Regular Paper

Anonymous On-Demand Position-based Routing in Mobile Ad-hoc Networks

Sk. Md. Mizanur Rahman,[†] Atsuo Inomata,[††]
Masahiro Mambo[†] and Eiji Okamoto[†]

Due to the infrastructure-less, dynamic, and broadcast nature of radio transmissions, communications in mobile ad-hoc networks, MANETs, are susceptible to malicious traffic analysis. After performing traffic analysis, an attacker conducts an intensive attack (i.e., a target-oriented attack) against a target node specified by the traffic analysis. Because of the degradation of both throughput and security of routing, traffic analysis and its subsequent target-oriented attack are known as serious problems in regards to MANETs. Basically, position information of routing nodes is very sensitive data in MANETs, where even nodes not knowing each other establish a network temporarily. It is desirable that position information is kept secret. All of these problems are especially prominent in position-based routing protocols of MANETs. Therefore a new position-based routing protocol, which keeps routing nodes anonymous, thereby preventing possible traffic analysis, is proposed. The proposed scheme uses a time-variant temporary identifier, Temp ID, which is computed from the time and position of a node and used for keeping the node anonymous. Only the position of a destination node is required for the route discovery, and the Temp ID is used for establishing a route for sending data. A receiver dynamic-handshake scheme is designed for determining the next hop on-demand by using the Temp ID. The level of anonymity and the performance of this scheme were evaluated. The evaluation results show that the proposed scheme ensures the anonymity of both route and nodes and robustness against a target-oriented attack and other attacks. Moreover, this scheme does not depend on node density as long as nodes are connected in the network.

1. Introduction

Mobile ad hoc networks, MANETs, are finding ever-increasing applications in both military and civilian systems owing to their self-configuration and self-maintenance capabilities. Many of these applications, such as military battlefield operations, homeland-security scenarios, law enforcement, and rescue missions are security sensitive. As a result, security in MANETs has recently been drawing much attention[1].

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[2]. Sensitive information includes the identities of communicating parties, network traffic patterns[1], and their changes. The leakage of such information is often devastating in security-sensitive scenarios. For example, an unexpected change of the traffic pattern in a military network may indicate a forthcoming action, a chain of commands, or a state change of network alertness[3]. It may also reveal the locations of command centers or mobile VIP nodes, which will enable the adversaries to launch pinpoint attacks on them. In contrast to active attacks, which usually involve the launch of denial of service or other more “visible” and aggressive attacks on the target network, traffic analysis is a kind of passive attack, which is “invisible” and difficult to detect. It is therefore important to design countermeasures against such malicious traffic analysis.

The shared wireless medium of MANETs introduces opportunities for passive eavesdropping on data communications. Adversaries can easily overhear all messages “flying in the air” without physically compromising nodes. Several methods for withstanding eavesdropping and other kinds of traffic analysis have been investigated. One attempt is to prevent the wireless signals from being intercepted or even detected by developing LPI/LPD (low probability of interception/low probability of detection) communication techniques. Examples of such techniques include spread-spectrum modulation, effective power control, and directional

[†] Graduate School of Systems and Information Engineering, University of Tsukuba
[††] Japan Science and Technology Agency

antennas[4]. However, it is impossible to completely avoid signal detection in open wireless environments. The second method relies on the use of traffic padding, i.e., inserting dummy packets into the network[5] to camouflage the real traffic pattern. However, this approach adds significant extra load to the network and consumes scarce network resources. The third method is to perform end-to-end encryption and/or link encryption on data traffic. However, this only prevents adversaries from accessing traffic contents. Adversaries can still carry out traffic analysis based on the bare network-layer and/or MAC addresses, both of which are unprotected and unencrypted in common ad-hoc routing protocols such as Ad hoc On-Demand Distance Vector (AODV)[6]. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)[7], and the de facto MAC protocol IEEE 802.11.

Research on ad-hoc networks has resulted in a number of routing protocols suitable for MANETs[8]. Most current researches on MANET routing are focused on *topology-based* protocols. These protocols use information about links in the network to perform packet forwarding and are generally classified as either table-driven or on-demand. The on-demand scheme is more familiar than the table-driven one because it does not involve extra computation like routing table maintenance of the table-driven scheme.

Meanwhile *position-based* routing protocols are known to be a good alternative to on-demand topology-based protocols in many cases[9],[10]. Position-based routing protocols use a node's geographical position to make routing decisions, resulting in improved efficiency and performance. Therefore, nodes of these protocols are required to obtain their own geographical position and the geographical position of the destination. Generally, this information is obtained via global positioning system (GPS) and location services.

Most traditional topology-based MANET protocols were designed with reliability and performance in mind. Unfortunately these protocols were not designed to be secure and do not defend against malicious attacks. AODV and DSR, two protocols under consideration for standardization by the IETF MANET Working Group, are both vulnerable to a number of attacks, including impersonation, modification, and fabrication[11]. Position-based MANET routing protocols[9],[12],[13] are also vulnerable to such attacks, as they focus on improving performance while disregarding security issues. In addition, these protocols lack cryptographic techniques to protect location information exchanged between nodes, revealing the exact location of nodes to anyone within range. In a high-risk environment, this is unacceptable. Cryptographic techniques must be employed to protect position information in these protocols if they are to be used in a high-risk MANET.

If position information can be safely protected, not only efficiency but also security of MANET routing is improved. Lack of privacy in traditional position-based ad-hoc routing algorithms is mainly caused by extensive position-information exposure[18].

To achieve communication anonymity and security in any node density network, we propose a new position-based routing protocol, called an anonymous on-demand position-based routing (AODPR), which keeps routing nodes anonymous. In AODPR, the position of the destination is encrypted with a common key of nodes, and this encrypted position is used for the routing. Information is thus not disclosed to nodes not composing the ad-hoc network. In AODPR, a route is discovered by a *dynamic handshake* mechanism, which dynamically determines the next hop. For this purpose, a route-request message is sent from the source towards the position of the destination.

