Defense Against Injecting Traffic Attacks in Wireless Mobile Ad-Hoc Networks

Wei Yu and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—In ad-hoc networks, nodes need to cooperatively forward packets for each other. Without necessary countermeasures, such networks are extremely vulnerable to injecting traffic attacks, especially those launched by insider attackers. Injecting an overwhelming amount of traffic into the network can easily cause network congestion and decrease the network lifetime. In this paper, we focus on those injecting traffic attacks launched by insider attackers. After investigating the possible types of injecting traffic attacks, we have proposed two sets of defense mechanisms to combat such attacks. The first set of defense mechanisms is fully distributed, while the second is centralized with decentralized implementation. The detection performance of the proposed mechanisms has also been formally analyzed. Both theoretical analysis and experimental studies have demonstrated that under the proposed defense mechanisms, there is almost no gain to launch injecting traffic attacks from the attacker's point of view.**

***Index Terms*—Ad-hoc networks, attack models, network security.**

I. Introduction

A MOBILE ad-hoc network is a group of mobile nodes without a fixed network infrastructure, and nodes can communicate with other nodes out of their direct transmission ranges by cooperatively forwarding packets for each other. Since ad-hoc networks can be easily deployed as needed, they have a wide of range of applications. However, before ad-hoc networks can be successfully deployed, security concerns must be resolved first [1]–[8]. In this paper, we study a class of powerful attacks: injecting traffic attacks. Specifically, attackers inject an overwhelming amount of traffic into the network in an attempt to consume valuable network resources and, consequently, degrade the network performance. Since nodes need to cooperatively forward packets for other nodes in ad-hoc networks, such networks are extremely vulnerable to injecting traffic attacks, especially those launched by insider attackers.

Roughly speaking, injecting traffic attacks can be classified into two types: 1) query-flooding attack and 2) injecting data packets attack (IDPA). Due to the changing topology or traffic pattern, nodes in ad-hoc networks may need to frequently update their routes, which may require broadcasting route query messages. Then attackers can broadcast query messages with a very high frequency to consume valuable network resources. We call such attacks query-flooding attacks. Besides query-flooding attacks, attackers can also inject an overwhelming amount of data packets into the network to request other nodes to forward. When other nodes process and forward these packets, their resources (e.g., energy) are wasted. We call such attacks injecting data-packet attacks (IDPA). Since, in general, the size of a data packet is much larger than the size of a route query message, and the injection rate of data packets is usually much higher than the injection rate of route query messages, the damage that can be caused by injecting data packet attacks is usually more severe than by query-flooding attacks.

To defend against query-flooding attacks, we can limit the amount of queries that each node can initiate. Although this may degrade the network performance in a certain degree, this method can effectively limit the damage that can be caused by query-flooding attacks. On the other hand, if nodes in the network cannot know other nodes' data packet injection rates, it will become extremely hard or even impossible to detect injecting data packet attacks. In this work, we focus on the scenario that nodes in the network belong to the same authority and pursue some common goals. Therefore, each node's traffic injecting pattern can usually be estimated by at least a subset of nodes in the network, such as those sinks in ad-hoc sensor networks. To handle injecting traffic attacks in ad-hoc networks where nodes belong to different authorities and pursue different goals, interested readers please refer to [9].

In this paper, we first propose a set of fully distributed defense mechanisms which can effectively detect injecting data-packet attacks. The proposed mechanisms can even work well when attackers can use advanced transmission techniques, such as directional antennas, to avoid being detected. We then derive the theoretical upperbounds for the probability that attackers can successfully launch injecting data packet attacks without being detected. The results show that from the attackers' point of view, the best injecting data-packet attack strategy is to conform to their legitimate data packet injection rates. In other words, the best attacking strategy is not to launch injecting data-packet attacks. To decrease the storage overhead and further increase the attacker detection performance, we then propose a centralized defense mechanism with decentralized implementation. This is achieved by letting some nodes under strong protection perform attacker detection. Besides injecting data packet attacks, the query-flooding attacks have also been studied and the tradeoff between limiting the query rates and the system performance has been exploited.

Manuscript received June 1, 2006; revised February 23, 2007. This work was supported by the Army Research Office under URI Award DAAD19-01-1-0494. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Klara Nahrstedt.

W. Yu was with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: weiyu@isr.umd.edu). He is now with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: weiy@microsoft.com).

K. J. R. Liu is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: kjrliu@isr.umd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2007.897269

The rest of this paper is organized as follows. Section II reviews some related work. Section III describes the system model and investigates the possible types of injecting traffic attacks. Section IV describes the proposed fully distributed defense mechanisms. The theoretical analysis of the proposed distributed defense mechanisms is presented in Section V. In Section VI, a centralized detection mechanism with decentralized implementation is described. To confirm the effectiveness of the proposed mechanisms, we have conducted extensive simulation experiments which are presented in Section VII. Finally, Section VIII concludes this paper.

II. Related Work

To secure ad-hoc networks, the first step is to prevent attackers from entering the network through secure key distribution and secure route and neighbor discovery, such as [1], [5], [6], and [10]–[14]. In [1], Zhou and Haas investigated distributed certificate authorities in ad-hoc networks using threshold cryptography. In [4], Hubaux *et al.* developed the idea of self-organized public-key infrastructure similar to PGP in the sense that public-key certificates are issued by the users. The difference with PGP is that in their system, certificates are stored and distributed by the users. In [15], Capkun *et al.* discussed how to build security associations with the help of mobility in mobile ad-hoc networks.

Besides injecting traffic attacks, routing disruption attacks can also be a severe threat to ad-hoc networks. Roughly speaking, routing disruption attacks mean that attackers attempt to cause legitimate data packets to be routed in a dysfunctional way and, consequently, cause packets to be dropped or extra network resources to be consumed. In the literature, many schemes have been proposed to handle such attacks. For example, Papadimitratos and Haas [10] proposed a secure routing protocol for mobile ad-hoc networks that guarantees the discovery of correct connectivity information over an unknown network in the presence of malicious nodes. Sanzgiri *et al.* [11] considered a scenario that nodes authenticate routing information coming from their neighbors while not all of the nodes on the path will be authenticated by the sender and the receiver. Hu *et al.* [5] proposed Ariadne, a secure on demand ad-hoc network routing protocol, which can prevent attackers or compromised nodes from tampering with uncompromised routes that consist of uncompromised nodes. In [6] and [13], they describe how to defend against rushing attacks through secure neighbor discovery and how to apply packet leashes to defend against wormhole attacks. Later, Capkun and Habaux [16] investigated secure routing in ad-hoc networks in which security associations exist only between a subset of all pairs of nodes. Aad *et al.* [7] studied DoS resilience in ad-hoc networks, where two attacks are studied—black hole and JellyFish.

Once attackers have entered the network, the schemes based on secure key distribution and secure route discovery will become ineffective. In these situations, schemes based on monitoring traffic in the network can be used to detect malicious nodes and to limit the damage, such as [2], [3], and [17]–[21]. Initial work using these mechanisms was proposed by Marti *et al.* [3]. They considered the case that nodes agree to forward packets but fail to do so, and proposed two tools that can be applied upon source routing protocols—watchdog and pathrater. However, this system suffers some problems. First, many attacks can cause a malicious behavior from not being detected, such as ambiguous collisions, receiver collisions, limited transmission power, collusion, and partial dropping, and malicious nodes can easily propagate false information to slander good nodes. In [17] and [21], the authors extended the ideas in [3] and allowed the reputation to propagate throughout the network. However, since these schemes still rely on watchdog, they also suffer the same types of problems as [3]. Furthermore, once the reputation is allowed to propagate, attackers can also collude to frame up or blackmail other nodes. In [2], Zhang and Lee discussed intrusion detection in wireless ad-hoc networks. They examined the vulnerabilities of a wireless ad-hoc network, then introduced multilayer-integrated intrusion detection and response mechanisms. However, they have not described specific mechanisms to secure ad-hoc networks.

Some other related work appeared in [9] and [18]–[20]. In these papers, instead of cooperative ad-hoc networks, the authors considered the scenario that nodes in the network are selfish which are not willing to forward packets on the benefits of other nodes. They propose schemes to stimulate cooperation among selfish nodes based on a credit system or game theory. However, those schemes cannot handle the situations with the presence of malicious nodes, whose objective is to maximize the damage they cause to the network, instead of maximizing their own benefits obtained from the network.

III. Injecting Traffic Attacks

In this paper, we focus on ad-hoc networks with nodes belonging to the same authority and on pursuing some common goals. Nodes in such networks can be classified into two types: good and malicious. Good nodes will unconditionally help other good nodes to achieve the common goals, while malicious nodes will try to degrade the network performance as much as possible. Each node is equipped with a battery with limited power supply, communicates with other nodes through wireless connections, and can move freely inside a certain area. We focus on the most general scenario that good nodes use omnidirectional transmission techniques. However, in our setting, attackers are allowed to use directional transmission techniques, such as directional antennas [22] or adaptive beamforming [23], to improve their attacking capability.

