A Secure Ad-hoc Routing Approach using Localized Self-healing Communities *

Jiejun Kong†, Xiaoyan Hong*, Yunjung Yi†, Joon-Sang Park†, Jun Liu*, Mario Gerla†
†Department of Computer Science
University of California
Los Angeles, CA 90095
*Department of Computer Science
University of Alabama
Tuscaloosa, AL 35487
jkong@cs.ucla.edu, hxy@cs.ua.edu, {yjyi,jspark}@cs.ucla.edu, jliu@cs.ua.edu, gerla@cs.ucla.edu

ABSTRACT

Mobile ad hoc networks (MANETs) are vulnerable to routing attacks, especially attacks launched by non-cooperative (selfish or compromised) network members and appear to be protocol compliant. For instance, since packet loss is common in mobile wireless networks, the adversary can exploit this fact by hiding its malicious intents using compliant packet losses that appear to be caused by environmental reasons.

In this paper we study two routing attacks that use non-cooperative network members and disguised packet losses to deplete ad hoc network resources and to reduce ad hoc routing performance. These two routing attacks have not been fully addressed in previous research. We propose the design of "*self-healing community*" to counter these two attacks. Our design exploits the redundancy in deployment which is typical of most ad hoc networks; Namely, it counters non-cooperative attacks using the probabilistic presence of nearby cooperative network members.

To realize the new paradigm, we devise localized simple schemes to (re-)configure self-healing communities in spite of random node mobility. We develop a general analytic model to prove the effectiveness of our design. Then we implement our secure ad hoc routing protocols in simulation to verify the cost and overhead incurred by maintaining the communities. Our study confirms that the community-based security is a cost-effective strategy to make off-the-shelf ad hoc routing protocols secure.

Categories and Subject Descriptors

C.2.2 [**Computer-Commmunication Networks**]: Network Protocols—*Routing protocols*

General Terms

Security, Design, Measurement, Experimentation, Performance

*Part of the work is funded by ONR MINUTEMAN grant N00014-01-C-0016 and NSF NRT grant ANI-0335302.

MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.

Keywords

Self-healing ad hoc routing, Community-based security

1. INTRODUCTION

A mobile ad hoc network (MANET) is an infrastructureless mobile network formed by a collection of peer nodes using wireless radio. It can establish an instant communication structure for civilian and military applications. Unfortunately, the mobility and radio broadcast medium make MANETs very vulnerable to malicious attacks. First, outsiders (or non-network members) can monitor the open wireless medium to intercept legitimate traffic or to inject illegitimate traffic. Fortunately, cryptographic schemes can protect the network from such external attacks. Second, some previously cooperative mobile nodes may turn selfish due to various reasons (like resource deprivation); or, some mobile nodes with inadequate physical protection may be captured and compromised. Purely cryptographic countermeasures are not effective against compromised or selfish members because cryptographic trust is rendered to whoever owns the cryptographic keys, independent of node's networking behavior. The network must rely on non-cryptographic means like intrusion detection systems (IDS) to cope with these non-cooperative (compromised or selfish) members. However, as studied in [1], the non-cooperative members may try to hide their attacks under protocol-compliant behaviors. In this case, behavior discrimination is not an effective countermeasure. For example, it is very hard to discriminate between losses caused by normal network and environmental conditions and those caused by selfish and malicious behaviors as they could all appear to be protocol-compliant.

In this paper our goal is to propose a new intrusion protection mechanism, namely *community-based security*, and evaluate its effectiveness in defending ad hoc routing protocols against non-cooperative nodes. The basic idea is to mitigate the adverse (albeit seemingly protocol-compliant) actions of selfish and malicious nodes by distributing the network service in question (e.g., packet forwarding) to a community of neighboring nodes. We will call such a community the localized *"self-healing community"*. At the node level, the service provisioning is untrustworthy and is allowed to be disrupted. However, at the community level, the service provisioning becomes trustworthy—even if some of the community members are selfish or malicious, the self-healing service remains available and reliable if there is at least one "good" cooperative community member that can provide the needed service.

Clearly, there are challenges in realizing such self-healing communities in mobile networks. In particular,

- *Community creation and configuration*: A self-healing com-

munity can be created and configured anywhere and anytime in a manner compatible with state-of-art ad hoc routing protocols, but the related process should only incur reasonably low overhead.

- *Community reconfiguration*: The self-healing community must adapt to changes in the network topology and other dynamics. The impact of mobility, channel fluctuation, community member join and leave, and non-cooperative nodes must be addressed and resolved.

The contributions of this paper are in three areas:

1. *Development of a new network security concept based on "self-healing community"*. For each source-destination pair, the conventional "per node forwarding" is replaced by the "community forwarding" concept: a chain of self-healing communities along the path will forward the packet, where each community is comprised of multiple peer members, each of which can provide the needed service. This tolerates the presence of non-cooperative nodes and stops disruption attacks locally and immediately. The self-healing strategy has not been studied previously in the context of secure routing.
2. *Analytic study of the new security protection in mobile networks*. We develop a general model to verify the effectiveness of community-based security. The analytic study presents a new method to analyze secure routing schemes in various mobility models. In this paper we use the popular Random Way Point (RWP) model as the underlying mobility model, but the same analytic framework is applicable to other mobility models as well.
3. *Easy integration of community-based security with conventional routing schemes*. In this paper we use AODV [22] as the example. We implement the new design in network simulators and study the new design's impact on the underlying routing protocol.

The rest of the paper is organized as follows. In Section 2, we present security threats that have not been fully addressed by existing secure routing schemes. In Section 3, we describe the concept of "self-healing community" as well as related self-configuration and self-reconfiguration protocols. An analytic model is presented in Section 4 to verify the effectiveness of community-based secure routing. Simulation results in Section 5 confirm the efficiency of community-based secure routing. In Section 6 we compare our work with related work. Finally Section 7 concludes this paper.

2. PROBLEM: ROUTING DISRUPTION

On-demand routing in MANET In this paper we apply the "community" concept to protect on-demand routing. Though the attacks and countermeasures are also applicable to proactive routing schemes, they are not studied in this paper due to page limits. While proactive routing protocols exchange routing information even when there is no data transmission, the on demand approach pays the cost of routing overhead only when it is needed. An on-demand routing protocol is composed of two parts: *route discovery* and *route maintenance*. In route discovery, the source sends out a route request (RREQ) to the network when it needs a route to destination. A neighbor either forwards the RREQ if it does not know the route to the destination, or sends back the needed routing information to the source. Upon receiving one or more RREQs, the destination sends back at least one route reply (RREP) to the source. Contrary to RREQ flooding, an RREP message is typically forwarded by a limited set of chosen forwarders, which are called "*RREP forwarders*" (or "*RREP nodes*") in this paper. Although various on-demand routing protocols use different algorithms to process RREQ and RREP messages, the combination of RREQ and RREP processing establishes a route between the source and the destination. Due to mobility and network dynamics, an established route may be broken at any time. On-demand routing schemes use route error (RERR) notification to inform the source or the destination about the status. Then the source will initiate a new route discovery procedure to find new routes towards the destination. To overcome the overhead of a fresh restart from the source after each route outage, local recovery techniques are often applied (e.g., cached routes at neighbors are used when available).

Limitations of cryptographic protection Cryptography is an essential building block of network security. It relies on secrecy of keys, which are secret random variables maintained by each individual network member. Qualitative cryptographic algorithms ensure that any computationally bounded adversary cannot break the cryptosystem *if these secret keys are not compromised*.

However, purely cryptographic solutions cannot answer the challenge imposed by non-cooperative (either compromised or selfish) network members. The network has to rely on non-cryptographic means like intrusion detection system (IDS) to cope with these non-cooperative members. Unfortunately, as pointed out in [1], it is hard to differentiate various packet loss scenarios, for example, to identify those cases caused by natural reasons (e.g., channel interference or node mobility) and those cases caused by non-cooperative behaviors. A malicious sender can intentionally corrupt at least 1 random bit *before* packet transmission, then it is hard for a good receiver to judge whether the corruption is caused by environmental reasons or otherwise. A malicious node can also selectively drop some critical packets, so that its packet loss pattern appears to be random as expected. In on demand route discovery, a mobile node participated in RREQ forwarding may fail to forward RREP and data packets due to all kinds of reasons—random mobility, selfishness or maliciousness. There is no fail-safe method for loss discrimination between environment reasons and non-cooperative behaviors.

Not fully-addressed ad hoc routing threats Although many secure ad hoc routing protocols [12][27][20][33][2] have been proposed to secure on-demand routing schemes, the following security attacks are not fully addressed in the existing proposals.

