Defending against Wormhole Attacks in Mobile Ad Hoc Networks [*][†]

Weichao Wang **Bharat Bhargava** **Yi Lu** **Xiaoxin Wu**
wangwc, bb, yilu, wu @cs.purdue.edu
Department of Computer Sciences, Purdue University, W. Lafayette, Indiana

In ad hoc networks, malicious nodes can carry wormhole attacks to fabricate a false scenario on neighbor relations among mobile nodes. The attacks threaten the safety of ad hoc routing protocols and some security enhancements. We propose a classification of the attacks according to the format of the wormholes. It establishes a basis on which the detection capability of the approaches can be identified. The analysis shows that previous approaches focus on the prevention of wormholes between neighbors that trust each other. As a more generic approach, we present an end-to-end mechanism that can detect wormholes on a multi-hop route. Only trust between the source and the destination is assumed. The mechanism uses geographic information to detect anomalies in neighbor relations and node movements. To reduce the computation and storage overhead, we present a scheme, Cell-based Open Tunnel Avoidance(COTA), to manage the information. COTA achieves a constant space for every node on the path and the computation overhead increases linearly to the number of detection packets. We prove that the savings do not deteriorate the detection capability. The schemes to control communication overhead are studied. We show by simulations and experiments on real devices that the proposed mechanism can be combined with existent routing protocols to defend against wormhole attacks.

I. Introduction

As ad hoc networks are merging into the pervasive computing environment, security becomes a central requirement. Distributed node behaviour monitoring has been applied to enhance security. A system integrating watchdog and pathrater with the Dynamic Source Routing protocol (DSR) [1] is presented in [2]. In security enhanced Ad hoc On-demand Distance Vector protocol (AODV-S) [3], the neighbors collaboratively authorize a token to the node before it joins the network activities. The researchers have proposed several protocols that use hash chains or digital signatures to protect the integrity and authenticity of routing information. The Secure AODV protocol (SAODV) [4] adopts both mechanisms. Secure Efficient Ad hoc Distance vector routing (SEAD) [5] and Ariadne [6] use a variant of the Timed Efficient Stream Loss-tolerant Authentication (TESLA) [7] to accomplish authentication. A security-aware routing environment has been presented in [8].

Intrusion detection systems (IDS) have been adopted as the second line of defense to protect ad hoc networks. Zhang and Lee presented a generic multi-layer integrated IDS structure [9]. Bharghavan [10] and Haas *et al* [11] explored the security issues in wireless LANs and ad hoc networks respectively. An architecture that combines IDS with trust is presented by Alberts *et al* [12]. Intrusion detection using mobile agents is studied in [13].

In this paper, we focus on the detection of wormhole attacks in ad hoc networks. Since the mobile devices use a radio channel to send information, the malicious nodes can eavesdrop the packets, tunnel them to another location in the network, and retransmit them. This generates a false scenario that the original sender is in the

[*] This work is supported by NSF ANI 0219110, IIS 0242840, CERIAS, MOTOROLA, and CISCO URP.

[†] The paper is accepted to appear in Wiley Journal Wireless Communications and Mobile Computing (WCMC)

neighborhood of the remote location. The tunneling procedure forms a wormhole. It is conducted by collusive attackers. If a fast transmission path (e.g. a dedicated channel shared by attackers) exists between the two ends of the wormhole, the tunneled packets can propagate faster than those through a normal multi-hop route. This forms the "rushing attack" studied by Hu *et al* [14].

Wormhole attacks put severe threats to both ad hoc routing protocols and some security enhancements. In many routing protocols, mobile nodes depend on the neighbor discovery procedure to construct the local network topology. If the attackers tunnel the neighbor discovery beacons through wormholes, the good nodes will get false information about their neighbors. This may lead to the choice of a non-existent route. Zero-interaction authentication (ZIA) [15] is designed to protect the data on mobile devices from the illegal access. The files are decrypted only when an authentication token that is worn by the user can directly communicate to the device through a short-range wireless channel. If a wormhole exists between the token and the device, the data may be disclosed.

Some efforts have been put on this problem and encouraging results have been collected [16–18]. However, the classification in section II shows that the previous research focuses on the detection of closed wormholes. Half open and open wormholes, whose detection requires information exchanges beyond direct neighbors, have not been studied. In this paper, we propose an end-to-end mechanism that can detect wormholes along a multi-hop path on which the intermediate nodes do not necessarily trust each other. The mechanism combines authentication with location information. To prevent the destination node from being overwhelmed by the computation and storage overhead, a scheme called Cell-based Open Tunnel Avoidance (COTA) is introduced to manage the position records. COTA enables a node to pre-determine the resources that it wants to consume on wormhole detection. It reduces the overhead without hurting the detection capability. The frequency to conduct the detection is also studied to examine the communication overhead. The proposed mechanism can be combined with existent ad hoc routing protocols.

The contributions of this paper are:

- A classification of the wormholes is proposed. We divide the attacks into three groups (closed, half open, and open) according to the format of the tunnel and attacker's capability. The classification establishes a basis on which the detection capability of the approaches can be identified.
- We propose an end-to-end mechanism that can detect closed, half open, and open wormholes in ad hoc networks. Considering the limited resources available to a mobile node, we design COTA to manage the information. The communication overhead introduced by the proposed mechanism is also studied.
- Simulation is conducted to evaluate the overhead, detection capability, and accuracy of the proposed mechanism. Experiments on off-the-shelf mobile devices are conducted to examine the practicability of this method in real world.

The remainder of this paper is organized as follows: Section II describes the wormhole attacks and their impacts on ad hoc network security. The classification is given out. In section III, we review the previous work. Section IV presents and justifies our assumptions. Section V describes the end-to-end mechanism and COTA in detail. The problems such as detection capability and parameter determination are studied. Section VI studies the frequency to conduct wormhole detection and the method to control communication overhead. Section VII investigates the robustness of the proposed mechanism. Section VIII presents the simulation results. Section IX discusses the future work and section X concludes the paper.

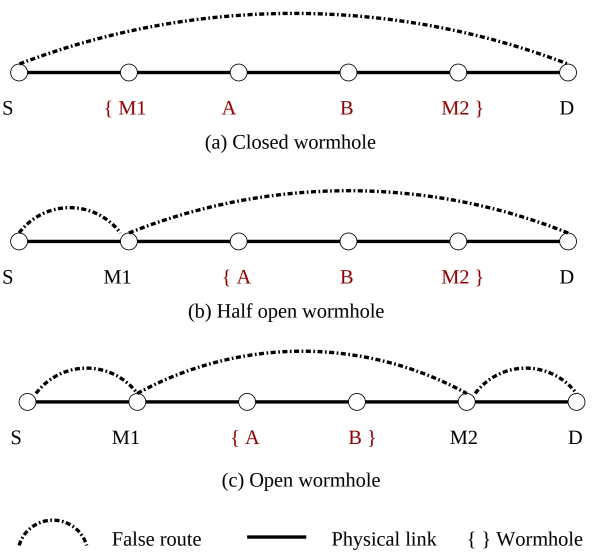
II. Problem Statement

In a wormhole attack, if the malicious nodes have a dedicated channel, the tunneling procedure can be conducted in real time. Since the packets are resent in the exactly same way, encryption or authentication alone cannot prevent the attacks. Other nodes cannot tell whether the packets are from the real originator or from the resender. A group of collusive attackers can form a wormhole that has as many ends as the number of malicious nodes.

Wormhole attacks put severe threats to ad hoc routing protocols. In the protocols that use distance vector technique, such as Ad hoc On-demand Distance Vector protocol (AODV) [19] and Destination-Sequenced Distance Vector protocol (DSDV) [20], the hop count of a path affects the choice of routes. A pair of attackers can form a long tunnel and fabricate the false scenario that a short path exists between the source and the destination. The fake path will attract the data traffic. As soon as the packets are absorbed to the wormhole, the attackers can either drop them or compromise them. The attacks can also do harm to the hierarchical routing protocols such as Hierarchical State Routing protocol (HSR) [21], Clusterhead Gateway Switch Routing protocol (CGSR) [22], and Adaptive Routing using Clusters (ARC) [23]. They may confuse the clustering procedure and lead to a wrong topology. If a wormhole controls the link between two clusterheads or a link close to the root of the routing hierarchy, it can partition the network.

The safety and effectiveness of some security enhancements for ad hoc networks would be improved if wormholes can be defended. The example on ZIA has been shown in section I. Another example shows the impacts of such attacks on the distributed monitoring of node misbehaviours. In AODV-S [3], the neighbors collaboratively authorize a token to the node before it joins the network activities. If a wormhole exists beside the misbehaved node, the attackers can selectively tunnel the good-looking packets to the remote side. The good nodes at the remote side monitor all these packets and can not detect any security violations. The new token will be authorized. This may conflict with the conclusion drawn by the real neighbors. We can settle this embarrassment through preventing wormholes.

The classification of such attacks will facilitate the design of detection methods. According to whether the attackers are visible on the route, we classify the wormholes into three types: closed, half open, and open. The examples that include two malicious nodes are shown in Figure 1. For the clearance of the remainder of the paper, we use $M1$ and $M2$ to represent the malicious nodes, S and D to represent the good nodes as source and destination, and A, B, *etc.* as the good nodes on the route.



(a) Closed wormhole

(b) Half open wormhole

(c) Open wormhole

Figure 1: The classification of wormholes.

The nodes between the curly-braces ("{}") are those on the path but invisible to S and D because they are in a wormhole. We borrow the words "closed" and "open" from the definition of an interval, where "closed" means "start from and include", and "open" means "start from but not include". In Figure 1-(a), $M1$ and $M2$ tunnel the neighbor discovery beacons from S to D, and vice versa, so S and D think that they are direct neighbors. Both $M1$ and $M2$ are in the wormhole. In Figure 1-(b), $M1$ is a neighbor of S and it tunnels its beacons through $M2$ to D. Only one malicious node is visible to S and D. In an open wormhole, both attackers are visible to S and D, as shown in Figure 1-(c).

The previous research focuses on the prevention of closed wormholes. The mechanisms that only examine direct neighbors cannot guarantee the detection of the other two types. For example, S and $M1$, and D and $M2$ are real neighbors in Figure 1-(c). This does not impact the establishment of an open wormhole between $M1$ and $M2$. Therefore, an end-to-end mechanism must be designed to defend against the half open and open wormholes.