According to the common system goal, each node may be required to generate a sequence of packets to be delivered to certain destinations. For example, in wireless ad-hoc sensor networks, each node may need to periodically send the sensed information back to the data sinks. We say a source-destination pair legitimate if this pair is required by the common system goals. For each legitimate source-destination pair (s, d) in the network, we assume that the number of packets that is required to be delivered by this pair until time t is $N_{s,d}(t)$. In general, the exact value of $N_{s,d}(t)$ may not be known *a priori* by the other nodes in the network. To overcome this difficulty, in this paper, we make an assumption that the upperbound of $N_{s,d}(t)$, denoted by $f_{s,d}(t)$, can be estimated by some other nodes in the network. From now on, $f_{s,d}(t)$ will be referred to as the upperbound of

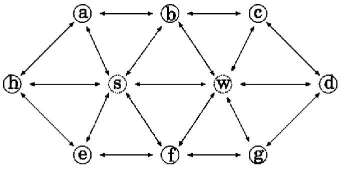


Fig. 1. Example of long-route attack.

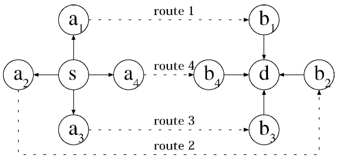


Fig. 2. Example of multiple route attack.

the traffic injection rate associated with the source-destination pair (s, d).

In this paper, we mainly focus on insider attackers. That is, all nodes in the network are legitimate, regardless of whether they are good or malicious. To handle outside attackers, access control and secret communication channels can usually work well. We assume that each node has a public/private key pair. We also assume a node can know or authenticate other nodes' public keys. However, no node will disclose its private key to the others unless it has been compromised. To maintain the confidentiality and integrity, each packet may be encrypted and signed by its sender when necessary. Without loss of generality, we simply assume that all data packets have the same size.

As mentioned before, in this paper, our focus is to defend against injecting traffic attacks, or more specifically, IDPA and query-flooding attacks. We first consider the possible ways that IDPA can be launched by attackers s and d with s being the source and d being the destination. The simplest way, which is called simple IDPA, is that s picks a route R to d and injects an overwhelming amount of packets into the network, with the injection rate being much higher than the legitimate upperbound $f_{s,d}(t)$.

In the second way, which is called long-route IDPA, the source s picks a very long route to inject data packets into the network. For example, as in Fig. 1, s can pick the route "$s \rightarrow w \rightarrow c \rightarrow b \rightarrow a \rightarrow h \rightarrow e \rightarrow f \rightarrow g \rightarrow d$" to send packets from s to d, and meanwhile keep the number of injected packets below the legitimate upperbound $f_{s,d}(t)$. By acting in this way, s and d can achieve the same effect as increasing its traffic injection rate.

In the third and advanced way, which is called multiple routes IDPA, the source s picks multiple routes to d and simultaneously injects traffic into the network via these routes. For example, as shown in Fig. 2, s uses four routes "$s \rightarrow a_1 \rightarrow \cdots \rightarrow b_1 \rightarrow d$," "$s \rightarrow a_2 \rightarrow \cdots \rightarrow b_2 \rightarrow d$," "$s \rightarrow a_3 \rightarrow \cdots \rightarrow b_3 \rightarrow d$," and "$s \rightarrow a_4 \rightarrow \cdots \rightarrow b_4 \rightarrow d$" to inject packets into the network. By doing it this way, the traffic can be distributed among multiple routes such that for each route, the packet injection rate is no higher than the legitimate upperbound $f_{s,d}(t)$ though the total number of injected packets can be much higher than the legitimate upperbound $f_{s,d}(t)$. Moreover, the attackers can also take advantage of advanced transmission techniques, such as directional antenna and beamforming, to avoid being detected.

Besides injecting data packets, attackers can also inject an overwhelming amount of query messages into the network to request other nodes to forward, which is called the query-flooding attacks. The advantage of query-flooding attacks lies in that for each query, more nodes in the network will be involved to process and forward packets compared to injecting data packets. Although a query message is usually much smaller than a data packet, when the query frequency is very high, query-flooding attacks can still cause severe damage to the network.

IV. Defense Mechanisms

In general, to detect whether a node has launched injecting traffic attacks, the detectors have to base on what they have observed. For example, a node can be marked as launching injecting traffic attacks only if it has been observed by some other nodes that it has injected too much traffic (higher than their legitimate bounds), or it has sent traffic to illegitimate destinations. Therefore, the following mechanisms will be required by any defense system to combat injecting traffic attacks.

- A robust packet delivery mechanism where for each packet injected by a node, this node cannot deny that this packet is from it and no other nodes can generate the same packet without colluding with it. This is addressed in Section IV-A.
- A robust traffic monitoring mechanism to count the number of packets injected by each node in the network. This is addressed in Section IV-B.
- A robust detection mechanism to detect injecting traffic attacks based on the observed information. This is addressed in Section IV-C.

A. Route Discovery and Packet Delivery

Since source routing has been widely used in mobile ad-hoc networks, and can greatly facilitate the attacker detection, in this paper, we focus on source routing. Specifically we will adopt dynamic source routing (DSR) [24] as the underlying routing protocol to perform route discovery and maintenance. On the other hand, to defend against possible routing-related attacks, the following security enhancements will be incorporated into the baseline DSR protocol.

- When a node s initiates a route discovery to destination d, besides the source-destination pair, the route request packet should also includes a unique ID associated with this request and the sequence number corresponding to the last data packet that s has sent to d. In this paper, the following format is used for each route request packet:

$$\{s, d, id_s(s,d), seq_s(s,d), \mathrm{sign_s(s, d, id_s(s,d), seq_s(s,d))}\}.$$

  Here, $id_s(s,d)$ is the sequence number of this route request packet, which has an initial value of 1 and is required to be increased by 1 after each route request has been issued by the pair (s,d). $seq_s(s,d)$ is the sequence number of the last data packet that the pair (s,d) has injected into the network. $\mathrm{sign_s(s, d, id_s(s,d), seq_s(s,d))}$ is the signature generated by s based on the message $\{s, d, id_s(s,d), seq_s(s,d)\}$.

- When a good node x receives a route request packet with s being the source and d being the destination, x first checks whether the following conditions can be satisfied:
  1) the source-destination pair (s, d) is legitimate;
  2) all signatures are valid;
  3) $id_x(s,d) < id_s(s,d)$, where $id_x(s,d)$ is the largest route request sequence number corresponding to the source-destination pair (s,d) that x has observed before;
  4) $seq_x(s,d) \leq seq_s(s,d)$, where $seq_x(s,d)$ is the largest data-packet sequence number corresponding to the pair (s,d) that x has observed before;
  5) no nodes appended to the route request packet have been detected as malicious by x;
  6) less than L_{maxhop} relay nodes have been appended to the query packet, where L_{maxhop} is a system parameter indicating the maximum number of relays that any route is allowed to have.
  7) x has not forwarded any route request for the source-destination pair (s,d) in the last $T_x(s,d)$ interval, where $T_x(s,d)$ is the minimum route request forwarding interval specified by x to indicate that x will not forward more than one route request for (s,d) in any $T_x(s,d)$ interval.

  If all of the above conditions can be satisfied, we call such a route request as a valid request. In this case, x will assign the value of $id_s(s,d)$ to $id_x(s,d)$, assign the value of $seq_s(s,d)$ to $seq_x(s,d)$, append its own address to the route request packet and sign the whole packet, and rebroadcast the updated route request. If only the first four conditions can be satisfied, x will simply update the values of $id_x(s,d)$ and $seq_x(s,d)$ using $id_s(s,d)$ and $seq_s(s,d)$. In all other situations, x will just discard this route request, and perform necessary attacker detection. Assume the request is the received valid route request message that x has decided to forward, then the following format will be used for x to append its own address

$$\{\text{request}, x, \mathrm{sign_x}(\text{request}, \mathrm{x})\}$$

Once a source has decided to send a packet to a certain destination using a certain route, a data-packet delivery transaction should be initiated. The proposed data-packet delivery mechanism works as follows. Suppose that node s is to send a packet to destination d through the route R with the payload *msg* and the sequence number $seq_s(s,d)$. s first generates two signatures sig_h and sig_b, with sig_h generated based on the message $\{R, seq_s(s,d)\}$ and sig_b generated based on the message $\{R, seq_s(s,d), MD(msg)\}$ where $MD()$ is a digest function such as SHA-1 [25]. The format of the packet to be sent is as follows:

$$\{R, seq_s(s,d), sig_h, msg, sig_b\}. \quad (1)$$

We refer to $\{R, seq_s(s,d), sig_h\}$ as the header of the packet, and refer to $\{msg, sig_b\}$ as the body of the packet. Next, s transmits this packet to the next node on route R, and increases the value of $seq_s(s,d)$ by 1. The advantage of generating two signatures will be explained later.