ATTACK 1. ***(RREQ resource depletion)*** *A malicious node can attempt to deplete network resource by repetitively initiating superfluous RREQ. In this attack, an attacker sends RREQ packets, which the underlying on-demand routing protocol floods throughout the network. If the attacker is not a network member, cryptographic authentication can be added to RREQ packets to filter out those forged route discovery requests. However, if the attacker is a compromised or selfish network member, the cryptographic countermeasures are ineffective.*

An RREQ rate limit approach is proposed in [12][23] to reduce number of RREQ packets each node is allowed to initiate. *In this paper we seek to achieve this goal without compromising routing performance*. Approaching the ideal case, where a routing protocol only incurs one initial RREQ flood for each end-to-end connection, the community-based design significantly reduces the number of RREQ floods that each node initiates.

ATTACK 2. ***(RREP packet and data packet loss)*** *A malicious or selfish node may cause the loss of certain critical packets. In a route discovery procedure initiated by a good network member, an attacker can use "wormhole attack" [12] or "rushing attack" [14]*

to surpass other nodes with respect to the underlying routing metric. Then it is highly likely the attacker is selected en route. When the RREP comes back it may not forward or may forward a corrupted one. The result is equivalent to RREQ resource depletion attack, except now the RREQ initiator is not the one to blame. Also an attacker can severely degrade data delivery performance by (selectively) dropping data packets [1].

We will show how the proposed self-healing approach can counter all such attackers, including non-cooperative RREP forwarders and data forwarders. When an RREP or data packet is lost, the damaged route is locally healed within minimal latency.

3. COMMUNITY-BASED SECURE ROUTING PROTOCOLS

3.1 Network assumptions

At the routing layer, community-based security is applicable to a broad variety of routing schemes. Backward compatibility is one of our design goals. Given an underlying on-demand routing scheme (e.g., AODV [22], ARAN [27], DSR [15], Ariadne [12]), all original RREQ/RREP packet formats and packet forwarding requirements are preserved in our design. This will make it possible to seamlessly integrate the proposed community-based paradigm with most existing ad hoc routing protocols.

At the link layer, we assume that a node can always monitor ongoing transmissions even if the node itself is not the intended receiver. This typically requires the network interface stay in the "receive mode" (i.e., promiscuous reception mode) during all transmissions, which is less energy efficient than listening only to packets directed to oneself. We also assume radio transmission is omnidirectional and radio links are symmetric; that is, if a node X is in transmission range of some node Y, then Y is in transmission range of X. This can be enforced by three-way handshake (e.g. TCP style SYN-ACK-ACK) in secure neighbor detection.

At the physical layer, transmissions are vulnerable to jamming. Fortunately, mechanisms like erasure coding, spread spectrum, and directional antenna have been extensively studied as means of improving resistance to jamming. In addition, jamming attackers are more easily to be detected and counter-attacked. In this work we consider packet loss attacks, which are more "covert" than jamming attacks. We explore *physical* node redundancy in a self-organizing network as a method to stop route disruptions. We assume that in a network locality there are redundant network members with high probability. These peer members will have identical capabilities and responsibilities in community-based communication. No centralized control or hierarchical control is assumed.

3.2 Network security assumptions

We assume all packet transmissions (including control, data packets and their ACKs) are protected by data origin authentication service. *Every packet is authenticated and the packet sender's identity is unforgeable* (of course, only for uncompromised senders). This can be implemented by signing each packet by the sender's certified digital signature or using efficient symmetric key protocols like TESLA [24][12]. Therefore, the adversary cannot forge packet transmissions from uncompromised nodes, and cannot launch Sybil attack [9] by faking uncompromised nodes' identities.

We also assume that the ad hoc nodes are equipped with hardware needed by packet leashes [13] or Brands-Chaum protocols [6]. Hence by secure distance bounding, any pair of topological neighbors in ad hoc routing are indeed physical neighbors.

3.3 Design principles

Localized and immediate self-healing When a packet forwarder is a non-cooperative node that loses the packet, we use a localized, immediate and efficient self-healing scheme to elect a substitution within minimal time. The "healed" path is a close approximation of the shortest path discovered by the original on-demand route request. Extra self-healing overheads are incurred only in the localized apposite areas around the damaged links.

Limit the frequency of flooding (either network-wide or limited-scope) Control packet flooding, either network-wide or limited-scope, incurs tremendous energy expense and wireless channel contention. Malicious nodes can explore this feature to deplete needed network resource. Ideally, we seek to realize a secure routing paradigm that only requires a single initial RREQ flood per end-to-end connection, despite of unpredictable node mobility and wireless packet losses.

Explore useful information embedded in the initial RREQ floods Our secure routing schemes are not feasible if the abundant information embedded in the expensive RREQ floods is not fully used. Critical information acquired from the initial RREQ floods, such as the recent neighborhood snapshot, is useful to heal damaged on-demand routes afterwards.

End-to-end maintenance Due to the possible presence of malicious nodes, the intermediate forwarders cannot be trusted. Therefore, the two ends of a connection should pay reasonable cost to maintain the in-between self-healing communities (whose shape degenerates due to node mobility). End-to-end maintenance may include monitoring end-to-end data delivery ratio, implementing end-to-end probing, maintaining fresh routes, and finding new routes when a community en route is empty (e.g., has been completely compromised).

3.4 Community-based security (CBS)

Configuring and reconfiguring self-healing communities is the central part of our community-based scheme. For each end-to-end connection, a chain of self-healing communities along the shortest path are established to thwart route disruption. This section details how a secure community at each forwarding step is created and how the secure communities are maintained facing network dynamics and possible attacks.

3.4.1 Self-healing community overview

The concept of "self-healing community" is based on the observation that wireless packet forwarding typically relies on more than one immediate neighbor to relay packets. Figure 1 shows the simplest case that node B relays packets from node A to node C. Typically, node B is within the intersection of node A and C's radio range while A and C *cannot* hear each other. In principle, all nodes within the "moon"-shape intersection can relay packets from A to C. Nodes in such an intersection[1] form our *self-healing community*. Figure 2 depicts a chain of self-healing communities along a multi-hop path. Community-based security explores node redundancy at each forwarding step so that the conventional per-node based forwarding scheme is seamlessly converted to a new per-community based forwarding scheme. *CBS does not require unusually high node redundancy*—a self-healing community is functional as long

[1]The actual community area in our design is the further intersection of the "moon"-shape and B's one-hop transmission circle. For the clarity of presentation we spare B's one-hop circle in all depictions in this paper.

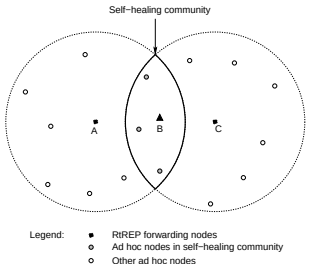

Figure 1: A self-healing community between a 2-hop source and destination pair

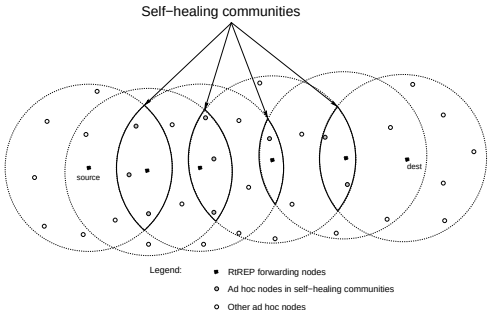

Figure 2: Packet forwarding self-healing communities along a multi-hop path

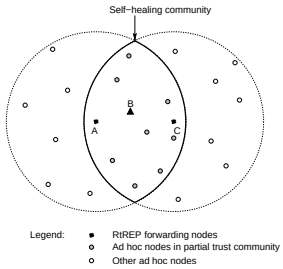

Figure 3: An inappropriate self-healing community

as there is at least one cooperative "good" node is in the community. Intuitively, a self-healing community is a "big virtual node" that replaces a single forwarding node in conventional routing schemes (Figure 4).

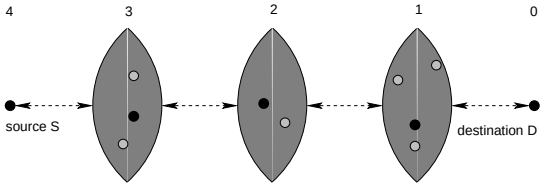

Figure 4: Self-healing communities as "big" virtual nodes

3.4.2 Self-healing route discovery

A self-healing community must be formed properly. As a comparison to Figure 1, Figure 3 shows an inappropriate community between A and C. Because A and C are one-hop neighbors, it is inefficient to introduce an extra forwarder B and pay the overhead to configure the community around B. To avoid such improper community configurations, we slightly change the underlying on-demand routing protocol's RREQ packet format, so that when B forwards its RREQ packet, it adds its immediate upstream A in the RREQ packet. The new RREQ packet format is[2]:

$$\langle RREQ, \underline{upstream\_node}, \ldots \rangle$$

where the underlined part is newly added. The distributed Algorithm A specifies each autonomous node's action during the RREQ phase. The distributed algorithms B and B_V specify how RREP forwarding can be healed by nearby network members en route.