This paper is organized as follows. In Section 2, the preliminaries are described. In Section 3, AODPR fundamental definitions are given. In Section 4, the AODPR protocol is described. In Section 5, anonymity achievements and security analysis are given. In Section 6, performances of AODPR are analyzed. Finally, Section 7 describes conclusions.

2. Preliminaries

2.1 Privacy and security notions

The key notions on privacy associated with MANETs are summarized as follows.

Identity Privacy: Identity privacy means no one knows the real identity of the nodes in the network. We are especially interested in discussing identity privacy of entities involved in packet transmission, namely, the source, intermediate nodes and the destination.

Location Privacy: Requirements for location privacy are as follows: (a) no one knows the exact location of a source or a destination, except themselves; (b) other nodes, typically intermediate nodes in the route, have no information about their distance, i.e., the number of hops, from either the source or the destination. It is said that a protocol satisfying (a) achieves weak location privacy, and a protocol satisfying both (a) and (b) achieves strong location privacy.

***Route Anonymity*:** Requirements for route anonymity are as follows: (a) adversaries either in the route or out of the route cannot trace packet flow back to its source or destination; (b) adversaries not in the route have no information on any part of the route; (c) it is difficult for adversaries to infer the transmission pattern and motion pattern of the source or the destination.

As in normal networks, attacks on MANETs are categorized as either passive or active.

***Passive Attacks*:** Passive attack typically involves unauthorized "listening" to the routing packets or silently refusing execution of the function requested. This type of attack might be an attempt to gain routing information from which the attacker could extrapolate data about the positions of each node in relation to the others. Such an attack is usually impossible to detect, since the attacker does not disrupt the operation of a routing protocol but only attempts to discover valuable information by listening to the routed traffic.

***Active Attacks*:** Active attacks are meant to degrade or prevent message flow between nodes. They can cause degradation or a complete halt in communications between nodes. Normally, such attacks involve actions performed by adversaries, e.g., replication, modification, and deletion of exchanged data.

The traffic analysis we are interested in is usually passive. After performing traffic analysis, an adversary can set a target node and conduct an intensive attack against the node. We call such an attack "target-oriented". Such attacks are often active. The followings are examples of active attacks.

***DoS*:** Multiple adversaries in co-operation or one adversary with enough power can set a specific node as a target in order to exhaust the resource of that node. That is to identify a node and make a target to that specific node.

***Wormhole Attacks*:** In wormhole attack, an attacker records a packet in one location of the network and sends it to another location through a tunnel 23) made between the attacker's nodes. Afterwards, the packet is retransmitted to the network under his control.

***Rushing Attack*:** Existing on-demand routing protocols forward a request packet that arrives first in each route-discovery. In the rushing attack, the attacker exploits this property of route discovery operation. If the route requests forwarded by attackers arrive at a target node earlier than other route requests, any route discovered by this route discovery includes a hop via the attacker. In general, an attacker can forward a route request more quickly than legitimate nodes can, so he can enter a route. Such a route cannot be easily detected.

Since nodes in MANETs move dynamically, adversaries cannot conduct active attacks without knowing the location or name of nodes. It thus often happens that traffic analysis is conducted passively at first and active attacks are conducted later. Therefore, we set our goal to establish a protocol that is secure against passive attacks in terms of the privacy notions explained above and also secure against the three active attacks described above.

2.2 Related work

There are some recent proposals 14)~18) taking care of privacy in MANETs. In Ref. 14), a secure dynamic distributed topology-based routing algorithm (SDDR) based on the onion-routing protocol 15) for ad-hoc wireless networks has been proposed. The anonymity-related properties achieved with 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. 16), Kong, et al. design an Anonymous On-Demand Routing (ANODR) based on topology. Similar to Hordes 17), 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' 17) multicast mechanism.

Compared to Ref. 14), ANODR gives a more comprehensive analysis of the anonymity and security properties achieved, and provide detailed simulation results. In addition, ANODR is more efficient than SDDR at the data-transmission stage. However, similar to SDDR in Ref. 14), ANODR does not provide identity anonymity and strong location privacy. To the best of our knowledge some protocols are also

Table 1 Comparison of security-related properties.

Routing protocol Sec. properties	SDDR	ANODR	AODPR (proposed)
Identity privacy *	√	×	√
Identity privacy **	×	√	√
Weak location privacy	√	√	√
Strong location privacy	×	×	√
Route anonymity	×	√	√
DoS attacks	××	××	√√
Wormhole attacks	√√	√√	√√
Rushing attacks	√√	√√	√√

×: Not achieved √: Achieved
××: Not protected √√: Protected
* : Identity privacy of source and destination
**: Identity privacy of forwarding nodes in route

Table 2 Comparison of routing strategies.

Routing protocol Strategy		SDDR	ANODR	AODPR (Proposed)
Routing strategy	Broadcast	Yes	Yes	Yes
	RREQ	Flooding	Flooding	Convergence

RREQ: Route request packet
Flooding: Network-wide flooding process
Convergence: Converging on a destination

dependent on node density [21].

Concerning the rushing attack, an existing on-demand routing protocol, such as AODV [6], DSR [7], location-aided routing (LAR) [24], Ariadne [25], secure AODV [26], a secure routing protocol for ad-hoc networks (ARAN) [27], AODV secured with statistically unique and cryptographically verifiable (SUCV) [28] and secure routing protocol (SRP) [29] are all susceptible to rushing attack.

Although SDDR and ANODR are topology-based, these protocols guarantee many privacy properties, as shown in **Table 1**. The proposed AODPR and other two protocols are described in Table 1 and **Table 2** with respect to security and routing strategies, respectively. Detailed discussions of these security properties are given in Section 5.