When a node x detects that a certain packet is to be transmitted by a certain node y, x first decodes and checks the header of the packet. Assume $\{R, seq_s(s,d), sig_h\}$ is the header of the transmitted packet, x needs to continue receiving and decoding the body of the packet only if all of the following conditions can be satisfied:

1) the signature sig_h is valid;
2) x belongs to the route R and is the target of this transmission;
3) no nodes on route R have been detected as malicious by x;
4) $seq_s(s,d) > seq_x(s,d)$;
5) route R has no more than L_{maxhop} relays;
6) x has agreed to participate on this route before and the route has not expired, where each route will be set with an expiration time.

If all of the above conditions can be satisfied, x will continue receiving and decoding the body of the packet, assuming it is $\{msg, sig_b\}$. If the signature sig_b is valid, x will forward the packet to the next node on the route, and update the value of $seq_x(s,d)$ using $seq_s(s,d)$.

B. *Traffic Monitoring*

Traffic monitoring is an indispensable component to detect possible injecting traffic attacks. In this paper, each node will keep monitoring its neighbors' transmission activities using the proposed header watcher mechanism. Specifically, when a node x detects that a neighbor y is transmitting a data packet, no matter whether x is the receiver of this transmission or not, x will try to receive and decode the packet header sent by y. Actually this is needed in most wireless networks: without decoding the header, how can a node know whether a packet targets it or not? The uniqueness of the proposed header watcher mechanism lies in that each node will also check the validity of the signature for the packet header. If the signature of the packet header is valid, x will put the packet header into the set $List(s,d,x)$ in x's records, which will be used later to detect whether s has launched injecting traffic attacks.

Unlike the "watchdog" mechanism introduced in [3], which requires a node to buffer all of the packets that it has sent or forwarded and to keep monitoring its neighbors' transmission activities in order to check whether those packets have been forwarded by them, the "header watcher" mechanism proposed in this paper only requires a node to monitor the packet headers in its neighborhood. Since only packet headers need to be received and decoded, and since the header of a packet is much shorter than the body of a packet, a lot of effort can be saved compared to the watchdog mechanism which requires receiving, decoding, and comparing the whole packet.

In general, if all packet headers received by node x are recorded, with the increase of x's staying time in the network, more storage will be required. Actually, in our scheme, for each legitimate source-destination pair (s,d), only those packet headers received after the last valid route request issued by (s,d) need to recorded by x. In other words, only those packet headers whose sequence numbers are larger than the sequence number broadcast by s in its last valid route request packet need to be recorded. With this modification, the storage requirement becomes very small and does not increase over x's staying time in the network. In Section VI, we will also show how to modify the schemes to further decrease the storage requirement.

C. Injecting Traffic Attack Detection

In this paper, each good node in the network will perform an injecting traffic attack detection based on what it has observed. Specifically, for each source-destination pair (s, d) with $List(s, d, x)$ being nonempty in a good node x's records, the following detection rules will be used by x to check whether s has launched injecting traffic attacks.

- Rule 1: x will mark s as malicious if $List(s, d, x)$ is not empty and the source–destination pair (s, d) is illegitimate.
- Rule 2: x will mark s as malicious if x has received a request issued by an illegitimate source destination (s, d).
- Rule 3: For any packet header $\{R, seq_s(s, d), sig_h\}$ in $List(s, d, x)$, x will mark s as malicious if route R has more than L_{maxhop} relays.
- Rule 4: x will mark s as malicious if x has detected that two valid packet headers exist $\{R, seq_s(s, d), sig_h\}$ and $\{R', seq'_s(s, d), sig'_h\}$ in the set $List(s, d, x)$ with $seq_s(s, d) = seq'_s(s, d)$ but $R \neq R'$.
- Rule 5: Let $seq_{\max}(s, d)$ be the largest sequence number corresponding to the source–destination pair (s, d) at time t (i.e., $seq_{\max}(s, d) = f_{s,d}(t)$ at time t), x will mark s as malicious if x has detected that a sequence number exists$seq_s(s, d)$ in $List(s, d, x)$ with $seq_s(s, d) > seq_{\max}(s, d)$.

The first two rules imply that only legitimate source-destination pairs can inject packets into the network. Rule 3 implies that no routes should have more than L_{maxhop} relays. Rule 4 handles multiple route IDPA. Rule 5 handles attackers who inject more packets than they should. In summary, rules 4 and 5 are used to prevent attackers from injecting more packets than they are allowed to by associating each packet with a unique sequence number. That is, no two packets from the same traffic pair should have the same sequence number, and the sequence number has to increase monotonically.

Once x has detected that s is launching injecting traffic attacks, x will also inform the other nodes in the network by broadcasting an ALERT message which includes evidence such as the corresponding packet headers. When other good nodes have received the ALERT message, after necessary verification (i.e., signatures are valid), they will also mark s as malicious.

Next, we analyze the effects of possible impersonation attacks that can be launched by attackers. Under the proposed mechanisms, in order to impersonate a good node s that has not been compromised, an attacker m has to first record the packets that s has transmitted, then later forwards/broadcasts these packets. Specifically, there are two situations.

- Situation 1: m recorded a query packet issued by s and rebroadcasted it later. However, since this query packet has been seen by all other nodes in the network due to the flooding nature of query message, no nodes will further process this query packet.
- Situation 2: m recorded a data packet issued by s and forwarded it later. However, since nodes on the route associated with this data packet will only process this packet at the most one time, forwarding this packet at time t_1 by m cannot cause damage to other nodes.

In summary, an impersonation attack cannot cause further damage to good nodes in the network. Furthermore, it can be readily checked that as long as s is good and has not been compromised, the probability that x will mark s as malicious is 0. That is, the false alarm ratio of the above detection rules is 0.

D. Overhead Analysis

Now we analyze the overhead associated with the above defense mechanisms. According to the above description, since each good node solely bases its own observation to conduct attacker detection, there is no extra communication overhead. The computation overhead mainly comes from generating and verifying the signatures for each sent and received packet, or specifically, the computation overhead comes from generating and verifying the signatures for packet headers. Compared to the packet body, the length of a packet header is much smaller; therefore, the extra computation overhead is also small. Meanwhile, when applying rule 4 and 5 to perform attacker detection, a node also needs to go through the header records it has stored, which may also incur some extra computation overhead. However, since the list of records is usually small, the extra computation overhead is not significant at all.

Now we analyze the storage overhead. The main drawback of the above proposed defense mechanism lies in that it requires some extra memory, while in some mobile nodes, storage may be a precious resources. For each good node, it needs to store the set of legitimate source-destination (SD) traffic pairs as well as the upperbounds of their traffic injection rate. Meanwhile, for each source-destination pair, it also needs to store the set of received packet headers from this node pair's last valid route request. If there are too many legitimate source-destination pairs, the storage overhead can be huge. However, in many ad-hoc network applications, the number of legitimate source-destination pairs is usually limited, such as in wireless ad-hoc sensor networks. Further, in mobile ad-hoc networks, route requests need to be issued very frequently; therefore, the number of packet headers that each node needs to store is also limited. In Section VI, we will discuss how to further reduce the storage overhead by proposing some centralized detection mechanisms with decentralized implementation.

V. Theoretical Analysis

According to the secure route discovery procedure described in Section IV-A, a good node x will only forward, at most, one route request in any time interval $T_x(s, d)$ for any legitimate SD pair (s, d), and will not forward route requests for any illegitimate SD pairs; therefore, the total damage that can be caused by attackers launching query flooding attacks is bounded. Next, we analyze the effects of IDPA. Assume that node s is malicious and tries to launch IDPA with d being the destination of the packets injected by s. To avoid being detected immediately, the SD pair (s, d) must be legitimate and d must be malicious too, otherwise, s can be easily detected by d as malicious. According to Section III, there are three possible ways to launch IDPA: simple IDPA, long-route IDPA, and multiple-route IDPA.

We first consider simple IDPA. According to Section IV-A, in order for good nodes to forward packets for s, s has to increase the sequence number $seq_s(s,d)$ by one after each packet delivery. Unless all nodes on the selected route are malicious, which makes no sense, the good nodes on route R can easily detect that s is launching IDPA by comparing the received packets' sequence number with $f_{s,d}(t)$ defined in Section IV-C. That is, when launching simple IDPA, the attackers can be immediately detected and can cause negligible damage.

If s launches long-route IDPA, since more good nodes will be involved, s can cause similar damage as launching simple IDPA. However, as described in Section IV-A, the maximum allowable number of hops per route is bounded by L_{maxhop}, and good nodes will drop all packets with the associated number of hops more than L_{maxhop}. Therefore, the damage is upperbounded by $f_{s,d}(t)L_{\mathrm{maxhop}}$.

Finally, we consider the multiple-route IDPA. To avoid being detected immediately, the packet injection rate to each route must conform to $f_{s,d}(t)$, and the selected routes must be node-disjoint, that is, no selected routes should share any common good node except s and d; otherwise, if a good node x lies in more than one route from s to d, it can easily detect whether s and d have launched multiple-route IDPA. Meanwhile, the packets passing through the same route should have different sequence numbers in order for good nodes on the route to forward them. Based on whether s allows packets in different routes to share the same sequence numbers and what transmission techniques s will use, there are three cases.