Algorithm A: During an RREQ fbod, a node just received an authentic RREQ packet $V \rightarrow *$ for the current route discovery:
1 Insert V in my soft-state neighbor set;
2 U := the $upstream\_node$ fi eld in the RREQ packet;
3 In my soft state, record U as V's upstream;
4 IF {(I have never forwarded the RREQ) AND
5 (I have not heard U in my neighborhood during the RREQ)}
6 Record V as my RREQ upstream for the connection;
7 Process the packet according to the underlying routing protocol;
8 Locally rebroadcast the RREQ packet.

[2]Due to page limit we do not include detailed packet formats of the underlying on-demand routing protocol. Interested parties may check [23] for AODV and [16] for DSR. Note that in DSR the upstream node is already in its forwarding list, thus RREQ packet format is unchanged for community-based DSR.

Algorithm B: During RREP, a node $V'(\neq V)$ just heard the coming-back RREP packet $E \rightarrow V$ for the current route discovery:
01 Insert E in my soft-state neighbor set;
02 W := the RREQ upstream node recorded for V;
03 WHILE {(My soft state for the connection is still alive) AND
04 (Both V and W are in my soft-state neighbor set) AND
05 ((V didn't correctly forward RREP within the bounded window)
06 OR (W didn't correctly ACK within the bounded window))}
07 Wait for an autonomously decided random time;
08 IF (During the waiting period nobody has taken over)
09 Send RREP packet: $V' \rightarrow W$ (i.e., I try to take over).
10 ELSE
11 V := the node who is forwarding the RREP packet.

Algorithm B_V: During RREP, a node V just received the coming-back RREP packet $E \rightarrow V$ for the current route discovery:
1 Insert E in my soft-state neighbor set;
2 Record E as my RREP upstream for the connection.
3 W := the RREQ upstream node according to my soft-state;
4 Send RREP packet: $V \rightarrow W$.

Let's use Figure 1 to describe a simple example of self-healing route discovery. If B is a malicious forwarder, B can use rushing attack to make C believe that the best path between source A and destination C goes through B. Therefore, C will unicast back an RREP packet to B. Fortunately, even though the malicious B will drop the RREP packet or send a corrupted RREP packet, the other cooperative nodes in the community area will be able to identify the situation and try to take over as the forwarder.

- First, during RREQ phase any cooperative node B_c in the community area already remembered $V = B$ as its one-hop neighbor and $U = A$ as V's upstream node.
- Second, during RREP phase any such cooperative B_c can detect that $V = B$ fails to forward within a bounded window. For example, in 802.11, the bounded window is a heuristic estimation of B's exponential backoff window. If B_c is very near B and hears all B's receptions, then the initial backoff windows size is 32 (i.e., 0..31), or doubled after each collision. However, this is not always true and some of B's receptions cannot be heard by B_c due to hidden terminals. To count B's deferring, B_c can add an extra defer time $\tau_{defer} = \frac{l}{w}$ to the estimated window where l is the estimated packet size (e.g. l =1500bytes) and w is the link capacity (e.g. 11Mbps for 802.11b).

  Once the estimated window expires, B_c tries to take over no matter what happened to B (e.g., selfishness, maliciousness, hidden terminal, route outage due to mobility, etc.).
- Third, multiple B_c nodes may compete to forward the RREP packet. Similar to the random delay imposed in the DSR and

AODV's RREQ forwarding design, each node uses an autonomous random delay to alleviate the chance of collision. Nevertheless, this design does not completely eliminate take-over collisions. When collisions occur, the node $W = A$ determines who wins by sending back a unicast ACK, that is, the one who is ACKed by A is the one who successfully takes over.

- Finally, as depicted in Figure 5, ACKs to the unicast control packets play an important role in solving ambiguities in community configuration. At the link layer, a unicast is always ACKed in 802.11. To make our design more general, at the network layer we have implemented dedicated short ACKs for RREP packets (also for other unicast control packets, i.e., PROBE, PROBE_REP and data packets piggybacked with probing message described in Section 3.4.4. Due to page limit, see our technical report [17] for the full-fledged design of Algorithm A, B and B_V that uses network layer ACKs).

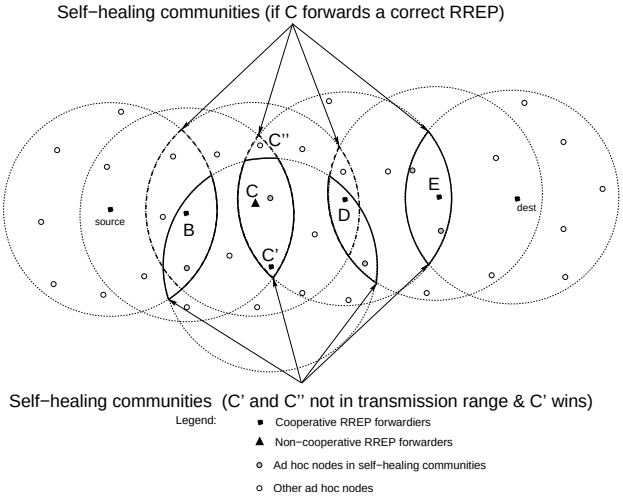

Figure 5: ACK solves ambiguity in take-over collisions

If S and D are more than two hops away, then the single-hop self-healing procedure described above is executed from D to S inductively. It is guaranteed a correct RREP comes back to S if at least one cooperative node physically presents in every community area en route.

3.4.3 Configuration of self-healing communities

A chain of self-healing communities is configured during the self-healing RREP phase. Each node must maintain a 2-bit membership flag in its on-demand soft-state for an S-D connection. Each RREP forwarder sets its membership flag to 2. A node overhearing three *consecutive* RREP ACKs sets its membership flag to 1. This is because a self-healing community member must be in the transmission range of exactly three RREP forwarders: the immediate upstream forwarder, the forwarder in the same community, and the immediate downstream forwarder. As a result, a new field is added to the existing RREP packet format:

$$\langle RREP, \underline{hop\_count}, \dots \rangle$$

where the underlined part is a counter added for the purpose of evaluating consecutiveness. The field is set to 0 by the destination D, then increased by one by each RREP forwarder. From the three consecutive hop count values, any community member can identify the index corresponding to its own community (i.e., the middle one). For example, if a mobile node overhears three RREP packets (of the same connection) with consecutive hop_count values 2, 3, and 4 in the strict order specified, then it can conclude it is in the community indexed by 3. Finally, to correctly maintain the communities immediately next to the destination D, a community member only need to hear two consecutive RREP ACKs and check whether D is involved in the packets.

3.4.4 Reconfiguration of self-healing communities

The self-healing communities lose shape due to mobility and other network dynamics. For each S-D connection, we use end-to-end probing to reconfigure self-healing communities. The probing interval T_{probe} is adapted with respect to network dynamics. The following intuitive example explains our essential design motives. Instead of using constrained flooding described in the example, the real end-to-end probing employs the same "self-healing unicast" design like the one used in Algorithms B and B_V. Therefore, the RREQ rate limit approach proposed in [12][23] is practical and causes no major routing performance degradation in CBS.

EXAMPLE 1. ***(Proactive probing by constrained flooding - An inefficient variant of community reconfiguration)*** *Suppose the two ends of a connection employ constrained RREQ floods rather than network-wide floods after RREP phase. In every constrained RREQ flood, only those nodes whose community flags for the connection are non-zero (i.e., set to 1 or 2) forward the RREQ packet as usual. This way, as the needed flags have been set previously in RREP phase (or previous probing rounds), the constrained RREQ floods* only *incur forwarding overhead in the community areas. Ideally, if T_{probe} is small enough, the constrained RREQ floods can maintain ad hoc routes just like network-wide floods, but with much less RREQ forwarding overhead per flood.*

We firstly describe how T_{probe} is selected in practice following a heuristic design. Whenever a take-over action happens, the taking-over node B_c also sends a short report to the source S

$$\langle TAKE\_OVER\_REPORT, (S, D, seq\#), B_c, B \rangle$$

where $(S, D, seq\#)$ identifies the end-to-end connection and B is the forwarding node being taken over. T_{probe} is initialized to be $\frac{R}{v}$ where R is the well-known one-hop transmission range and v is the estimated average node mobility speed. The quantity $\frac{R}{v}$ estimates the time of next link outage due to node mobility. The source decreases its T_{probe} by $\tau_{dec} = 100$ms upon receiving such a take-over report, and increases T_{probe} by $\tau_{inc} = 10$ms if no take-over report is received in the most recent second.

