$\mathcal{GVG}-\mathcal{RP}$: A Net-centric Negligibility-based Security Model for Self-organizing Networks

Dr. Jiejun Kong
Department of Computer Science
University of California, Los Angeles, CA 90095
jkong@cs.ucla.edu

Abstract

We present a rigorous approach to building a secure self-organizing mobile ad hoc network (MANET). In a highly dynamic environment like MANET, it is impossible to ensure absolute security to protect everything. We have to speak of the ”infeasibility” of breaking the security system rather than the ”impossibility” of breaking the same system. More formally, security is defined on the concept of ”negligible”, which is asymptotically sub-polynomial with respect to a pre-defined system parameter n. Intuitively, the parameter n in modern cryptography is the key length. The crypto-system's security is broken if the adversary's capability is of exponentials of n, and the efficiency of all related algorithms is measured in polynomials of n.

We adopt the same formal security notion in ad hoc network security research. In network security, the network scale (i.e., number of network members) N replaces the role of key length n in cryptography. If a security scheme can be devised to ensure that the probability of security failure is negligible, then the larger the network scale is or the more complex the network system is, the more secure the network is. In other words, given a negligibility-based protection against a specific security attack, larger or more complex systems are favored over smaller or simpler systems. Intuitively, this is consistent with the evolution theory where more complex entities probabilistically emerge from and likely survive longer than their less complex counterparts.

In this paper, we use “rushing attack” as the exemplary security attack to disrupt mobile ad hoc routing. We show that “rushing attack” is a severe attack against on-demand ad hoc routing schemes. Fortunately, “localized forwarding community area” is an available countermeasure to ensure that the failure probability of packet forwarding is negligible. This demonstrates the usefulness of our negligibility-based network security model. We expect to augment the pool of negligibility-based protections and explore the general notion in other types of networks.

Keywords—Net-centric Security = Negligibility + Scalability

I. 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, such self-organizing networks are very vulnerable to non-cooperative behaviors from the inside and malicious attacks from the outside. Although purely cryptographic countermeasures are effective against outsiders, they are not effective against insiders because cryptographic trust is rendered to whoever owns the cryptographic keys, independent of node's networking behavior. The network must rely on new network-centric protections to cope with these non-cooperative insiders. In this paper, our goal is to carry out a formal rigorous treatment of the secure routing problem against insider's disruption, which is a critical problem in network-centric security.

First, formal security is defined by the adversary model, the cost of attack and the cost of defense. Since 1970s, modern cryptography has abandoned the assumption that the adversary has available infinite resource to break the defense system, and assumes instead that any practical adversary is a polynomially-bounded probabilistic algorithm. In a complex and probabilistic system, we speak of the ”infeasibility” of breaking the security system rather than the ”impossibility” of breaking the same system. Security is defined on the concept of ”negligible”, which is asymptotically sub-polynomial with respect to a pre-defined system parameter n. Intuitively, the parameter n in cryptography is the key length, and efficiency of all related algorithms is measured in polynomials of n.

Second, we apply the same formal security notion in network security research. In network security, the network scale (i.e., number of network members) N replaces the role of key length n in cryptography. In a dynamic environment like MANET, it is impossible to ensure absolute security to protect everything. We have to adopt a *probabilistic* security framework. Once a negligibility-based protection is found in network design, we seek to prove that the probability of security breach decreases exponentially toward 0 as the network scale increases polynomially. In other words, just like the key length's impact on crypto-systems, the larger the network scale is or the more complex the network system is, the more secure the network is. This concludes that larger or more complex systems are favored over smaller or simpler systems in terms of survivability (assuming a negligibility-based protection is feasible).

Third, in MANET, we propose a concept of “$\mathcal{GVG}$ *polynomial-time*” protocol/algorithm as the formal model of a secure ad hoc scheme. Given a “global virtual god” ($\mathcal{GVG}$) that virtually oversees the network, the number of protocol steps is polynomially bounded by the input parameter N (the number of network members in the bounded network area). In $\mathcal{GVG}$'s view, no one in the system has exponential or other super-polynomial capability measured in N. Both the legitimate routing scheme and the adversary's attack scheme are measured in polynomials of N (i.e., $poly(N)$):

1) *Distributed scalable network assumption*: Each legitimate node's capability is bounded by $poly(N)$. This network assumption clearly differentiates those centralized non-scalable distributed systems from the self-organizing networks studied in our work. In the former case, a centralized server can beat a polynomially/linearly increasing network component and accomplish the needed network function. In contrast, in the latter case no single node in the network is able to accomplish the network function (e.g., routing) provided by a polynomially growing network component.
2) *Polynomially-bounded adversary*: The adversary is allowed to capture and compromise a fraction θ of N (as $\theta \cdot N$ is a polynomial of N) network members. θ is the node compromise probability that captures the hardness of breaking a mobile node's physical protection. Moreover, node compromise does not increase the captured node's capability beyond the polynomial bound. This way, as the sum/product of all adversary's capability is yet another $poly(N)$ (because sum/product of polynomials is another polynomial), the aggregation of all compromised nodes' capability is within $poly(N)$. Thus the adversary *cannot* thwart a security scheme which reduces the probability of security failure to negligible.

Finally, we show that “localized greedy recovery” is a negligibility-based protection against “rushing attack” [11]. For any mobile node following *any* mobility probability distribution function (PDF) in the bounded network area, our formal model illustrates that the probability of an empty forwarding *area* is negligible. Thus in order to achieve routing security in the negligibility-based framework, an anti-disruption secure routing scheme can choose to implement a greedy recovery to ensure that routing is feasible as long as there is at least one honest node in the forwarding area. This effectively thwarts rushing attackers.

The rest of the paper is organized as follows. Section II describes related security work in MANET. In Section III we present the negligibility-based model to formally specify network-centric security in mobile ad hoc networks. Next in Section IV we use the formal model to show the reason why "rushing attack" [11] is a severe attack against ad hoc routing protocols. Section V describes a practical community-based countermeasure to defend against rushing attack. Finally Section VI summarizes the paper.

II. Related work

A. On-demand routing

Most routing protocols in ad hoc networks fall into two categories: proactive routing and reactive routing (aka., on demand routing) [5]. In proactive ad hoc routing protocols like OLSR, TBRPF and DSDV, mobile nodes constantly exchange routing messages which typically include node identities and their connection status to other nodes (e.g., link state or distance vector), so that every node maintains sufficient and fresh network topological information to allow them to find any intended recipients at any time.

On the other hand, on-demand protocols generally have lower overhead and faster reaction time than other types of routing based on periodic (proactive) mechanisms. AODV [20] and DSR [13] are common examples. They are better suited for most ad hoc applications. Unlike their proactive counterparts, on demand routing operation is triggered by the communication demand at sources. Typically, an on demand routing protocol has two components: *route discovery* and *route maintenance*. In route discovery phase, the source seeks to establish a route towards the destination by flooding a route request (RREQ) message, then waits for the route reply (RREP) which establishes the on-demand route. In the route maintenance phase, nodes on the route monitor the status of the forwarding path, and report to the source about route errors. Optimizations could lead to local repairs of broken links.

B. Secure ad hoc routing

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 [24] or symmetric key based protocols (e.g., TESLA [22])[9] 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" [10], "rushing attack" [11] and various resource depletion attacks [16], 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. In rushing attack, malicious nodes increase the chance to be forwarder by rushing RREQ forwarding. Once the attacking nodes are en route, they can launch various attacks against data delivery.

