science and Technology Agency, Tokyo, Japan

Abstract. The main characteristic of a mobile ad-hoc network is its infrastructure-less, highly dynamic topology, which is subject to malicious traffic analysis. Malicious intermediate nodes in wireless mobile ad-hoc networks are a threat concerning security as well as anonymity of exchanged information. To protect anonymity and achieve security of nodes in mobile ad-hoc networks, an anonymous on-demand routing protocol, termed RIOMO, is proposed. For this purpose, pseudo IDs of the nodes are generated considering Pairing-based Cryptography. Nodes can generate their own pseudo IDs independently. As a result RIOMO reduces pseudo IDs maintenance costs. Only trust-worthy nodes are allowed to take part in routing to discover a route. To ensure trustiness each node has to make authentication to its neighbors through an anonymous authentication process. Thus RIOMO safely communicates between nodes without disclosing node identities; it also provides different desirable anonymous properties such as identity privacy, location privacy, route anonymity, and robustness against several attacks.

Keywords: Ad-hoc network, Anonymity, Routing, Pairing-Based Cryptography, Security.

1 Introduction

Conventional wireless mobile communications are normally supported by a fixed wire/ wireless infrastructure. In contrast, mobile ad-hoc networks, MANETs do not use any fixed infrastructure. So, the shared wireless medium MANETs, introduces opportunities for passive eavesdropping on data communications. Thus traffic analysis is one of the most subtle and unsolved security attacks against MANETs. By definition, it is an attack such that an adversary observes network traffic and infers sensitive information of the applications and/or the underlying system [1].

Anonymity and/or privacy is an important criteria for securing ad-hoc network communication. Anonymity ensures that a user may use a resource or service without disclosing the user's identity. Thus anonymity requires that other users or subjects are unable to determine the identity of a user bound to a subject or operation [2]. If anonymity is the stronger the less is known about the

F. Stajano et al. (Eds.): ICUCT 2006, LNCS 4412, pp. 140–149, 2007.

linking to a subject. As a result, adversaries fail to make correlation between the eavesdropped traffic information and the actual network traffic patterns. Thus traffic analysis attack can be efficiently defeated. In this paper an anonymous on-demand routing protocol, called RIOMO, is proposed. In RIOMO, every node can generate its own pseudo IDs dynamically and independently based-on pairing-based cryptography, without making communication with the system administrator. Thus pseudo IDs maintenance cost is reduced compared to the previous proposed method namely MASK by Zhang et al., [3]. A route is discovered without disclosing the nodes IDs for successful communication.

The remainging of this paper is organized as follows. In section 2, preliminaries are described. In section 3, RIOMO architecture and design are given. In section 4 RIOMO protocol is described. In section 5 anonymity achievements and security analysis are given. Finally, section 6 describes conclusions and future works.

1.1 Related Work and Our Contributions

The proposed protocol RIOMO is exclusively based on pairing-based cryptographic properties. There is also another approach of anonymous communication based on pairing-based cryptography proposed by Zhang et al., [3], called MASK. In MASK, system administrator generates a large set of pseudo IDs for every node, thus every node has a fixed pseudo ID set and it should be large enough, otherwise there is a chance of finding pseudonym linking by the intruders. To keep strong anonymity in MASK, every node should have to manage an extremely large enough number of pseudo IDs set provided by the system administrator, which is costly for ad-hoc network communication in terms of extra task for nodes namely IDs maintenance cost. In this paper we explicitly show, by using only one pseudo ID taking from system administrator, nodes can generate their own pseudo IDs, independently and dynamically. It is the first approach to achieve anonymity by using only one pseudo ID taking from the system administrator in ad-hoc network. With pairing based IBE properties and random number nodes can generate their own pseudo IDs dynamically, which also provide strong security properties.

There are some other proposals [4,5,6,7] taking care of privacy. In [4], a secure dynamic distributed routing algorithm (denoted as SDDR in this paper) for ad hoc wireless networks is proposed based on the onion routing protocol [5]. The anonymity-related properties achieved in this algorithm include weak location privacy and route anonymity. However, it ignores one important part of privacy in mobile ad-hoc networks, namely identity anonymity, and it cannot provide strong location privacy.

In ref.[6], Kong et al. design an Anonymous On-Demand Routing (ANODR) based on topology. Similar to Hordes [7], ANODR also applies multicast/ broadcast to improve recipient anonymity. ANODR is an on-demand protocol, and is based on trapdoor information in the broadcast. These features are not discussed in regards to Hordes' [7] multicast mechanism.