As frequent take-over actions indicate more network dynamics or more non-cooperative behaviors, the heuristic scheme seeks to maintain fresher self-healing communities by issuing more probing requests. Meanwhile it also seeks to decrease probing overhead when the self-healing communities en route are relatively stable. As a result, even if the number of network-wide RREQ floods for each connection is not 1 (as in the ideal case), this heuristic scheme significantly reduces the network-wide flooding frequency. This implies RREQ rate-limit proposal [12][23] is practical in community-based security.

We then describe the probing protocol details. The source S is responsible to keeping the on-demand route alive because it knows whether there is further data transmission. For every T_{probe}, the source S sends out a PROBE packet.

$$\langle PROBE, (S, D, seq\#), hop\_count \rangle.$$

Upon receiving a PROBE message, the destination D replies with a PROBE_REP packet.

$$\langle PROBE\_REP, (D, S, seq\#), hop\_count \rangle.$$

PROBE and PROBE_REP unicast forwarding follows the same self-healing procedure like Algorithms B and B_V (due to page limit, see our technical report [17] for more details). The self-healing communities en route are reconfigured by monitoring the hop_count field. That is, a node who forwards the PROBE or PROBE_REP message sets its membership flag to 2 (i.e., the forwarding member), and any node overhearing three consecutive ACKs should set its membership flag to 1 (i.e., the non-forwarding member). The hop_count field, which is increased by 1 at each stop, is similar to the same field in RREP packets to evaluate consecutiveness in packet transmission.

Since both PROBE and PROBE_REP are short messages, an optimization technique is to piggyback them on active data traffic (clearly, the connection identifier field $(S, D, seq\#)$ is not needed in piggybacked data packets). Moreover, due to wireless channel contentions and errors, it is possible that a *de facto* non-forwarding member fails to overhear at least one of the three ACKs (of RREP, PROBE, PROBE_REP or piggybacked data packets) in the current probing round. Fortunately, this unlucky node has the chance to rectify its incorrect membership flag in the next round.

3.4.5 Self-healing data delivery

Community-based data delivery is a combination of conventional node-based data forwarding plus community-based healing. At the source, the source node is unambiguously the current forwarder. At each intermediate stop, the most recent control packet forwarder (of RREP, PROBE, PROBE_REP or piggybacked data packet) is supposed to be the current data forwarder. The current forwarder plays the role of "*core*" in its self-healing community. However, if this node fails to forward data packet due to maliciousness, selfishness, or network dynamics, members in the same self-healing community will make up.

Algorithm C: During data delivery, a node just overheard a unicast data packet $E \rightarrow V$ for an $S - D$ end-to-end connection:

```
1  Insert E in my soft-state neighbor set;
2  W := my next stop (according to the underlying routing protocol);
3  WHILE {(My soft state for connection S − D is still alive) AND
4      (My community flag in the soft state is set) AND
5      (V didn't correctly forward within the bounded window)}
6        Waits for an autonomously decided random time;
7        IF (During the waiting period nobody has forwarded correctly)
8              Unicast the data packet to W.
```

Note that Algorithm C requires make-up but no take-over and no network layer ACKs for unicast data packets. Another design choice is to follow Algorithm B so that unicast data packets are not different from unicast control packets in CBS. Although this ensures per-hop reliability and thus significantly changes the network's data forwarding behavior, it may be a good choice when per-hop data packet loss ratio is huge (e.g., when either the channel error rate or the ratio of non-cooperative nodes is approaching 1).

3.5 Discussions

Soft-state design In MANET routing an adaptive soft-state strategy is used to cope with the highly dynamic network. For example, routing states are maintained using timeouts, and all unicasts and ACKs are tried for a threshold number of time (then errors will be reported to upstream if necessary). In the community-based secure routing, various new routing states are added to the underlying routing protocol's soft-state. These include the current neighbor set, community membership flag, community index (i.e., from hop_count), and probing interval T_{probe}. These records are maintained in the same way that DSR and AODV maintain their routing states.

Wormhole attack Compared to brute-force jamming, wormhole attack [13] is more "covert" in nature and harder to detect. A wormhole attacker tunnels messages received in one location in the network over a low latency link and replays them in a different location. In MANET, a typical countermeasure against wormhole attackers is to verify neighbor relation. This is due to the fact that radio propagation speed is the maximum in physics. Hence wormholes shorter than one-hop transmission range impose less threat as the original transmission (which is to be replayed by the short-range wormhole devices) features better routing metrics. (1) Physical layer countermeasures, such as RF watermarking, seek to prevent wormholes by increasing the difficulties to capture the signal patterns. The data bits are transferred in some special modulating method known only to the neighbor nodes. (2) Packet leash is a solution proposed by Hu, Perrig and Johnson for wormhole detection [13]. The leash is the information added into a packet to restrict its transmission distance. It requires either geographical location service support, or time synchronization amongst neighboring nodes. In the geographical leashes, the location information and loosely synchronized clocks together verify the neighbor relation. In the temporal leashes, the TIK protocol efficiently bounds a packet's transmission distance given tightly synchronized clocks. (3) An approach to detect wormholes without clock synchronization is proposed by Capkun et al. [31]. Every node is assumed to be equipped with a nano-second hardware that can use variants of Brands-Chaum protocol [6] to securely measure one-hop distance bound. (4) Another approach is based on the use of directional antennas. In [11], neighboring nodes examine the directions of the received signals from each other and a shared witness. Only when the directions of both pairs match, the neighbor relation is confirmed.

In this paper we assume that the network is already protected by either packet leashes or variants of Brands-Chaum protocol. This way, any pair of topological neighbors in ad hoc routing are indeed physical neighbors.

Directional and variable-power transmissions As described in [1], malicious nodes may use directional antenna and variable-power transmissions to attack ad hoc routing. In the context of community-based secure routing, the essence of such attacks is to break our design assumption that all nodes use omnidirectional radio with (nearly) identical transmission range.

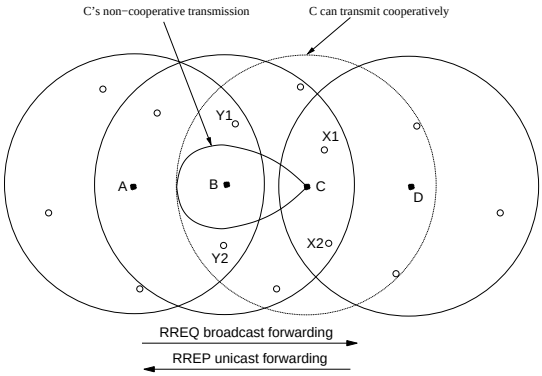

Figure 6: Attackers using directional or variable-power transmissions

Fortunately, such misbehavior can be fixed in community-based secure routing. We divide the discussion into two major cases: in CASE I the malicious nodes do *not* use directional or variable power transmission during the initial RREQ flooding procedure; and in CASE II they do.

In CASE I, the initial RREQ flood is done as described in previous sections. As depicted in Figure 6, the destination D is cooperative with its own connections so it uses standard omni-directional radio in its transmissions. Then every cooperative forwarder en route behaves the same way, until the moment the RREP packet meets a non-cooperative forwarder. Without loss of generality, let's assume the non-cooperative node is C, who uses a directional (or variable-power) transmission to unicast back the RREP packet to its RREQ upstream node B.

- Algorithm B is triggered on the *de facto* community members $X1$ and $X2$ once they receive the cooperative RREP transmission $D \rightarrow C$.
- Now that both C (the receiver in cooperative RREP transmission $D \rightarrow C$) and B (recorded RREQ upstream node of C) are in $X1, X2$'s neighbor lists (e.g., the list is maintained during the initial RREQ flood), Algorithm B (lines 03—06) ensures that $X1, X2$ will initiate the take-over process no matter what C does.

In summary, CASE I is already addressed in Algorithm B. After the initial RREQ flood, each node knows its current neighbors and also upstream node of every neighbor. This already provides two transmission circles needed in community formation. With the RREP coming back from a cooperative node, all three transmission circles needed are actualized and the associated community is formed.

In CASE II, a malicious node V uses directional antenna in RREQ forwarding. Such misbehavior can be countered by secure neighbor detection protocols, which have been studied in many secure routing literatures [14][20]. In the example depicted in Figure 6, C has to authenticate itself to its one-hop neighbors using a secure neighbor detection protocol when it roams into a new neighborhood, otherwise nobody will forward its packets (including RREQ packets). At the time when the cooperative RREP transmission ($D \rightarrow C$ in our example) is in the air, the *de facto* community members $X1$ and $X2$ sees that C is the intended RREP forwarder, but in their soft-state C's upstream node is unknown because they did not hear C's directional RREQ forwarding. In this case each of them will take over and forward RREP to its own RREQ upstream node recorded for the current route discovery. This results in a furcation and eventually multiple healed paths toward the source. The source can choose to use the best one (e.g., using the first coming-back RREP) in unicast routing. Similar to this case, RREP furcation occurs whenever RREQ upstream node is unknown (e.g., $X1$ and $X2$ may miss C's RREQ forwarding due to hidden terminals even when C is cooperative).

