Modeling Ad-hoc Rushing Attack in a Negligibility-based Security Framework

Jiejun Kong*, Xiaoyan Hong†, Mario Gerla‡

*Scalable Network Technologies, Inc.
6701 Center Drive West, Suite 520
Los Angeles, CA 90045

†Dept. of Computer Science
University of Alabama
Tuscaloosa, AL 35487

‡Dept. of Computer Science
University of California
Los Angeles, CA 90095

jkong@scalable-networks.com, hxy@cs.ua.edu, gerla@cs.ucla.edu

ABSTRACT

In this paper, we propose a formal notion of network security for ad hoc networks. We adopt a probabilistic security framework, that is, security is defined by a polynomially bounded adversary model, the cost of attack and the cost of defense. 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 x. Intuitively, the parameter x in cryptography is the key length n. We apply the same bounds in ad hoc network security research, but in regard to *scalability* from now on. We propose an $\mathcal{RP}$ (n-runs) complexity class with a global virtual god oracle ($\mathcal{GVG}$) to model a general class of network protocols. In $\mathcal{GVG}$-$\mathcal{RP}$ (n-runs) class, the network scale (i.e., number of network members) N replaces the role of key length n in cryptography. From our formal rigorous treatment, we show that "rushing attack" is a severe attack that can reduce the success probability of common ad hoc routing schemes to negligible.

Fortunately, countermeasures can be devised to answer this challenge. (1) Common network protocols are *not* designed to ensure that probability of security failure is negligible. In such designs, the system's security is not related to scalability. There is no asymptotic security guarantee in the network design; (2) We seek to devise security schemes to ensure that the probability of security failure is *negligible* in regard to network *scale*. In Theorem 2, we present an asymptotic invariant for scalable networks: "a polynomial-time network algorithm that ensures negligible probability of security failure at each step would stay in the state of ensuring negligible probability of security failure globally". This invariant demonstrates the existence of asymptotic security guarantee in ad hoc networks. It leads to the design of community-based secure routing to defend against rushing attacks. Nevertheless, it is unknown to us whether the ideal invariant can be practically implemented.

WiSe'06, September 29, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-557-6/06/0009 ...$5.00.

Categories and Subject Descriptors

C.2.2 [**Computer-Commmunication Networks**]: Network Protocols—*Protocol Verification*

General Terms

Security, Theory, Verification

Keywords

Randomized network algorithms, Randomized Turing Machine, Negligibility, Sub-polynomial, Scalability, Asymptotic invariant

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. Since nodes must rely on other nodes in data forwarding and routing in MANET, the adversary has abundant opportunities to attack the cooperative network protocols. Amongst various attacks against cooperative ad-hoc routing, "rushing attack" [11] is one of the most practical scheme requiring nearly no extra cost (e.g., no need to expend exquisite hardware and expensive energy consumption). Such a "bottomline" attacking strategy is expected to be included in any adversary's travel kit to attack a wireless network. Therefore, assessing the damage caused by "rushing attack" and looking for needed countermeasure are essential requirements to protect mobile ad-hoc networks deployed in hostile environments.

1.1 Our contributions

First, we adopt a formal approach to characterize a general category of network algorithms/protocols, which belongs to the family of Monte Carlo algorithms with 1-side or 2-side errors. In Monte Carlo algorithms, algorithm execution is accomplished efficiently but with probabilistic deviations. In particular, the proposed algorithm/protocol class is a special class of Monte Carlo algorithm ending in *polynomial-time* (polynomial-step or polynomial-stop in this paper) and with *negligible deviation* $\epsilon(x)$ from the ideal case (as defined by a Las Vegas algorithm). Moreover, the deviation stays as negligible in polynomial steps $poly(x)$. That is, the negligible deviation $\epsilon(x)$ is an asymptotic invariant in terms of an input metric x, which is N, the number of network members in this paper.