Compared to ref.[4], ANODR is more efficient than SDDR at the data-transmission stage. However, similar to SDDR in [4], ANODR does not provide identity anonymity and strong location privacy. RIOMO and other two protocols are described in Table 1 with respect to the Anonymity and security related properties. For anonymity related properties ✓: indicates property is achieved, and blank indicates property is not achived, *: indicates identity privacy of source and destination, **: indicates identity privacy of forwarding nodes in route. For security related properties ✔: indicates attack is protected and blank indicates not protected. Detailed discussions of these properties are given in Section 5.

Table 1. Comparison of anonymity and security related properties

Anonymity and security properties	Routing protocol		
	SDDR	ANODR	RIOMO (proposed)
Identity privacy*	✓		✓
Identity privacy**		✓	✓
Weak location privacy	✓	✓	✓
Strong location privacy			✓
Route anonymity		✓	✓
DoS attacks			✔
Wormhole attacks	✔	✔	✔
Rushing attacks	✔	✔	✔

2 Preliminaries

In this section, we just describe some preliminaries and mathematical properties which are useful to understand our proposed protocol.

2.1 Bilinear Maps

Let G_1 be an additive group and G_2 be a multiplicative group of the same prime order q. Let P be an arbitrary generator of G_1. *(aP denotes P added to itself a times)*. Assume that discrete logarithm (DL) problem is hard in both G_1 and G_2. We can think G_1 as a group of points on an elliptic curve over F_q, and G_2 as a subgroup of the multiplicative group of a finite field F_{q^k} for some $k \in Z_q^*$. A mapping $\tilde{e} : G_1 \times G_1 \rightarrow G_2$, satisfying the following properties is called a cryptographic bilinear map.

- *Bilinearity:* $\tilde{e}(aP, bQ) = \tilde{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$. This can be restated in the following way. For $P, Q, R \in G_1, \tilde{e}(P + Q, R) = \tilde{e}(P, R)\tilde{e}(Q, R)$ and $\tilde{e}(P, Q + R) = \tilde{e}(P, Q)\tilde{e}(P, R)$.
- *Non-degeneracy:* If P is a generator of G_1, then $\tilde{e}(P, P)$ is a generator of G_2. In other words, $\tilde{e}(P, P) \neq 1$.
- *Computable:* A mapping is efficiently computable if $\tilde{e}(P, P)$ can be computed in polynomial-time for all $P, Q \in G_1$.

Modified Weil Pairing [8]and Tate Pairing [9]are examples of cryptographic bilinear maps.

2.2 Diffie-Hellman Problems

With the group G_1 described in section 2.1, we can define the following hard cryptographic problem applicable to our proposed scheme.

- *Discrete Logarithm (DL) Problem:* Given $P, Q \in G_1$, find an integer n such that $P = nQ$ whenever such integer exists.
- *Computational Diffie-Hellman (CDH) Problem:* Given a triple $(P, aP, bP) \in G_1$ for $a, b \in Z_q^*$, find the element abP.
- *Decision Diffie-Hellman (DDH) problem:* Given a quadruple $(P, aP, bP, cP) \in G_1$ for $a, b, c \in Z_q^*$, decide whether $c = ab \bmod q$ or not.
- *Bilinear Diffie-Hellman (BDH) Problem:* Given a quadruple $(P, aP, bP, cP) \in G_1$ for some $a, b, c \in Z_q^*$, compute $\tilde{e}(P, P)^{abc}$.

Groups where the *CDH* problem is hard but *DDH* problem is easy are called GAP Diffie-Hellman (*GDH*) groups. Details about *GDH* groups can be found in [10].

3 RIOMO Architecture and Design

In RIOMO, system administrator does not take part in routing rather it has the following tasks during the boot strap of the network.

- Determines two groups G_1, G_2, of the same prime order q. We view G_1 as an additive group and G_2 as a multiplicative group as discussed in section 2.1.
- Determines bilinear map $g : G_1 \times G_1 \rightarrow G_2$, collision resistant cryptographic hash functions H_1 and H_2, where $H_1 : \{0,1\}^* \rightarrow G_1$, a mapping from arbitrary-length strings to points in G_1 and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^\mu$, a mapping from arbitrary-length strings to μ-bit fixed length output.
- Generates system's secret $\omega \in Z_q^*$, where $Z_q^* = \{y | 1 \leq y \leq q-1\}$. Any one in the network does not know ω except system administrator. System administrator also uses this secret to generate the secret point of the non-adversary nodes.