Network-based countermeasures must be devised to answer the new challenges. To defeat rushing attackers, Hu et al. [11] 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 an honest member. In this countermeasure, a community-based solution is implicitly applied only at RREQ phase, but not in RREP and data delivery phases. In [16], a fully community-based solution is proposed to implement anti-disruption routing in all on-demand routing phases. In local recovery query [26], the forwarders need to *cooperatively* query a larger recovery area to fix a damaged link. Though this cooperative assumption does not apply to non-cooperative members in our adversary model, the proposed scheme implements a localized greedy recovery protocol for anti-disruption purposes. Nevertheless, none of the related work has used a formal network security model studied in this paper.

In multi-path routing [23][17] , more paths parallel (albeit some of them are near) to the optimal path are maintained, a disrupted path is replaced by another node-disjoint or link-disjoint path rather than fixed locally. However, the solution's complexity is huge because it is hard to find node-disjoint or link-disjoint paths in a MANET and to maintain these paths. Moreover, it incurs extra overheads to maintain redundant paths other than the optimal path and to deliver data on these non-optimal paths. Papadimitratos and Haas [19] 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 nodes on a probed path. The security model behind these multi-path routing schemes is not based on the formal concept of negligibility.

III. Formal treatment of network security in MANET

In this section we propose a concept of "$\mathcal{GVG}$*-polynomial time*" protocol/algorithm as the formal model of a secure ad hoc scheme. Given a "global virtual god" ($\mathcal{GVG}$) that virtually oversees the network, the number of protocol steps is polynomially bounded by the number of network members N. The notions used are listed below:

N	network scale (# of nodes in the network)
$\|x\|$	the cardnality of a set x
τ	least network time granularity (e.g., 1 nano-sec)
$\alpha = poly(N)$	α is a polynomial of N
$\Sigma < O(poly(N))$	Σ is asymptotically less than $poly(N)$
A	the area size of the entire network area
a	the area size of an average node "position"
L	the size of the largest mobile node's storage

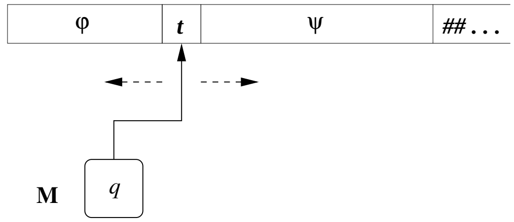

Fig. 1. **1-tape Turing Machine M in configuration (q, φ, t, ψ)**

A. Applied Turing Machines and Complexity classes

Turing Machines A Turing machine consists of a tape, a head, a state register, and an action table. According to the number of used tapes Turing machine is classified into two classes, namely 1-tape and k-tape Turing machine. We define now formally Turing machine.

Definition 1: A Turing machine is a septuple $M = (Q, \Gamma, \Sigma, q_I, \#, F, \delta)$, where

- Q is a finite set of states.
- Γ is a finite set of the tape alphabet.
- $\Sigma \subseteq \Gamma$ is a finite set of the input alphabet.
- $q_I \in Q$ is the initial state.
- $\# \in (\Gamma - \Sigma)$ is the blank symbol.
- $F \subseteq Q$ is the set of final or accepting states.
- δ is the transition set. For 1-tape Turing Machine, δ is

$$\delta : Q \times \Gamma \leftarrow Q \times \Gamma \times \{L, R\},$$

while for k-tape Turing Machine, δ is

$$\delta : Q \times \Gamma^k \leftarrow Q \times (\Gamma \times \{L, R, S\})^k$$

Here L is left shift, R is right shift, and S is stationary without shift. □

Using 1-tape Turing Machine as an example, as depicted in Figure 1, a *configuration*, or *instantaneous description*, of M is a quadruple

$$(q, \varphi, t, \psi), \quad \varphi\psi \in \Gamma*, \ t \in \Gamma, \ q \in Q$$

in which the rightmost symbol of ϕ is not $\#$. The string of symbols $\varphi t \psi$ is called the *tape* of the configuration. If $\varphi = \lambda$ and $q = q_I$, the configuration is an *initial configuration* of M.

Upon each left (or right) *move*, the current symbol t under the tape head is replaced by t', and the tape head is moved to the immediate left (or right) of the replaced symbol. Then M's current state q is replaced by q'. If a machine enters a state $q' \in F$ or has no moves from a given configuration, the configuration is *dead*. Otherwise, we say that

$$(\lambda, q_I, t, \psi) \Longrightarrow (\varphi', q', t', \psi')$$

is a *computation* of M, if M has a sequence of moves leading from the initial configuration (λ, q_I, t, ψ) to the final configuration $(\varphi', q', t', \psi')$, and call the computation *halted* if the final configuration is dead.

Definition 2: A Turing Machine is *deterministic Turing Machine (DTM)* if at most one move is possible from each configuration in the machine's transition set δ.

A Turing Machine is *non-deterministic Turing Machine (NDTM)* if more than one move is possible from each configuration in the machine's transition set δ.

A Turing Machine is *probabilistic Turing Machine (PTM)* if it is NDTM and the different moves are taken with certain probabilistic distribution. □

A probabilistic Turing machine is a non-deterministic Turing machine which randomly chooses between the available transitions at each point with certain probability. As a consequence, a probabilistic Turing machine can (unlike a deterministic Turing Machine) have stochastic results; on a given input and instruction state machine, it may have different run times, or it may not halt at all; further, it may accept an input in one execution and reject the same input in another execution.

A common reformulation of PTM is a DTM with an added *random tape* full of random bits, which are pre-determined by an oracle's coin-flips and placed on the tape to replace the DTM's own coin-flips in decision. The DTM with added random tape is equivalent to the PTM if the oracle's coin-flips and the DTM's (assumed-to-be) coin-flips follow the same probabilistic distribution.

Complexity classes used in our study Like modern cryptography, our net-centric security notion is based on "non-deterministic" and "probabilistic" algorithms. In modern cryptography, probability of security failure (e.g., inverting a one-way function, distinguishing cryptographically strong pseudorandom bits from truly random bits) is defined on the concept of "negligible", which is *asymptotically* sub-polynomial with respect to a pre-defined system parameter n. Intuitively, the parameter n in cryptography is the key length.

Definition 3: **(Negligible):** A function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}$ is *negligible* if for every positive polynomial $poly(n)$, and all sufficiently large n's (i.e., there exists N_c, for all $n > N_c$),

$$\epsilon(n) < \frac{1}{poly(n)}. \quad \square$$

The negligibility-based security is against a polynomially bounded adversary, such as $\mathcal{RP}$ and $\mathcal{BPP}$. In all cases, *if per-step probability of security failure is negligible, the overall probability of security failure after polynomial steps (implemented by the polynomially bounded adversary) stays as negligible*. Intuitively, negligibility is an asymptotic fix-point for polynomial-time algorithms. $\mathcal{RP}$ and $\mathcal{BPP}$ are defined when uniformly distributed randomness (aka. coin-flips, coin-tosses) is introduced (on the "random tape" in the equivalent DTM). Every problem in $\mathcal{RP}$ is bounded with one-side negligible errors, while every problem in $\mathcal{BPP}$ is bounded within two-side negligible errors. These errors stay as negligible against any polynomial-time algorithm.

Let x be the input in the polynomial size of a system parameter n, let $M(x)$ be the random variable denoting the output of a PTM M. Let

$$\textsf{Pr}[M(x) = y] = \frac{|\{d \in \{0,1\}^{t_M(x)} : \ M_d(x) = y\}|}{r^{t_M(x)}}$$

where d is a truly random coin-flip, $t_M(x)$ is the polynomial number of coin-flips made by M on input x, and $M_d(x)$ denotes the output of M on input x, when d is the outcome of its coin-flips (i.e., the random tape of an equivalent DTM).