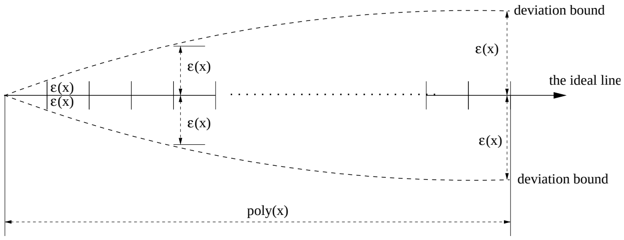

Figure 1: Polynomial-time Monte Carlo algorithm family with negligible deviations $\epsilon(x)$ (2-side errors depicted)

In a dynamic environment like MANET, it is impossible to ensure absolute security to protect everything. We have to adopt a *probabilistic* security framework. Ács, Buttyán and Vajda [2] have accomplished the earliest work to apply probabilistic indistinguishability based analysis in ad hoc routing research. Nevertheless, [2]'s approach is based on cryptographic indistinguishability which measures security in regard to key length n. Our approach is different from this earlier work as we seek to prove that the failure probability of a secure routing protocol is negligible in regard to a *network metric* x, which is **the network scale** N in this paper. In other words, the larger the network scale is, the more secure the network is in an asymptotic manner. This concludes that larger systems are favored over smaller or simpler systems (assuming a negligibility-enforcing mechanism is feasible) in terms of survivability. Intuitively, this is consistent with the evolution theory where more complex entities probabilistically emerge from and likely survive longer than their less complex competitors.

Second, in particular for MANETs, we propose a concept of "$\mathcal{GVG}$ *polynomial-time algorithm*" (or "$\mathcal{GVG}$ polynomial-step protocol") 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 steps in any (legitimate/adversarial) protocol is polynomially bounded by an input parameter N (the number of network members in the bounded network area).

1. **Distributed scalable network assumption**: Each legitimate node has resources or capabilities bounded by $O(poly(N))$. This implies that centralized non-scalable distributed systems are different from the self-organizing networks studied in this work. In former case, a centralized server or a set of servers bounded by a constant number $O(1)$ can accomplish the needed network function. In contrast, in latter case *no* $O(1)$ nodes in the network are able to accomplish the network function (e.g., ad hoc routing) provided by a network whose size is measured in $O(poly(N))$.
2. $\mathcal{RP}$ **(**n**-runs) model**: The efforts presented in [2] are based on cryptographic indistinguishability, which uses the $\mathcal{BPP}$ (Bounded-error Probabilistic Polynomial-time) class to measure indistinguishability between truly randomness and cryptographically strong pseudorandomness. In contrast, our model is based on the $\mathcal{RP}$ (Randomized Polynomial-time) class which measures success probability of a protocol execution (e.g., per-hop forwarding and multi-hop routing of a wireless packet) by ensuring negligible failure probability per step. These are two different components of the same problem: $\mathcal{BPP}$ indistinguishability is computation-centric while our $\mathcal{GVG} - \mathcal{RP}$ protocol execution is network-centric. Although both of them are probabilistic polynomial-time algorithms, one is defined on key length n to address computational cryptanalysis and the other is defined on network scale N to address node's non-cooperative behaviors.

   In addition, the $\mathcal{RP}$ n-runs model is different from the basic $\mathcal{RP}$ 1-run model. In the $\mathcal{RP}$ 1-run model, the failure probability is biased toward 0, but can be any number less than 1/2. *In the* $\mathcal{RP}$ *n-runs model, the failure probability must already be negligible* $\epsilon(n)$ *at every step*, because it is impractical to repeat an $\mathcal{RP}$ 1-run execution n times in a mobile network to do the $\mathcal{RP}$ amplification.
3. **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. While each network node's capability is polynomially bounded in terms of network scale N, the adversary is also polynomially bounded. It cannot thwart a negligibility-based $\mathcal{GVG} - \mathcal{RP}$ security scheme with non-negligible probability.

Third, we show that an ideal implementation of "localized coordination community" [17] is one of the feasible negligibility-enforcing primitives. For any mobile node, no matter what kind of continuous node presence PDF (probability distribution) it has in the bounded network area, our formal model illustrates that the probability of a forwarding area with no honest nodes is negligible[1]. Thus in order to achieve routing security in the negligibility-based framework, a secure routing scheme can choose to implement a greedy coordination to ensure that security goal is achieved as long as there is at least one honest node in the forwarding area. Our efforts have shown that a negligibility-enforcing primitive can provide protection for both privacy-preserving routing [18][15] (where the community is called "motion-MIX") and routing integrity [16][17]. This is very similar to the case that one-way function is a security primitive for both data privacy (e.g., encryption) and data integrity (e.g., message authentication code).