Thus the system parameters $< G_1, G_2, g, H_1, H_2 >$ are known to the non-adversary nodes. System administrator also provides the following parameters for nodes, regarding their IDs and secret points.

- Provides each node, a secret point SP_R, with respect to the node's real ID ID_R, which is defined as $SP_R = \omega H_1(ID_R)$. The Source and the destination use their corresponding secret point in the route discovery phase to authenticate each other. For a given set of $< ID_R, SP_R >$ no one can determine the system secret ω as we discussed in section 2.1 and 2.2.

– Provides each node a pseudo ID IDP_i, and their corresponding secret point SPP_i, which is defined as $SPP_i = \omega H_1(IDP_i)$; if $i \neq j$ then $IDP_i \neq IDP_j$ as well as $SPP_i \neq SPP_j$. For a given set of $< IDP_i, SPP_i >$ no one can determine the system secret ω.

With the above information any node can generate its own *pseudo* IDs and the corresponding secret points randomly in every session in communication. Let's check for a node, namely K; K has received its pseudo ID IDP_K and the corresponding secret point $SPP_K = \omega H_1(IDP_K)$ from the system administrator. Now, K is able to generate its own pseudo ID $ID_{PK} = R_K H_1(IDP_K)$, and the corresponding secret point $SP_{PK} = R_K SPP_K = R_K \omega H_1(IDP_K) = \omega R_K$-$H_1(IDP_K) = \omega ID_{PK}$, where R_K is a random generated by K; this relation also holds the previous cited property in section 2.1 and 2.2, that is no one can determine the system secret ω for a given set of pseudo ID and the corresponding secret point , $< ID_{PK}, SP_{PK} >$. Thus a node can generate its own pseudo IDs and corresponding secret points when it is needed.

4 RIOMO Protocol

4.1 Anonymous Neighbor Authentication

When a node wants to join in the network or moves to a new place, it has to authenticate within its neighbor nodes. Say, Alice has received her pseudo ID IDP_A, and the corresponding secret point $SPP_A = \omega H_1(IDP_A)$, i.e.,$< IDP_A, SPP_A >$ from the system administrator. She can join in the network by authenticating within her neighbor nodes or if she moves another place in the network different from her current place, she also needs to authenticate her within her neighbor, to avoid a *target oriented* attack. If Alice wants to change her pseudo ID different from her current pseudo ID without moving her place, she also needs to authenticate her current pseudo ID within her neighbor. For this purpose she generates pseudo ID $ID_{PA} = R_A H_1(IDP_A)$, and corresponding secret point $SP_{PA} = R_A SPP_A = R_A \omega H_1(IDP_A) = \omega R_A H_1(IDP_A) = \omega ID_{PA}$, where R_A is a random generated by Alice; she also generates a random R_{RA} which is used to generate verification codes Ver_0^* and Ver_1. Alice broadcasts her pseudo ID ID_{PA}, and random R_{RA} within her neighbor region. One of her neighbor, let's say Bob, makes a response with his pseudo ID ID_{PB}, and generated random R_{RB} and verification code Ver_0 as shown in Figure 1. If Alice is a valid node then $Ver_0^* = Ver_0$, and $Ver_1^* = Ver_1$ holds, thus she can be a member and she is identified as ID_{PA}, within her neighbor. Alice and Bob use their session key $K_{AB} = g(SP_{PA}, ID_{PB}) = g(ID_{PA}, ID_{PB})^\omega$ and $K_{BA} = g(SP_{PB}, ID_{PA}) = g(ID_{PB}, ID_{PA})^\omega$; thus $K_{AB} = K_{BA}$ corresponding their pseudo IDs, ID_{PA} and ID_{PB} respectively. No one within Alices neighbor can recognize her as Alice because she is using her pseudo ID and she is changing her pseudo ID time to time. Thus the nodes can hide their IDs in the network and always seem new to each other. Any adversary node can not be a member within its neighbor, because it has to pass the verification process "?$(Ver_1^* = Ver_1)$"

which is not possible to generate without the knowledge of the system secret. Similar way all nodes in the network can authenticate anonymously within their neighbors and generate their corresponding session key. Thus nodes in the network maintain their neighbor table with their pseudo IDs and corresponding session key.