Collaborative adversarial RREP forwarders It is possible that there are collaborative adversarial nodes in the network. It is easy to see that Algorithms A, B, and C are unaffected if the adversarial nodes are not consecutive RREP forwarders (i.e., at most every other RREP forwarder is non-cooperative). However, if two or more *consecutive* RREP forwarders are non-cooperative, we need to devise new countermeasures. Again let's use Figure 6 in our discussion. At the time when the cooperative RREP $D \rightarrow C$ is transmitted, X ($X = X1$ or $X = X2$) will take over if either C's ACK to D or $C \rightarrow B$ forwarding cannot be heard by X:

- (X knows that B is C's RREQ upstream node) X does a cooperative RREP transmission $X \rightarrow B$. Inductively, just like what happened amongst D, C and X, now the cooperative node X replaces the cooperative node D, B replaces C, and a cooperative community member Y near B will play the role of X.
- (X does not know that B is C's RREQ upstream node) This case is similar to directional RREQ forwarding attack. X will forward RREP to its own RREQ upstream node. This results in a furcation and eventually multiple healed paths toward the source.

Therefore, the collaborative C and B cannot attack any X in C's community (due to the cooperative $D \rightarrow C$ transmission). However, they can attack a community member Y in B's community following this procedure: (1) C quickly uses at least two directional transmissions to unicast the RREP $C \rightarrow B$ to node B and all nodes in C's community area. No other nodes hear the RREP. (2) B uses one directional transmission to unicast its ACK back only to C's community area. (3) B uses one directional transmission to unicast RREP $B \rightarrow A$ only to C's community area. The RREP is indeed lost rather than delivered to A. Although this requires delicate timing and transmission techniques, it is at least a feasible attack. To defend against this attack, a community node (e.g., X in this example) should perform a secure neighbor detection right after its flag is set to 1. In the exchanged messages, the node must tell each of its neighbor that it just heard RREP $C \rightarrow B$ and B's ACK to C. Therefore, since RREP $C \rightarrow B$ is authenticated by C (according to network security assumption) and unforgeable, this effectively triggers Y's execution of Algorithm B.

Packet modification attacks CBS adds one $upstream$ field in RREQ and one hop_count field in RREP. A malicious node can attack CBS route discovery by using a random node ID in the $upstream$ field. This attack can be countered by a secure neighbor detection scheme discovering 2-hop neighborhood rather than 1-hop neighborhood. This requirement incurs more neighbor detection overhead to defend against a more insidious adversary.

The hop_count field can be protected by a simple intrusion detection system monitoring the field. Since every community member can hear three/two consecutive RREP packets (e.g., $D \rightarrow C$, $C \rightarrow B$, $B \rightarrow A$), anomaly in the hop_count field can be easily detected.

Key management In community-base secure routing, cryptographic keys are shared between two neighboring communities rather than two neighboring nodes. They are used in encryption and message authentication to protect privacy and integrity against external adversary. In particular, packets transmitted to a community member must be seen in cleartext by other community members so that they can monitor misbehaviors. Recently Deng et al. [8] proposed a cluster key design based on a global key and a distributed echo-back scheme. This key management design can be used to protect our self-healing communities—the RREQ/RREP route discovery phases are protected by the global key, then each RREP forwarder treats its community as a cluster in [8]. Currently we are investigating more resilient methods to avoid the use of global key and more efficient methods to reduce the incurred key management overhead.

Energy efficiency Community-based security requires each ad hoc node to constantly monitor its neighborhood (including configured communities if there is any). This implies the network interface must be ready for packet reception all the time. Real measurements [10] have shown that various network interfaces consume much less energy in the "receive mode" than in the "transmit mode", and on some popular interface cards the energy consumed in the "receive mode" is comparable to the "idle mode" (though

lack of standard definition, "idle mode" typically refers to an energy efficient quasi-active mode so that the device's energy consumption is minimal and the device can be made active in minimal latency). Various schemes [30] have also been proposed to significantly decrease energy expense for the "receive mode" without affecting wireless packet reception guarantee.

4. EFFECTIVENESS STUDY

In this section we use an analytic model to verify the effectiveness of community-based secure routing. We define a quantity "*effectiveness gain*" EG to quantify the advantage of CBS.

4.1 Underlying spatial model

We divide the network area into a large amount of small (virtual) grids, so that the grid size is even smaller than the physical size of the smallest network member. This way, each grid is either empty, or is occupied by a single node. Also because the network area is much larger than the sum of all mobile nodes' physical size, the probability that a grid is occupied by a mobile node is very small.

Now a bionomial distribution $B(n,p)$ defines the probabilistic distribution of how these grids are occupied by each mobile ad hoc node. Here n, the total number of grids, is very large; and p, the probability that a grid is occupied by the single node, is very small. When n is large and p is small, it is well-known that a bionomial distribution $B(n,p)$ approaches Poisson distribution with parameter $\lambda = n{\cdot}p$. Hence this bionomial spatial distribution is translated into a *spatial Poisson point process* [7] to model the random presence of the network nodes. In other words, suppose that L events occur in area $\mathcal{A}$ (here an event is an ad hoc node's physical presence). If the node density $\rho_{_L} = \frac{|L|}{\mathcal{A}}$ (where $|\cdot|$ denotes the cardnality of a set, and $\rho_{_L} = |L| \cdot \rho_1$ if nodes roam independently) is equivalent to a random sampling of $\mathcal{A}$ with rate $\rho_{_L}$. Let x denote the random variable of number of related network member nodes. Then the probability that there are exactly k nodes in a specific area $\mathcal{A}$ is

$$Pr[x=k] \;=\; \frac{(\rho_{_L}\mathcal{A})^k}{k!}{\cdot}e^{-\rho_{_L}\mathcal{A}} \qquad (1)$$

The choice of ρ_1 depends on the underlying mobility model. Some stochastic mobility models which directly choose a destination direction rather than a destination point and allow a bound back or wrap-around behavior at the border of the system area are able to achieve a uniform spatial distribution [3]. However, the others are not. Let's use random way point (RWP) model, the most popular one currently used in simulation studies, as the underlying mobility model. The probability of mobile node's spatial distribution in RWP model has been extensively analyzed in various literatures [4] [5] [25]. For a network deployed in a bounded system area, let the random variable $\Omega = (X,Y)$ denote the Cartesian location of a mobile node in the network area at an arbitrary time instant t. The spatial distribution of a node is expressed in terms of the probability density function

$$\rho_1 \;=\; f_{XY}(x,y)$$
$$= \lim_{\delta\to 0} \frac{Pr[(x-\frac{\delta}{2} < X \leq x+\frac{\delta}{2}) \wedge (y-\frac{\delta}{2} < Y \leq y+\frac{\delta}{2})]}{\delta^2}$$

The probability that a given node is located in a subarea $\mathcal{A}'$ of the system area $\mathcal{A}$ can be computed by integrating ρ_1 over this subarea

$$Pr[\text{node in } \mathcal{A}'] \;=\; Pr[(X,Y)\in\mathcal{A}'] \;=\; \iint_{\mathcal{A}'} f_{XY}(x,y)d\mathcal{A}$$

where $f_{XY}(x,y)$ can be computed given geometric properties of the network. For example, as suggested in [5], we can use the analytical expression

$$\rho_1 \;=\; f_{XY}(x,y) \;\approx\; \frac{36}{a^6}\left(x^2-\frac{a^2}{4}\right)\left(y^2-\frac{a^2}{4}\right)$$

for a square network area of size $a{\times}a$ defined by $-a/2 \leq x \leq a/2$ and $-a/2 \leq y \leq a/2$.

Therefore, the node density $\rho_{_L}$ is a *location dependent* variable. In particular for the random waypoint model, $\rho_{_L}$ is higher at the central area and lower at the boundary area [4][5]. For location dependent distributions, the probability of (1) that there are exactly k nodes in a subarea $\mathcal{A}'$ of the system area $\mathcal{A}$ (with respect to a tiny unit area) is changed to

$$Pr[x=k] \;=\; \iint_{\mathcal{A}'}\left(\frac{\rho_{_L}^k}{k!}{\cdot}e^{-\rho_{_L}}\right)\,d\mathcal{A}$$

where $\rho_{_L}$ is the node's spatial distribution function with respect to the underlying mobility model.

4.2 Geometric properties of self-healing community

Here we assume ideal circular radio coverage for the ease of analysis. Also in Section 3 we enforce the policy that two 2-hop forwarders cannot overhear each other, that is, the minimal distance between them is larger than 1-hop transmission range. Figure 7 depicts the maximum case when the distance between two 2-hop forwarders is $(1+\epsilon){\cdot}R$ (where ϵ is a negligible quantity). On the other hand, Figure 8 depicts the minimum case when the distance between two 2-hop forwarders is $(2-\epsilon){\cdot}R$.