Wormhole detection needs to be conducted when a neighbor relation or a route is first established. Besides, it must be conducted repeatedly during the lifetime of the neighbor relation or the route because the nodes are moving and wormholes can be formed dynamically. The detection frequency impacts the overhead and the detection accuracy. This problem is studied in section VI.

III. State of Art

The wormhole attackers encapsulate the received packets into their packets and tunnel them to another location. Ironically, this technique was first introduced to overcome some difficulties in networking. IP-tunnel is used in Mobile IP to transfer packets from home agents to remote agents [24]. In Virtual Private Networks (VPN), the tunnels are used to connect two LANs, allowing them to share data, but keeping them behind the firewalls [25]. IP-tunnels are also used in Multi-Protocol Label Switch (MPLS) [26].

Wormhole attacks threaten the safety of Internet routing protocols such as RIP2 [27]. But the attacker needs to have a total control on a router, which is not easy to achieve. Furthermore, through using static routes, such attacks can be prevented. Because of the dynamic membership and frequent topology changes, a node in ad hoc networks does not have this luxury.

Wormhole attacks on mobile ad hoc networks were independently discovered by Dahill *et al* [28], Hass *et al* [29], and Hu *et al* [16]. To defend against them, some efforts have been put on hardware design and signal processing techniques. If the data bits are transferred in some special modulating method known only to the neighbor nodes, they are resistant to the closed wormholes. Another approach, RF watermarking, works in the similar way. It modulates the radio waveform in a specific pattern to accomplish authentication. Both mechanisms will be compromised if the malicious nodes can accurately capture the signal patterns. Neither of them can prevent half open or open wormholes.

The adoption of directional antenna [30, 31] by mobile devices can raise the security levels. A solution that uses such equipments to defend against closed wormholes has been presented in [18]. The nodes examine the directions of the received signals from each other and a witness. Only when the directions of both pairs match, the neighbor relation is confirmed.

Another potential solution is to integrate the prevention methods into Intrusion Detection Systems. The traffic monitoring module of IDS will find that the ends of wormholes act as packet sinks: many data packets which are not destined to them will lose their tracks at these nodes. The joint response generated by the neighbors of the malicious node will expose the anomalous traffic pattern.

Some mechanisms proposed to locate the position of a mobile node in an indoor environment [32–34] can be applied to prevent wormholes. For example, both the original packet and the resent one will be captured by the location sensors and two conflicting positions of the same node will be detected. Either the good nodes or a centralized controller will discover this anomalous result. However, it will not be easy to port such methods to outdoor environments.

One approach to detect closed wormholes without clock synchronization is proposed by Capkun *et al* in [17]. Every node is assumed to be equipped with a special hardware that can respond to a one-bit challenge without any delay. The challenger measures the round trip time of the signal with an accurate clock to calculate the distance between the nodes. The probability that an attacker can guess all bits correctly decreases exponentially as the number of challenges increases.

Another approach to detect closed wormholes is packet leash, which was proposed by Hu, Perrig and Johnson [16]. The leash is the information added into a packet to restrict its transmission distance. In the geographical leashes, the location information and loosely synchronized clocks together verify the neighbor relation. In temporal leashes, the packet transmission distance is calculated as the product of signal propagation time and the speed of light. Both mechanisms use lightweight hash chains to authenticate the nodes [7]. The Message Authentication Code (MAC) can be calculated in real time.

One advantage of packet leashes is the low computation overhead. Similar to geographical leashes, the end-to-end mechanism proposed in this paper assumes the knowledge of location information and loosely synchronized clocks. It can be deployed in the environments where geographical leashes can be used to detect closed, half open, and open wormholes.

IV. Assumptions and Notations

IV.A. Network assumptions

We assume that the links among nodes are bidirectional. Two neighbor nodes can always send packets to each other. This assumption will hold under most conditions in the real environment. Many medium access control protocols [35, 36] are also based on this assumption.

The security threats to mobile ad hoc networks come from all layers. The malicious node can jam the physical layer. There have been approaches using spread spectrum [37] to provide resistance to such attacks. There are also Deny of Service (DoS) attacks on the medium access control layer [38]. For example, if a malicious node keeps sending noises and causes collisions, the communication within the neighborhood will be paralyzed. The fairness control mechanisms such as time division multiple access [39] can avoid one attacker eating up all available bandwidth. In this paper we will not discuss solutions to those attacks.

The network itself acts in Byzantine manners [40]. It may drop or corrupt packets. The packets can be duplicated or forwarded out of order.

IV.B. Node assumptions

A node can locate its geographic position. There have been off-the-shelf devices, such as Global Positioning System (GPS) [41], that can provide accurate positioning service. The accuracy of position information will affect the detection capability of the proposed mechanism. For example, if the direct communication range among neighbors is d and the maximum error of the distance between two locations is δ, they introduce a relative error up to δ/d. The mechanisms such as GPS can locate the position with the accuracy of 15 feet.

In certain environments, using the distance between two nodes to verify the neighbor relation cannot guarantee the detection. For example, even if an obstacle (e.g. a building) exists between two nodes and blocks all signals, a position based mechanism may still consider them as neighbors. To detect such conditions, the nodes need a more accurate signal propagation model in that environment. It is able to determine whether the two nodes can actually communicate to each other when the positions are given. In this paper, we assume that any two nodes having a distance shorter than d can communicate to each other.

Each node has a clock, and we assume that the clocks are loosely synchronized with the maximum relative error Δ. By this we mean that the difference between the clocks of any two nodes is smaller than Δ if we sample the time at the same moment.

The mobile nodes have limited computation resources. But they can accomplish the operations required by the security mechanisms, such as the calculation and verification of digital signatures, encryption using symmetric keys, and calculation of the MAC code.

We do not assume that a node can control the longest buffering time of a packet before it is forwarded. We do not assume trusted hardware such as tamper-proof wireless cards. They may restrict the deployment of the proposed mechanism.

IV.B.1. Assumption justification

A lot of research in ad hoc networks has been conducted based on the position awareness assumption. They cover from routing [42–44], energy consumption [45, 46], to location based services [47, 48]. Some work has been conducted based on loose synchronization assumption, such as geographical leashes [16], Secure Tracking of Node Encounters (SECTOR) [17], Ariadne [6], and Light-weight Hop-by-hop Authentication Protocol (LHAP) [49]. In this part, we justify our assumptions based on the available techniques and application scenarios.

Location-based services using GPS are ready to launch with the progresses that reduce expense, size, weight, and power consumption of such devices. Motorola has unveiled one kind of GPS chip set whose unit cost is about $10 [50]. It is cheap and small enough to fit in PDAs and cell phones. Sprint PCS alone has sold over 8.8 million GPS-enabled handsets by March, 2003 [51]. The positioning device is becoming a common part of the wireless node for both military and civilian usage.

A lot of efforts have been put on clock synchronization in ad hoc and sensor networks. There have been solutions based on LORAN-C [52], WWVB [53], Reference Broadcast [54], TINY/MINI-SYNC [55], Tree-based solution [56], and GPS [57]. A good survey can be found in [56]. In this paper we only assume loosely synchronized clocks among nodes, for example, $\Delta \leq 1 second$. If every node is equipped with GPS, its clock is accurate enough to satisfy this requirement. The mobile nodes do not need extra hardware or a complex protocol to get more accurate time.

The security of GPS has been studied for some time. Solutions for anti-jamming and anti-spoofing civilian GPS signals have been proposed in [58–60]. Here we do not present the details. GPS signals can be weak or biased in an indoor environment. But as discussed in section III, a more accurate positioning scheme in such an environment can be implemented to defend against wormholes.

IV.C. Key setup

The safety of the end-to-end mechanism relies on the secrecy and authenticity of the keys stored in nodes. The authentication can be accomplished by pairwise secret keys, digital signatures, or a variant of TESLA [7].

The advantage of pairwise keys is that the nodes can use symmetric cryptographic methods and avoid the expensive operations such as exponential computation. It is very important to the mobile nodes with limited resources. The disadvantages are two folds. First, if there are n nodes in the network, we need to set up $n(n-1)/2$ keys. This can impact the scalability of the proposed mechanism. Second, it will be difficult to authenticate multicast or broadcast packets. Although there has been research on allowing multiple users to share a group key, it will put threats to the whole scheme if one node is compromised [61].

If digital signatures are used, we only need to set up n keys. The method supports the authentication of multicast traffic and has the non-repudiation property. But the computation of a digital signature can be three orders of magnitude slower than that of the symmetric mechanisms [62]. With the emergence of high speed signature algorithms [63, 64] and more powerful portable processors, computing digital signatures will cause less overhead on the mobile nodes.

With the assumption of loosely synchronized clocks, if the end-to-end delay can be accurately predicted, a variant of TESLA can be applied to accomplish authentication. We only need to set up n keys and the nodes do not have to do the expensive computations. The disadvantage is that the packets may not be authenticated immediately. The receiver needs to temporarily buffer the packet until the key is disclosed. This may require more storage space.

To set up the keys, either centralized mechanisms, such as a key distribution center or a Public Key Infrastructure (PKI), or a pre-load method during initialization, can be applied. A survey of key establishment in mobile networks can be found in [65]. Another solution proposed in [66] applies a PGP-like mechanism to distribute the keys.

IV.D. Model of attackers

The attackers do not have the computation power to crack the secret keys. To tunnel the packets beyond a long distance without interfering with the signals sent by the good nodes, the attackers have a dedicated, speed-of-light communication channel. The attackers have a total control over the wormholes. They can choose to tunnel a packet through or drop it without knowing the content of the packet.

IV.E. Notations

For the clearance of the remainder of the paper, we give out the following notations:

If pairwise keys are applied, K_{AB} and K_{BA} are same and both represent the secret key between node A and node B. $MAC_K(M)$ represents the MAC code computed over message M with key K. If digital signatures are applied, $Sign_A(M)$ represents the signature of A over message M. All nodes that have the public key can verify the signature.

Every node can locate its geographic position. The position of node A is P_A. The maximum error of the distance between two positions is δ. If node A and node B read their positions at the same time, the real distance between them $d_{AB} \leq ||P_A - P_B|| + \delta$. The clocks are loosely synchronized. The difference between the clocks of any two nodes is smaller than Δ. We also assume that v is the upper bound of the velocity of node movement.