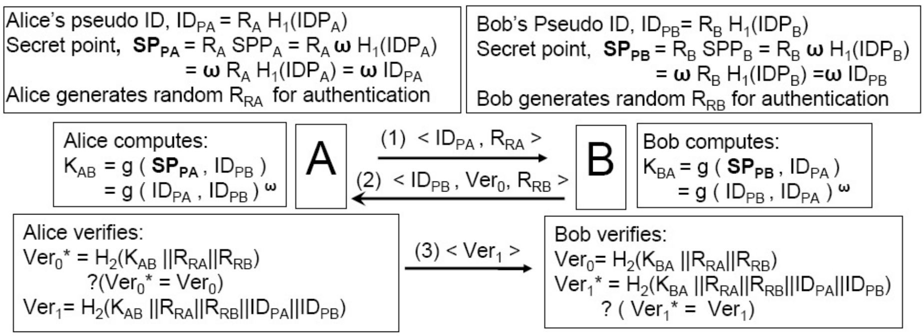

Fig. 1. Anonymous neighbor authentication process for two neighbor nodes Alice and Bob

4.2 Control Packets

RIMIO uses route request packet RRQ, and route reply packet RRP, to find a route in the network. To discover a route and to receive a response it uses RRQ and RRP respectively.

Route Requeat Packet *RRQ*:

ID_{PSE}	$RRQSeqNO$	ID_S	ID_D

ID_{PSE}: Sender pseudo ID ID_{PSE}, it is the pseudo ID of the current sender. When sender broadcasts a RRQ packet it puts its own pseudo ID in this field. Thus $ID_{PSE} \neq ID_S$, but when the source is a sender then $ID_{PSE} = ID_{PSO} \neq ID_S$, here ID_S is the source's real ID and ID_{PSO} is the source's pseudo ID which we discussed in section 3.

RRQSeqNO: Route request sequence number is used for identifying each route-request and corresponding route-reply packet from each other. It is generated by the source uniquely when source wants to communicate with a destination. $RRSeqNO = H(ID_{PSO}||Time)$, where, H is a collision resistant hash function known to all non adversary nodes in the network, ID_{PSO} is a pseudo ID of the source, and Time is the calendar time when source generates RRQ packet. This field remains unchanged for the corresponding RRP generated by the destination.

ID_S: Source's ID ID_S, it is the source's real ID. Source generates a route request packet and puts its real ID in this field, and pseudo ID ID_{PSO}, in ID_{PSE} field; thus for source $ID_{PSE} = ID_{PSO}$ but $ID_{PSE} \neq ID_S$. It is used by the destination to make a sign in route reply packet.

ID_D: Destination's ID ID_D, it is the destination's real ID.

Route Reply Packet *RRP*:

ID_{PSE}	ID_{PRE}	*RRQSeqNO*	$Sign_D$

ID_{PRE} : Receiver's pseudo ID; on the path from the destination to the source when RRP packet travels ID_{RPE} defines the next node who receives RRP packet.

SignD : Destination's Sign; when destination replies to source through intermediate nodes, it generates a sign, so that no one can forge. $Sign_D$ =-$H_2(K_{DS}||RRQSeqNO)$, where K_{DS} is a session key between the source and the destination, and generated by the destination as $K_{DS} = g(\omega H_1(ID_D)$,-$H_1(ID_S)) = g(H_1(ID_D), H_1(ID_S))^\omega$.

Destination also uses its session key K_{DS}, to decrypt data, which sent by the source encrypted with source's session key K_{SD}, where $K_{SD} = g(\omega H_1(ID_S)$,-$H_1(ID_D)) = g(H_1(ID_S), H_1(ID_D))^\omega$.

4.3 Route Discovery and Route Reply

On route discovery and route response procedures nodes maintain their corresponding tables. When a node receives a *RRQ* packet it broadcasts within its neighbor and when it receives a *RRP* packet, it sends the *RRP* corresponding to the receiver. RIOMO is described in terms of its functionalities which are described below.