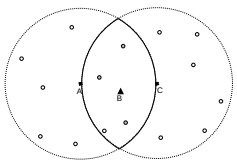

Figure 7: Self-healing community: maximum case

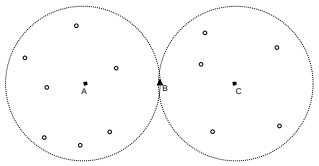

Figure 8: Self-healing community: minimum case

In the minimum case the area is approximately 0. And in the maximum case the area occupied by the self-healing community is approaching

$$\mathcal{A}_{heal}^{max} \;=\; \left(\frac{2\pi}{3}-\sqrt{3}\right)R^2.$$

Suppose the RREQ procedure is a truly random process where the distance between 2-hop forwarders randomly distributes over the range between $(1+\epsilon){\cdot}R$ and $(2-\epsilon){\cdot}R$. We expect the size of a self-healing community is the average case:

$$E(\mathcal{A}_{heal}) \;\approx\; \left(\frac{\pi}{3}-\frac{\sqrt{3}}{2}\right)R^2.$$

Therefore, the probability that the expected self-healing community area $E(\mathcal{A}_{heal})$ has exactly k nodes is

$$Pr[x=k] \;=\; \iint_{E(\mathcal{A}_{heal})}\left(\frac{\rho_{_L}^k}{k!}{\cdot}e^{-\rho_{_L}}\right)\,d\mathcal{A}.$$

4.3 Spatial model with adversarial presence

We adopt a probabilistic adversarial model. Amongst L authenticated network members, there are $\theta{\cdot}L$ non-cooperative nodes and $(1-\theta){\cdot}L$ cooperative nodes. If the network is protected by cryptographic authentication schemes (e.g., by KDC in Ariadne [12] or

by certification in ARAN [27]), non-network member nodes cannot join the network to be forwarders. Here θ becomes the *non-cooperative ratio* that quantifies the number of compromised or selfish network members.

Let y denote the random variable of number of cooperative network members in the expected self-healing community area. The probability that the expected area has k cooperative nodes is

$$Pr[y=k] \;=\; \iint_{E(\mathcal{A}_{heal})} \frac{\left((1-\theta)\cdot\rho_L\right)^k}{k!}\cdot e^{-(1-\theta)\cdot\rho_L}\; d\mathcal{A}$$

In a community-based secure routing scheme, the per-hop route discovery success ratio is

$$\begin{aligned} P_{community} &= Pr[y\geq 1] \;=\; 1 - Pr[y=0] \\ &= \iint_{E(\mathcal{A}_{heal})} \left(1 - e^{-(1-\theta)\rho_L}\right)\; d\mathcal{A}. \end{aligned} \quad (2)$$

4.4 Effectiveness gain

In regular on-demand routing, if (at least) one non-cooperative "bad" node presents in the community area and launches rushing attack [14] in route discovery, then with a high probability p_{rush} the bad node will be selected as an RREP forwarder. For simplicity of analysis, let's assume $p_{rush} = 1$ and the bad forwarder drops its RREP packet. Let z denote the random variable of number of non-cooperative network members in the expected self-healing community area. The probability that the expected area has k non-cooperative nodes is

$$Pr[z=k] \;=\; \iint_{E(\mathcal{A}_{heal})} \frac{\left(\theta\cdot\rho_L\right)^k}{k!}\cdot e^{-\theta\cdot\rho_L}\; d\mathcal{A}$$

In a regular on-demand routing scheme, the per-hop route discovery success ratio is computed from knowing all nodes in the forwarding area are cooperative. The ratio is only

$$\begin{aligned} P_{regular} &= \sum_{k=1}^{\infty} Pr[y=k, z=0] \\ &= Pr[y\geq 1]\cdot Pr[z=0] \\ &= \iint_{E(\mathcal{A}_{heal})} \left((1 - e^{-(1-\theta)\rho_L})\cdot e^{-\theta\rho_L}\right)\; d\mathcal{A}. \end{aligned} \quad (3)$$

The per-hop routing *effectiveness gain* for the community-based secure routing is defined as the ratio between the two routing probabilities: the self-healing one with community-based security, and the regular one without the protection.

$$EG \;=\; \frac{P_{community}}{P_{regular}} \;=\; \frac{1}{e^{-\theta\rho_L}}.$$

EG is a simple metric that does not depend on the size of self-healing community and the number of hops. Figure 9 illustrates EG for a very small non-cooperative ratio in a scalable network. The effectiveness gain is huge. It is even more tremendous when either network scale or non-cooperative ratio increases.

5. SIMULATION STUDY

We implement community-based security routing scheme on top of AODV (denoted as CBS-AODV) in QualNet [28], a detailed packet-level network simulator. Our evaluation will investigate: (1) the impact of internal adversaries on the performance and the resilience of community forwarding against rushing attack and black hole attack. As a comparison, we also implemented part of *Rushing Attack Prevention (RAP)* scheme [14] (denoted as RAP-AODV), namely, a node buffers a few received RREQs belonging to the same flooding and replays by randomly picking up one RREQ from the buffered ones; (2) the impact of node mobility on community-forwarding scheme under these attacks. We also implemented *constrained flooding* (Section 3.4.4, denoted as "CBS-AODV,cons_flood") for comparison.

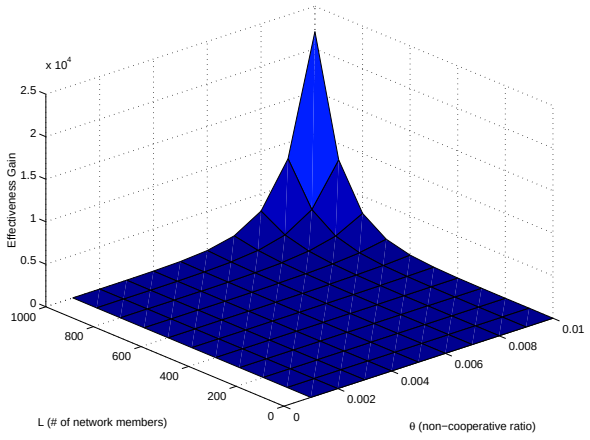

Figure 9: Effectiveness gain EG (with normalized ρ_1)

In our simulation scenario, 150 nodes are randomly placed within a field of size 2400m×600m. The nodes move according to RWP model [15]. Simulations use CBR (Constant Bit Rate) application where each session lasts for 2 minutes and generates data packets of 512 bytes at a rate of 4 packets per second. The source-destination pairs are chosen randomly from all the nodes. During total 15 minutes simulation time, five CBR sessions are constantly maintained. We use IEEE 802.11b DCF at MAC layer and two-ray ground propagation model at physical layer. Network devices have link bandwidth at 2Mbits/sec and 250 meter power range. The results are averaged over several simulation runs conducted with various random seeds.

The following metrics are used for measurement. *(i) packet delivery ratio*: the ratio between the number of data packets received and those originated by the sources. *(ii) routing overhead*: total bytes of routing control packets. For CBS-AODV, new types of control packets are all calculated. *(iii) average end-to-end packet latency*: the average time from when the source generates the data packet to when the destination receives it. Community makeup back off delay is included for CBS-AODV. *(iv) average route acquisition latency*: the average latency for discovering a route. *(v) number of triggered route request flooding*: the number of route search flooding initiated by the sources. This metric is used to show that using the community forwarding and self-healing community maintenance, recourse depletion attack through excessive control packet flooding can be limited,

5.1 Impact of non-cooperative ratio θ

To investigate the impact of non-cooperative members using a combined strategy of rushing attack and black hole attack, we use static network scenarios to emphasize only on the impact of non-cooperative ratio θ. We vary the ratio (θ) from 0 to 10% (e.g., if θ = 10, 15 nodes (0.1 * 150 nodes) are non-cooperative). With the increase of the ratio, more non-cooperative members will place themselves on the routing paths through rushing attacks and hence to perform black hole attacks on data packets and on RREP pack-

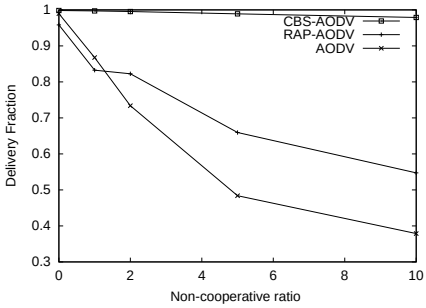

Figure 10: Data Packet Delivery Ratio

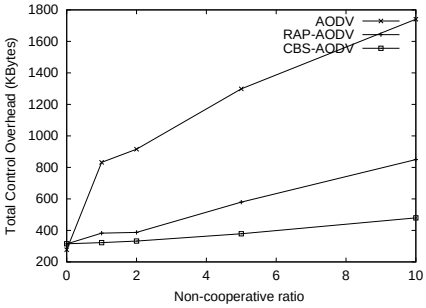

Figure 11: Control Overhead (Kbytes)

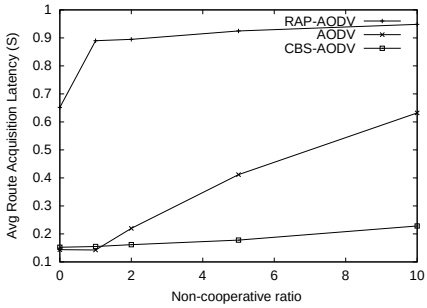

Figure 12: Average Route Acquisition Latency (S)

ets. For RAP-AODV, we use the same parameters as used by the authors [14].