Case 1) s dose not allow packets on different routes to share the same sequence numbers. Since $seq_s(s,d) \leq f_{s,d}(t)$ is required to let s avoid being detected immediately, in this case, s has no extra gain compared with launching simple IDPA.

Case 2) s allows packets on different routes to share the same sequence numbers, and transmits packets omnidirectionally. Since s's neighbors will keep monitoring s's packets transmission, they can easily detect that some packets sent by s through different routes use the same sequence number, which indicates that s is launching IDPA. Therefore, if s can only transmit packets omnidirectionally, s should not launch multiple-route IDPA.

Case 3) s allows packets on different routes to use the same sequence numbers, and can transmit packets using directional transmission techniques. Since now s's neighbors cannot receive s' transmission not targeting on them, they have little chance to directly detect that s is launching IDPA. However, since good nodes in the network use omnidirectional transmission techniques, the probability that s can successfully launch multiple-route IDPA without being detected still approaches 0, as will be shown next.

Next, we derive the upperbounds for the probability that s is able to successfully pick n node-disjoint routes to inject data packets without being detected immediately, as illustrated in Case 3. We consider the most general situation that the destination d does not know the exact locations of those nodes within its transmission range, and all d's neighbors are good nodes. Given a node x and a certain area S, we say that x is randomly deployed inside S according to the 2-D uniform distribution, that is, for any subarea $S_1 \subset S$ we have $P(x \in S_1 | x \in S, S_1 \subset S) = S_1/S$. Then, we have the following theorem.

Theorem 1: Suppose that N good nodes are independently deployed inside a large area of S according to the 2-D uniform distribution. Suppose that all of these N nodes use omnidirectional transmission techniques and r is their common maximum transmission distance. Suppose that the SD pair (s,d) collude to launch IDPA with s using directional transmission technique and s and d not knowing the exact location of the nodes inside d's receiving range (which is r). If the defending mechanisms described in Section IV are used by good nodes, then the probability $P(n,r,N)$ that the two attackers can successfully pick n node-disjoint routes to launch multiple-route IDPA without being detected immediately is upperbounded by

$$P(n,r,N) \leq \left(\frac{3\sqrt{3}}{4\pi}\right)^{\binom{n}{2}} \sum_{k=n}^{N} P_1(k,N) \left(n \left(\frac{3\sqrt{3}}{4\pi}\right)^{\binom{n-1}{2}} \right)^{k-n} \tag{2}$$

where $P_1(k,N)$ is defined as follows:

$$P_1(k,N) = \binom{N}{k} \left(\frac{\pi r^2}{S}\right)^k \left(1 - \frac{\pi r^2}{S}\right)^{N-k}. \tag{3}$$

Before proving Theorem 1, we first prove the following lemmas.

Lemma 1: Assume N nodes are independently deployed inside an area of S according to the 2-D uniform distribution. For any node x inside subarea $S_1 \subset S$ and for any subarea $S_2 \subset S_1$, we have

$$P(x \in S_2 | x \in S_1, S_2 \subset S_1 \subset S) = \frac{S_2}{S_1}. \tag{4}$$

Proof:

$$\begin{aligned} &P(x \in S_2 | x \in S_1, S_2 \subset S_1 \subset S) \\ &= \frac{P(x \in S_2, x \in S_1 | S_2 \subset S_1 \subset S)}{P(x \in S_1 | S_2 \subset S_1 \subset S)} = \frac{P(x \in S_2 | S_2 \subset S)}{P(x \in S_1 | S_1 \subset S)} = \frac{S_2}{S_1}. \end{aligned}$$

That is, the conditional distribution of x in S_1 is independent of S, which is also the 2-D uniform distribution. ■

Lemma 2: Assume nodes x and y are independently deployed inside a certain area S according to the 2-D uniform distribution. Given $x \in S_1 \subset S$ and $y \in S_1 \subset S$, and given any subareas $S_x \subset S_1$ and $S_y \subset S_1$, we have

$$\begin{aligned} &P(x \in S_x, y \in S_y | x \in S_1, y \in S_1, S_x \subset S_1, S_y \subset S_1) \\ &\quad = P(x \in S_x | x \in S_1, S_x \subset S_1) P(y \in S_y | y \in S_1, S_y \subset S_1). \end{aligned} \tag{5}$$

Proof: Since the deployment of x and y are independent of each other, we have

$$\begin{aligned} &P(x \in S_x, y \in S_y | x \in S_1, y \in S_1, S_x \subset S_1, S_y \subset S_1) \\ &= P(x \in S_x | x \in S_1, S_x \subset S_1, y \in S_y \subset S_1) * \\ &\quad P(y \in S_y | y \in S_1, S_y \subset S_1, x \in S_1, S_x \subset S_1) \\ &= P(x \in S_x | x \in S_1, S_x \subset S_1) P(y \in S_y | y \in S_1, S_y \subset S_1). \end{aligned}$$

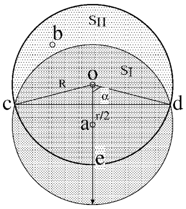


Fig. 3. Illustration of proof of Lemma 4.

That is, the distribution of x and y inside S_1 are independent of each other. ■

Lemma 3: Let S be a circular area with o being the center and R being the radius. Assume that node x lies in S and $P(A \in S_1 | A \in S, S_1 \subset S) = S_1/S$. Let $d(x)$ denote the random variable of the distance from x to o, then

$$P(d(x) = r | x \in S) = \begin{cases} \frac{2r}{R^2} & 0 \leq r \leq R \\ 0 & r > R \end{cases}. \tag{6}$$

Proof: For any $0 < r \leq R$, we have

$$P(d(x) = r | x \in S) = \lim_{\Delta \to 0} \frac{\frac{\pi r^2}{\pi R^2} - \frac{\pi (r-\Delta)^2}{\pi R^2}}{\Delta} = \frac{2r}{R^2}. \tag{7}$$

For any $r > R$, we have $x \notin S$, which implies $P(d(x) = r | x \in S) = 0$. ■

Lemma 4: Let S be a circular area with o as its center and R as its radius. Given that two nodes a and b are independently deployed in S according to the 2-D uniform distribution, we have

$$P(|ab| > R | a \in S, b \in S) = \frac{3\sqrt{3}}{4\pi} \tag{8}$$

where $|ab|$ denote the distance between a and b.

Proof: We use Fig. 3 to help illustrate the proof. Let r denote the distance from a to o, let C_o denote the circle with o being the center and R being the radius, and let C_a denote the circle with a being the center and R as the radius. Let c and d be the intersecting points between the two circles C_o and C_a, and let $\alpha = \angle coa = \angle doa$. Let $S_I(r)$ denote the intersecting area inside both circles C_o and C_a with $|oa| = r$, and let $S_{II}(r)$ denote the area of S subtracted by $S_I(r)$. Then, we have

$$P(|ab| > R \big| a \in S, b \in S) = \int_0^R \frac{2r}{R^2} \frac{S_{II}(r)}{S} \mathbf{d}r \tag{9}$$

where (9) comes from Lemma 4. We first calculate $S_I(r)$

$$S_I(r) = 2\left(R^2 \arccos \frac{r}{2R} - \frac{r}{2}\sqrt{R^2 - \left(\frac{r}{2}\right)^2}\right) \tag{10}$$

where $\alpha = \arccos(r/2R)$. Then, $S_{II}(r)$ can be calculated as

$$S_{II}(r) = R^2 \left(\pi - 2\arccos \frac{r}{2R} - \frac{r}{R^2}\sqrt{R^2 - \left(\frac{r}{2}\right)^2}\right). \tag{11}$$

By integrating (11) into (9), we have $P(|ab| > R \big| a \in S, b \in S) = 3\sqrt{3}/4\pi$. ■

Lemma 5: Assume that n nodes $A = \{a_1, \ldots, a_n\}$ are independently deployed inside a circular area S according to the 2-D uniform distribution with R being the radius, then we have

$$P(|a_i a_j| > R : \forall a_i, a_j \in A) \leq P(|a_1 a_2| > R)^{\binom{n}{2}}. \tag{12}$$

Proof:

$$\begin{aligned} &P(|a_i a_j| > R : \forall a_i, a_j \in A) \\ &= P(|a_1 a_2| > R, \ldots |a_1 a_n| > R, \ldots, |a_{n-1} a_n| > R) \\ &= P(|a_1 a_2| > R \big| |a_1 a_3| > R, \ldots |a_{n-1} a_n| > R) * \\ &\quad P(|a_1 a_3| > R, \ldots, |a_{n-1} a_n| > R) \\ &= P(|a_1 a_2| > R \big| |a_1 a_i| > R, |a_2 a_i| > R : \forall 3 \leq i \leq n) * \\ &\quad P(|a_1 a_3| > R, \ldots, |a_{n-1} a_n| > R). \end{aligned}$$