3. AODPR fundamentals

3.1 Position management

Known position-based routing protocols [18)~21)] use a position/location management scheme, called a virtual home region-based distributed secure position service (DISPOSER). In this scheme, each node has its own virtual home region (VHR), which is a geographical region around a center specific to the node. The center is fixed center and anyone can identify it by taking a concatenation of two publicly known values, namely, the node's ID and position information regarding the center of the whole network, as input to a publicly known hash function. There are position servers PSs for each node in the network. PSs of a node N exist only inside the VHR of N and manage position information of N as follows. To report the position of N to its PSs, N executes a region-based broadcast [20] in the VHR if N stays inside its VHR. If N stays out of its VHR, N sends a packet containing position information of N and the center of N. The latter position information is used for determining which node forwards the packet. Once the packet reaches a node in the VHR, the node executes a region-based broadcast. After the region-based broadcast the PSs can store the latest position information of N. To retrieve position information of N, a source sends a request packet in the direction of the center of N. When the packet reaches a node in the VHR of N, the node executes a sequential searching method [20] and finally the packet reaches one of the PSs. The source authenticates itself to the PS, and then the PS provides the required position information. Using this position information, the source can establish a path from him to the destination. PSs are determined from the node density, the size of the VHR, the robustness of the system, and so on, and the number of the PSs is set in an appropriate value that makes the sequential search more cost-effective than the region-based broadcast and the management cost of the position information low enough. More details on the VHR are described in Refs. 18) and 20).

The PS of our proposed scheme has an additional property: PS provides a source with additional information to enhance the authenticity and secrecy of services provided by the PS. Before describing this scheme, we define two notations: *Position information* denotes a pair composed of position and time, and legitimate nodes denote nodes which have registered with PS and received a common key $\mathrm{C_K}$ form PS.

In contrast to ordinary PS, our PS provides a source with a common key $\mathrm{C_K}$ for all legitimate nodes, public key PK of the destination, *position information* of the destination, authentication information *Auth*, and a *Token*.

When a node joins a network, it is registered the PS and gets a common key $\mathrm{C_K}$ and a pair of public key PK/secret key SK from the PS.

When a node updates its *position informa-*

tion and sends it to the PS, it generates a *random number* and sends it together with its *position information* to the PS. This *random number* is used for generating *Auth*, where *Auth* = [H$_1$ (Destination's *Position*, Destination's *random number*)] and H$_1$ is a global hash function. The notation A = [B, C, . . . , Z] means variable A is substituted by the concatenation of B, C, . . . , Z. Later, at the route discovery phase *Auth* is used for authenticating the destination to the source.

To obtain the *position information* of the destination from the PS, the source has to send a signed position request to PS with a *route-request sequence number RRQSeqNo*. After verification of the signature, the PS responds to the source's request with the *position information* of the destination, *Auth*, public key $\mathrm{P_K}$ of the destination, and a *Token* defined as *Token* = [H_{PS} (Sender Temp ID, Receiver Temp ID), Time, *RRQSeqNo*], where H_{PS} is a local hash function defined by the PS. *Position information* is used for generating the temporary Identifier Temp ID. In contrast, *Position* is used only for routing, and it is encrypted by $\mathrm{C_K}$ in the route-request phase.

A sender keeps *Auth* received from the PS for a session of communication. At the last phase of the route-discovery procedure, destination will reply with a *route-reply message RRPMsg* for its authentication to the sender: *RRPMsg* = [$Sig_{SK_{Dest}}$ (*Auth*)], where Sig_{SK} is a digital signature function under secret key *SK*, and SK_{Dest} is the SK of the public/secret key pair of the destination. With this *RRPMsg* the sender authenticates the destination.

A *Token* is sent in the last phase of data transmission to the destination. At the end of the communication, the destination sends this *Token* to PS, so that PS can determine whether the communication between the source and the destination is valid. If a node takes the *position information* of the destination and does not make a data transmission, then PS will not supply any further position information to that node.

3.2 Dynamic handshaking

A kind of handshaking, called dynamic handshaking, which is established from the ending point to the beginning point, is defined here as shown in **Fig. 1**. At first, node A sends a signal for node D via B. B will response to A after getting a response from C. That means A will wait for a certain time. The whole handshaking process is performed from the ending to the beginning.

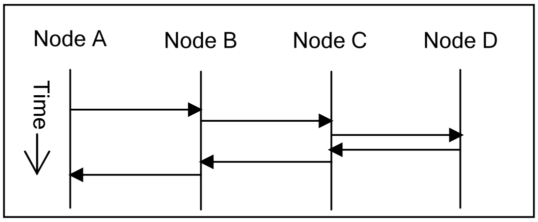

Fig. 1 Dynamic handshaking.

3.3 Control packets of AODPR

Three control packets are used for route discovery of AODPR: *Route Request Packet RRQ*, *Route Reply Packet RRP* and Fail Packet *Fail*. These packets are described in *Appendix*. Here only a few fields of the *RRQ* are described.

Sender Temp ID: For every session of communication, a source generates its temporary ID Temp ID, computed as Temp ID = [*H* (Position, Time, PK)], where *H* is a global hash function known to all legitimate nodes in the network, Position is the position of the source, Time is the present time, and PK is a public key of the source. Temp ID uniquely identifies the source in each session of communication and is dynamically changed from session to session and from hop to hop. When nodes staying within the sender's radio range receive the *RRQ* packet, they will become new senders or forwarders and update the Temp ID into their own Temp ID, which is generated in a similar way to that mentioned above.

For successful identification, the Temp ID should be unique for each session of communication. To this end, *H* should be collision resistant. Theoretically proven collision-resistant hash functions are slow; thus, in practice, hash functions that are expected to be collision-resistant, such as Message Digest algorithm 5 (MD5)[31] and Secure Hash Algorithm 1 (SHA-1)[32], are used instead. The probability of finding a collision for MD5 w.r.t 128-bit output and that for SHA-1 w.r.t 160-bit output have been estimated as, on average, 2^{64} and 2^{80}, respectively. As long as these probabilities hold, it is difficult to find the same Temp ID for different nodes in each session of communication

Position of Destination (PD): The geographical position ($\mathrm{X_T}, \mathrm{Y_T}$) of the destination, taken from PS and encrypted by $\mathrm{C_K}$.