Figure 10 shows that the delivery ratios are drastically impaired for AODV and RAP-AODV when the number of attackers increases, while it remains high for CBS-AODV. This drastic change verifies the analytic predictions (Figure 9). For RAP-AODV, the delivery ratio is higher than regular AODV, but cannot be restored to CBS-AODV's level since the chance of rushing attack cannot be completely eliminated through randomization in RREQ forwarding. In addition, Figure 12 verifies that RAP-AODV's route acquisition delay is much higher than CBS-AODV and AODV due to the added latency in RREQ forwarding. Figure 11 verifies that both AODV and RAP-AODV generate higher routing overhead when there are more non-cooperative nodes in the network.

Figure 13 and Figure 12 collectively illustrate the delay performance. The impacts are two folds. First, with the community security support, initial route acquisition latency is small for CBS-AODV since dropped RREP packets will be backed up by community nodes. But for AODV, sources have to re-send RREQ packets when RREPs are not received or not received in time. For RAP-AODV, buffering RREQ packets at each stop greatly slows down the time of propagating RREQ, hence results in a high route acquisition latency. Second, when packet losses occur, community nodes make up the lost transmissions after a short time period. This mechanism produces very high packet delivery ratio at the cost of slightly prolonged end-to-end delay. The end-to-end delay in CBS-AODV stays at a relatively stable level. But for RAP-AODV, Figure 13 shows longer end-to-end delay and a decreasing trend. This is due to longer route acquisition latency and degradation in packet delivery, respectively. As packet delivery degrades, those successfully delivered packets are the ones that delivered to closer destinations on average, so the end-to-end latency decreases. This trend is also observed for AODV. Our results on average path lengths validate this reasoning (not shown here due to page limit) by showing that AODV reduces path length from 4.36 to 3.61 for this simulation configuration while CBS-AODV remains within the range between 4.34 to 4.53 on average.

Figure 14 shows the portion of forwarding that is performed by original RREP nodes and the portion of forwarding that is performed by community nodes. It is clear that with increasing attacker ratio, the RREP forwarders fail more in forwarding, while the community nodes forward more packets to take over. Figure 15 demonstrates that the rate of needed network-wide RREQ floods stays at a relatively stable level in CBS-AODV, but not in AODV and RAP-AODV. This verifies that imposing RREQ rate limit is a practical design for CBS-AODV, but not for AODV or RAP-AODV.

5.2 Impact from mobility

Our second set of experiments examine the impact of node mobility. The attacker ratio is set at 1% in all mobile scenarios. We vary the node mobility from stationary to a speed of 10 m/s (same for minimum and maximum speeds in RWP model [32]). The pause time is set to 30s. The proactive probing rate for CBS-AODV and "CBS-AODV, cons_flood" is identical. This configuration seeks to show that probing by constrained flooding can cope with mobility without incurring network-wide floods. Only in some extreme cases, a source has to re-initiate a network-wide RREQ flood to rebuild the route.

In Figure 16, CBS-AODV's delivery ratio slightly degrades when mobility increases. In the extreme case, all old community members roam out of range during a probing interval. Then the current route completely breaks. Intuitively, delivery ratio degrades because the probability of the occurrence of the extreme case increases as node mobility increases. The inefficient variant "CBS-AODV, cons_flood" exhibits similar delivery ratio performance like CBS-AODV. It is slightly worse than CBS-AODV because 802.11 broadcasts and constrained flooding incur more channel contention and packet loss in the community areas (of the current probing round). Moreover, Figure 17 shows that "CBS-AODV,cons_flood" is inefficient in terms of routing overhead. Figure 17 also verifies the intuition that the overall control overhead increases, as CBS-AODV adapts its probing interval to a shorter period when mobility increases. Nevertheless, CBS-AODV incurs less control overhead than AODV with respect to the increasing mobility.

Figure 18 studies the impact of RREQ rate control to resist resource depletion attack. The simulations run at mobility of 10m/s. Each source is not allowed to send more than one RREQ flood within the minimum RREQ intervals shown on the x-axis. The figure shows the combined impact of mobility and non-cooperative ratio (in this case it is the compromised ratio because selfish nodes would not waste their own energy to initiate RREQs). The results show that CBS-AODV copes with the reduced RREQ flood rate better than AODV under both high mobility and various attacker ratios. For AODV under 5% attackers ratio and high mobility, the curve is mostly flat. This is because the delivered packets are mostly close to the source nodes, then the RREQ rate limit control does not significantly change the protocol performance.

6. COMPARISON TO RELATED WORK

Recently many solutions are proposed for ad hoc routing schemes to mitigate the problem of routing disruption. To resist attacks from non-network members, either public key based digital signatures [27] or symmetric key based protocols (e.g., TESLA [24])[12]

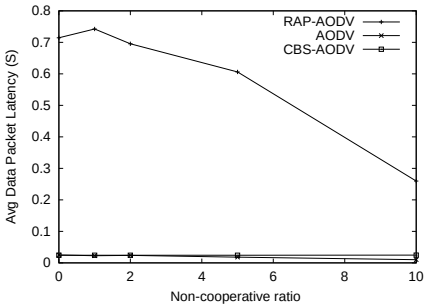

Figure 13: Average End-to-end Latency

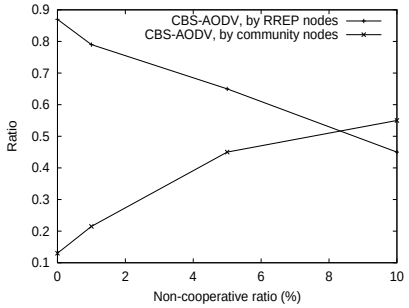

Figure 14: RREP forwarding ratio

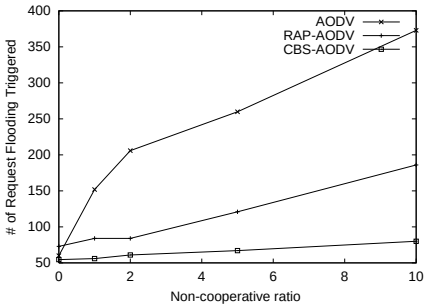

Figure 15: # of needed network-wide RREQ floods (if not limited by rate control)

are used to differentiate legitimate members from external adversaries. Afterwards network members refuse to accept or forward any unauthenticated packet. However, such cryptographic countermeasures cannot fully answer the routing disruption challenge. As demonstrated in "wormhole attack" [13], "rushing attack" [14] and the resource depletion attacks studied in this paper, malicious nodes can easily disrupt ad hoc routing without breaking the cryptosystems in use. A wormhole attacker tunnels messages received in one location in the network over a low latency link and replays them in a different location. The attacking nodes can selectively let routing messages get through. Then the "wormhole" link has higher probability to be chosen as part of multi-hop routes due to its excellent packet delivery capability. Once the attacking nodes know they are en route, they can launch various attack against data delivery. In rushing attack, malicious nodes increase the chance to be forwarder by rushing RREQ forwarding. Then they can launch similar attacks used by wormhole attackers.

Network-based countermeasures must be devised to answer the new challenges. To defeat rushing attackers, Hu et al. [14] proposed to form local communities by a secure neighborhood discovery protocol. In a local community, RREQ forwarding is delayed and randomized so that an RREQ rushing attacker cannot dominate other members during the RREQ phase. Route disruption is mitigated because the chance of selecting a rush attacker on a path equals the chance of selecting a good member. Our self-healing design adopts a different approach. We implement a faster RREQ phase, then in the RREP phase we explore the presence of good network members to heal a damaged route on the fly. Such self-healing feature has not been explored in previous secure routing research to counter malicious nodes. To resist wormhole attackers, our design relies on countermeasures like packet leashes [13] and secure distance bounding [31][6].