Definition 4: **(Randomized Polynomial-time, $\mathcal{RP}$):** We say that L is recognized by the probabilistic polynomial-time Turing Machine M with negligible single-side errors if

- for every $x \in L$ it holds that $Pr[M\ accepts\ x] \geq \frac{1}{2} + \frac{1}{poly(n)}$ or $Pr[M\ accepts\ x] \geq 1 - \frac{1}{poly(n)}$ for every polynomial $poly(n)$.
- for every $x \notin L$ it holds that $Pr[M\ accepts\ x] = 0$.

$\mathcal{RP}$ is the class of languages that can be recognized by such a probabilistic polynomial time Turing Machine. □

Definition 5: **(Bounded-Probability Polynomial-time, $\mathcal{BPP}$):** We say that L is recognized by the probabilistic polynomial-time Turing Machine M with negligible double-side errors if

- for every $x \in L$ it holds that $Pr[M\ accepts\ x] \geq \frac{1}{2} + \frac{1}{poly(n)}$ or $Pr[M\ accepts\ x] \geq 1 - \frac{1}{poly(n)}$ for every polynomial $poly(n)$.
- for every $x \notin L$ it holds that $Pr[M\ accepts\ x] \leq \frac{1}{2} - \frac{1}{poly(n)}$ or $Pr[M\ accepts\ x] \leq \frac{1}{poly(n)}$ for every polynomial $poly(n)$.

$\mathcal{BPP}$ is the class of languages that can be recognized by such a probabilistic polynomial time Turing Machine. □

In below, we will show that the probability of security failure decreases exponentially toward 0 when the corresponding network metrics increase linearly. In this paper, the network scale (i.e., number of network members) N replaces the key length n in cryptography. N becomes the critical system parameter in *network-centric security*. As a result, in negligibility-based cryptography, the longer the key length is, the more asymptotically secure a cryptosystem is; In negligibility-based network security, the larger the network scale is, the more asymptotically secure the network is.

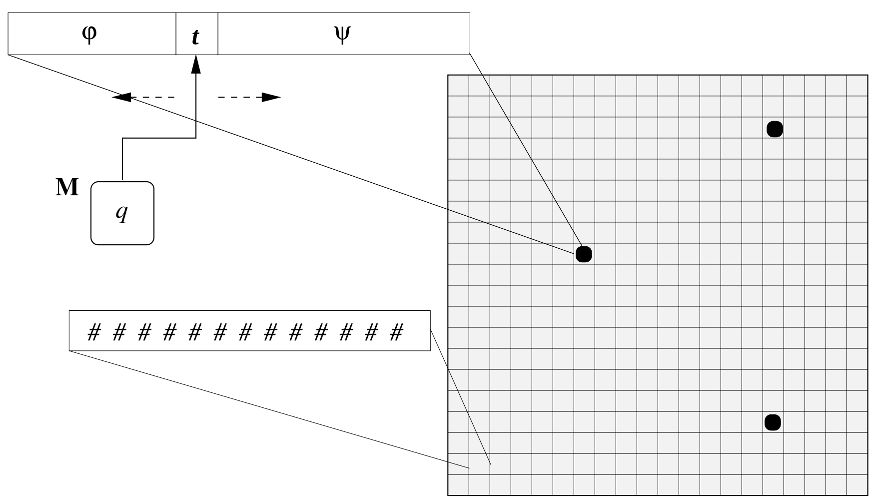

Fig. 2. **A $\mathcal{GVG}$ Probabilistic Turing Machine ($\mathcal{GVG}$ PTM) to model mobile nodes in a finite square area with a large number of node "positions". The figure shows that $N = 3$ of η ($N \ll \eta < O(poly(N))$) such "positions" have been taken by $N = 3$ mobile nodes. Each empty "position" is filled with a tape of $poly(N)$ blank symbols, and the blank tape is replaced with a mobile node's tape once the corresponding position is taken, or the tape goes back to the blank tape upon the node's leaving of the position. If the largest tape length of each mobile node can carry is $L < O(poly(N))$, then the $\mathcal{GVG}$ PTM's *consummate tape* length is $\eta{\cdot}L$. The $\mathcal{GVG}$ PTM's tape head always parks at the place corresponding to the current symbol of the first mobile node (the node with least node index). All random decisions, such as the mobile node's mobility patterns, are "as if" decided by the $\mathcal{GVG}$ using coin-flips. In theory, the $\mathcal{GVG}$ does all symbol processing and coin-flipping operations and its operation speed is fast enough to process all symbols on its tape within the least network time granularity τ**

B. *Modeling mobile networks: a $\mathcal{GVG}$ PTM approach*

We propose to use a special form of PTM to model the probabilistic stochastic behaviors of a mobile network. The fundamental idea is to use a *global virtual god* ($\mathcal{GVG}$) to handle the PTM's control states, while each mobile node is only treated as a tape carrier.

As depicted in Figure 2, the entire network area is of finite size A. The finite network area A is divided into large number of small tiles of size a, and each tile is smaller than the physical size of any mobile node. In other words, each tile is virtually a node "position" to stand on. The number of node "positions" $\eta = \frac{A}{a}$ is quite large. It is nevertheless a finite number. In practice, $\eta = \frac{A}{a}$ is a large constant, but is always asymptotically less than $poly(N)$, that is, $\eta < O(poly(N))$.

Tape Each mobile node functions as a carrier of a *moving tape* of polynomial size of the network scale N. That is, each mobile node carries a tape of $O(poly(N))$ bits. A moving tape is intuitively the memory snapshot of the corresponding mobile node. Let $L < O(poly(N))$ be the size of the largest moving tape. An empty node "position" is occupied by a blank tape of L blank symbols. This blank tape is replaced with a node's moving tape once the corresponding position is taken by the node, or the tape goes back to the blank tape upon the node's leaving of the position. If the largest tape length of each mobile node can carry is $L < O(poly(N))$, then the $\mathcal{GVG}$ PTM's *consummate tape* length is $\eta{\cdot}L$, which is $< O(poly(\eta)){\cdot}O(poly(N))$, thus $< O(poly(N))$.

Control state operations Each mobile node's decision of network operation (e.g., packet forwarding and routing), though autonomous in nature, can be translated into an equivalent form *as if all the decisions are made by the $\mathcal{GVG}$ using coin-flips*. Along the timeline, there exists a minimal time granularity τ such that any Turing Machine operation latency less than τ will make *no* difference in network protocol execution. For $\eta > N$, we assume that the $\mathcal{GVG}$ can make decisions for all mobile nodes and emulate all the decisions globally within the granularity τ.

The mobile nodes are indexed from 1 to N. At the beginning/end of each τ time granularity, the PTM's tape head always parks at the place corresponding to the current symbol of the first mobile node (with node index 1). During a τ interval, the PTM processes every mobile node's tape one by one (treating the corresponding node as a puppet of the $\mathcal{GVG}$).

Environmental randomness As to environmental conditions, for each network operation (e.g., packet forwarding and routing), the $\mathcal{GVG}$ emulates the physical condition (e.g., air humidity and obstacles that affect wireless radio transmission) in a perfect manner, and precisely moves each packet from one forwarding node to another. That is, the packet content is deleted from the sending node's moving tape, and the received packet content is added to the proper place of the receiving node's moving tape. In the eyes of the $\mathcal{GVG}$, any packet forwarding is simply a movement of a set of tape symbol from one place of its consummate tape to another place.