Given $|a_1 a_i| > R$ and $|a_2 a_i| > R$ for any $3 \leq i \leq n$, we can draw a circle with a_i being the center and R being the radius. To conform to the statement that "$\forall a_i, a_j \in A, |a_i a_j| > R$," both a_1 and a_2 cannot lie inside the intersecting area between this circle and the circle with o being the center. That is, a_1 and a_2 are now restricted in an area of $S' \subset S$ smaller than S. So the probability that $|a_1 a_2|$ is larger than R under such restrictions will become smaller than without such restrictions. That is

$$\begin{aligned} &P(|a_1 a_2| > R \big| |a_1 a_i| > R, |a_2 a_i| > R) \\ &\qquad \leq P(|a_1 a_2| > R : \forall 3 \leq i \leq n). \end{aligned} \tag{13}$$

Following the same arguments, we can have:

$$P(|a_i a_j| > R : \forall a_i, a_j \in A) \leq \prod_{1 \leq i < j \leq n} P(|a_i a_j| > R). \tag{14}$$

Since there are a total of $\binom{n}{2}$ items in the product, and nodes in A are symmetric, we can conclude that (12) holds. ■

Lemma 6: Assume $n + m$ nodes $\{a_1, \ldots, a_n, b_1, \ldots, b_m\}$ are independently deployed inside a circular area S according to 2D uniform distribution with R being the radius. Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$, then we have

$$\begin{aligned} &P(|a_i b_l| > R \text{ or } |a_j b_l| > R : \forall a_i, a_j \in A, b_l \in B, i \neq j) \\ &\qquad \leq \left(nP(|a_1 b_1| > R)^{n-1}\right)^m. \end{aligned} \tag{15}$$

Proof: Let $A_i = A - \{a_i\}$. Given any $b \in B$, to say "$|a_i b| > R$ or $|a_j b| > R : \forall a_i, a_j \in A, a_i \neq a_j$" is equivalent to say "there exists at least one A_i with $|xb| > R$ for any $x \in A_i$", that is

$$\begin{aligned} &P(|a_i b| > R \text{ or } |a_j b| > R : \forall a_i, a_j \in A, a_i \neq a_j) \\ &= P((|xb| > R : \forall x \in A_1) \text{ or } \ldots \text{ or } (|xb| > R : \forall x \in A_n)) \\ &\leq \sum_{i=1}^{n} P(|xb| > R : \forall x \in A_i) \\ &= nP(|xb| > R : \forall x \in A_1) \\ &\leq nP(|a_1 b| > R)^{n-1}. \end{aligned}$$

Due to the symmetry and independence of the m nodes in B, we can conclude that (15) holds. ■

Now Theorem 1 can be proved as follows.

Proof: Let C_d denote the circle with d being the center and r being the radius. For s and d to successfully pick n node-disjoint routes to launch multiple-route IDPA without being detected immediately, they need to pick at least n distinct nodes inside C_d, one for each route, to act as the last intermediate nodes on these routes. Since s and d do not know the exact locations of the nodes inside C_d, these n nodes can only be randomly selected. It is easy to see that the following three necessary conditions must be satisfied in order for the attackers to succeed.

C1. There exists at least n nodes inside C_d, otherwise, s and d can never have n node-disjoint routes between them.

C2. Given that there are $k \geq n$ nodes inside C_d, and that s and d are to randomly select n nodes among them to act as the last intermediate node for these n node-disjoint routes, then for any two nodes among the n nodes selected by s and d, no node should lie in the other nodes' transmission range. Otherwise, if any two of the n nodes lie in each other's transmission range, they can easily detect that s is launching a multiple-route IDPA.

C3. Given that the n nodes have been selected by s and d, there should exist no other good nodes (nodes excluding the selected n good nodes) which can simultaneously lie in any two of these n nodes' transmission range. Otherwise, if one such node exists, then it can easily detect that s is launching a multiple-route IDPA.

Let $P_1(k, N)$ denote the probability that there are k nodes inside C_d, $P_2(n, r, k)$ denote the probability that the condition C2 can be satisfied given that the n nodes are randomly selected among $k \geq n$ nodes inside C_d, and $P_3(n, r, k, N)$ denote the probability that the condition C3 can be satisfied given there are $k \geq n$ nodes inside C_d and the n nodes have been determined by s and d. It is easy to see that

$$P(n, r, N) \leq \sum_{k=n}^{N} P_1(k, N) P_2(n, r, k) P_3(n, r, k, N). \quad (16)$$

Since nodes are independently deployed inside S according to the 2-D uniform distribution, we can immediately have

$$P_1(k, N) = \binom{N}{k} \left(\frac{\pi r^2}{S}\right)^{k} \left(1 - \frac{\pi r^2}{S}\right)^{N-k}. \quad (17)$$

Given that k nodes lie in C_d, according to Lemma 1 and Lemma 2, it is equivalent to say that these k nodes are independently deployed inside C_d according to the 2-D uniform distribution. According to Lemma 4 and Lemma 5, we can have

$$P_2(n, r, k) = \left(\frac{3\sqrt{3}}{4\pi}\right)^{\binom{n}{2}}. \quad (18)$$

To simplify the analysis, we consider a modified version of condition C3: given any two nodes among the selected n nodes, there should exist no other good nodes inside C_d but not belonging to these n nodes which can simultaneously lie in these two nodes' transmission range. That is, only a small subset of the applicable nodes is considered. Let $P_3'(n, r, k, N)$ denote the probability that the modified condition C3 can be satisfied given there are $k \geq n$ nodes inside C_d and the n nodes have been determined by s and d, then we must have $P_3(n, r, k, N) \leq P_3'(n, r, k, N)$. According to Lemma 4 and Lemma 6, the probability that the modified condition C3 can be satisfied is upperbounded by

$$P_3'(n, r, k, N) \leq \left(n \left(\frac{3\sqrt{3}}{4\pi}\right)^{\binom{n-1}{2}} \right)^{k-n}. \quad (19)$$

By combining the above results, we can conclude that (2) as well as Theorem 1 holds. ■

Theorem 2: The probability that two colluding attackers s and d can successfully pick 6 or more node-disjoint routes to launch multiple-route IDPA without being detected immediately is 0.

Proof: For the attackers s and d (assuming s is the source and d is the destination) to simultaneously pick 6 routes to launch multiple-route IDPA, it needs to pick 6 nodes within d's receiving range, that is, the circular area C_d with d being the center and r the radius. Let $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ denote the set of 6 selected nodes by s and d that lie inside C_d. One necessary condition for the attackers to succeed is that for any $a_i, a_j \in A$, we must have $|a_i a_j| > r$ for any $a_j \in A$ and $a_j \neq a_i$. Now we show that it is not achievable. If $a_i, a_j \in A$ exists with $\angle a_i d a_j = 0$, then we must have $|a_i a_j| \leq r$. Next, we only need to consider the situations that for any $a_i, a_j \in A$, $\angle a_i d a_j \neq 0$. For each node $a_i \in A$, we draw a radial originating from d and passing a_i, and let a_i' be the intersecting point between the radial da_i and the circumference of the circle C_d. Any two radials will partition the circular area C_d into two sectors. We say a sector is singleton if none of the nodes in A lie inside this sector (including the arc but excluding the two radials). It is easy to say that the six nodes will partition the circle into six singleton sectors. To satisfy the above necessary condition, the angle of each singleton sector should be more than $\pi/3$: if the angle of a singleton section is no more than $\pi/3$, let a_i be the node on one side of this sector, and a_j be the node on the other side of this sector, then for any point x that lies in the segment da_i' and any point y that lies in the segment da_j', we must have $|xy| \leq r$. Since we have six singleton sectors, and each singleton sector has an angle of more than $\pi/3$, the summed angle is more than 2π, which contradicts the fact that a circle is 2π. Given this conclusion, it is trivial to show that more than six routes is also not achievable. ■

We have also evaluated through experiments the upperbounds of the success ratio for two colluding attackers s and d to launch multiple-route IDPA with s using directional transmission technique. Given a rectangular area of $20r \times 20r$, we put d in the center of the area. At each round of the experiment, we independently deploy $400r^2\rho$ nodes inside the area according to 2-D uniform distribution and randomly pick n nodes inside d's receiving range, where ρ is referred to as the node density. We say (s, d) may succeed only if all of the three necessary conditions presented in the proof of Theorem 1 are satisfied. For each configuration of route number n and node density ρ, 10^7 experiments have been conducted, and the upperbounds are obtained as the ratio of the total success number over the total number of experiments.