Multi-path routing [26][18] and route fix using local recovery query [29] are alternative choices of community-based routing. In multi-path routing, more paths parallel (albeit some of them are near) to the optimal path are maintained, a damaged path is replaced by another path rather than fixed locally. It incurs extra overheads to maintain paths other than the optimal path and to deliver data on those non-optimal paths. In local recovery query, the forwarders need to *cooperatively* query a larger recovery area to fix a damaged link. This cooperative assumption does not apply to non-cooperative members studied in this work. In general, our approach is very different from existing approaches—we build localized self-healing communities *on* the optimal path to counter non-cooperative nodes. In the context of secure routing, Papadimitratos and Haas [21] studied a multi-path approach to mitigate route disruption attacks. By encoding data packets into erasure codes, the destination is able to recover the source's data upon receiving a threshold subset of encoding symbols that have been delivered along the multiple paths. Awerbuch et al. [2] proposed a multi-path evaluation and probing scheme to detect malicious packet forwarders. If a malicious forwarder cannot differentiate the data packets without probing piggybacks from those with, then the source can pinpoint the range of failure on a path. Nevertheless, none of the related work adopts our localized approach to secure the optimal path discovered by the underlying ad hoc routing protocol.

Local monitoring also improves ad hoc routing security. In secure neighbor detection schemes [14][20], mobile nodes constantly gather knowledge about its current neighborhood. Each node must prove its network membership as well as its local presence. Control and data packets are only forwarded for verified neighbors. As we mentioned in Section 3.5, these secure neighborhood detection schemes help community-based routing to subdue attackers with directional transmission capability. In neighbor monitoring, any wireless node can use "watchdog" [19] or passive acknowledgement [16] to detect its neighbors' forwarding misbehaviors. Inside a self-healing community, members monitor each other's behavior. We devised autonomous algorithms to guide each member's actions and reactions upon detecting non-cooperative events. Routing integrity is achieved if at least one cooperative member is monitoring when a routing misbehavior occurs.

7. CONCLUSIONS

In this paper we have studied how non-cooperative network members can threat the secure routing protocols by various means. In particular, they can deplete network resource and reduce the routing performance to minimum. These security threats have not been fully addressed in previous research. We propose the concept of "*self-healing community*" and show how to use this concept to defend against the new security threats. Our design explores redundancy in deployment, an inherent feature of ad hoc networking, to let nearby cooperative network members counter the attacks launched by the non-cooperative nodes.

We rely on localized simple schemes and end-to-end probing to configure and reconfigure "self-healing communities". Ad hoc routes are healed locally within minimal latency. In the ideal case, only a single initial RREQ flood is needed for each end-to-end connection. In practice, even though this ideal case is impractical, the RREQ flooding frequency is minimized. By an analytic model we show the effectiveness gain of community-based secure routing is tremendous. Then we design and simulate secure ad hoc routing protocols to verify the cost and overhead incurred by reconfigurable self-healing communities. Our study verifies that it is effective and

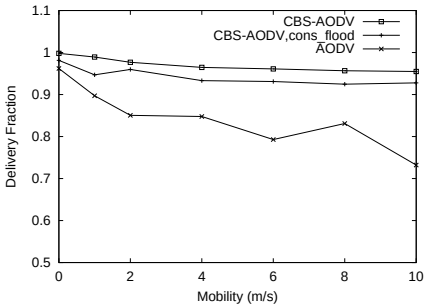

Figure 16: Packet Delivery Ratio

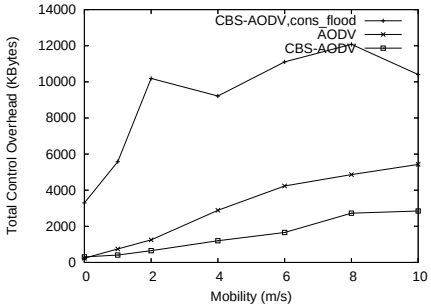

Figure 17: Control Overhead (Kbytes)

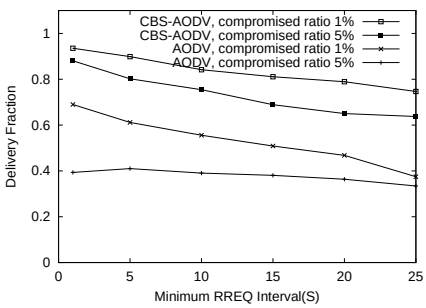

Figure 18: RREQ Flood Rate Limit Control

efficient to use the new paradigm to secure common ad hoc routing protocols.

8. REFERENCES

[1] I. Aad, J.-P. Hubaux, and E. W. Knightly. Denial of Service Resilience in Ad Hoc Networks. In *ACM MOBICOM*, pages 202–215, 2004.

[2] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. In *First ACM Workshop on Wireless Security (WiSe)*, pages 21–30, 2002.

[3] C. Bettstetter. Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects. *ACM Mobile Computing and Communication Review*, 5(3):55–67, 2001.

[4] C. Bettstetter, H. Hartenstein, and X. Perez-Costa. Stochastic Properties of the Random Waypoint Mobility Model. *ACM/Kluwer Wireless Networks, Special Issue on Modeling and Analysis of Mobile Networks*, 10(5):555–567, 2004.

[5] C. Bettstetter and C. Wagner. The Spatial Node Distribution of the Random Waypoint Mobility Model. In *German Workshop on Mobile Ad Hoc Networks (WMAN)*, pages 41–58, 2002.

[6] S. Brands and D. Chaum. Distance-Bounding Protocols (Extended Abstract). In T. Helleseth, editor, *EUROCRYPT'93, Lecture Notes in Computer Science 765*, pages 344–359, 1993.

[7] N. Cressie. *Statistics for Spatial Data*. John Wiley and Sons, 1993.

[8] J. Deng, R. Han, and S. Mishra. Intrusion Tolerance and Anti-Traffi c Analysis Strategies for Wireless Sensor Networks. In *IEEE International Conference on Dependable Systems and Networks (DSN)*, pages 594–603, 2004.

[9] J. Douceur. The Sybil Attack. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, 2002.

[10] L. M. Feeney and M. Nilsson. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. In *IEEE INFOCOM*, 2001.

[11] L. Hu and D. Evans. Using Directional Antennas to Prevent Wormhole Attacks. In *Network and Distributed System Security Symposium (NDSS)*, 2004.

[12] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks. In *ACM MOBICOM*, pages 12–23, 2002.

[13] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks. In *IEEE INFOCOM*, 2003.

[14] Y.-C. Hu, A. Perrig, and D. B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. In *ACM WiSe'03 in conjunction with MOBICOM'03*, pages 30–40, 2003.

[15] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.

[16] D. B. Johnson and D. A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), April 2003.

[17] J. Kong, X. Hong, J.-S. Park, Y. Yi, and M. Gerla. L'Hospital: Self-healing Secure Routing for Mobile Ad-hoc Networks. Technical Report CSD-TR040055, Dept. of Computer Science, UCLA, January 2005.

[18] M. K. Marina and S. R. Das. Ad Hoc On-demand Multipath Distance Vector Routing. In *IEEE ICNP*, pages 14–23, 2001.

[19] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *ACM MOBICOM*, 2000.

[20] P. Papadimitratos and Z. J. Haas. Secure Routing for Mobile Ad Hoc Networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 2002.

[21] P. Papadimitratos and Z. J. Haas. Secure Data Transmission in Mobile Ad Hoc Networks. In *Second ACM Workshop on Wireless Security (WiSe)*, pages 41–50, 2003.

[22] C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *IEEE WMCSA'99*, pages 90–100, 1999.

[23] C. E. Perkins, E. M. Royer, and S. Das. Ad-hoc On Demand Distance Vector (AODV) Routing. `http://www.ietf.org/rfc/rfc3561.txt`, July 2003.

[24] A. Perrig, R. Canetti, D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5(2):2–13, 2002.

[25] G. Resta and P. Santi. An Analysis of the Node Spatial Distribution of the Random Waypoint Model for Ad Hoc Networks. In *ACM Workshop on Principles of Mobile Computing (POMC)*, pages 44–50, 2002.

[26] P. Sambasivam, A. Murthy, and E. M. Belding-Royer. Dynamically Adaptive Multipath Routing based on AODV. In *Med-Hoc-Net*, 2004.

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

[28] Scalable Network Technologies (SNT). QualNet. `http://www.qualnet.com/`.

[29] C. Sengul and R. Kravets. Bypass Routing: An On-Demand Local Recovery Protocol for Ad Hoc Networks. In *Med-Hoc-Net*, 2004.

[30] E. Shih, P. Bahl, and M. Sinclair. Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In *ACM MOBICOM*, pages 160–171, 2002.

[31] S. Čapkun, L. Buttyán, and J.-P. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 21–32, 2003.

[32] J. Yoon, M. Liu, and B. Noble. Sound Mobility Models. In *ACM MOBICOM*, pages 205–216, 2003.

[33] M. G. Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. In *First ACM Workshop on Wireless Security (WiSe)*, pages 1–10, 2002.