V. Detecting Wormhole Attacks

V.A. Basic end-to-end mechanism

Design goals

In the end-to-end wormhole detection mechanism, we only assume that the source and the destination of a route trust each other. This assumption holds under most conditions. If on-demand routing protocols are used, the source and the destination can negotiate the parameters such as the frequency to send wormhole detection packets through the routing request and reply. If geographical routing protocols are used, the destination can pre-determine the parameters and submits them to the location servers. The source will get them when acquiring the position of the destination.

As shown in section II, wormholes need to be examined repeatedly during the route's lifetime. The detection information can be attached to the routing packets or the data packets. If the frequency of data packets is too low, wormhole detection packets can be sent separately. For multiple applications that use the same route, only one group of detection packets need to be sent. No detection is required for a route if it is no longer in use.

Compared to the mechanisms that only detect closed wormholes, the end-to-end mechanism has two special features:

First, every intermediate node will attach its timestamps and positions to the detection packets, but all examining operations are conducted by the destination node. In this way it can detect the conflicting information sent by the attacker to different neighbors. However, a scheme must be designed to prevent the destination node from being overwhelmed by the overhead.

Second, cross packet examination must be adopted by the end-to-end mechanism. Examining the packets independently can miss some wormholes. For example, in the half open wormhole shown in Figure 1-(b), $M1$ can declare that it receives a packet at the position of $M1$ and forwards it at the position of $M2$. As long as the distance d_{M1M2} and the declared buffering time t satisfy $d_{M1M2} \leq v \times t$, examining the single packet cannot find any anomaly.

In the cross packet examination, if a node declares its position P_1 at its clock time t_1, and P_2 at its clock time t_2, the destination can estimate its average moving speed and examines whether it is lying. If

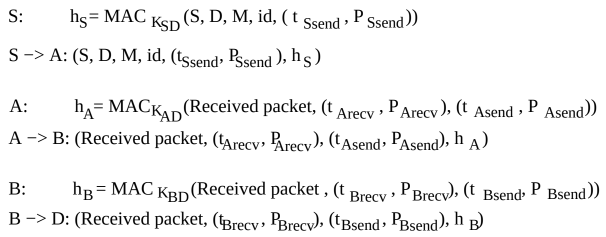
$$\frac{||P_1 - P_2|| - \delta}{||t_1 - t_2|| + \Delta} > v \tag{1}$$

the node lies about its positions and there is a wormhole on the path.

Delivery of detection packets

We assume that the pairwise keys have been deployed. Every node A on the path has a secret key K_{AD} with the destination D. D knows the path length in hops and the identity of every node along the path. This can be achieved through the routing protocols such as DSR [1] or the explicit attachment of node identify to the routing packets.

When a wormhole detection packet is sent from the source, it contains seven fields: source address S, destination address D, the message M, a sequence number id for this route, the local time t_{Ssend} when the packet is sent, the position P_{Ssend} at that time, and the MAC code $MAC_{K_{SD}}(\ S, D, M, id, t_{Ssend}, P_{Ssend})$. M can be a routing packet, a data packet, or void if the detection packet is sent separately. Every time af-



S: $h_S = MAC_{K_{SD}}(S, D, M, id, (t_{Ssend}, P_{Ssend}))$

S −> A: $(S, D, M, id, (t_{Ssend}, P_{Ssend}), h_S)$

A: $h_A = MAC_{K_{AD}}$(Received packet, $(t_{Arecv}, P_{Arecv}), (t_{Asend}, P_{Asend})$)

A −> B: (Received packet, $(t_{Arecv}, P_{Arecv}), (t_{Asend}, P_{Asend}), h_A$)

B: $h_B = MAC_{K_{BD}}$(Received packet , $(t_{Brecv}, P_{Brecv}), (t_{Bsend}, P_{Bsend})$)

B −> D: (Received packet, $(t_{Brecv}, P_{Brecv}), (t_{Bsend}, P_{Bsend}), h_B$)

Figure 2: The delivery of a wormhole detection packet in the end-to-end mechanism.

ter sending a detection packet for this route, the source increases id by 1. Examining the received ids, D can find out how many detection packets have been dropped.

When an intermediate node A forwards the packet, it attaches two ⟨time, position⟩ pairs: $\langle t_{Arecv}, P_{Arecv} \rangle$ when it receives the packet, and $\langle t_{Asend}, P_{Asend} \rangle$ when it forwards the packet. Then it attaches $MAC_{K_{AD}}(M_A)$, where M_A includes both the received packet and the two attached pairs. Figure 2 shows an example of the delivery procedure.

Two potential problems exist in the proposed mechanism. First, how can a node get the accurate sending time and calculate it into the MAC code? Second, how does the signal propagation time affect the detection? For the first problem, the end-to-end mechanism has a weaker requirement on the accuracy of the sending timestamp than packet leashes. So we can adopt either of the two approaches [16] for temporal leashes.

For the second problem, taking the signal propagation time t_{prop} into account, we have the following equation for the sending time t_{send} at this node and the receiving time t_{recv} at the next hop:

$$||t_{recv} - t_{send}|| \leq \Delta + t_{prop} \qquad (2)$$

If the direct communication range between neighbors is known, we can estimate $\Delta' = \Delta + t_{prop}$. The sending time and the receiving time of the same transmission always satisfy:

$$||t_{recv} - t_{send}|| \leq \Delta' \qquad (3)$$

Detection operations at the destination

When D receives a detection packet, it will check the following items: (a) All MAC codes are calculated correctly. (b) The neighbor nodes are within the direct communication range when the packet is passed. (c) The average moving speed of a node between it receives and forwards the packet does not exceed v. (d) The sending and receiving time of the same transmission satisfy equation 3. (e) The new ⟨time, position⟩ pair and the previous pairs of the same node do not conflict as shown in equation 1.

Items (a), (b), (c) and (d) focus on a single packet. The MAC codes guarantee the authenticity and integrity of the ⟨time, position⟩ pairs. Since the MAC codes cover the whole packet (including the information attached by previous nodes), it is very difficult for the attacker to record a part of the packet and conduct resend attacks. If the attacker resends the whole packet, it is the same as the duplicate packet generated by the network. Item (b) will detect the closed wormholes on the route.

Item (e) implements the cross packet examination. Through calculating the average moving speed of the intermediate nodes using their current and previous ⟨time, position⟩ records, the destination node monitors their movements. Items (c), (d) and (e) together prevent the malicious node from cheating the wormhole detection mechanism by declaring fake positions.

The wormhole detection packets may get lost because of the unreliable network. They can also be intentionally dropped or corrupted by the malicious nodes. Using the sequence number, the destination node can monitor how many detection packets have been lost. If a wormhole is detected, or q consecutive detection packets are all lost, the destination node will broadcast a message which notifies the source to abort the current route and activate the reinitiation. The relation between the value of q and the position information density is discussed in section VI. The same condition will happen if a route becomes expired and no more detection packets are sent. Under this condition, the source will ignore the broadcast message.

Compared to previous approaches, the end-to-end mechanism does not let the intermediate nodes verify the neighbor relations by themselves. On the contrary, all examinations are conducted by the destination. The proposed mechanism can detect closed wormholes because the good nodes at different ends of the tunnels will not lie about their positions. For half open and open wormholes, if the malicious nodes send their real positions in the detection packets, the fake neighbor connections will be discovered because they are longer than the direct communication range. To avoid being detected, a smart attacker can buffer the packets and declare that it moves to the other end of the wormhole and forwards the data. This attack can be prevented by introducing packet lifetime into the network. It will restrict the longest moving distance of a node during the delivery of a single packet so that the length of the tunnel will be limited. It will also guarantee the position information density of the intermediate nodes as shown in section VI.

Restricted by the positioning and clock synchronization errors, the end-to-end mechanism could introduce false alarms into the system. For example, the attacker will be able to tunnel the packets δ beyond d without being detected if the destination considers the positioning error when calculating the distance between two points. On the contrary, if the destination ignores the error, some connections having a distance close to d will be wrongly accused as wormholes. With the progresses in the positioning and synchronization techniques, the error rate will become smaller.

V.B. Overhead of basic end-to-end mechanism

Since the mobile devices have limited resources, the end-to-end mechanism, as a security enhancement, must consider the communication, computation, and storage overhead.

Every intermediate node attaches two $\langle$time, position$\rangle$ pairs and a MAC code to each detection packet. If pairwise keys are used, it has been shown in [16] that a PDA can accomplish 220K times MAC code calculation in one second. So there is not much computation overhead at the intermediate nodes. And they do not need to store any information.

The communication overhead includes byte overhead and packet overhead. If we use an eight-byte timestamp (if the time unit is $1ns$, it covers more than 500 years), an eight-byte position (on the surface of the Earth, it locates a position within $0.1m$), and an eight-byte MAC code, every intermediate node will attach 40 bytes to the packet. If the maximum transmission unit (MTU) is 1,500 bytes and M is 1,000 bytes, twelve nodes can attach their information.

In this way the byte overhead will increase fast to the path length. It does not scale well to long routes. To control the byte overhead, the destination node can choose a part of the intermediate nodes to attach their information. It changes the selected nodes to guarantee that everyone is examined. If k consecutive nodes on the route are chosen, it is the same as an end-to-end detection on these k nodes. More details of this method and its impact on the detection capability are studied in section VI. The packet overhead is determined by the frequency to send the detection packets. This question is also studied in section VI.

All examining operations are conducted by the destination node. We assume that the path length is l, and m packets carrying the detection information arrive at the destination. It needs $O(l)$ operations to examine a single packet. For every intermediate node, there will be $2m$ $\langle$time, position$\rangle$ pairs. If the cross packet examination calculates the average moving speed between every two pairs, there will be $O(m^2)$ operations for each node. So the total computation overhead for the m packets will be $O(lm + lm^2)$.

For the same reason, the destination needs to store $2m$ $\langle$time, position$\rangle$ pairs for every intermediate node. The required storage space will be $O(lm)$.

The lm^2 entry in the computation overhead and lm entry in the storage space put challenges to the mobile nodes with limited resources. For example, if a route is ten-hop long and 1,800 wormhole detection packets are received, the destination needs to store 36K $\langle$time, position$\rangle$ pairs (about 580 Kbytes), and conducts 65M operations. As the number of routes increases, the node cannot afford the required resources. In the next section, we propose a mechanism called COTA that requires $O(c_1 l)$ space and $O(c_2 lm)$ computation overhead (where c_1 and c_2 are constant), but having the same wormhole detection capability as the end-to-end mechanism.