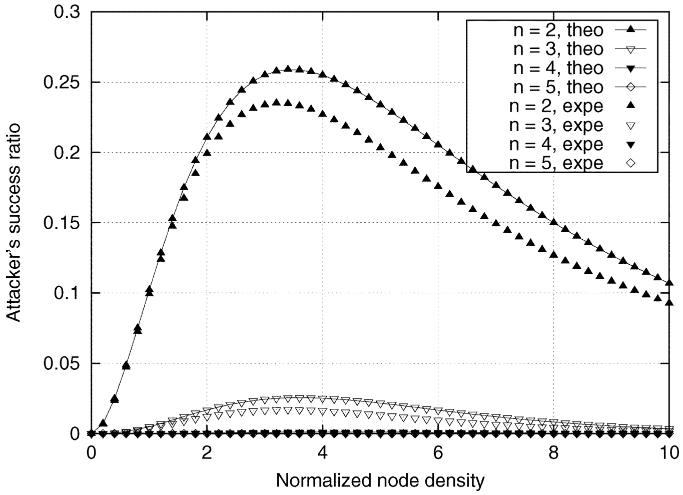


Fig. 4. Upperbounds of attackers' success probability to launch injecting packet attack via two node-disjoint routes under a lossless channel.

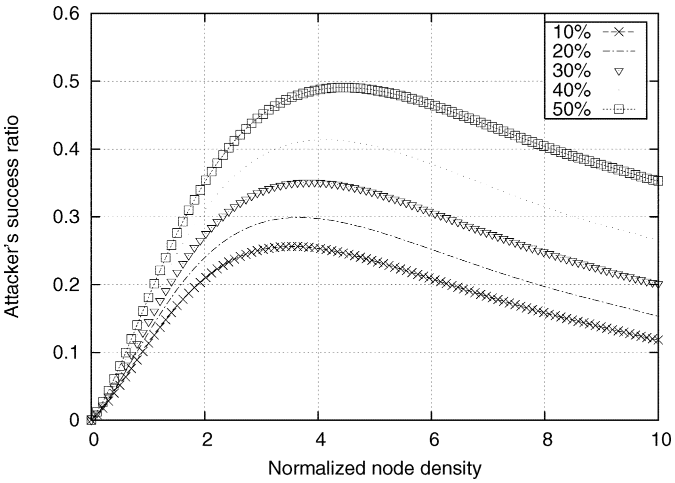


Fig. 5. Upperbounds of attackers' success probability to launch injecting packet attack via two node-disjoint routes under the lossy channel.

Both experimental and theoretical upperbounds are plotted in Fig. 4, where "theo" denotes the theoretical upperbounds obtained using (2), "expe" denotes the experimental upperbounds obtained through experiments described above, and "n" denotes the number of node-disjoint routes to be picked by the malicious SD pair (s, d). In Fig. 4, the normalized node density is defined as the average number of nodes inside an area of πr^2. Since both the theoretical and experimental upperbounds corresponding to $n = 4$ and $n = 5$ are almost equal to 0 across all illustrated node densities (e.g., for $n = 4$, all values are less than 2 $\times 10^{-3}$), the four curves associated with $n = 4$, 5 have almost overlapped into one single curve, which is the lowest curve illustrated in Fig. 4. For $n = 2$, 3, we can see that the success ratio increases first with the increase of node density until it arrives at a peak, then decreases with the further increase of node density, which is consistent with (2). The reason is as follows: with the increase of the node density, the probability P_1 that the condition C1 can be satisfied increases monotonically from 0 to 1, the probability P_2 that the condition C2 can be satisfied keeps unchanged, while the probability P_3 that the condition C3 can be satisfied decreases monotonically from 1 to 0, and when ρ is small, the value of P_1 dominates the bound, while when ρ is large, the value of P_3 dominates the bound. From Fig. 4, we can also see that gaps exist between theoretical results and experimental results. The reason is that when we calculate the probability of condition C3 being satisfied, only a subset of applicable nodes have been considered, which make the theoretical upperbounds a little bit looser (higher) than the experimental upperbounds.

In the above experiments, we have assumed that all packets can be successfully received as long as the distance between the transmitter and receiver is no more than the transmission range r. However, in reality, the channel is usually lossy, and not all packets can be successfully received. This can be taken advantage of by the attackers to increase their success probability. Fig. 5 illustrates the experimental results of the attackers' success probability to launch an injecting packet attack via two node-disjoint routes under lossy channels. Specifically, each curve corresponds to a certain packet-loss ratio. From these results, we can see that the lossy channel can certainly increase the attackers' success probability. However, we can also see that even half of the packets have been lost, the maximum possible success ratio is still no more than 50% even for two node-joint routes.

The above upperbounds are evaluated based on a fixed topology, that is, the set of links $E(t)$ keeps unchanged for all time index t. However, due to node mobility, $E(t)$ will change over time t; therefore, s needs to frequently update routes. Then, after several route updates, the probability that s still has not been detected as malicious will be very small. For example, assume that each route update is independent, after 5 times of route updates, even for $n = 2$, the probability that s has not been detected as malicious is less than 0.06%. That is, attackers have a negligible chance to flee. In summary, when the malicious SD pair (s, d) tries to launch IDPA, to avoid being detected and to maximize the damage, the optimal strategy is to use only one route to inject data packets by conforming to both the maximum hop number L_{maxhop} and the legitimate rate $\lambda_{s,d}$, which is equivalent to say that the optimal strategy is to not launch IDPA.

VI. Centralized Detection With Decentralized Implementation

The defense system described in Section IV is fully distributed. However, the drawback of this system is that it may have relatively high storage complexity. Meanwhile, each node needs to have prior knowledge of the set of legitimate traffic pairs, which may not be available to all nodes in general. Next, we describe a modified version of the proposed defense system. In the modified version, instead of performing attacker detection by itself, each good node will report the observed information to certain nodes which we called centralized detectors; then, the centralized detectors will perform attacker detection based on the collected traffic information. In general, the centralized detectors will be under stronger protection than those normal nodes and may have more powerful computation capability and more storage.

The detailed description of the modified defense system is as follows. First, the route discovery and packet delivery procedure are the same as described in Section IV-A. Second, the monitoring mechanism is still the proposed header watcher mechanism as described in Section IV-B. To reduce the storage overhead, we made the following modification: for each good node, instead of storing all listened valid packet headers, most times it does not need to store any packet headers locally, but only needs to store the following three tuples (traffic pair, sequence number, route) associated with each listened valid packet header. A good node needs to record a whole packet header only if it has been requested by the detectors to do so, as to be explained next. Furthermore, instead of reporting each listened packet header information separately, each good node will report the listened packet header information in a batch mode, that is, each report consists of a lot of listened packet header information. Assume in the previous fully distributed mechanism a good node needs to store n number of packet headers with each having l bytes (l is usually more than 100 B for a route request with ten relays considering the extra signatures), then in the modified defense mechanism, it only needs to store $n \times \tilde{l}$ bytes where $\tilde{l}$ is usually much smaller than l. For example, for a route with ten relays, each node ID uses 8 b, and the sequence number uses 32 b, $\tilde{l}$ is only 14 B. Further, normal nodes do not need to know who are legitimate SD pairs or their legitimate traffic injection rates.

For those centralized detectors, their job is to perform injecting traffic attack detection by applying similar detection rules as described in Section IV-C. The major difference lies in that when the centralized detector performs injecting traffic attack detection, there are usually two steps. In the first step, the detector will check whether a node has injected two packets with the same sequence number or whether a sequence number is larger than specified upperbound based only on the collected partial packet header information, that is, without checking the packet header signatures. If any of the two conditions has been satisfied, the detector will then request those nodes that report such information to submit full packet headers. That is, the centralized detector needs solid evidence in order to mark a node as an attacker.

Now, we use an example to illustrate the modified detection procedure. Assume that node a has reported a sequence number seq_1 and route R_1 associated with traffic pair (s, d), and node b has reported a sequence number seq_2, and route R_2 associated with traffic pair (s, d). After the centralized detector has received these reports, it will find that $seq_1 = seq_2$ but $R_1 \neq R_2$. Then, the detector has reason to suspect that s has launched injecting traffic attacks. When this occurs, the detector will ask node a and b to report the full packet headers next time such that it can collect concrete evidence to charge s.

From the above description, we can see that although the attacker detection is performed in a centralized way, the monitoring is still fully distributed. Now we analyze the detection performance of the modified defense system. It is easy to see that either simple IDPA or long-route IDPA can be easily detected. Meanwhile, for the multiroute IDPA, requiring packets sent via different route to use a different sequence number has no gain from the attacker's point of view, and allowing packets sent via a different route to use same sequence number will be detected immediately when the omnidirectional transmission technique is used.

TABLE I
SIMULATION PARAMETERS

Number of Good Nodes	100
Number of Malicious Nodes	0-50
Minimum Velocity (v_{min})	2 m/s
Maximum Velocity (v_{max})	10 m/s
Average Pause time	300 seconds
Dimensions of Space	1500m × 1500m
Maximum Transmission Range	300 m
Average Packet Inter-Arrival Time	1 seconds
Data Packet Size	1024 bytes
Link Bandwidth	1 Mbps

Now we focus on the scenario that attackers allow packets to be sent via different routes using the same sequence number, and the directional transmission technique is used to avoid being detected. Given that an attacker s picks n node-disjoint routes to simultaneously inject packets and packets on different routes will share the same set of sequence numbers as long as at least two nodes on the selected routes are good, it is easy to check with zero probability that s can avoid being detected. In other words, attackers have no chance to launch IPDA without being detected. In other words, under the modified defense mechanism, the attackers' success probability is much lower compared to the previous fully distributed defense mechanism, which is the major advantage of the modified mechanisms.