PTM as DTM with random tape If we use DTM rather than PTM to model the network protocol execution, the $\mathcal{GVG}$ can pre-cast many coin-flips to emulate the probabilistic events in the network, and place the result of the coin-flips to an added consummate random tape. These probabilistic events include mobile node's probabilistic moving pattern, probabilistic application requests to create and destroy end-to-end sessions, probabilistic packet forwarding and queuing events, probabilistic packet transmission collision, etc. The totally number of coin-flips (or the length of the consummate random tape) is $< O(poly(\eta))\cdot O(poly(N))$, thus $< O(poly(N))$.

Definition We formally define $\mathcal{GVG}$ Polynomial-time Probabilistic Turing Machine in below.

Definition 6: A *$\mathcal{GVG}$ Polynomial-time Probabilistic Turing Machine ($\mathcal{GVG}$-PPTM)* is an octuple

$$M = (N, \mathcal{GVG}(Q,r), \Gamma, \Sigma, q_I, \#, F, \delta),$$

where

- N is a pre-defined system parameter. N quantifies the size of the $\mathcal{GVG}$-PPTM's input and output. For any configuration (q, φ, t, ψ), $\varphi\psi \in \Gamma*$, $t{\in}\Gamma$, $q{\in}Q$ on any single tape of the machine, $|\varphi|, |\psi| < O(poly(N))$.
- $\mathcal{GVG}(Q,r)$ is an oracle with finite set of states Q and a probabilistic coin-flip sequence r (i.e., the random tape input of an equivalent DTM). $|Q|$ and $|r|$ are $< O(poly(N))$.
- Γ is a finite set of the tape alphabet.
- $\Sigma \subseteq \Gamma$ is a finite set of the input alphabet.
- $q_I{\in}Q$ is the initial state.
- $\#{\in}(\Gamma - \Sigma)$ is the blank symbol.
- $F{\subseteq}Q$ is the set of final or accepting states.
- δ is the transition set. For $\mathcal{GVG}$-PPTM having the consummate tape, δ is

$$\delta : Q \times \Gamma {\leftarrow} Q \times \Gamma \times \{L, R\}.$$

  Here L is left shift and R is right shift.

We say that L is recognized by the $\mathcal{GVG}$-PPTM M with negligible errors if

- for every $x{\in}L$ it holds that $Pr[M\ accepts\ x] \geq 1 - \frac{1}{poly(N)}$ for every polynomial $poly(N)$;
- for every $x{\notin}L$ it holds that $Pr[M\ accepts\ x] = 0$.

$\mathcal{GVG} - \mathcal{RP}$ is the class of languages that can be recognized by such a $\mathcal{GVG}$-PPTM. □

For every $x{\in}L$, $Pr[M\ accepts\ x]$ means "probability of protocol success", while its complement $Pr[M\ rejects\ x]$ means "probability of protocol failure". In $\mathcal{GVG} - \mathcal{RP}$, the former one must be $1 - \epsilon(N)$ and the latter one must be $\epsilon(N)$ in terms of network scale N. Note that here we have set the threshold to 0 to denote the surviving probability of a network protocol (rather than to $\frac{1}{2}$ to denote indistinguishability with the truly random half-half outcomes of coin-flips). As unintended network operations should always fail, so far we do not need double-side errors, thus spare the need to define $\mathcal{GVG} - \mathcal{BPP}$.

Example 1: **(Snapshot ad hoc routing)** In a (connected) *snapshot*[1] of a mobile network running AODV or DSR routing, nodes can be viewed as "puppets" of the $\mathcal{GVG}$. Based on the random coin-flips (or the random tape of an equivalent DTM) that simulate the probabilistic application demand, $\mathcal{GVG}$ initiates an RREQ flood on a source node. In the worst case, all mobile nodes organize into a linear chain topology, thus the route discovery procedure ends in $2{\cdot}N < O(poly(N))$ hops. When the corresponding RREP symbols come back to the source node, $\mathcal{GVG}$ enters a final acceptance state to finish[2] the on-demand

[1]Equivalently, as already assumed in AODV and DSR, the node mobility speed must be within a reasonable bound in any mobile scenarios, such that the there is at least an RREQ forwarder can forward the coming back RREP at the RREP forwarding moment.

[2]Here only route establishment is discussed. For mobile scenarios, a naive route maintenance plan is to periodically re-run this snapshot ad hoc routing scheme. As demonstrated in [16], the source and the destination can do periodic constrained flooding or recoverable unicast probes to gain better performance.

route discovery protocol. As an analogy, a $\mathcal{GVG}$ is a theoretic ideal entity corresponding to network simulators like NS2, QualNet/GloMoSim and OPNET. The only difference is that $\mathcal{GVG}$ can run perfect simulation beyond the finest time granularity.

Here every language string x starts from the RREQ symbols at the source node (on the consummate tape) and ends at the RREP symbols at the same source node (on the consummate tape). $x \in L$ means that there is indeed a physical route between the source node and the intended destination node in the network snapshot. Due to route disruption attacks, it is possible that the RREP symbols fail to come back at the source node for these $x \in L$.

For a secure routing protocol in $\mathcal{GVG}-\mathcal{RP}$, the probability of route discovery success *Pr*[RREP received at source] must be $1 - \epsilon(N)$, while the probability of route discovery failure *Pr*[RREP not received] must be $\epsilon(N)$. However, as we illustrate in Section IV, AODV and DSR are not in $\mathcal{GVG} - \mathcal{RP}$ under severe routing attacks like the "rushing attack"[11]. *To be in $\mathcal{GVG} - \mathcal{RP}$, we must ensure that the probability of per-hop forwarding failure is negligible*. Then the overall probability of routing failure of $O(poly(N))$ hops/steps would stay as negligible due to the mathematical properties of negligibility. □

Discussion In $\mathcal{GVG} - \mathcal{RP}$, no one in the system has exponential or other super-polynomial capability measured in N. In other words, both the legitimate routing scheme and the adversary's attack scheme are bounded by $poly(N)$.

First, each legitimate node has resources or capabilities bounded by $poly(N)$. This network assumption clearly differentiates those centralized infrastructure systems from the self-organizing infrastructureless networks studied here. In the former case, a centralized server can beat a polynomially growing network component and furnish the needed network function. Hijacking such a VIP node compromises the network system. In contrast, in the latter case no single node in the network is able to accomplish the network function (e.g., routing) provided by a polynomially/linearly increasing network. Hijacking a fraction of nodes doesn't crash the networked function as long as the remaining fraction of nodes are still functioning.

Second, the adversary is allowed to capture and compromise a fraction θ of N (as $\theta \cdot N$ is a polynomial of N) network members. θ is the node compromise probability that captures the hardness of breaking a mobile node's physical protection. Moreover, node compromise does not increase the captured node's capability beyond the polynomial bound. Each compromised node's capability is also at a polynomial level of a legitimate node. This way, as the sum/product of all adversary's capability is yet another $poly(N)$ (because sum/product of polynomials is another polynomial), the aggregation of all compromised nodes' capability is less than $O(poly(N))$. Thus the adversary cannot thwart a security scheme which reduces the probability of security failure to negligible[3].

Finally, *exponential capability* in our negligibility-based model means the capability to overwhelm an entire sub-area of the network despite there are honest nodes in the area. This is beyond the capability of capturing a subset of legitimate network members. The mathematical reasons are described below.

C. Underlying spatial model

As described above, we divide the network area A into a large amount of small (virtual) tiles of size a, so that the tile size is even smaller than the physical size of the smallest network member. This way, each tile is either empty, or is occupied by a single node. Also because the network area is much larger

[3]For an exponentially decreasing quantity, by *L'Hospital*'s rule, $\frac{poly(N)}{e^N}$ is negligible as N increases linearly/polynomially.

than the sum of all mobile nodes' physical size, the probability that a tile is occupied by a mobile node is very small.