V.C. Cell-based Open Tunnel Avoidance

COTA avoids to record and compare all $\langle$time, position$\rangle$ pairs. It divides the whole area into same-sized cells (hexagon), and divides the time into same-length slots. COTA only stores the first received $\langle$time, position$\rangle$ pair of every node that falls into the same cell and the same slot. Through adjusting the cell size and slot length, a node can control the efforts that it wants to put on the wormhole detection. The whole structure is a three dimensional cube of $\langle$time, position$\rangle$ records, and the format of the index is $\langle$node identity, cell number, slot number$\rangle$. When a new $\langle$time, position$\rangle$ pair of a node arrives, the destination selects from each cell a record of that node that has the shortest time difference from the new timestamp and calculates the average moving speed. In this way some pairs belonging to the same node are not compared. In section V.D we prove that after adding an offset to equation 1, COTA has the same detection capability as the end-to-end mechanism.

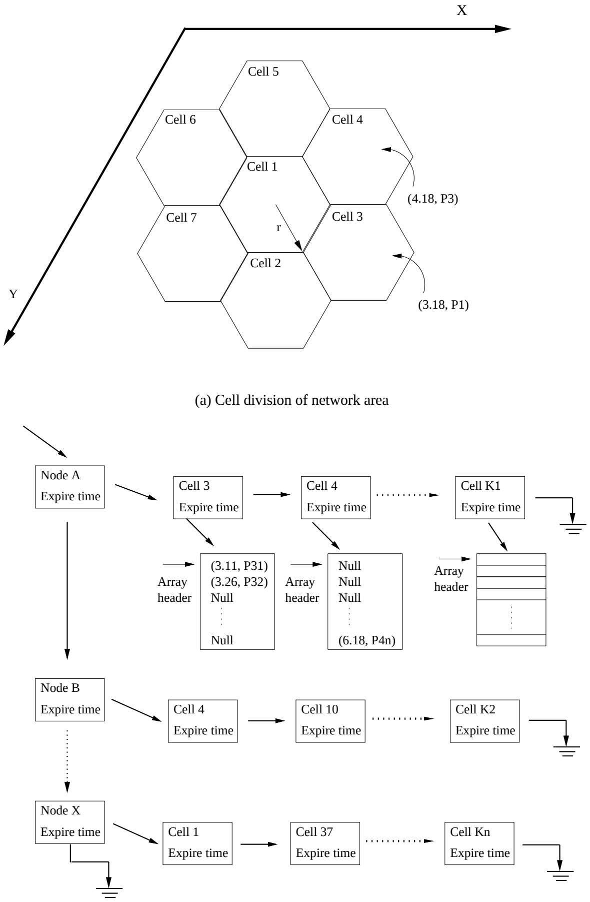
We also define the lifetime T_{life} of a packet. If a packet has traveled in the network for a time longer than T_{life}, it will be discarded by the destination node. Since the clocks are loosely synchronized, COTA can estimate the packet traveling time. To guarantee that the packet arrives at the destination D within its lifetime, the sending time at S and the receiving time at D should satisfy $(T_{Drecv} - T_{Ssend}) \leq T_{life} + \Delta$.

Two advantages have been brought to COTA by the definition of packet lifetime. First, it restricts the number of time slots that COTA needs to store for every intermediate node. If the slot length is T, the destination node only needs to store at most $(T_{life} + \Delta)/T + 1$ records for every intermediate node in every cell. Second, it restricts the longest moving distance of a node during the delivery of a single packet. It prevents the attacker in an open or half open wormhole from buffering the packet for a long time and declaring that it moves to the new position and forwards the packet.

An example of cell division and the data structure is shown in Figure 3. Every record contains a $\langle$time, position$\rangle$ pair. We assume that the time slot is 100 ms. For one intermediate node, the destination will only store one record in every slot in a cell. Now assuming that another record of node A with the content $\langle t_1$ = 3.18 sec, position = $P_1\rangle$ is received, and it falls into cell 3. Since there is already a record of node A in cell 3 in the slot 3.1 – 3.2 second, the new entry will not be recorded. Then the destination will select from each cell the record of node A that has the shortest time difference from the new timestamp. In this example, they will be (3.11, P_{31}) in cell 3 and (6.18, P_{4n}) in cell 4. If the selected pair is represented as $\langle t_2, P_2\rangle$, D will calculate the average moving speed of node A between these two positions by

$$\bar{v} = \frac{max(0, ||P_1 - P_2|| - \delta)}{||t_1 - t_2|| + \Delta} \qquad (4)$$

If $\bar{v} > v$, we know that A sent false information and there is a wormhole on the route.



(a) Cell division of network area

(b) Data structure organization

Figure 3: Cell division and data structure of COTA in one destination node.

Now let us consider another example. A record $\langle t_3$ = 4.18, position = $P_3\rangle$ of node A is received, and it falls into cell 4. There is no record of node A in cell 4 in the slot 4.1 – 4.2 second. So the new entry will be recorded by the destination. Then D will select the record of node A from each cell that has the shortest time difference from the new timestamp and calculates the average speed. In this case, they will be (3.26, P_{32}) in cell 3 and (6.18, P_{4n}) in cell 4.

COTA only stores and compares a part of the ⟨time, position⟩ records to reduce the storage and computation overhead. There are two questions that we have to ask: (1) can COTA detect all anomalies that the end-to-end mechanism can detect? (2) how much space and computation do we save? We discuss the answers in the next two sections.

V.D. Detection capability of COTA

COTA simplifies the end-to-end mechanism to reduce the storage and computation overhead. But it may miss the detection of some anomaly. An example is given out as follows. We have four ⟨time, position⟩ pairs of node M_1 as $\langle t_{M_1 1}, P_{M_1 1}\rangle$, $\langle t_{M_1 2}, P_{M_1 2}\rangle$, $\langle t_{M_1 3}, P_{M_1 3}\rangle$, and $\langle t_{M_1 4}, P_{M_1 4}\rangle$. The first two pairs fall into cell 1 and slot 1. The second two pairs fall into cell 2 and slot 2. We calculate the average moving speed between every two pairs using equation 4. The speed between pair two and four is faster than v, but the speed between any other two pairs is not. If we record and compare all pairs, we will find the anomaly and detect the wormhole. However, if COTA is applied and pair one and three arrive first, they will be recorded. Later when pair two and four arrive, they will not be stored and they are never compared to each other. COTA does not detect this anomaly.

However, adding an offset to equation 4 will enable COTA to detect all wormholes that the end-to-end mechanism can detect. The proof is given as follows.

Lemma 1 *: If COTA uses*

$$\frac{max(0, ||P_{new} - P_{select}|| - \delta + 2r + vT)}{||t_{new} - t_{select}|| + \Delta} \quad (5)$$

to calculate the average moving speed between the new and the selected ⟨time, position⟩ pairs, it can detect all wormholes that can be detected by the end-to-end mechanism. In equation 5, δ is the maximum error of the distance, r is the radius of cells, v is the highest speed of nodes, T is the length of a time slot, and Δ is the clock error.

Proof 1 *: We assume that we have two pairs, $\langle t_1, P_1\rangle$ and $\langle t_2, P_2\rangle$, of the same node that show the anomaly*

$$\frac{||P_1 - P_2|| - \delta}{||t_1 - t_2|| + \Delta} > v \quad (6)$$

And without losing generality, we assume that $\langle t_1, P_1\rangle$ is received by the destination first. When $\langle t_2, P_2\rangle$ arrives, there are two possible conditions:

Condition I: $\langle t_1, P_1\rangle$ is recorded by COTA.

If $\langle t_1, P_1\rangle$ is selected to be compared with $\langle t_2, P_2\rangle$, the average moving speed (after adding $2r + vT$) is faster than v, and the anomaly will be detected.

If it is not selected, there must be another pair $\langle t_3, P_3\rangle$ that falls into the same cell as P_1 but has a shorter time difference from t_2. So we have $||t_3 - t_2|| \leq ||t_1 - t_2||$. Since P_3 and P_1 are in the same cell, the longest distance between the two positions is 2r. So we have:

$$\begin{aligned} ||t_3 - t_2|| &\leq ||t_1 - t_2|| \\ ||P_3 - P_2|| + 2r &\geq ||P_1 - P_2|| \end{aligned} \quad (7)$$

Combining equation 6 and 7, we must have

$$\frac{||P_3 - P_2|| - \delta + 2r}{||t_3 - t_2|| + \Delta} > v \quad (8)$$

Condition II: If $\langle t_1, P_1\rangle$ is not recorded by COTA, there must be another pair $\langle t_3, P_3\rangle$ that is recorded. It falls into the same cell and same slot as $\langle t_1, P_1\rangle$, but it arrives earlier. So we have:

$$\begin{aligned} ||P_3 - P_2|| + 2r &\geq ||P_1 - P_2|| \\ ||t_3 - t_2|| - T &\leq ||t_1 - t_2|| \end{aligned} \quad (9)$$

If $\langle t_3, P_3\rangle$ is selected to be compared with $\langle t_2, P_2\rangle$, combining equation 6 and 9, we will get:

$$\frac{||P_3 - P_2|| - \delta + 2r + vT}{||t_3 - t_2|| + \Delta} > v \quad (10)$$

If $\langle t_3, P_3 \rangle$ is not selected to be compared with $\langle t_2, P_2 \rangle$, there must be another recorded pair $\langle t_4, P_4 \rangle$ that falls into the same cell but has a shorter time difference. So we have:

$$||t_4 - t_2|| - T \leq ||t_3 - t_2|| - T \leq ||t_1 - t_2|| \quad (11)$$

Combining equation 6 and 11, we will get

$$\frac{||P_4 - P_2|| - \delta + 2r + vT}{||t_4 - t_2|| + \Delta} > v \quad (12)$$

Combining all conditions shown in equation 6, 8, 10, and 12, we have:

$$\frac{||P_{select} - P_{new}|| - \delta + 2r + vT}{||t_{select} - t_{new}|| + \Delta} > v \quad (13)$$

□

After introducing the offset $2r + vT$ into COTA, we guarantee that it can detect all wormholes that the end-to-end mechanism can detect. We define $2r + vT$ as the $sensitivity$ of COTA. An optimistic node can use COTA as in equation 4. It will allow all neighbors that are within the direct communication range to exchange packets. But in some cases, a real attacker is able to tunnel a packet as far as $2r + vT$ beyond the range. A more conservative node can use the format shown in equation 5. It will guarantee the detection capability, but in some cases it will introduce false positive alarms. The impact of false alarms will be studied through simulation in section VIII.