Compared to the fully distributed defense system described in Section IV, the storage overhead of the modified defense system can be dramatically reduced, but some extra communication overhead is introduced since each node needs to report the centralized detector. However, since the size of each report is very small compared to the data packet, the extra communication overhead is negligible. For example, if the average packet size is 1000 B, and the report size is 20 B, then the increased overall traffic is only 2%. If the memory resource is more precious than the communication resource, the modified detection scheme should be preferred.

Until now, we have assumed that each good node will keep listening to all of the packet transmission in its neighborhood. Next, we show how to further decrease the overhead by letting nodes selectively listen to packet transmissions, with negligible degradation of the detection performance. Specifically, each node can selectively listen to its neighbors' transmission with a certain probability p, which we called probabilistic monitoring. That is, a packet transmission event occurs in a good node's neighborhood; with only probability p, this node will monitor this transmission and report the observation to the centralized detector. Now, when an attacker has injected n packets with the same sequence number via n node-disjoint routes (where $n > 1$), with no more than probability $p(n) = (1-p)^n + p(1-p)^{n-1}$, this attacker can avoid being detected. Furthermore, after the attacker has injected k packets, the probability that it will not be detected will be decreased to $p(n)^k$, which goes to 0 with the increase of k. By applying probabilistic monitoring, the communication overhead can be further decreased by $1 - p$, while the detection performance only suffers negligible degradation.

One possible drawback of such centralized detection mechanism is that the detector itself can also become attackers' target.

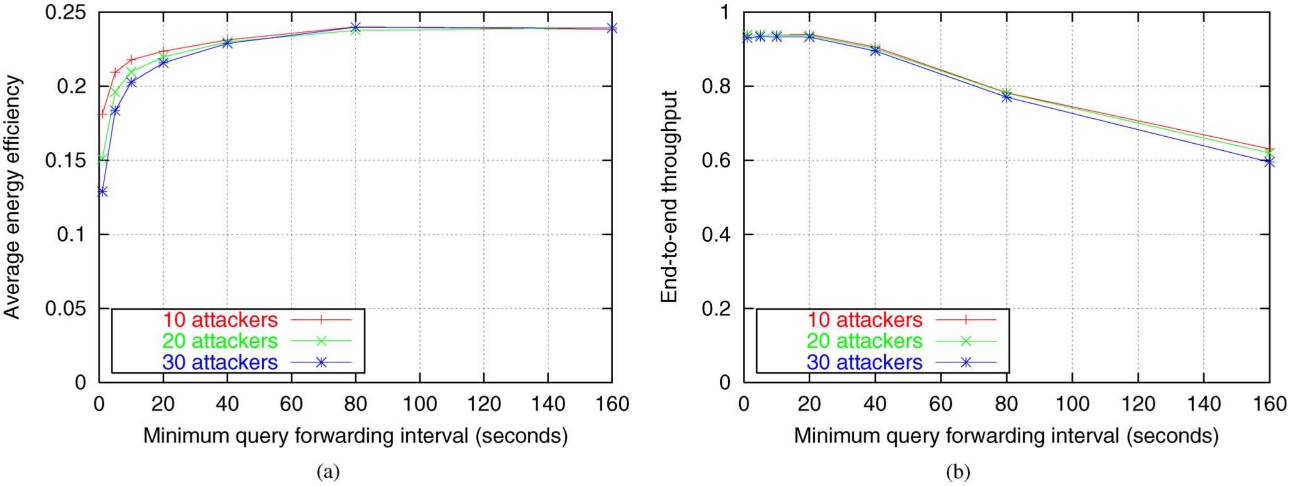


Fig. 6. Limiting route request rate versus system performance. (a) Energy efficiency. (b) End-to-end throughput.

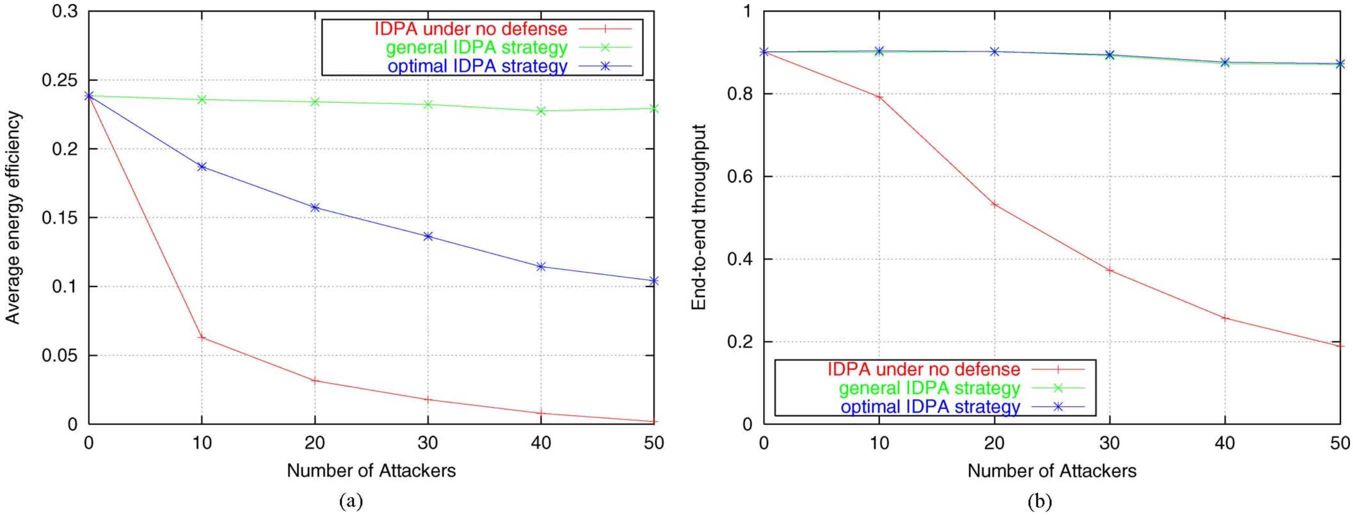


Fig. 7. Effects of IDPA under different configurations. (a) Energy efficiency. (b) End-to-end throughput.

Besides increasing the protection level, one can also increase the number of centralized detectors. For example, if there are 2 detectors in the network, even one has been compromised, the other still work well. In this case, for each node, it can either submit report to both detectors, or each time randomly pick one to submit, where the later is equivalent to reducing p by half.

VII. Simulation Studies

In our simulations, nodes are randomly deployed inside a rectangular area, and each node moves according to the modified random waypoint model [26] where a node starts at a random position, waits for a duration called the pause time that is modeled as a random variable with exponential distribution, then randomly chooses a new location and moves toward the new location with a velocity uniformly chosen between $v_{\min}$ and $v_{\max}$. The physical layer assumes that two nodes can directly communicate with each other successfully only if they are in each other's transmission range. The MAC layer protocol simulates the IEEE 802.11 Distributed Coordination Function (DCF) with a four-way handshaking mechanism [27]. Some simulation parameters are listed in Table I.

In the simulations, 50 good nodes are selected as the packet generators, and each will randomly pick a good node to send packets; therefore, the total number of SD pairs are 50. For each malicious node, it will also randomly pick another malicious node as the destination to inject packets. All SD pairs (either good or malicious) are set to be legitimate, and for each pair, packets are generated according to a Poisson process with a prespecified traffic rate known by all nodes, where the average packet interarrival time is 1 s. We set $f_{s,d}(t)$ to be $t+3$ for any SD pair (s,d). For malicious nodes who launch injecting traffic attacks, they will increase the average packet injection rate by 10 times. Also, all data packets have the same size and, in average, each route request packet has size 100 B.

In our simulations, each configuration has been run 20 independent rounds using different random seeds, and the results are averaged over all 20 rounds. For each round, the simulation time is set to be 5000 s. We use average energy efficiency and end-to-end throughput as metrics to measure the network performance. Here, the average energy efficiency is defined as the total number of good nodes' successfully delivered packets over the total amount of energy spent by all good nodes, and the end-to-end throughput is defined as the total number of good nodes' successfully delivered packets over the total number of good nodes' packets that need to be sent. When we calculate the energy efficiency, only transmission energy consumption has been considered. One reason is that transmission energy consumption plays a major role in overall energy consumption, and another reason is that receiving energy consumption may vary dramatically over different communication systems due to their different implementations. We assume that the transmission energy needed per data packet is normalized to be 1.

We first investigate the tradeoff between limiting the route request rate and system performance although the performance also depends on other factors, such as the mobility pattern, the number of nodes in the network, the average number of hops per route, etc. To better illustrate the tradeoff between limiting the route request rate and system performance, the other parameters are set to be fixed. However, similar results can also be obtained by changing these parameters.