Now a binomial distribution $B(\eta, p)$ defines the probabilistic distribution of how these tiles are occupied by each mobile ad hoc node. Here $\eta = \frac{A}{a}$, the total number of "positions", is very large but $< O(poly(N))$; and p, the probability that a tile is occupied by the single node, is very small. When η is large and p is small, it is well-known that a binomial distribution $B(\eta, p)$ approaches Poisson distribution with parameter $\rho_1 = \eta \cdot p$. Hence this binomial spatial distribution is translated into a *spatial Poisson point process* [6] to model the random presence of the network nodes. In other words, ρ_1 can be treated as a mobile node's arrival rate of each standing "position". Moreover, suppose that N events occur in area $\mathcal{A}$ (here an event is a mobile node's physical presence), $\rho_{_N} = \frac{N}{\mathcal{A}}$ (where $\rho_{_N} = N \cdot \rho_1$ if N nodes roam independently and identically distributed) is equivalent to a random sampling of $\mathcal{A}$ with rate $\rho_{_N}$.

Let x denote the random variable of number of mobile nodes in any network area concerned:

- (*Uniform* ρ_1) the probability that there are exactly k nodes in a specific area $\mathcal{A}'$ following a uniform distribution model is
$$\textit{Pr}[x = k] \;=\; \frac{(N \cdot \rho_1 \cdot \mathcal{A}')^k}{k!} \cdot e^{-N \cdot \rho_1 \cdot \mathcal{A}'}. \tag{1}$$
- (*Non-uniform* ρ_1) More generally, in arbitrary distribution models including non-uniform models, the arrival rate is *location dependent*. The probability that there are exactly k nodes in a specific area $\mathcal{A}'$ is
$$\textit{Pr}[x = k] \;=\; \iint_{\mathcal{A}'} \left(\frac{(N \cdot \rho_1)^k}{k!} \cdot e^{-N \cdot \rho_1} \right) \mathrm{d}\mathcal{A}. \tag{2}$$

Discussion on mobility PDF ρ_1: Our study is based on the mobility PDF ρ_1 that captures an average mobile node's mobility presence in the bounded network area. This is more general than a study based on a specific mobility model like random walk and random waypoint models, since any node mobility model can be transformed into its corresponding mobility PDF ρ_1 as shown below.

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

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

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

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

where $f_{XY}(x, y)$ can be computed by a stochastic analysis of an arbitrary mobility model. Let's use random waypoint (RWP) model as an example. As computed in [3], 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 RWP model in a square network area of size $a \times a$ defined by $-a/2 \leq x \leq a/2$ and $-a/2 \leq y \leq a/2$. In [12], extensive simulation study of the RWP model has been used to empirically verify the correctness of the analytic conclusion.

In a nutshell, mobility PDF is a more general notion than a specific model like the RWP model. In Theorem 1, we will prove that our analysis is valid for *any* mobility PDF, not only a specific mobility model.

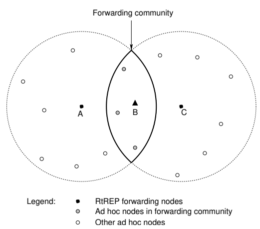

Fig. 3. **A forwarding community between a 2-hop source and destination pair**

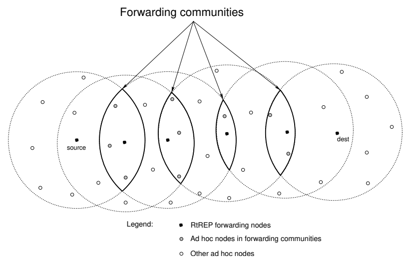

Fig. 4. **Forwarding communities along a multi-hop path**

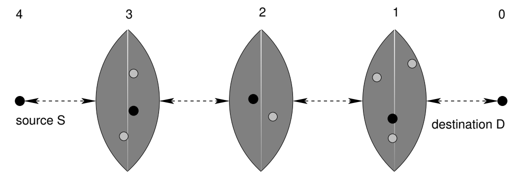

Fig. 5. **Forwarding communities as "big" virtual areas**

D. Negligibility-based Network Security

In order to ensure negligibility-based secure routing, at first we must ensure that per-hop security failure probability is negligible (with respect to network scale N). One candidate solution is "localized greedy recovery" which means that, when packet forwarding hop is rendered per *forwarding community area* rather than per node, secure ad hoc routing is feasible as long as there is at least one honest forwarder per hop.

The concept of "forwarding community area" is based on the observation that wireless packet forwarding typically relies on more than one immediate neighbor to relay packets. Figure 3 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 form the forwarding community area. Figure 4 depicts a chain of forwarding communities along a multi-hop path. Intuitively, a forwarding community is a "big virtual area" that replaces a single forwarding node in conventional routing schemes (Figure 5).

$\mathcal{GVG}$ **negligibility:** Now we show that an ideal implementation of forwarding communities ensures that the probability of per-step forwarding failure and the probability of polynomial-step routing failure are negligible.

Theorem 1: ($\mathcal{GVG}$-negligible at per hop/step) A routing protocol X is *$\mathcal{GVG}$-negligible* at per hop/step if the probability of packet forwarding failure is negligible with respect to the network scale N. A secure routing protocol which ideally implements forwarding communities is $\mathcal{GVG}$-negligible.

Proof: Let $\mathcal{A}'$ denote the expected size of a forwarding community area, and let y denote the random variable of number of honest network members in the expected forwarding community area. The probability that the expected forwarding area has k honest nodes is

$$Pr[y = k] \;=\; \iint_{\mathcal{A}'} \frac{((1-\theta)\cdot N\cdot\rho_1)^k}{k!}\cdot e^{-(1-\theta)\cdot N\cdot\rho_1}\ \mathrm{d}\mathcal{A}$$

In a secure routing scheme that ideally implements forwarding communities, the per hop/step probability of failure is

$$P_{stepfail} = \textit{\textbf{Pr}}[y=0] = \iint_{\mathcal{A}'} e^{-(1-\theta)\cdot N\cdot\rho_1}\ \mathrm{d}\mathcal{A}.$$

The mobility PDF ρ is arbitrary in our study, thus could be location dependent and becomes a function of the location area $\mathcal{A}$. Therefore, double integrals must be used here (or triple integrals in case of 3D scenarios). Fortunately, because e^x is a fixed point in differential and integral calculus, such integrals do not change the magnitude of order, that is, $\frac{\mathrm{d}e^x}{\mathrm{d}x} = e^x$ and $\int e^x\ \mathrm{d}x = e^x + C = O(e^x)$. In a nutshell, exponential quantities and polynomial quantities are unchanged in magnitude of order through these integrals. And this concludes that the probability of security failure per hop/step $P_{stepfail}$ is negligible with respect to the network scale N **in any mobility pattern PDF** ρ_1. □

Theorem 2: ($\mathcal{GVG}$-negligible at each step implies $\mathcal{GVG}$-negligible in polynomial-steps) A routing protocol X of polynomial hops/steps is $\mathcal{GVG}$-negligible if it is $\mathcal{GVG}$-negligible at each hop/step.

Proof: By assumption, X has $p(N)$ steps, where $p(N)$ is a positive polynomial. Given that per-step security failure probability is $P_{stepfail}$, the probability of security failure of the entire protocol $P_{polyfail}$ is

$$P_{polyfail} \ = \ 1 - (1 - P_{stepfail})^{p(N)}.$$

By assumption, $P_{stepfail}$ is negligible, thus is asymptotically less than any $\frac{1}{(p(N)+1)\cdot q(N)}$, where $q(N)$ is a positive polynomial and $(p(N)+1)\cdot q(N)$ is also a positive polynomial. In other words, there exists a positive integer $N_c > 0$, such that $P_{stepfail} < \frac{1}{(p(N)+1)\cdot q(N)}$ for all $x > N_c$. Then we have

$$(1 - P_{stepfail})^{p(N)} > \left(1 - \frac{1}{(p(N)+1)\cdot q(N)}\right)^{p(N)} > e^{-\frac{1}{q(N)}}$$

since $(1 - \frac{1}{x})^{x-1} > e^{-1}$ for all $x > 1$.