V.E. Parameter determination

There are three parameters in COTA: packet lifetime, cell size, and time slot length. The choices of the parameters impact not only the detection capability of COTA, but also the storage and computation overhead. We now discuss them separately.

The choice of packet lifetime is directly related to the end-to-end delay in ad hoc networks. Estimating this delay accurately is a difficult problem because it is affected by traffic load, path length, movement model, network size, and other features. There have been theoretical analyses of this problem in simplified network scenarios [67–70]. The results provide valuable guidelines for the design of protocols. However, the various simplifying assumptions do not always hold in real network settings.

Another approach is to conduct real measurements. The results collected from simulations have been shown in [71–74]. The typical value of the average delay is a few hundred milliseconds in the studied network scenarios. In this paper, we assume the similar network conditions, thus the similar average delay is also assumed. The analysis in [73] shows that after the mode of the curve, the probability density function of delay decreases exponentially as time increases. To control the fraction of detection packets that are discarded because of unexpected long delay, we set T_{life} an order of magnitude longer than the average delay. The typical value of T_{life} is a few seconds. When an approach for more accurate delay estimation appears, we can replace this method without impacting other components of COTA.

Now let us consider the required storage space and computation operations of COTA. We assume that the packet lifetime is T_{life}. The destination node only needs to record the ⟨time, position⟩ pairs that were sent in the past $T_{life} + \Delta$. During this period, the possible position of a good node must be within a circle with the diameter $v(T_{life} + \Delta)$ (use the position when $t = (T_{life} + \Delta)/2$ as the center, and $v(T_{life} + \Delta)/2$ as the radius). If there are no wormholes, the number of cells that have active records for the node is at most

$$\frac{\pi(v(T_{life} + \Delta)/2)^2}{1.5\sqrt{3}r^2}. \quad (14)$$

In each cell, at most $(T_{life} + \Delta)/T + 1$ records are stored. So the total number of records stored by the destination for one node is at most

$$\frac{\pi v^2 (T_{life} + \Delta)^2}{6\sqrt{3}r^2}\left(\frac{T_{life} + \Delta}{T} + 1\right) \simeq \frac{\pi v^2 (T_{life} + \Delta)^3}{6\sqrt{3}r^2 T} \quad (15)$$

If the path length is l, the destination needs to store at most $l \times Equation(15)$ records for one route.

When a new record arrives, the destination will select from each cell a record of that node to compare with it. There are at most $(T_{life}+\Delta)/T+1$ records in each cell for the node. If a linear search is applied, the new record needs at most

$$\frac{\pi(v(T_{life}+\Delta)/2)^2}{1.5\sqrt{3}r^2} \times (\frac{T_{life}+\Delta}{T}+1) \quad (16)$$

operations to accomplish the cross packet examination. If a balanced tree is used to store the records in each cell for a node, the examination in one cell can be accomplished in $\log_2((T_{life}+\Delta)/T+1)+1$ operations. But the maintenance overhead for the tree structure may outrun the benefits. In the following analysis, we assume that the linear search is applied. If there are m COTA packets for the route and the path length is l, the total number of operations required by COTA is

$$2ml \times Equation(16) \quad (17)$$

We examine a practical example to show the savings of COTA. We assume that r is 12 m, T_{life} is 5 seconds, v is 20 m/s, T is 200 ms, l is 10 hops, and 1,800 wormhole detection packets are received. The destination node needs to store at most 9 K $\langle$time, position$\rangle$ pairs. If a linear search is used to locate the record in every cell, the destination needs to conduct 32.5 M operations. Compared to the overhead of the end-to-end mechanism shown in section V.B, COTA saves 75% on space and 50% on computation. Examining equation 15 and 16, we find that when $m > \frac{\pi v^2(T_{life}+\Delta)^3}{6\sqrt{3}r^2T}$, COTA will save in both space and computation. As the number of wormhole detection packets increases, COTA can save more because the required space does not change and the computation overhead increases at a linear speed.

If the $sensitivity$ of COTA, $2r+vT$, is predetermined, we can choose suitable values of r and T to minimize the required storage space and the computation overhead. From equation 15 and 16, we find that both requirements are minimized at the same values of r and T. Figure 4 shows the consumed space for one intermediate node for different $sensitivity$ values. It can also be viewed as the curve of computation overhead.

We assume that $2r+vT=X$, where X is a constant. Equation 15 is minimized when

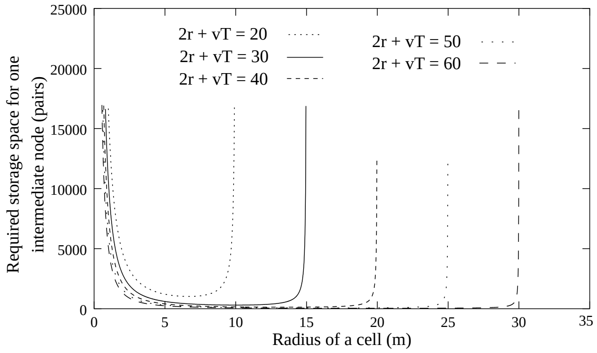


Figure 4: Storage space of COTA with different values of $sensitivity$.

$$\frac{\partial}{\partial r}(r^2(X-2r))^{-1}=0 \quad (18)$$

The optimal value of r equals to $\frac{1}{3}X$. It shows that the best choices of the cell size and the slot length increase linearly when the value of $sensitivity$ becomes larger. This allows a mobile node to pre-determine the resources that it wants to consume to defend against wormhole attacks. If we take the results back, we find that COTA has achieved its design goals: for every intermediate node, a constant storage space and the computation overhead that increases linearly to the number of the detection packets.

From Figure 4 we find that the storage and computation overhead of COTA is relatively stable within a large interval of r. Therefore, when the chosen r and T are biased from the optimal values, the overhead does not increase sharply. Since the resources available to mobile nodes vary greatly, this enables the nodes to make flexible choices on the parameters. Table 1 compares the overhead between the optimized COTA and the end-to-end mechanism. It assumes a 10-hop route and 1,800 wormhole detection packets. The values of T_{life} and v are the same as before.

Table 1: Overhead comparison between the end-to-end mechanism and COTA.

	Basic end-to-end		Optimized COTA	
sensitivity	storage(pair)	operation	storage(pair)	operation
30m	36 K	65 M	5.20 K	18.66 M
40m	36 K	65 M	2.25 K	8.12 M
50m	36 K	65 M	1.12 K	4.04 M
60m	36 K	65 M	0.70 K	2.49 M
70m	36 K	65 M	0.43 K	1.57 M

V.F. Data structure maintenance

COTA uses a complex data structure. If it cannot be efficiently maintained, the overhead caused by the insertion/deletion of records and the collection of unused space will cancel out all benefits described in previous sections. We are going to answer two questions about COTA: (1) How to organize the stored records? (2) How to efficiently recollect the space of the expired records?

A practical data structure in the destination node is shown in Figure 3-(b). All nodes that have active records are put into a link list. The destination remembers the expiration time of the latest ⟨time, position⟩ pair of every node. It is used in space collection. Every node has a link list, which stores the cells that have active records. Each cell also remembers the expiration time of the latest pair in it. In each cell, an array of records are maintained. The size of the array is $(T_{life} + \Delta)/T + 1$. Every entry has a bit to identify whether the data is available in it. To avoid memory copy as time elapses, the array is used in a cyclical way and a pointer is used to identify the current array header.

When a new ⟨time, position⟩ pair of a node arrives, the destination first locates the cell list of that node. Then for each cell of that node, it uses a linear search to locate the record that has the shortest time difference from the new timestamp.

This linear search is also used to abort those expired records and to recollect the space if all records in that cell have expired. It resets the available bits of those expired entries and moves the array header according to current time. If the latest pair in a cell has expired, COTA can take the memory back without looking into the array because all records must have expired.

This space collection method works well if new records of a node keep arriving. On the contrary, if a node is not on any active routes and no new records for it arrive, the space occupied by it cannot be efficiently recollected. To overcome this difficulty, the destination node needs to periodically traverse the node list to examine the expiration time of the latest record of every node. If the latest record has expired, all space occupied by that node can be recollected.

The maintenance of the data structure will increase the space requirement within a limited range. If all pointers are four-byte and timestamps are eight-byte, the required space will increase less than 10%. Resetting the available bits and moving the array headers can be accomplished when COTA traverses the array of each cell. They do not add much computation overhead. Referring to the results shown in Table 1, we find that COTA can still save a lot of resources.

V.G. Experiments on real devices

In this section, we present some results on the computation efficiency of COTA. They are collected through experiments on real mobile devices. We assume that symmetric cryptographic operations are used.

We define the procedure that locates the record in an array who has the shortest time difference from the new timestamp and calculates the average moving speed as a $unitcheck$. It corresponds to the operations that examine a new record in one cell. It also clears out the expired records in that cell. We design a test program that executes 100 million $unitchecks$ and run it on a Compaq iPAQ 3630 with 206 MHz CPU and 64M RAM. Different values of $sensitivity$ are examined and the size of an array is determined by the optimal value of the time slot length. The computation efficiency of COTA is shown in Table 2.

Table 2: Computation efficiency of COTA.

Sensitivity (m)	20	30	40	50	60
Optimal size of an array	19	13	10	8	7
$Unitcheck$ / sec	1.05M	1.54M	2.01M	2.54M	2.86M

For example, when the $sensitivity$ is 30 m, the device can execute 1.54 million $unitchecks$ in one second with the optimal values of cell size and time slot length. Therefore, as the destination of a 10-hop route that receives 5 wormhole detection packets per second, the node will use 0.28% of its CPU to check the records in these packets. If it uses 5% of its CPU to conduct wormhole detection, it can support 17 such routes simultaneously.

VI. Controlling Communication Overhead

The design of COTA allows a mobile node to predetermine the storage and computation resources that it wants to put on wormhole detection. In this section, we focus on the communication overhead. We study packet overhead and byte overhead separately.

VI.A. Frequency to conduct wormhole detection

Wormhole detection needs to be conducted repeatedly during the lifetime of a route. The detection frequency determines the packet overhead. It also impacts the intervals between the received $\langle$time, position$\rangle$ pairs of the intermediate nodes.