Fig. 6 illustrates the tradeoff between limiting the route request rate and network performance. In this set of simulations, all malicious nodes will only inject route request packets and will not inject any data packets or launch routing disruption attacks. We assume that all good nodes have the same minimum route request forwarding interval denoted by $T^{\min}$, but all malicious nodes will set their route request rate to be 1/s. From Fig. 6(a), we can see that with the increase of $T^{\min}$ from 1 to 80 s, the energy efficiency of good nodes also increases, and is kept almost unchanged from 80 to 160 s. The reason is that when $T^{\min}$ is small, attackers can waste good nodes' energy by injecting a lot of route request packets and to request others to forward. Fig. 6(b) shows that with the increase of $T^{\min}$ from 1 s to 20 s, the end-to-end throughput of good nodes is kept almost unchanged, while with the increase of $T_{\min}$ from 80 s to 160 s, the end-to-end throughput of good nodes drops almost linearly. These results also motivate us to pick $T^{\min}$ to be 40 s in the following simulations.

Fig. 7 shows the simulation results under various types of IDPA. Here, "IDPA under no defense" denotes that attackers just launched simple IDPA and the underlying system has not launched any defending mechanism. "General IDPA strategy" denotes that attackers launch IDPA but the mechanisms described in Section IV have been launched, where both multiple-route IDPA and long-route IDPA have been simulated. Specifically, half of the attackers have launched multiple-route IDPA who will try to pick as many as possible node-disjoint routes to inject packets, though for each route, it will conform to the legitimate traffic injection rate. Another half of the nodes will try to launch long-route IDPA that will try to pick the longest possible routes to inject traffic. "Optimal IDPA strategy" denotes that attackers will use only one route to inject data packets which conforms both to the maximum hop number $L_{\text{maxhop}} = 10$ and to the legitimate maximum packet injection rate, and the mechanisms described in Section IV have been launched. In other words, here "optimal IDPA strategy" can also be regarded as no IPDA attack at all.

From Fig. 7(a), we can see that when there is no defending mechanisms for IDPA, even simple IDPA can dramatically degrade the energy efficiency of good nodes. When the defending mechanisms described in Section IV are employed, from an attackers' point of view, launching IDPA does not have any gain in decreasing the energy efficiency of good nodes. However, if attackers apply the optimal IDPA strategy, they can still degrade the energy efficiency of good nodes. From Fig. 7(b), we can see that without employing necessary defending mechanisms, with the increase of the number of attackers, even simple IDPA can dramatically degrade the end-to-end throughput of good nodes due to the congestion they cause. When the defending mechanisms described in Section IV are employed, launching IDPA has almost no effects on the performance of good nodes' end-to-end throughput.

VIII. Conclusion

In this paper, we have studied the possible injecting traffic attacks that can be launched in mobile ad-hoc networks, and proposed a set of mechanisms to defend against such attacks. Both query flooding attacks and injecting general data packets attacks have been investigated. Furthermore, for injecting general data packets attacks, the situations that attackers may use some advanced transmission techniques, such as directional antennas or beamforming, to avoid being detected have also been studied. Two set of defense mechanisms have been proposed, one is fully distributed, while the other is centralized with decentralized implementation. Our theoretical analysis has shown that when the proposed mechanisms are used, the best strategy for attackers is not to launch injecting traffic attacks. Extensive simulation studies have also agreed with our theoretical analysis.

References

[1] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Netw. Mag.*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.

[2] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proc. 6th Annu. Int. Conf. Mobile Computing and Networking*, Boston, MA, Aug. 2000, pp. 275–283.

[3] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. 6th Annu. Int. Conf. Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000, pp. 255–265.

[4] J.-P. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," in *Proc. 2nd ACM Int. Symp. Mobile Ad Hoc Networking Computing*, Long Beach, CA, May 2001, pp. 146–155.

[5] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," presented at the 8th Annu. Int. Conf. Mobile Computing and Networking, Atlanta, GA, Sep. 2002.

[6] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," presented at the ACM Workshop on Wireless Security, San Diego, CA, Sep. 2003.

[7] I. Aad, J. P. Hubaux, and E. Knightly, "Denial of service resilience in ad hoc networks," in *Proc. 10th Annu. Int. Conf. Mobile Computing and Networking*, Philadelphia, PA, Sep. 2004, pp. 202–215.

[8] W. Yu and K. J. R. Liu, "Secure cooperative mobile ad hoc networks against injecting traffic attacks," in *Proc. 2nd Annu. IEEE Commun. Soc. Conf. Sensor and Ad Hoc Communications and Networks*, Sep. 2005, pp. 55–64.

[9] W. Yu and K. J. R. Liu, "Attack-resistant cooperation stimulation in autonomous ad hoc networks," *IEEE J. Selected Areas Commun. Special Issue Autonomic Commun. Syst.*, vol. 23, no. 12, pp. 2260–2271, Dec. 2005.

[10] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," presented at the SCS Commun. Networks Distributed Systems Modeling Simulation Conf., Jan. 2002.

[11] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," presented at the Int. Conf. Network Protocols, Nov. 2002.

[12] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," presented at the ACM Workshop on Wireless Security, Sep. 2002.

[13] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks," presented at the IEEE Infocom, 2003.

[14] Y.-C. Hu, A. Perrig, and D. B. Johnson, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," *Ad Hoc Netw. J.*, vol. 1, pp. 175–192, 2003.

[15] S. Capkun, J.-P. Hubaux, and L. Buttyan, "Mobility helps security in ad hoc networks," in *Proc. MobiHOC*, Annapolis, MD, Jun. 2003.

[16] S. Capkun and J.-P. Hubaux, "BISS: Building secure routing out of an incomplete set of security associations," presented at the WiSe, San Diego, CA, Sep. 2003.

[17] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the CONFIDANT protocol," in *Mobihoc*, 2002, pp. 226–236.

[18] L. Buttyán and J.-P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *Mobile Netw. Appl.*, vol. 8, no. 5, pp. 579–592, Oct. 2003.

[19] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," presented at the IEEE INFOCOM, 2003.

[20] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in wireless ad hoc networks," presented at the IEEE INFOCOM, 2003.

[21] P. Michiardi and R. Molva, "Core: A COllaborative REputation mechanism to enforce node cooperation in mobile ad hoc networks," presented at the IFIP Commun. Multimedia Security Conf., 2002.

[22] J. D. Kraus and R. J. Marhefka*, Antennas: for All Applications*, 3rd ed. New York: McGraw-Hill, 2002.

[23] S. Haykin*, Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[24] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks, mobile computing," in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, 1996, ch. 5, pp. 153–181.

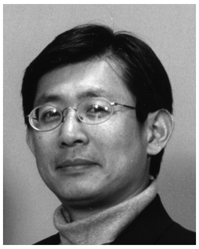
[25] *Secure Hash Standard*, , 1995, Federal Inf. Process. Std. Publ. 180-1.

[26] J. Yoon, M. Liu, and B. Noble, "Sound mobility models," presented at the MobiCom, San Diego, CA, Sep. 2003.

[27] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std. 802.11-1007, IEEE Comput. Soc. LAN MAN Standards Committee, Inst. Elect. Elect. Eng.

Wei Yu received the B.S. degree in computer science from University of Science and Technology of China (USTC), Hefei, China, in 2000, the M.S. degree in computer science from Washington University, St. Louis, MO, in 2002, and the Ph.D. degree in electrical engineering from University of Maryland, College Park, in 2006.

From 2000 to 2002, he was a Graduate Research Assistant at Washington University. From 2002 to 2006, he was a Graduate Research Assistant with the Communications and Signal Processing Laboratory and the Institute for Systems Research, University of Maryland. He joined Microsoft Corporation, Redmod, WA, in 2006. His research interests include network security, wireless communications and networking, game theory, wireless multimedia, and pattern recognition.



K. J. Ray Liu (F'03) is Professor and Associate Chair, Graduate Studies and Research, of Electrical and Computer Engineering Department, University of Maryland, College Park. His research contributions encompass broad aspects of wireless communications and networking, information forensics and security, multimedia communications and signal processing, bioinformatics and biomedical imaging, and signal processing algorithms and architectures. He was the Editor-in-Chief of *IEEE Signal Processing Magazine* and the founding Editor-in-Chief of the *EURASIP Journal on Applied Signal Processing*.

Dr. Liu is Vice President—Publications and on the Board of Governor of IEEE Signal Processing Society. He is the recipient of many honors and awards including best paper awards from the IEEE Signal Processing Society (twice), IEEE Vehicular Technology Society, and EURASIP; IEEE Signal Processing Society Distinguished Lecturer, EURASIP Meritorious Service Award, and National Science Foundation Young Investigator Award. He also received various teaching and research awards from the University of Maryland, including the Distinguished Scholar–Teacher award, Poole and Kent Company Senior Faculty Teaching Award, and the Invention of the Year award.