According to Lagrange mean value theorem, for a function $f(x)$ continuous on $[a, b]$, there exists a $c \in (a, b)$ such that $f(b) = f(a) + f'(c)\cdot(b-a)$ for $0 < a < b$. Then let $f(x) = e^{-x}$, there exists a $\xi \in (0, z)$, such that $e^{-z} = 1 + (-e^{-\xi})\cdot z > 1 - z$. Thus we have

$$(1 - P_{stepfail})^{p(N)} > e^{-\frac{1}{q(N)}} > 1 - \frac{1}{q(N)}.$$

Therefore, for any polynomial $q(N)$ and sufficiently large N,

$$P_{polyfail} \ = \ 1 - (1 - P_{stepfail})^{p(N)} < \frac{1}{q(N)}.$$

Thus the probability for the source to receive RREP and enter the final state is $1 - P_{polyfail} > 1 - \epsilon(N)$. □

Another proof of Theorem 2 using Chernoff bounds is available in standard cryptography literatures, such as [8]. Therefore, by Definition 6, such a protocol implementation with ideal forwarding communities is in $\mathcal{GVG} - \mathcal{RP}$.

Discussion: from ideal to practical A major contribution of this paper is to show the existence of the formal negligibility-based framework for network security. However, it is an open challenge to implement the formal framework in the real world. There are pre-conditions to realize an ideal implementation of the above-mentioned "forwarding community areas". We believe that *data origin authentication*, *secure neighbor detection*, *distance bounding* and *anti-jamming* services are prerequisites. (1) All packet transmissions (including control, data packets and their ACKs) are protected by data origin authentication service. For those honest senders, every packet is authenticated and the packet sender's identity is unforgeable. This can be implemented by signing each packet by the sender's certified digital signature or using efficient symmetric key protocols like TESLA [22][9]. Therefore, the adversary cannot forge packet transmissions from honest nodes, and cannot launch Sybil attack [7] by faking honest nodes' identities; (2) Secure neighbor detection protocols [11][18] must ensure that 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 single-hop three-way handshake (e.g. TCP style SYN-ACK-ACK) protocol with data origin authentication. On every honest node, packets received from undetected thus unauthenticated neighbors are dropped immediately; (3) Ad hoc nodes are equipped with hardware needed by packet leashes [10] or Brands-Chaum protocols [27][4]. Hence by secure distance bounding, any pair of topological neighbors in ad hoc routing are indeed physical neighbors; (4) 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.

IV. The severeness of rushing attack

A. *Rushing attack in intuition*

In on-demand routing, the source node initiates a Route Discovery process to find the target node. If the corresponding RREQs forwarded by the attacker are the first to reach each neighbor of the target, then any route discovered by this Route Discovery will include a hop through the attacker. In order to beat regular nodes in terms of forwarding latency and link speed, the attacker must acquire a relatively small latency and a relatively large link speed. Unfortunately, both can be done easily in ad hoc routing without the need of having access to vast resources.

First, Medium Access Control (MAC) protocols generally impose delays between when the packet is handed to the network interface for transmission and when the packet is actually transmitted. In a MAC using collision-free time division (like TDMA), for example, a node must wait until its time slot to transmit, whereas in a MAC using collision-based multiple access (like CSMA), a node generally performs some type of backoff to avoid collisions. In addition, because RREQ packets are broadcast, and collision detection for broadcast packets is difficult, routing protocols often impose a randomized delay in RREQ forwarding. Therefore, even if the MAC layer does not specify a delay, on-demand protocols generally specify a delay between receiving an RREQ and forwarding it, in order to avoid collisions of the RREQ packets. A rushing attacker can easily ignore delays at either the MAC or routing layers and becomes the "best" forwarder in terms of latency.

Second, a rushing attacker can ignore legitimate packet items in its network queues and MAC queues, thus gains advantage over legitimate nodes in terms of link speed. Another way that a rushing attacker can obtain an advantage in forwarding speed is to keep the network interface transmission queues of nearby legitimate nodes full. This can be realized, for example, in a secure ad hoc network relying on inefficient cryptography—the attacker can keep other nodes busy authenticating wireless packets, thus slowing their ability to forward legitimate RREQs.

Finally, once a rushing attacker is chosen as a forwarder en route, it may cause the loss of certain critical packets. It can choose to drop the coming-back RREP or to forward a corrupted RREP. After a timeout, the RREQ initiator must re-flood the network again and again. This is a transformed resource depletion attack, except the RREQ initiator is not the one to blame. Also a rushing attacker can severely degrade data delivery performance by (selectively) dropping data packets [1].

B. *Rushing attack as a formal and severe routing attack*

In this section, we use the negligibility-based model to prove that rushing attack is a severe attack against regular on-demand routing. We show that the *probability of forwarding success* at per-hop is negligible, thus the *probability of routing success* at any multi-hop path of $O(poly(N))$ size is negligible by Theorem 2.

As specified previously, there are N authenticated network members in the network, amongst them there are $\theta{\cdot}N$ dishonest rushing attackers and $(1-\theta){\cdot}N$ honest members. Let z denote the random variable of number of dishonest attackers in an arbitrary area $\mathcal{A}'$. The probability that there are k dishonest rushing attackers in the area $\mathcal{A}'$ is

$$\textit{\textsf{Pr}}[z=k] \;=\; \iint_{\mathcal{A}'} \frac{(\theta{\cdot}N\cdot\rho_1)^k}{k!}{\cdot}e^{-\theta\cdot N\cdot\rho_1}\;\mathrm{d}\mathcal{A}$$

In a regular on-demand routing scheme, the per-hop RREP forwarding success ratio, namely the per-hop route discovery success ratio, is computed from knowing all nodes in the forwarding area are honest. One rushing attacker will deprive the chance for other nodes to be the RREP forwarder. The per-hop success ratio is only

$$\begin{aligned} P_{hopsuccess} &= \textit{\textsf{Pr}}[y{\geq}1]\cdot\textit{\textsf{Pr}}[z=0] \\ &= \iint_{\mathcal{A}_{avg}} \left((1-e^{-(1-\theta)\cdot N\cdot\rho_1}){\cdot}e^{-\theta\cdot N\cdot\rho_1}\right)\;\mathrm{d}\mathcal{A} \\ &= (1-\epsilon(N)){\cdot}\epsilon(N)\;<\;\epsilon(N), \end{aligned}$$

where $\mathcal{A}_{avg}$ denotes the average size of the forwarding area (i.e., the intersection of three consecutive RREP transmission circles) and $\epsilon(N)$ denotes a negligible quantity with respect to N.

Since e^x is unchanged by differentials and integrals, the per-hop success ratio is negligible given an arbitrary node mobility PDF ρ_1, and the per-path success ratio is negligible by Theorem 2 for any path of the size $O(poly(N))$. This concludes that rushing attack is a severe routing attack that can reduce the success ratio of regular on-demand routing schemes to negligible. Proposed countermeasures against rushing attack include RAP [11] using forwarding communities during RREQ phase and CBS [16] using forwarding communities in all on-demand routing phases.

V. A PRACTICAL IMPLEMENTATION OF COMMUNITY-BASED SECURITY

In this section we describe a practical implementation to approach the ideal form of "forwarding community area".