An intermediate node cannot control the longest buffering time of a detection packet in it. However, the source and the destination can determine the frequency to send such packets when the route is established. Through adjusting this frequency, the destination can control the longest interval between the received timestamps of an intermediate node. If the interval between the timestamps of the source in any two consecutive detection packets is shorter than t_{int}, and at least one of q consecutive detection packets will reach to the destination, the longest interval T_{int} between the received timestamps of the source satisfies $T_{int} \leq qt_{int}$. If the lifetime of a packet is T_{life}, we have:

Lemma 2 *: If the longest interval between the received timestamps of the source is T_{int}, the longest interval between the received timestamps of any intermediate node is $T_{int}+T_{life}+2\Delta$. Here Δ is the error of clock, and T_{life} is the packet lifetime.*

Proof 2 *: Let's consider the $\langle time, position\rangle$ pairs of an intermediate node A received by the destination. The pairs received in the same detection packet have the same sub-index. If we randomly select a $\langle time, position\rangle$ pair, there are two possible conditions:*

Condition I: the selected pair records a receiving event. We assume that it is $\langle t_{Arecv1}, P_{Arecv1}\rangle$. There must be a record, $\langle t_{Asend1}, P_{Asend1}\rangle$, that remembers the sending of this packet. And we have $t_{Asend1} \epsilon (t_{Arecv1}, t_{Arev1} + T_{life}]$. Therefore, when searching in the direction that the time increases, we must be able to find another record of A within the interval T_{life}.

The longest interval between the received timestamps of the source is T_{int}. Therefore, there must be a received record of the source $\langle t_{SourceSend}, P_{SourceSend}\rangle$ that satisfies $t_{SourceSend} \epsilon [t_{Arecv1} - T_{life} - T_{int} - \Delta, t_{Arecv1} - T_{life} - \Delta)$. The receiving and sending timestamps of this packet at node A must be smaller than t_{Arecv1}. Considering the clock error, when searching in the direction that the time decreases, we must be able to find one record of A within the interval $T_{life} + T_{int} + 2\Delta$.

Condition II: the selected pair records a sending event. Similar analysis can be conducted.

Therefore, the longest interval between the received timestamps of any intermediate node is $T_{int} + T_{life} + 2\Delta$.

□

Since the position information is received together with the timestamps, this interval also determines the density of the position information of an intermediate node. For example, if the interval is 10 seconds and T_{life} and Δ are set as before, T_{int} = 3 seconds. If the probability that a packet gets lost or corrupted on the path is P, the probability that all q detection packets are lost is P^q if the events are independent. For example, if the probability that a detection packet gets lost is 25% and q = 5, the probability that five consecutive wormhole detection packets are all lost is less than 0.1%. The source needs to send one detection packet every 0.6 second. The end-to-end mechanism will not add packet overhead if the data packet density is larger than 2 packet/second.

VI.B. Controlling byte overhead

The analysis in V.B shows that the end-to-end mechanism does not scale well to very long routes if every intermediate node attaches its $\langle$time, position$\rangle$ pairs. To avoid this problem, the destination can choose a part of the nodes to attach their records. The selected intermediate nodes are switched to guarantee that every neighbor relation is examined. For example, the destination can ask the first $\lceil \frac{l}{2} + 1 \rceil$ nodes on the route to attach their records for q detection packets, then the second $\lceil \frac{l}{2} + 1 \rceil$ nodes on the route will attach their records for q detection packets. The two halves of the nodes are overlapped to avoid the detection gap. In this way the byte overhead will decrease for more than 50%. However, the longest interval between the received timestamps of an intermediate node will become $2T_{int} + T_{life} + 2\Delta$. This may allow a wormhole to exist for a longer time before it is detected.

VII. Security Analysis

Because of the error of position, the error of clock, and the $sensitivity$ of COTA, there exist false alarms in the end-to-end mechanism. If the node adopts a conservative policy, some real neighbor relations will be considered as wormholes. The destination will abort the route and activates the routing re-initiation. This leads to the increase in routing overhead and the average delay. On the contrary, if an optimistic policy is adopted, the real neighbor relations will not be wrongly accused. However, the attackers will be able to tunnel the packets beyond the direct communication range without being detected. In section VIII we study both conditions through simulations.

A malicious node can eavesdrop the wormhole detection packets. The position information attached by the intermediate nodes is protected by the MAC codes. The malicious node cannot send fake information in other node's name to fabricate a wormhole on the route. Every intermediate node calculates the MAC code based on the whole packet. An attacker cannot take a part of the detection packet and conducts the resend attack. If it resends the whole packet, it is the same as a duplicate packet generated by the unreliable network. For multiple connections that have the same source and destination and use the same route, only one group of wormhole detection packets need to be sent. A malicious node cannot overwhelm the destination by establishing multiple connections at the same time. The adoption of COTA allows the destination to control the overhead on each route. It helps to defend against the Distributed Deny of Service (DDoS) attacks generated by a group of collusive attackers.

A wormhole tries to fabricate a non-existent route. The malicious node cannot drop the detection packets to avoid being discovered. If a violation is detected in the received information, the destination will broadcast the anomaly condition. The source receiving this message will try to establish a new route. When a wormhole is detected, the end-to-end mechanism does not try to identify the attacker. Therefore, the malicious node cannot frame a good node.

VIII. Simulation and Results

We study the practical impacts of the proposed wormhole detection mechanism through simulation. The experiments are deployed using ns2 [75]. Two properties are of special interest: false alarm ratio and communication overhead.

VIII.A. Simulation setup

We assume that pairwise keys have been deployed. AODV [19] is chosen as the routing protocol and is updated to combine with the detection mechanism. When the source initiates the routing request (RREQ) packet, it proposes its choices of the parameters such as the $sensitivity$ and the frequency to send out wormhole detection packets. When an intermediate node forwards the routing request, it will attach its identity. The source and the intermediate nodes will protect the attached information using keyed hash codes. When the destination receives the request, it knows the identities of all intermediate nodes. The destination reviews the proposed parameters. If they satisfy its requirements, the destination will accept them. Otherwise, it will propose its choices in the routing reply (RREP). Besides the parameters, the destination will also determine the starting value of the sequence number for the detection packets. When the source receives the reply, it will start to send out the data packets and the detection packets.

RREQ and RREP accomplish the route discovery and try to settle the wormhole detection parameters. There are chances that the source and the destination cannot agree with each other on the parameters. A separate round of negotiation can be conducted after the route is established and before the data packets are transferred. A method similar to the negotiation procedure in the Internet Key Exchange (IKE) protocol [76] can be adopted.

Table 3 lists the simulation parameters of ns2.

Table 3: Simulation Parameters.

Simulation duration	1000 seconds
Simulation area	1000 * 1000 m
Number of mobile nodes	50
Transmission range	250 m
Movement model	Random waypoint
Highest node speed	8 – 20 m / s
Number of connections	30
Traffic type	CBR (UDP)
Data payload	512 bytes
Packet rate	2 pkt / s
Node pause time	0 second

The node moving speed covers a range from human jogging to vehicle riding in the country field. We assume that the moving speeds of the nodes are uniformly distributed from 0 to the highest value. The sources and the destinations of the connections are randomly picked from the mobile nodes. For each examined condition, five connection scenarios and four node movement scenarios are generated and the average values of the simulation results are shown in figures.

The parameters of the detection mechanism are as follows: T_{life} = 5 seconds, T_{int} = 3 seconds, Δ = 1 second, and δ = 10m. The longest interval between two consecutive detection packets sent by the source is 0.6 second, which is a little longer than the average interval between data packets. Therefore, most of the wormhole detection information can be attached to data packets. We define the longest distance that a node can move in one second as the unit distance. It can be calculated as $v \times 1second$. Different values of $sensitivity$ (from 0.5 to 2.5 times of the unit distance) are examined.

Since the wormhole detection mechanism is a security enhancement, its false alarm ratio is of special interest. Both false positive and false negative mistakes are studied. The storage and computation overhead has been discussed extensively in section V. In this part we focus on the communication overhead in packets and bytes.

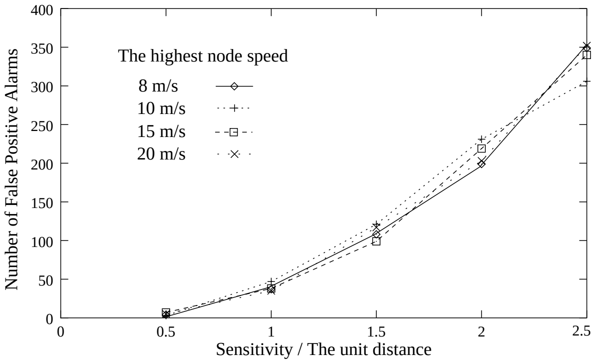


Figure 5: False positive alarms: no wormholes, sensitivity / the unit distance changes.

VIII.B. Results on false alarms

Figure 5 and 6 illustrate the false positive alarms in an environment where no wormhole exists. In Figure 5, all destination nodes adopt the proposed mechanism as in equation 5 to guarantee the detection capability. The curves show the relation between the number of false alarms and the $sensitivity$. Four values of the highest speed, from 8m/s to 20m/s, are examined. The values of r and T are determined to minimize the computation and storage overhead. The value of the $sensitivity$ increases from 0.5 to 2.5 times of the unit distance. From Figure 5, we find that as the ratio increases, the number of false alarms increases faster than a linear function. Another interesting point is that the curves for different speeds stay close to each other. In the proposed mechanism, it is the ratio between the $sensitivity$ and the unit distance, instead of the absolute value of it, that determines the number of false positive alarms.

False positive alarms lead to breaks of existent routes and an increase in communication overhead. To lower the number of such mistakes, equation 5 can be updated to the following format:

$$\frac{max(0, ||P_{new} - P_{select}|| - \delta + offset)}{||t_{new} - t_{select}|| + \Delta} \quad (19)$$

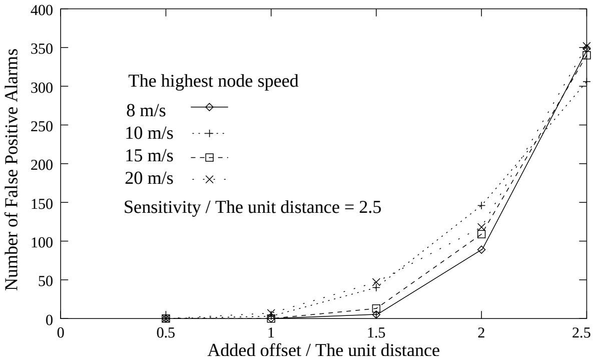


Figure 6: False positive alarms: no wormholes, added offset / the unit distance changes.