Number of Hops (NH): NH is the minimum number of hops that an *RRQ* packet travels to find a route from the source to the destination. NH is estimated by the *source*. It is changed by the source when the source tries to find a route with a new estimation. It is also encrypted by $\mathrm{C_K}$.

Temporary Number of Hops (Temp NH): At the beginning of route discovery, Temp NH is initiated as NH by the source, Temp NH = NH and it is encrypted with $\mathrm{C_K}$ by the source. After receiving the *RRQ* packet by legitimate nodes, it is updated. Update means decrementing by *one*, i.e., Temp NH = Temp NH - 1. When the *RRQ* packet travels from node to node it is updated each time by each node. Moreover, the nodes perform encryption/decryption operations and vice-versa by $\mathrm{C_K}$.

4. AODPR protocol

4.1 Parameters description

In certain environments, such as stadiums, classrooms, disaster areas, and battle fields, node placements and their corresponding density can be defined as follows.

Quadratic placement means that a node is connected in its radio range with its neighbors in all four compass directions from its center (Fig. 4): thus, their corresponding densities are approximated as $\mu_{quad} \approx \sqrt{n}/[\{\pi + (\sqrt{3}/2) + 1\} \times R^2]$, where n is the number of nodes to make the connection and R is the radius of the maximum radio-range coverage of each node of the ad-hoc network. When any node sends a packet within its radio range, the other nodes within its radio range can receive the packets. *Line placement* means that a node can be connected to any node in a line via intermediate nodes. *Least placement* means that a node can reach another node with just one connection to its neighbor (Fig. 5).

At first, we describe the estimation of NH by the source for different placements of the nodes in the network. The source estimates NH on the basis of the density of the nodes in the network, and NH is the highest when node density is the lowest and vice-versa. NH is thus proportional to $1/\mu$, where μ is the density of the nodes.

For *line placement*, NH = D/R, where R is the radius of the maximum radio-range coverage of each node of the ad-hoc network, D is the distance from the source to the destination, $\mathrm{D} = \sqrt{(X_T - X_S)^2 + (Y_T - Y_S)^2}$, where $(\mathrm{X_S}, \mathrm{Y_S})$ and $(\mathrm{X_T}, \mathrm{Y_T})$ are the source's and destination's positions, respectively. In this placement, NH is the minimum number of hops, from the source to the destination, estimated by the source.

For μ_{qaud} it is assumed that $\mathrm{NH} = \mathrm{f(L, B)/R}$, where f is a linear function in L and B, length L is the horizontal distance from the source to destination, and breadth B is the vertical distance from the source to destination.

For *least placement*, it is assumed that $\mathrm{NH} = (\mathrm{k} \times \mathrm{g(C)})/\mathrm{R}$, where k is a constant and a function of L/R or B/R; and g is an exponential function in circumference C of the area of the network. In this placement NH is the maximum number of hops, from the source to the destination.

4.2 AODPR overview

The AODPR protocol is described in detail with respect to the functionalities of the nodes.

Source: The source sends a request to the PS for the position information of the destination when it wants to communicate with the destination. AODPR is thus an on-demand protocol. The source generates its own Temp ID, *RRQSeqNo* and estimates NH and the maximum number of hops.

After receiving the destination's position, the source estimates NH and assigns this NH to Temp NH. It then source sends an *RRQ* packet within its radio range and waits to receive a *response*, which is either *RRP* or *Fail* during time $2 \times TTL$, where TTL denotes *time to leave* and is estimated by the source from $TTL = ($*traveling time for one hop*$) \times ($*number of hop*$)$.

- If the source receives *RRP*, by decrypting *RRPMsg* of *RRP*, it tries to find a match with *Auth*. If a match is found, it stores the corresponding *RRQSeqNo*, NH, receiver's Temp ID and status (i.e., "yes") in its routing table. It then sends data encrypted by the destination's public key. Lastly sender sends *Token* to the destination so that destination can inform the PS of this communication.
- If it receives a *Fail* packet, it stores the corresponding *RRQSeqNo*, NH, and status ("no") to its routing table, and again tries with a new estimated NH.
- If it does not receive any *response* and TTL is exceeded, it stores *RRQSeqNo*, NH and status ("no") in its routing table, and again tries with a new estimated NH.

As a result of this procedure, if the source fails to find the destination with an estimated

NH, it tries with the next estimated NH until it finds the route. In this way, it can try with the minimum to the maximum estimated NH. Moreover, the maximum number of hops can be varied for different placements.

***Intermediate nodes or Forwarders*:** If a node receives a packet *RRQ* but it is not the destination it is a *forwarder* and becomes a new sender. Forwarder F generates its own Temp ID and calculates distance $D_r(F)$ between F and its destination T by $D_r(F) = \sqrt{(X_T - X_F)^2 + (Y_T - Y_F)^2}$ from the forwarder's position $(\mathrm{X_F}, \mathrm{Y_F})$ and destination's position $(\mathrm{X_T}, \mathrm{Y_T})$. F then updates Temp NH by Temp NH = Temp NH - 1. It compares this updated Temp NH with $D_r(F)/R$ and makes the following decision, as shown schematically in **Fig. 2**.

- If $D_r(F)/R \leq$ Updated Temp NH, forwarder F forwards the packet to its radio region and keeps the route information.
- If $D_r(F)/R >$ Updated Temp NH, forwarder F discards the packets.

After forwarding a packet, the forwarder waits to receive a *response* for time 2 × TTL_1, where TTL_1 is computed from TTL_1 = (*traveling time for one hop*) × (*updated number of hops*).

- If the forwarder receives *RRP*, it just forwards it on the reverse path and keeps the route information.
- If the forwarder receives *Fail*, it also forwards it on the same reverse path and keeps the route information.
- If it does not receive any *response* and its waiting time exceeds TTL_1, it generates *Fail* and forwards it on the reverse path.