Route Discovery. Every node in the network maintains its neighbor table with their pseudo IDs and corresponding session keys. When a source wants to communicate with a destination it generates a *RRQ* and broadcasts this *RRQ* within its neighbor to find a route, thus RIOMO is an on-demand routing protocol. By receiving a *RRQ*, a node checks ID_D and *RRQSeqNO*, of the *RRQ* and makes the following decisions:

- If the node is the destination i.e., ID_D matches with its real ID then it do the following tasks:
  - It keeps $< RRQSeqNO, ID_{PSE} >$ in its routing table; this ID_{PSE} becomes ID_{PRE} for *RRP*, generated by the destination. By replacing destination's own pseudo ID in the ID_{PSE} field of *RRQ*, it broadcasts *RRQ*, within its neighbor. The purpose of this extra broadcast is to make attackers fool.
  - It generates a RRP with its own pseudo ID ID_{PSE}, receiver's pseudo ID ID_{PRE} already discussed above, makes a sign $Sign_D$ discussed in section 4.2 and sends to the receiver. Notice that *RRQSeqNO* is unchanged.
- If the node is not the destination and *RRQSeqNO*is new, it keeps *RRQSeqNO*, corresponding pseudo ID ID_{PSE} in its routing table, this information $< RRQSeqNO, ID_{PSE} >$ is used by the node in the route reply procedure; this ID_{PSE} becomes a receiver pseudo ID ID_{PRE} in the route reply procedure. The node becomes a new sender and it puts its own pseudo ID in the ID_{PSE} field of the *RRQ* and this *RRQ* within its region.

Route Reply. It is just a reverse path traverse of a *RRP* explored by a *RRQ*. When a *RRQ* reaches to the destination it generates a *RRP* and forwards it in the reverse path as we discussed above. If a node receives a *RRP*, it checks *RRQSeqNO* in its routing table then updates receiver's pseudo ID ID_{PRE}, with an appropriate ID_{PSE} (i.e., from whom it receives the corresponding *RRQ* with the same *RRQSeqNO*), and sends in the reverse path. If source receives a *RRP* it generates $Sign_S = H_2(K_{SD}||RRQSeqNO)$ and verify $Sign_D$. If $Sign_S = Sign_D$ the source sends data in the explored path by encrypting with its session key K_{SD}.

4.4 Working Procedure in Brief

1. Nodes make authentication of their neighbor nodes and maintain their neighbor table. Thus only the trusted nodes can take part in authentication.
2. On Route discovery phase, source generates a *RRQ* and sends within its neighbor. If the destination is not within its neighbor then neighbor nodes become new sender. By replacing their own pseudo IDs broadcast within their own neighbor region. They also maintain this information in routing table as we discussed in section 4.3.
3. If the node is the destination it generates a *RRP* and sends in the reverse path as we discussed in section 4.3
4. Receiving *RRP*, source checks the authenticity of the destination, by comparing $Sign_S$ and $Sign_D$. If success then sends data in the explored path. Source and destination will use their corresponding session key for encryption and decryption as discussed in section 4.2 and 4.3.

5 Anonymity Achievement and Security Analysis

When an *RRQ* and *RRP* travel from node to, every node generates a large bit random sequence corresponding to the fields of *RRQ* and *RRP*. By extracting random bits from the fields of the packets, every node pads their own random bit sequence, and replaces their own pseudo IDs to the ID_{PSE} accordingly. Thus the packets appear new when it moves from node to node. Also the fields (except ID_{PSE}, ID_{PRE}) are encrypted with corresponding session keys, thus it is also protected from intruders.

Identity Privacy. In RIOMO the identities of the nodes are represented by their pseudo IDs which are changed by the nodes in each session of communication. Pseudo IDs are also generated by using random numbers, hash functions as we discussed in section 3, also the control packets are encrypted so no one can recognize who is actual source and/or destination in a route request, route reply phase. Thus identity privacy of nodes is achieved in the network.

Location Privacy. If there is extra information added to control packets when the packets are forwarded form node to node; by observing the route request and the route response packets an attacker can estimation about the distance between the source and the destination. Thus, an attacker can set an attack regarding location privacy.

In our scheme, nodes do not know anything about the locations and identities of the other nodes in the network. So, no nodes in the network can determine the distance from them to the source and to the destination; they also do not know about the starting point of a packet traveling in the network. Only in a session the nodes know pseudo IDs of its neighbor region. Thus RIOMO ensures location privacy.

Route Anonymity. Current attacks on route anonymity are based on traffic analysis [11]. The general theory behind these kinds' of attacks is to trace or to find a path in which packets are moving. For these purpose the malicious nodes mainly looks for common information which are not changing in a packet during movements of control packets. As a result, the adversaries can find or to estimate the route from source to the destination. In RIOMO all the control packets appear new to the network, when it travels form node to node. Because every time random bits are extracted and padded during movements of the control packets as we discussed at the beginning of this section. Thus route anonymity is achieved of a path.