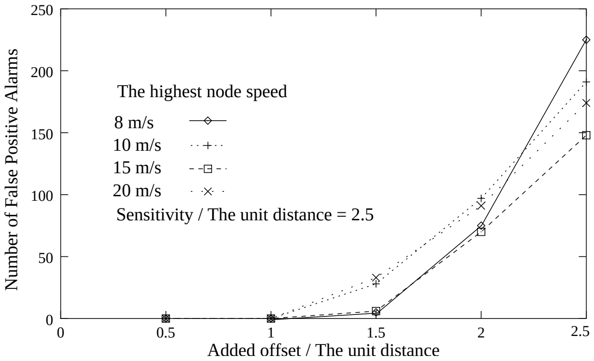


Figure 7: False positive alarms: wormholes exist, added offset / the unit distance changes.

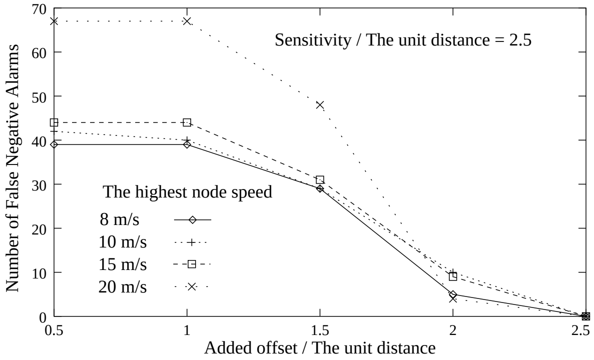


Figure 8: False negative alarms: wormholes exist, added offset / the unit distance changes.

in which the added offset is a value between 0 and the $sensitivity$ to achieve different tradeoffs between the detection capability and false alarm ratio. Figure 6 illustrates the relation between the number of false positive alarms and the added offset. The network setup is the same as in Figure 5. The ratio between the $sensitivity$ and the unit distance is 2.5. The number of false alarms increases as the added offset increases. Compared to Figure 5, fewer false positive alarms are introduced in Figure 6. The curves for different speeds are close to each other.

When the added offset is smaller than the $sensitivity$, it causes fewer false positive mistakes. However, as the analysis shows in section V, it may also miss the detection of some real wormholes. The following experiments are conducted to study this impact. Among the 50 nodes in the network, two pairs of nodes are randomly selected as "potential attackers". Each pair has a long-range, out-of-band wireless channel that can be used to construct a wormhole. The attackers' channels do not interfere with each other or the channel used by the good nodes. To construct the wormholes that cannot be detected when the added offset is 0, the longest distance that the attackers can tunnel beyond the direct communication range is the $sensitivity$. Therefore, the attackers will form a wormhole only when the distance between them falls into that interval. Other simulation parameters are the same as in Figure 6. Figure 7 and 8 illustrate the false positive and negative mistakes as the added offset changes.

In Figure 7, the curves are similar to those in Figure 6. Since some real wormholes are introduced into the system, fewer false positive mistakes are made. In Figure 8, as the added offset increases, fewer and fewer real wormholes are missed by the detection mechanism. Through adjusting the choices of the $sensitivity$ and the added offset, a mobile node can achieve a better tradeoff between the false alarm ratio and the detection capability.

VIII.C. Results on communication overhead

The following experiments explore the extra communication overhead introduced by the wormhole detection mechanism. We study both packet overhead and byte overhead. To establish the baseline for the comparison, we also examine the overhead caused only by the routing protocol when the detection mechanism is not enabled. Figure 9 and 10 illustrate the results collected from an environment in which no wormholes exist.

As stated in section VIII.A, most of the detection information is attached to the data packets. The extra packet overhead is primarily caused by the route re-discovery procedure after the detec-

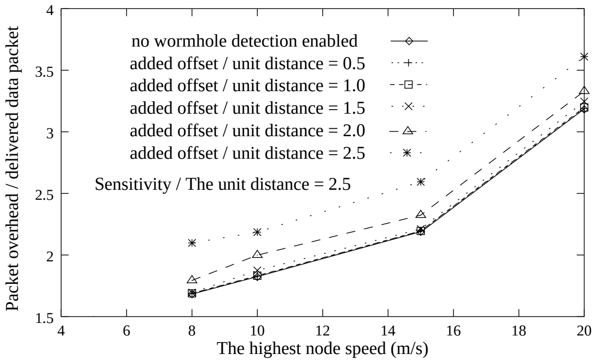


Figure 9: Packet overhead of the wormhole detection mechanism.

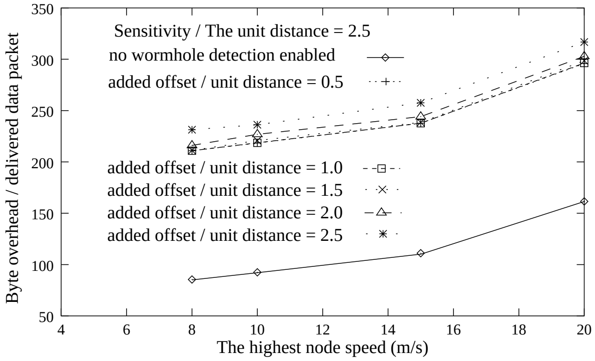


Figure 10: Byte overhead of the wormhole detection mechanism.

tion of a "wormhole"(could be a false alarm). From Figure 9 we find that as the added offset increases, more packet overhead is introduced. But the increase is small when the ratio between the added offset and the unit distance is small. Through adjusting the values of $sensitivity$ and the added offset, a node can control the increase of packet overhead.

The increase in byte overhead shown in Figure 10 is more notable. The extra bytes are primarily caused by the $\langle$time, position$\rangle$ pairs attached by the intermediate nodes, so the number of false alarms does not impact them to a large extent. The curves for different values of the ratio are close to each other. But compared to the overhead when no wormhole detection is enabled, the increase is sharp. The justifications for this increase are two folds. First, the overhead is still reasonable for the applications in which the wormholes must be prevented. The communication peers can lower the relative byte overhead by increasing the length of data packets. Second, since the routes in the simulation are not long, we do not enable the byte overhead control method stated in VI.B. The adoption of that method may help the nodes to improve the efficiency.

IX. Discussion and Future Work

The analysis shows that the $sensitivity$ represents a tradeoff between the detection capability and the required resources. Many features can impact the choice of its "optimal" value and here we only explore a few of them. First, the $sensitivity$ is restricted by the positioning accuracy because two records having a distance shorter than δ could represent the same point in the network. Second, the $sensitivity$ is impacted by the movement patterns of the nodes. The example in V.D shows that COTA misses some anomalies because the records falling into the same slot and the same cell might be ignored. Therefore, taking the movement patterns into account, we can choose a suitable $sensitivity$ value and determine r and T to control the occurrence of such events. More experiments will be conducted to explore their relations so that we can predict the optimal interval of the $sensitivity$ given a certain scenario.

As illustrated in section V.C and VI.A, the definition of packet lifetime T_{life} can restrict the number of stored records at the destination node. The other contribution of T_{life} is to guarantee the record density of the intermediate nodes. It can prevent the attacker in an open or half open wormhole from declaring that it moves back and forth between two ends of the wormhole and forwards the packets. If the proposed mechanism can be integrated with IDS, this attack can be detected by examining the anomaly in the node movement patterns. Under that condition, the restriction of T_{life} can be loosened and replaced by a more generic time window duration.

In the proposed mechanism, wormhole detection is conducted in a pro-active manner. We may expect that a re-active wormhole detection mechanism will cause less communication and computation overhead. To enable this update, the mechanism needs to be combined with the intrusion detection systems so that it will be activated when suspicious conditions are discovered.

In the proposed mechanism every mobile node acts individually to detect wormholes. The computation overhead and detection accuracy can be further improved if the nodes can share the knowledge securely and cooperate on the detection operations. Mechanisms will be designed to enable this information exchange procedure and to verify the authenticity of the information. The mechanisms can also be applied to defend against wormhole attacks conducted by a group of collusive attackers.

Geographic based wormhole detection can be viewed as an example of the emerging Location Based Services (LBS). One security concern of LBS is the disclosure of the location and movement patterns of a mobile node. For example, in COTA, the destination has the real time location information of the source and every intermediate node. This may conflict with the privacy concerns of some users. A potential extension is to use a "shadow" position to replace the real item while keeping the distance among nodes the same. The work is motivated by the research in multi-dimensional scaling [77]. It allows the nodes to take advantages of LBS while achieving privacy preservation.

X. Conclusion

The classification of wormhole attacks on ad hoc networks constructs a basis on which the detection mechanisms can be examined and compared. It divides the attacks into three groups: closed, half open, and open. The previous solutions focus on the prevention of closed wormholes. This leads to the requirement of a more generic approach.

An end-to-end mechanism is presented that can detect closed, half open and open wormholes. To reduce the storage and computation overhead, we present a new scheme, COTA, to manage the detection information. It records and compares a part of the $\langle$time, position$\rangle$ pairs. With a suitable relaxation, COTA has the same detection capability as the end-to-end mechanism. Through adjusting the cell size and time slot length, a node can control the resources that it wants to put on wormhole detection.

The schemes to control communication overhead are studied. Through adjusting the frequency to send detection packets, the longest interval between position information of any intermediate node is guaranteed. To improve the scalability of the proposed mechanism, the destination can select a part of the nodes to attach their wormhole detection information.

The practicability of the proposed mechanism is examined using both simulation and experiments on real mobile devices. The false alarm ratio can be controlled by adjusting the parameters such as the $sensitivity$ and the added offset. When the $sensitivity$ is 30m, a Compaq iPAQ 3630 with 206M Hz CPU and 64M RAM can use 0.28% of its CPU to accomplish the wormhole detection on a 10-hop route. COTA does not depend on a specific authentication mechanism. It can be combined with other approaches such as TESLA to construct new wormhole detection mechanisms.

The immediate extensions to our work consist of two parts. First, we propose to combine the mechanism with the location based routing protocols. Second, new schemes to reduce the detection packet frequency and byte overhead are under development. They will lead to a generic, efficient approach that helps the ad hoc networks to defend against the wormhole attacks.

Acknowledgement

The authors would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions.

References

[1] D. Johnson, D. Maltz, and J. Jetcheva. *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Network.* Ad Hoc Networking, Addison-Wesley, 2001.

[2] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.