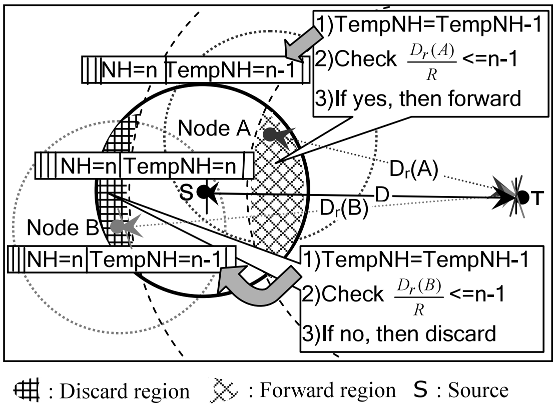

Fig. 2 Packet forwarding or discarding in intermediate nodes.

***Destination*:** The destination checks EM of *RRQ* to confirm the destination of *RRQ* (see Appendix A.1). Finally, it replies by *RRP* and keeps the route information.

AODPR protocol

Carrier sense multiple access with collision avoidance (CSMA/CA)[33] is used as the channel-access mechanism for control messages. A sender (a source or a forwarder) of an *RRQ* transmits the *RRQ* after sensing the channel and finding idle time for a distributed inter frame space (DIFS). When there is a collision, the sender retransmits the *RRQ* after a short inter frame space (SIFS). The same procedure is applicable for any node for the *RRP* as well as *Fail.*

***Initial procedure*:**

A source makes a signed position request to the PS, and receives required information $\mathrm{C_K}$, destination's *position information*, *Auth*, *Token*, and PK of the destination from the PS.

***Source's working procedure*:**

The source generates an *RRQ* and sends it to its radio region and waits to receive a *response* for time 2 × *TTL.*

If it receives a following *response*

- If source receives *RRP*, then it compares *Auth* with *RRPMsg* by decrypting it.
  - If it matches, then source sends data in the path and at last sends the *Token.*
  - If it does not match, then source discards this *RRP* and estimates a new NH and again tries this procedure until it receives a valid *RRP.*
- If source receives a *Fail* packet within time 2 × *TTL*, it estimates a new NH and again tries this procedure until it receives an *RRP that does not exceed the maximum number of hops for that environment.*

If the source does not receive any response and the waiting time exceeds 2 × *TTL*, the source estimates a new NH and again tries the above procedure until it receives an *RRP*. The source repeats this procedure as long as the NH of its packet is smaller than the *maximum number of hops for that environment.*

***Forwarder's or destination's working procedure*:**

On receiving an *RRQ*, a forwarder checks whether it is the destination or not.

If it is the destination, then it generates an *RRP* and sends this *RRP* on the reverse path.

If it is not the destination, then it forwards the *RRQ* and waits for time 2 × TTL_1

- If the forwarder receives a *RRP*, it keeps the route information and sends it on the reverse path.
- If the forwarder receives *Fail*, then it keeps the route information and sends it on the reverse path.

If the waiting time for the forwarder exceeds $2 \times TTL_1$ time, then the forwarder generates *Fail* and sends it on the reverse path.

5. Anonymity achievement and security analysis

When senders or forwarders forward any packets, they generate a large bit random number and use parts of that random bit sequence corresponding to the number of encrypted fields of the packet, i.e., *RRQ* and *RRP*. The packets are described in the appendix. They also specify all the fields with a specific bit number. They then pad the fields with random bits and encrypt these fields. When a packet reaches a node, the node first decrypts it, extracts the random bits from the fields, and pads these fields with its own random bits. All the fields of a packet are thus changed. As a result, when the packet moves from node to node it appears new to the network. This procedure is applicable to all the encrypted fields of all the packets. Encryption and decryption are performed as necessary when a packet moves from node to node.

In an ad-hoc security routing protocol, the most expensive operation is the public key operation [34]. To guarantee the anonymity in the AODPR, every node generates its Temp ID, which is a hash computation, and a random bit corresponding to the fields of the packets, and it finally performs symmetric encryption/decryption of the fields. These computations are not more computationally complex than those of some other ad-hoc security routing protocol [30].

***Identity Privacy*:** In AODPR the identities temp ID of the nodes are changing in each hop as a packet is forwarded. Location of destination is encrypted and padded with random bits. Also the temp ID is changed in each session of communication. As we discussed before in Section 3.3 the temp ID depends on not only the position of the node and the public key but also on time, so it is changeable within a hop range. So AODPR ensures identity privacy.

***Location Privacy*:** The general concept of the current attacks on the location privacy is to observe the route request and route response packets and to estimate the distance between the source and the destination from the traveling information added to the packet, i.e., how many hops it travels. In contrast to existing anonymous ad-hoc routing protocols, there is no extra traveling information added to the packets in our scheme, and estimating the distance between the source and the destination is not possible in a straightforward way. No node knows anything about the location and identity of the other nodes, including the source, and it does not know from where a packet starts to travel in the network. Even though all legitimate nodes can determine the distance from themselves to the destination and also know the temp ID of other nodes in the neighboring region, no one except the source can determine the distance from the source to the destination by using this information. Location privacy is thus achieved.

***Route Anonymity*:** Current attacks on route anonymity are based on traffic analysis [22]. The general theory behind these kinds of attacks is to trace or to find the path in which the packets are moving. For this purpose, a malicious node, mainly looks for unchangeable information i.e., common information in a packet, so that it can trace the movement of control packets. As a result, the adversaries can find or estimate the route from the source to the destination. In AODPR, all the control packets appear new (**Fig. 3**) in the network when packets moves form node to node. So no one can trace the path of the route. Route anonymity is thus achieved. A detailed description is given in the appendix.

***DoS*:** Multiple adversaries cooperatively or one adversary with enough power can exhaust the resource of a specific target node. To this end, adversaries need to identify a node and set that specific node as a target. In AODPR, identity privacy is achieved as discussed above and *DoS* can be protected.