A. Route discovery

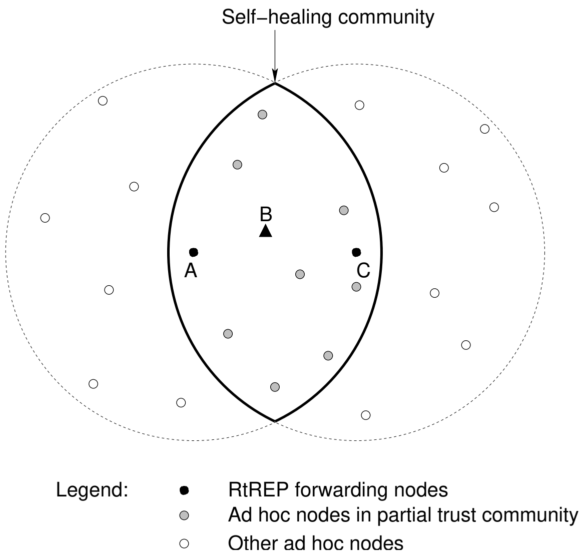

Fig. 6. **An inappropriate community**

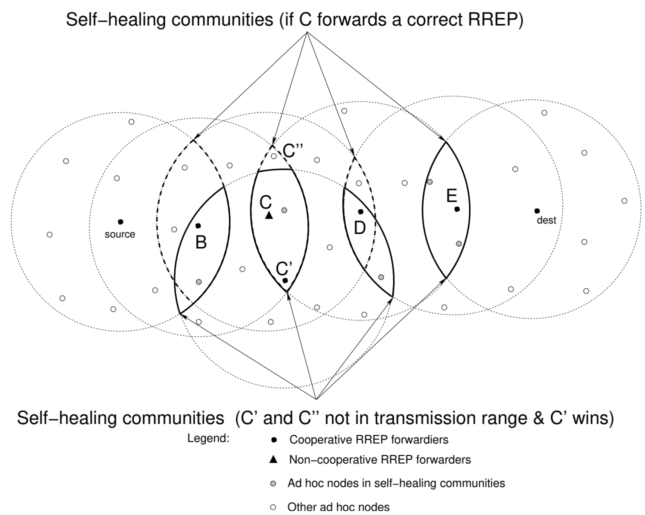

Fig. 7. **ACK solves ambiguity in take-over collisions**

A community must be formed properly. As a comparison to Figure 3, Figure 6 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[4]:

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

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.

[4]We do not include detailed packet formats of the underlying on-demand routing protocol. Interested parties may check [21] for AODV and [14] 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 A: During an RREQ flood, a node just received an authentic RREQ packet $V \rightarrow *$ for the current route discovery:

Insert V in my soft-state neighbor set;
U := the $upstream\_node$ field in the RREQ packet;
In my soft state, record U as V's upstream;
IF {(I have never forwarded the RREQ) AND
    (I have not heard U in my neighborhood during the RREQ)}
    Record V as my RREQ upstream for the connection;
    Process the packet according to the underlying routing protocol;
    Locally rebroadcast the RREQ packet.

Algorithm B: During RREP, a node $V'(\neq V)$ just heard the coming-back RREP packet $E \rightarrow V$ for the current route discovery:

Insert E in my soft-state neighbor set;
W := the RREQ upstream node recorded for V;
WHILE {(My soft state for the connection is still alive) AND
  (Both V and W are in my soft-state neighbor set) AND
  ((V didn't correctly forward RREP within the bounded window)
   OR (W didn't correctly ACK within the bounded window))}
    Wait for an autonomously decided random time;
    IF (During the waiting period nobody has taken over)
        Send RREP packet: $V' \rightarrow W$ (i.e., I try to take over).
    ELSE
        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:

Insert E in my soft-state neighbor set;
Record E as my RREP upstream for the connection.
W := the RREQ upstream node according to my soft-state;
Send RREP packet: $V \rightarrow W$.

Let's use Figure 3 to describe a simple example of 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 7, 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 V-C. Due to page limit, see our technical report [15] for the full-fledged design of Algorithm A, B and B$_V$ that uses network layer ACKs).

If S and D are more than two hops away, then the single-hop 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.

B. *Configuration of communities*

A chain of communities is configured during the 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 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:

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

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.

C. *Reconfiguration of communities*

The communities lose shape due to mobility and other network dynamics. For each S-D connection, we use end-to-end probing to reconfigure 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 " unicast" design like the one used in Algorithms B and B$_V$. Therefore, the RREQ rate limit approach proposed in [9][21] is practical and causes no major routing performance degradation in CBS.

Example 2: **(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 communities by issuing more probing requests. Meanwhile it also seeks to decrease probing overhead when the 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 [9][21] 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 procedure like Algorithms B and B_V (due to page limit, see our technical report [15] for more details). The 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.

D. 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 community. However, if this node fails to forward data packet due to maliciousness, selfishness, or network dynamics, members in the same 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:

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

E. Simulation Study

1) Simulation environment: We implement community-based security routing scheme on top of AODV (denoted as CBS-AODV) in QualNet [25], 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 [11] (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 V-C, denoted as "CBS-AODV,cons_flood") for comparison.

In our simulation scenario, 150 nodes are randomly placed within a field of size 2400m$\times$600m. The nodes move according to RWP model [13]. 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,

2) 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 packets. For RAP-AODV, we use the same parameters as used by the authors [11].

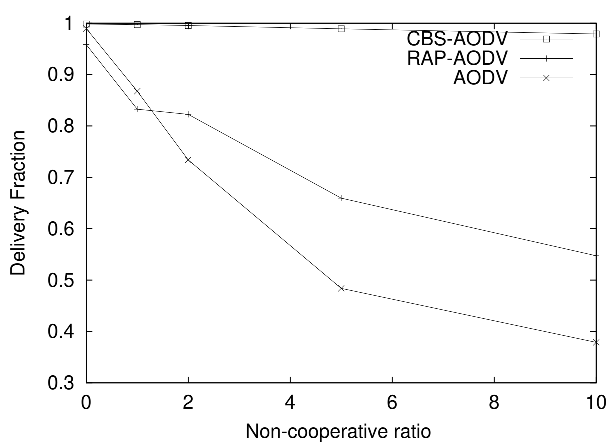

Fig. 8. **Data Packet Delivery Ratio**

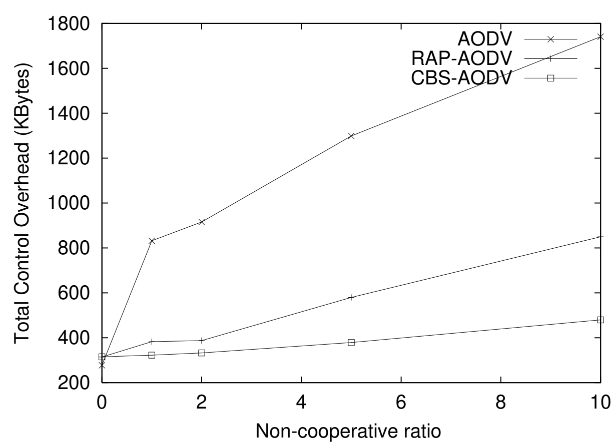

Fig. 9. **Control Overhead (Kbytes)**

Figure 8 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. 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 10 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 9 verifies that both AODV and RAP-AODV generate higher routing overhead when there are more non-cooperative nodes in the network.