[3] H. Yang, X. Meng, and S. Lu. Self-organized network-layer security in mobile ad hoc networks. In *Proceedings of ACM Workshop on Wireless Security (WiSe)*, 2002.

[4] M. Zapata and N. Asokan. Securing ad-hoc routing protocols. In *Proceedings of ACM Workshop on Wireless Security (WiSe)*, 2002.

[5] Y. Hu, D. Johnson, and Adrian Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proc. of the IEEE Workshop on Mobile Computing Systems and Applications*, 2002.

[6] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proc. of ACM MobiCom*, 2002.

[7] A. Perrig, R. Canetti, D. Song, and D. Tygar. Efficient and secure source authentication for multicast. In *Proc. of Network and Distributed System Security Symposium (NDSS)*, 2001.

[8] S. Yi, P. Naldurg, and R. Kravets. Security-aware ad hoc routing for wireless networks. In *Proc. of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2001.

[9] Y. Zhang and W. Lee. Intrusion detection in wireless Ad-Hoc networks. In *Proceedings of ACM MobiCom*, 2000.

[10] V. Bharghavan. Secure wireless LANs. In *Proc. of the ACM Conference on Computers and Communications Security*, 1994.

[11] Z. Zhou and Z. Haas. Securing Ad Hoc networks. *IEEE Networks*, 13(6):24–30, 1999.

[12] P. Albers and O. Camp. Security in Ad Hoc network: A general ID architecture enhancing trust based approaches. In *Proceedings of International Conference on Enterprise Information Systems*, 2002.

[13] O. Kachirski and R. Guha. Intrusion detection using mobile agents in wireless ad hoc networks. In *Proceedings of IEEE Workshop on Knowledge Media Networking (KMN)*, 2002.

[14] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proc. of the ACM Workshop on Wireless Security (WiSe)*, 2003.

[15] M. Corner and B. Noble. Zero-interaction authentication. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2002.

[16] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of INFOCOM*, 2003.

[17] S. Capkun, L. Buttyan, and J. Hubaux. Sector: Secure tracking of node encounters in multi-hop wireless networks. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks,* 2003.

[18] L Hu and D. Evans. Using directional antennas to prevent wormhole attacks. to appear in the Proceedings of Network and Distributed System Security Symposium (NDSS), 2004.

[19] C. Perkins and E. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[20] C. Perkins. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of SIGCOMM*, 1994.

[21] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang. A wireless hierarchical routing protocol with group mobility. In *Proc. of IEEE WCNC*, 1999.

[22] C. Chiang. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel. In *Proceedings of IEEE SICON*, 1997.

[23] E. Royer. Hierarchical routing in ad hoc mobile networks. *Wireless Communication and Mobile Computing, 2(5)*, 2002.

[24] C. Perkins, B. Woolf, and S. Alpert. *Mobile IP: Design Principles and Practices*. Printice Hall, 1998.

[25] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. A framework for IP Based Virtual Private Networks. IETF RFC 2764, 2000.

[26] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. IETF RFC 3209.

[27] S. Bellovin. Security problems in the TCP/IP protocol suite. *Computer Communications Review*, 19(2):32–48, April 1989.

[28] B. Dahill, B. Levine, E. Royer, and C. Shields. A secure routing protocol for ad hoc networks. Tech report 02-32, Dept. of Computer Science, University of Massachusetts, Amherst, 2001.

[29] P. Papadimitratos and Z. Haas. Secure routing for mobile Ad Hoc networks. In *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2002.

[30] Y. Ko, V. Shankarkumar, and N. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In *Proc. of INFOCOM*, pages 13–21, 2000.

[31] R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya. Using directional antennas for medium access control in ad hoc networks. In *Proceedings of ACM MobiCom*, 2002.

[32] N. Sastry, U. Shanker, and D. Wagner. Secure verification of location claims. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2003.

[33] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. of INFOCOM*, 2000.

[34] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personnel Communications, 4(5):42–47*, 1997.

[35] N. Jain, S. Das, and A. Nasipuri. A multichannel MAC protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of the 9th Int. Conf. on Computer Communications and Networks (IC3N)*, 2001.

[36] E. Jung and N. Vaidya. A power control MAC protocol for ad-hoc networks. In *Proceedings of ACM MOBICOM*, 2002.

[37] R. Pickholtz, D. Schilling, and L. Milstein. Theory of spread spectrum communications – a tutorial. *IEEE Trans. Comm.*, 1982.

[38] V. Gupta, S. Krishnamurthy, and M. Faloutsos. Denial of service attacks at the MAC layer in wireless ad hoc networks. In *Proceedings of Milcom*, 2002.

[39] Patrik Bjorklund, Peter Varbrand, and Di Yuan. Resource optimization of spatial TDMA in ad hoc radio networks: A column generation approach. In *Proceedings of INFOCOM*, 2003.

[40] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Trans. Program. Languages*, 4(3):382–401, 1982.

[41] P. Misra and P. Enge. *Global Positioning System, Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2001.

[42] Y. Ko and N. Vaidya. Location-aided routing(LAR) in mobile ad hoc networks. In *Proceedings of MobiCom*, 1998.

[43] S. Basagni, I. Chlamtac, and V.R. Syrotiuk. Dynamic source routing for ad hoc networks using the global positioning system. In *Proceedings of IEEE Wireless Communication and Networking Conference*, 1999.

[44] R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *IEEE Personal Communication*, pages 48–57, Feb 2001.

[45] Ya Xu, John S. Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. of ACM Mobile Computing and Networking*, pages 70–84, 2001.

[46] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, pages 246–257, 1984.

[47] T. Hodes and R. Katz. Composable ad hoc location based services for heterogeneous mobile clients. *Wireless Networks*, 5(5):411–427, 1999.

[48] J. Agre, A. Akinyemi, L. Ji, R. Masuoka, and P. Thakkar. A layered architecture for location-based services in wireless ad hoc networks. In *Proc. of IEEE Aerospace Conference*, 2002.

[49] S. Zhu, S. Xu, S. Setia, and S. Jajodia. LHAP:a lightweight hop-by-hop authentication protocol for ad-hoc networks. In *Proceedings of International Workshop on Mobile and Wireless Network*, 2003.

[50] Ashton Applewhite. What knows where you are? *IEEE Pervasive Computing*, Oct-Dec 2002.

[51] Federal communications commission(FCC) report 03-133. $http : //hraunfoss.fcc.gov /edocs\_public/attachmatch/FCC - 03 - 133A1.pdf$, 2003.

[52] D. Mills. A computer-controlled LORAN-C receiver for precision timekeeping. Technical report 92-3-1, Dept. of Electrical and Computer Engineering, University of Delaware, 1992.

[53] D. Mills. A precision radio clock for wwv transmissions. Technical report 97-8-1, Department of Electrical and Computer Engineering, University of Delaware, 1997.

[54] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the Fifth Symposium on Operating systems Design and Implementation*, 2002.

[55] M. Sichitiu and C. Veerarittiphan. Simple, accurate time synchronization for wireless sensor networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2003.

[56] Jana van Greunen and Jan Rabaey. Lightweight time synchronization for sensor networks. In *Proceedings of ACM Workshop on Wireless Sensor Networks and Applications*, 2003.

[57] B. Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins. *Global Positioning System: Theory and Practice*. Springer Wien, New York, 1997.

[58] A. Brown. High accuracy GPS and antijam protection using a p(y) code digital beam-steering receiver. In *Proceedings of ION GPS Conference*, 2001.

[59] L. Scott. Anti-spoofing and authenticated signal architectures for civil navigation systems. In *Proc. of ION GPS/GNSS Conference*, 2003.

[60] K. Deergha Rao. Anti-FM jamming in GPS receivers using a Kalman-type nonlinear adaptive filter. In *Proc of ION GPS/GNSS Conference*, 2003.

[61] P. Kruss. A survey of multicast security issues and architectures. In *Proceedings of 21st National Information Systems Security Conference*, 1998.

[62] M. Brown, D. Cheung, D. Hankerson, J. Hernandez, M. Kirkup, and A. Menezes. PGP in constrained wireless devices. In *Proceedings of the 9th USENIX Security Symposium*, 2000.

[63] N. Courtois, L. Goubin, and J. Patarin. Flash, a fast multivariate signature algorithm. In *Proceedings of Cryptographers' Track RSA Conference*, 2001.

[64] Guillaume Poupard and Jacques Stern. On the Fly signatures based on factoring. In *ACM Conference on Computer and Communications Security*, pages 37–45, 1999.

[65] C. Boyd and A. Mathuria. Key establishment protocols for secure mobile communications: A selective survey. *Lecture Notes in Computer Science*, 1998.

[66] J. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc network. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2001.

[67] B. Shrader, M. Sanchez, and T. Giles. Throughput-delay analysis of conflict-free scheduling in multihop ad-hoc networks. In *Proceedings of the 3rd Swedish Workshop on Wireless Ad-hoc Networks*, 2003.

[68] N. Bansal and Z. Liu. Capacity, delay and mobility in wireless ad-hoc networks. In *Proceedings of IEEE InfoCom*, 2003.

[69] G. Sharma and R. R. Mazumdar. Delay and capacity trade-offs for wireless ad hoc networks with random mobility. Tech Report of ECE Department, Purdue University and submit for review, 2003.

[70] Eugene Perevalov and Rick Blum. Delay limited capacity of ad hoc networks: Asymptotically optimal transmission and relaying strategy. In *Proceedings of IEEE InfoCom*, 2003.

[71] C. Perkins, E. Royer, and S. Das. Performance comparison of two on-demand routing protocols for Ad Hoc networks. In *Proceedings of INFOCOM*, 2000.

[72] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi. Performance comparison of two location based routing protocols for Ad Hoc networks. In *Proc. of the IEEE INFOCOM*, 2002.

[73] J. Kim and M. Krunz. Fluid analysis of delay performance for QoS support in wireless networks. In *Proc. of Seventh Annual International Conference on Network Protocols*, 1999.

[74] Y. Lu and B. Bhargava. Self-adjusting congestion avoidance routing protocol for ad hoc networks. Technical report, Purdue University, Department of Computer Sciences, February 2003.

[75] Network simulator – ns2. *http://www.isi.edu/nsnam/ns/*.

[76] D. Harkins and D. Carrel. The internet key exchange (IKE) protocol. IETF RFC 2409, 1998.

[77] W. Torgeson. Multidimensional scaling of similarity. *Psychometrika*, 30:379–393, 1965.