***Wormhole Attacks*:** In wormhole attack,

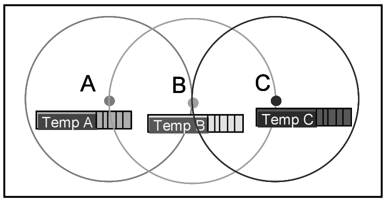

Fig. 3 Route anonymity model.

there could be a long distance for a packet to travel for finding the route from the source to the destination. In AODPR, the source and the forwarders wait for a *limited time*, *TTL or* TTL_1, for getting a response based on the estimated NH. If an attacker's response exceeds a *limited time*, it cannot be a forwarder within a routing path. If the attacker is a forwarder within a path limit and does not reply properly, this path no longer remains valid. The sender will try another path. A wormhole attack is therefore not effective in the case of AODPR.

***Rushing Attack*:** Many existing on-demand routing protocols only forward the request that arrives first from each route discovery. In a rushing attack, the attacker exploits this property of the operation of route discovery, and establishes a rushing attack. A more powerful rushing attacker may employ a wormhole to rush packets. By using the tunnel of a wormhole attack, the attacker can introduce a rushing attack. As shown above, AODPR can prevent a wormhole attack. It is thus also robust against a rushing attack.

6. Performance analysis

6.1 Theoretical analysis

In the case of AODPR, the source can determine the direct distance from him to any node connected in the network. Let the distance from the source to a node be D, so the number of hop given by $h = D/R$, where R is the radio-range coverage around a node. For route discovery, when a control packet travels from hop to hop, h is decremented by one. When a packet is forwarded to a specific node, the values of h will thus converge to a smaller value than its previous value. Let t be the time a packet needs to travel h number of hops, within time $2 \times t$, the source will receive a response. If the source does not receive any response, it will estimate new hop h_1 and will wait for a corresponding traveling time $2 \times t_1$. Thus, by consecutive estimation of new hops, the source reaches to the goal, as long as there is at least one path to reach the goal. If the density of the network is more than the quadratic-placement density μ_{quad} (**Fig. 4**), it can reach the goal directly. If there is a shield on the path, it is also informed to the source by sending a fail packet after a certain amount of time. The source therefore estimates a new hop number by increasing its value more than in the previous attempts. If the source fails again, it will try as previously

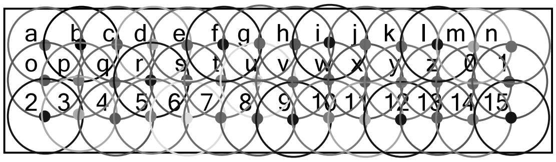

Fig. 4 Quad-placement-connected network.

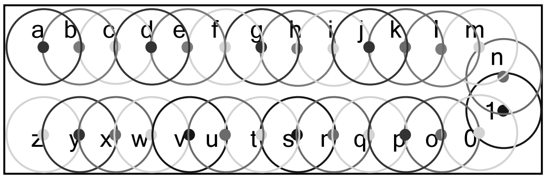

Fig. 5 Least-placement-connected network.

with a new estimate. If there is at least one path from the source to a node, then it can be found out, and successful communication is accomplished.

If the nodes in the network are at least-connected as shown in **Fig. 5**, the maximum hop count to reach the goal is $n - 1$, where n denotes the number of nodes in a network. Let us consider a path from node **a** to **z**, as an example path. At first **a** will calculate $D_r(a)$ from node **a** to **z** and also estimates NH, so that the packet travels according to the protocol for this NH. If the relation $D_r(F)/R >$ Updated Temp NH holds for a node in the path, that node discards the packet. After that, node **a** sends the packet with a new estimated NH, which is a value greater than the old NH. Either with current estimated NH or a new estimated NH on the consecutive estimation of NH, the relation $D_r(F)/R >$ Updated Temp NH does not hold for that node any more and the node finally forwards the packet. As long as the NH of a packet from **a** is smaller than the maximum hop count, this procedure will continue. By taking an appropriate value for the maximum hop count, the packet can reach from node **a** to node **z**. The simulation results of least connected nodes in a network are given in Section 6.2 with a different estimation of NH.

6.2 Simulation result

The reach ability in a network with least placement was simulated by varying the number of nodes, as shown in **Fig. 6**, under a C++ programming environment. The graph shows the number of trials with respect to the number of nodes, in different estimation. For all the estimation methods the source at first initializes NH = D/R. With this initial value the source

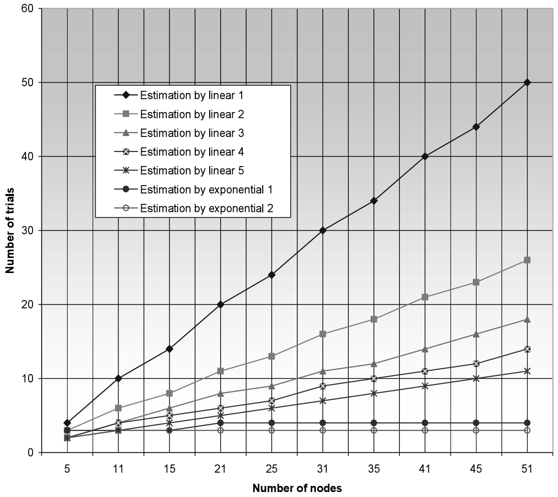

Fig. 6 Number of trials for different estimation methods to find a route for different numbers of nodes in a least-placement-connected network.

tries to reach the destination. If the source fails, it estimates a new NH value and tries to reach to the goal with this value. Each time the source tries to reach the goal, the trial number is counted. For estimating NH value, we experimented with seven estimation functions. For all the estimation functions the *estimation value* is initialized by NH = D/R. These functions are mainly defined in two ways, (i) linear or (ii) exponential described as follows.

Estimation by linear I ($I = 1$ to 5): After initializing the estimation value, it is incremented by I, so *estimation value* = *estimation value* + *I*. Detailed results for various I (I = 1 to 5) are shown in Fig. 6.