Figure 11 and Figure 10 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

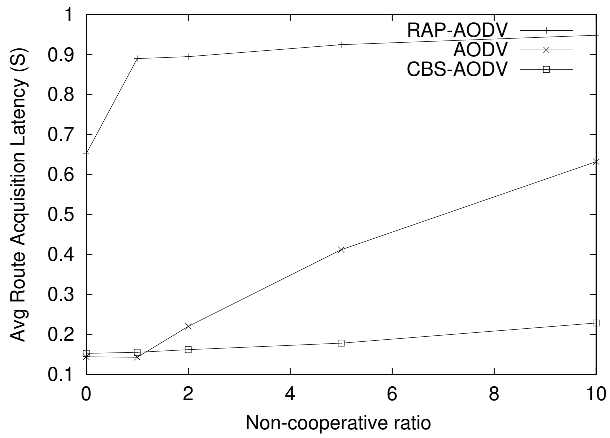

Fig. 10. **Average Route Acquisition Latency (S)**

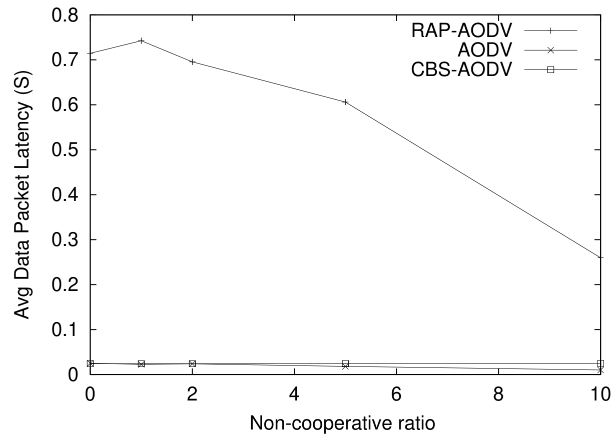

Fig. 11. **Average End-to-end Latency**

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 11 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.

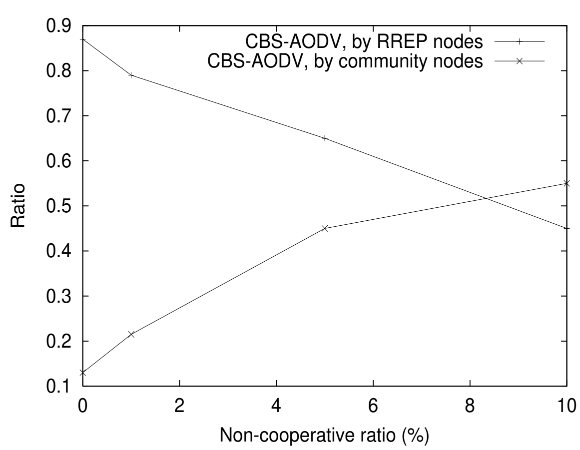

Fig. 12. **RREP forwarding ratio**

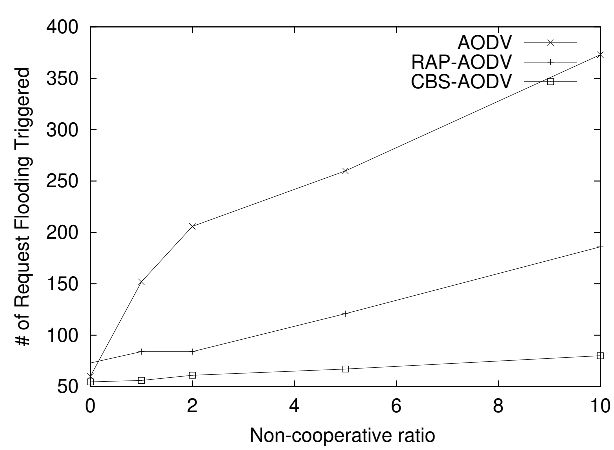

Fig. 13. **# of needed network-wide RREQ floods (if not limited by rate control)**

Figure 12 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 13 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.

3) 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 [28]). 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.

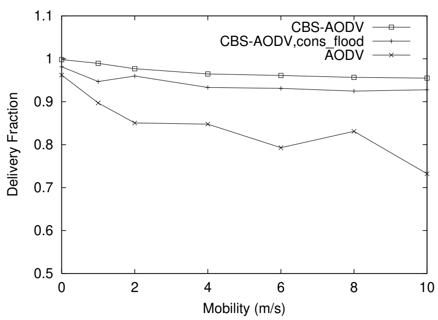

Fig. 14. **Packet Delivery Ratio**

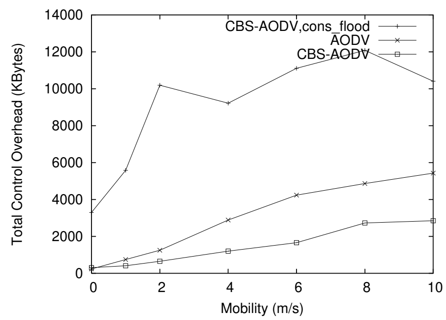

Fig. 15. **Control Overhead (Kbytes)**

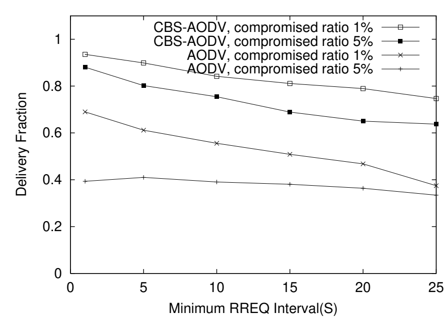

Fig. 16. **RREQ Flood Rate Limit Control**

In Figure 14, 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 15 shows that "CBS-AODV,cons_flood" is inefficient in terms of routing overhead. Figure 15 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 16 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.

The impact of network scale is limited by the simulator's capability. Currently, QualNet reports that it can simulate thousands of nodes on high-end servers, while OPNET and NS2 can only simulate a network of less than half of QualNet's scale on the same machine using the same network settings. We are currently trying to acquire high-end servers to simulate self-organizing networks at the scale of thousands of nodes or even tens of thousands of nodes. The related result will be reported when it is ready.

VI. Conclusions

In this paper we have presented a negligibility-based framework to formally model network-centric security problems in mobile ad hoc networks. We have used the same security notion of formal cryptography, except the input parameter is changed from key length to network scale. Our proposal is the first one to model network-centric security in a negligibility-based framework with network scale as the input parameter.

We use anti-disruption secure routing as an example of network-centric security. Non-cooperative network members can thwart mobile ad hoc routing by various means. In particular, "rushing attack" is proved to be a severe routing disruption attack that can reduce the probability of routing success to negligible. Fortunately, if we can reduce the probability of per-hop forwarding failure to negligible, then we can guarantee that the overall probability of routing failure is also negligible. A candidate solution to achieve the design goal is using "localized forwarding community areas" to replace conventional forwarding nodes. We prove that an ideal form of forwarding community can satisfy the negligibility-based model, thus a practical implementation approaching the ideal form is a valid protocol to defend against route disruptions. We expect to augment the pool of negligibility-based protections to include more countermeasures, and explore the general notion of network-centric negligibility in other types of networks.

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 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.

[4] 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.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *ACM MOBICOM*, pages 85–97, 1998.

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

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

[8] O. Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, 2001.

[9] 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.

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

[11] 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.

[12] Y.-C. Hu and H. J. Wang. A Framework for Location Privacy in Wireless Networks. In *ACM SIGCOMM Asia Workshop*, 2005.

[13] 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.

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

[15] 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.

[16] J. Kong, X. Hong, Y. Yi, J.-S. Park, J. Liu, and M. Gerla. A Secure Ad-hoc Routing Approach using Localized Self-healing Communities. In *ACM MOBIHOC'05*, pages 254–265, 2005.

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

[18] 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.

[19] 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.

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

[21] 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.

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

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

[24] 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.

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

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

[27] 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.

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