DoS. According to the target of attack, multiple adversaries can co-operate or one adversary with enough power can target to a specific node to exhaust the resource of the node. For this purpose the adversaries try to identify a node and set a target to that specific node. In RIOMO identity privacy is achieved; so one can identify a node make a target to attack. Thus DoS can be protected.

Wormhole Attack. In wormhole attack an attacker records a packet in one location of the network and sends it to another location making a tunnel [12] between the attacker's nodes, later packet is retransmitted to the network under its control. Thus there could be a long distance travel for a packet to find a route from the source to the destination. In RIOMO an attacker can not be a trusted member within its neighbor so it can not be an intermediate node in route discovery or route reply phase thus an attacker can not take part in the routing. So the affect of the wormhole attack is not effective in RIOMO.

Rushing Attack. By using the tunnel of wormhole attack an attacker can introduce rushing attack to rush packets. Existing almost all on-demand routing protocols suffers from rushing attack. As RIOMO can prevent wormhole attack so rushing attack is not effective in this protocol also.

6 Conclusions and Future Works

Anonymity is one of the important characteristics in securing a mobile ad-hoc network routing. In this paper an anonymous on-demand routing protocol, called RIOMO, is proposed, for preventing active as well as passive attacks. Nodes in RIOMO take only one pseudo ID from system administrator and generate their own pseudo IDs for anonymous communications. Thus pseudo IDs maintenance cost is reduced compare to the existing protocol. Moreover RIOMO ensures node privacy, route anonymity and location privacy and is robust against several known attacks. Comparison analysis and security properties are described. Further research is to consider performance analysis as well as implementation in a specific environment.

References

1. Y. Guan, X. Fu, D. Xuan, P. Shenoy, R. Bettati, and W. Zhao, NetCamo: Camouflaging Network Traffic for QoS-Guaranteed Mission Critical Applications, *IEEE Transactions on Systems, Man, and Cybernetics*, 31(4), July 2001, pp.253-265.
2. ISO99 ISO IS 15408, 1999, available at `http://www.commoncriteria.org`
3. Y. Zhang, W.Liu and W.Lou, Anonymous Communications in Mobile Ad Hoc Networks, In *IEEE Infocom 2005*, Miami, USA, March 13-17, 2005. The 24th Annual Conference Sponsored by IEEE Communications Society, available at `http://ece.wpi.edu/~wjlou/publication/INFOCOM05_Zhang.pdf`
4. K. El-Khatib, L. Korba, R. Song, and G. Yee, Secure dynamic distributed routing algorithm for ad hoc wireless networks, In *International Conference on Parallel Processing Workshops (ICPPW03)*, 2003.
5. M. G. Reed, P. F. Syverson, and D. M. Goldschlag, Anonymous connections and onion routing, *IEEE Journal on Selected Areas in Communications, Special Issue on Copyright and Privacy Protection*, 16(4), 1998, pp. 482494.
6. J. Kong and X. Hong, ANODR: ANonymous on demand routing with untraceable routes for mobile ad-hoc networks, In *Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc03)*, 2003, pp. 291302.
7. B.N. Levine, C.Shields, Hordes:a multicast based protocol for anonymity,*Journal of Computer Security* Volume 10, Issue 3, ISSN:0926-227X, 2002, pp. 213-240.
8. D. Boneh, M. Franklin, Identity Based Encryption from the Weil Pairing, *SIAM Computing*, Vol. 32, No. 3, Extended Abstract in Crypto 2001, 2003, pp. 586-615.
9. P. S. L. M. Berreto, H. Y. Kim and M. Scott, Efficient algorithms for pairing-based cryptosystems, *Advances in Cryptology - Crypto2002*, LNCS 2442, Springer-Verlag (2002), pp.354-368.
10. D. Boneh, B. Lynn and H. Shachum, Short signatures from the Weil pairing, *Advances in cryptology ASIACRYPT01*, Lecture Notes in Comput Sci. 2248 (2001), pp.514-532.
11. J.F.Raymond,, Traffic Analysis: Protocols, Attacks, Design Issues and Open Problems, in *Proceedings of PET 01*, Vol. 2009, LNCS, Springer-Verlag, 2001, pp. 10-29.
12. Y.C. Hu, A. Perrig, and D. B. Johnson, Packet leashes: A defense against wormhole attacks in wireless ad hoc networks, In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, 2003.