Estimation by exponential I ($I = 1, 2$): After initializing the estimation value, it is incremented as a power, so *estimation value* = (*estimation value*)$^{I+1}$. When I = 1, the source tries four times for 21 numbers of nodes to reach the goal and for 51 numbers of nodes, it tries four times, but the trial value for 5 to 15 numbers of nodes differs from the previous value and it is 3. When I = 2, the source tries three times to reach the goal for 21 numbers of nodes, and for 51 numbers of nodes, it also tries three times and it remains constant from any number of nodes from 5 to 51. Exponential 2 is thus the best estimation for a least-placement-connected network.

7. Conclusions

Anonymity is one of the important factors in securing a mobile ad-hoc network routing. We thus proposed an anonymous on-demand routing protocol, called AODPR, for preventing a target-oriented attack, which is applicable to most node densities in a network. Moreover, AODPR ensures node privacy, route anonymity, and location privacy and is robust against most known attacks. As a further research, we plan to make a theoretical analysis of an efficient estimation-function for any connected network, and simulation of that function in detail.

References

1) Lou, W. and Fang, Y.: A Survey on Wireless Security in Mobile Ad Hoc Networks: challenges and available solutions, Book chapter in *Ad Hoc Wireless Networking*, Kluwer (May 2003).

2) Guan, Y., Fu, X., Xuan, D., Shenoy, P., Bettati, R. and Zhao, W.: NetCamo: Camouflaging Network Traffic for QoS-Guaranteed Mission Critical Applications, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.31, No.4, pp.253–265 (July 2001).

3) DARPA: Research Challenges in High Confidence Networking (July 1998).

4) Berg, O., Berg, T., Haavik, S., Hjelmstad, J. and Skaug, R.: *Spread Spectrum in Mobile Communications*, IEEE (1998).

5) Jiang, S., Vaidya, N. and Zhao, W.: Prevent Traffic Analysis in Packet Radio Networks, *Proc. DARPA Information Survivability Conference and Exposition* (*DISCEX II'01*), Vol.II–Vol.2, pp.1153–1158 (June 2001).

6) Perkins, C., Belding-Royer, E. and Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561 (July 2003).

7) Johnson, D.B., Maltz, D.A. and Hu, Y.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), <draft-ietf-manet-dsr-09.txt> (Apr. 2003).

8) Royer, E. and Toh, C-K.: A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks, *IEEE Personal Communications Magazine*, pp.46–55 (Apr. 1999).

9) Ko, Y. and Vaidya, N.: Location-Aided Routing in Mobile Ad Hoc Networks, *Proc.4th International Conference on Mobile Computing and Networking*, Dallas, USA, pp.66–75 (1998).

10) Morris, R. and De Couto, D.: Location Proxies and Intermediate Node Forwarding for Practical Geographic Forwarding, Technical Report MIT-LCS-TR-824, MIT Laboratory for Computer Science (June 2001).

11) Dahill, B., Levine, B., Royer, E. and Shields, C.: A Secure Routing Protocol for Ad Hoc Net-

works, University of Massachusetts Technical Report 01-37 (2001).

12) Basagni, S., Chlamtac, I., Syrotiuk, V. and Woodward, B.: A Distance Routing Effect Algorithm for Mobility (DREAM), *Proc. 4th International Conference on Mobile Computing and Networking*, Dallas, USA, pp.76–84 (1998).

13) Karp, B. and Kung, H.: Greedy Perimeter Stateless Routing for Wireless Networks, *Proc. 6th International Conference on Mobile Computing and Networking*, Boston, USA, pp.243–254 (2000).

14) El-Khatib, K., Korba, L., Song, R. and Yee, G.: Secure dynamic distributed routing algorithm for ad hoc wireless networks, *International Conference on Parallel Processing Workshops* (*ICPPW'03*) (2003).

15) Reed, M.G., Syverson, P.F. and Goldschlag, D.M.: Anonymous connections and onion routing, *IEEE Journal on Selected Areas in Communications*, Vol.16, No.4, pp.482–494 (1998).

16) Kong, J. and Hong, X.: ANODR: Anonymous on- demand routing with untraceable routes for mobile ad-hoc networks, *Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (*MobiHoc'03*), pp.291–302 (2003).

17) Brian Neil Levine and Clay Shields: Hordes: a multicast based protocol for anonymity, *Journal of Computer Security* Vol.10, Issue 3, pp.213–240 (2002). ISSN: 0926-227X.

18) Wu, X.: DISPOSER: Distributed Secure Position Service in Mobile Ad Hoc Networks, Technical Report CSD TR # 04-027, Dept. Computer Sciences (2004).

19) Xue, Y., Li, B. and Nahrstedt, K.: A Scalable Location Management Scheme in Mobile Ad-Hoc Networks, *Proc. 26th IEEE Annual Conference on Local Computer Networks* (*LCN 2001*), Tampa, Florida, pp.102–111 (Nov. 2001).

20) Wu, X.: VDPS: Virtual Home Region based Distributed Position Service in Mobile Ad Hoc Networks, *Proc. ICDCS* (2005).

21) Wu, X. and Bhargava, B.: AO2P: Ad Hoc On-Demand Position-Based Private Routing Protocol, *IEEE Transactions on Mobile Computing*, Vol.4, No.4, pp.335–348 (July/Aug. 2005).

22) Raymond, J.-F.: Traffic Analysis: Protocols, Attacks, Design Issues and Open Problems, *Proc. PET 01*, LNCS, Vol.2009, pp.10–29, Springer-Verlag (2001).

23) Hu, Y.C., Perrig, A. and Johnson, D.B.: Packet leashes: A defense against wormhole attacks in wireless ad hoc networks, *Proc. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM 2003*) (2003).

24) Ko, Y. and Vaidya, N.: Location-Aided Routing (LAR) in Mobile Ad Hoc Networks, *Proc. Fourth International Conference on Mobile Computing and Networking* (*MobiCom'98*), pp.66–75 (Oct. 1998).

25) Hu, Y., Perrig, A. and Johnson, D.B.: Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks, *Proc. Eighth Annual International Conference on Mobile Computing and Networking* (*MobiCom 2002*), pp.12–23 (Sep. 2002).

26) Zapata, M.G. and Asokan, N.: Securing Ad Hoc Routing Protocols, *Proc. ACM Workshop on Wireless Security* (*WiSe 2002*) (Sep. 2002).

27) Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C. and Royer, E.B.: A Secure Routing Protocol for Ad hoc Networks, *Proc. 10th IEEE International Conference on Network Protocols* (*ICNP '02*) (Nov. 2002).

28) Castelluccia, C. and Montenegro, G.: Protecting AODV against Impersonation attacks, *Mobile Computing and Communications Review*, Vol.6, No.3, pp.108–109 (2002).

29) Papadimitratos, P. and Haas, Z.J.: Secure Routing for Mobile Ad Hoc Networks, *Communication Networks and Distributed Systems Modeling and Simulation Conference* (*CNDS 2002*) (Jan. 2002).

30) Carter, S. and Yasinasc, A.: Secure Position Aided Ad hoc Routing Protocol, *Proc.IASTED International Conference on Communications and Computer Networks* (*CCN02*), pp.329–334 (Nov. 2002).

31) Rivest, R.L.: The MD5 Message Digest Algorithm, Internet RFC 1321 (Apr. 1992).

32) NIST: FIPS 180-1, Secure hash standard, US Department of Commerce, Washing D.C. (Apr. 1995).

33) Technical Report on the IEEE 802.11 Protocol.

34) Zhang, Y.: Security in Mobile Ad-hoc networks, *Ad hoc Networks technologies and protocols*, Mohapatra, P. and Krishnamurthy, S. (Eds.), Springer, ISBN 0-387-22689-3, pp.249–268 (2005).

Appendix

Here packets are described. Common key $\mathrm{C_K}$ is used for encryption and decryption by all legitimate nodes. E_{C_K}: means encryption with $\mathrm{C_K}$.

A.1 Route Request Packet (*RRQ*)

Sender Temp ID	E_{C_K} (*RRQSeqNo*)	E_{C_K} (PD)	E_{C_K} (NH)	E_{C_K} (Temp NH)	E_{C_K} (EM)

For construction purposes when senders or forwarders forward any packet, they generate a large bit random number and make parts of that random bit corresponding to the number of fields of the packet. And they specify all the fields with a specific bit number. They then encrypt these fields by padding with random bits. When a packet reaches a node, the node first decrypts and extracts the random bits from the fields and pads their own random bits. As all the fields of a packet are changed, when a packet moves from node to node, it appears new to the network. This procedure is applicable to all the encrypted fields of all the packets. Encryption/decryption is performed as necessary when a packet moves from node to node.

RRQSeqNo**:** Route request sequence number *RRQSeqNo* generated by the source uniquely, for the uniqueness of a session.

Ensure Message (EM): This examines the genuineness of the destination. The source generates an EM when it receives the destination's position. EM = [H_2 (position of destination, time)], where H_2 is the global hash function.

A.2 Route Reply Packet (*RRP*)

E_{C_K} (RRQSeqNo)	Sender Temp ID	Receiver Temp ID	*RRPMsg*

Receiver Temp ID: For every session of communication, an intermediate node or the destination generates its Temp ID in the same procedure as the sender Temp ID. Temp ID is the only identification of a node in one session of communication. It is dynamically changeable from session to session. When packets are forwarded, this field is updated by nodes according to their own Temp ID.

A.3 Fail Packet, (*Fail*)

E_{C_K} (RRQSeqNo)	Sender TempID	Receiver Temp ID	E_{C_K} (NH)

(Received November 28, 2005)
(Accepted June 1, 2006)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.2, pp.524–536.)

Sk. Md. Mizanur Rahman received B.Sc. (Honors) and M.Sc. degrees in computer science, securing first class first with honors and received Gold Medal, from the Institute of Science and Technology (I.S.T), National University, Bangladesh, in 1997 and 1998, respectively. Since 1999, he had been in the I.S.T. as a Teaching Assistantship as well as a computer programmer in the Democracy-watch, Bangladesh. Since 2002, he had been in the I.S.T. as a Lecturer and in the Stamford University Bangladesh as a lecturer in the Department of Computer Science and Engineering since 2003. Currently, he is a Ph.D. degree student in the Graduate School of Systems and Information Engineering, University of Tsukuba, Japan. His research interests are security network, algorithms, cryptography etc. He is a member of BCS, Bangladesh, and a student member of Information Processing Society Japan (IPSJ).

Atsuo Inomata received M.S. and Ph.D. degrees in information science from Japan Advanced Institute of Science and Technology in 1999 and 2002, respectively. He worked and studied optical network architecture for Japan Telecom Information and Communication Laboratories since 2002. Since 2004, he has been a researcher at Japan Science and Technology agency (JST.GO.JP). His research interests are network security, cryptography theory, and hardware implementation.

Masahiro Mambo received a B.Eng. degree from Kanazawa University, Japan, in 1988 and M.S.Eng. and Dr.Eng. degrees in electronic engineering from Tokyo Institute of Technology, Japan in 1990 and 1993, respectively. He worked at Japan Advanced Institute of Science and Technology, JAIST, from 1993 to 1997 and at Tohoku University from 1997 to 2004. Then he joined University of Tsukuba in 2004. He is currently an associate professor of Department of Computer Science, Graduate School of Systems and Information Engineering. His research interests include information security, software protection, and privacy protection.

Eiji Okamoto received B.S., M.S. and Ph.D. degrees in electronics engineering from the Tokyo Institute of Technology in 1973, 1975, and 1978, respectively. He worked and studied communication theory and cryptography for NEC central research laboratories since 1978. From 1991, he was a professor first at Japan Advanced Institute of Science and Technology and then at Toho University. Now he is a professor at Graduate School of Systems and Information Engineering, University of Tsukuba. His research interests are cryptography and information security. He is an editor-in-chief of International Journal of Information Security, and an associate editor of IEEE Information Theory Society.