The rest of the paper is organized as follows. In Section 2 we present our formal model to show the reason why "rushing attack" [11] is a severe attack against ad hoc routing protocols, and why an ideal implementation of "localized forwarding community" is a qualified candidate solution. Section 3 describes related security work in MANET. Finally Section 4 summarizes the paper.

2. FORMALIZING RUSHING ATTACK

2.1 Rushing attack in intuition

Most routing protocols in ad hoc networks fall into two categories: proactive routing and reactive routing (aka., on demand routing) [6]. 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.

[1] In our model, integral calculus is applied on the arbitrary mobility PDF that is continuous in the network area. Then due to the fact that e^x is a fixed point in differential and integral calculus, such integrals do not change the magnitude of order. In other words, the orders exponential quantities $O(e^N)$ and polynomial quantities $O(poly(N))$ are unchanged by differentials and integrals.

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 [22] 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.

Unfortunately, the on-demand routing approach is vulnerable to various security threats. If the corresponding RREQs forwarded by the attacker are the first to reach each neighbor of the “real” forwarder, 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.

Third, once a rushing attacker is chosen as a forwarder en route, it may cause the loss of certain critical packets. The malicious losses are mixed into environmental random packet losses, thus hard to be identified. This way, the adversary 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].

Nevertheless, this intuitive explanation is *not* a formal research answer to the problem. To illustrate the network system’s complex behavior under rushing attacks, we need a more formal specification to identify the invariants in the scalable and probabilistic network system. In below we will present a new asymptotic invariant—“*a polynomial-step network protocol (i.e., a polynomial-time network algorithm) ensuring negligible probability of security failure at each step stays in the state of ensuring negligible probability of overall security failure in a scalable network*” (Theorem 2). To prove this we need the following formal model.

2.2 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.

2.2.1 ASSUMPTIONS

We assume that all packet transmissions (including control, data packets and their ACKs) are protected by data origin authentication service. *Every packet is authenticated and the (uncompromised) packet sender’s identity is unforgeable*. Unauthenticated packets are dropped at the receivers immediately. This can be implemented by signing each packet by the sender’s certified digital signature or using efficient symmetric key protocols like TESLA [23][9]. Therefore, the adversary cannot forge packet transmissions from uncompromised nodes, and cannot launch Sybil attack [8] by faking uncompromised nodes’ identities. These cryptographic assumptions ensure that the adversary cannot forge uncompromised nodes’ identities and wireless traffic with non-negligible probability in regard to a pre-defined key length n.

In regard to jamming, we assume that anti-jamming technologies like spread spectrum can prevent the adversary from destroying all traffic in an area of constant size. The adversary’s transmission range is hence *not* a design parameter here (as introducing this parameter in this paper will significantly complicate the analysis). In regard to wormhole attack [10], we assume that the network is already protected by either packet leashes [10] or variants of Brands-Chaum protocol [5][27], which ensure that any pair of communicating neighbors in ad hoc routing are indeed physical neighbors.

The notions used in this paper are listed below:

N	network scale (number 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

2.2.2 APPLIED TURING MACHINES AND COMPLEXITY CLASSES

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. □

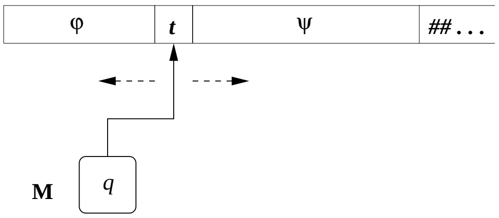

Figure 2: 1-tape Turing Machine M in configuration (q, φ, t, ψ)

Using 1-tape Turing Machine as an example, as depicted in Figure 2, 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 ψ 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 distributions.* □

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 x. Intuitively, the parameter x in cryptography is the key length n.

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

$$\epsilon(x) < \frac{1}{poly(x)}. \qquad \square$$

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

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

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, $\mathcal{RP}$ class):*** *We say that L is recognized by the probabilistic polynomial-time Turing Machine M with biased single-side errors if*

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

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

DEFINITION 5. ***(Bounded-error Probabilistic Polynomial-time, $\mathcal{BPP}$ class):*** *We say that L is recognized by the probabilistic polynomial-time Turing Machine M with biased double-side errors if*

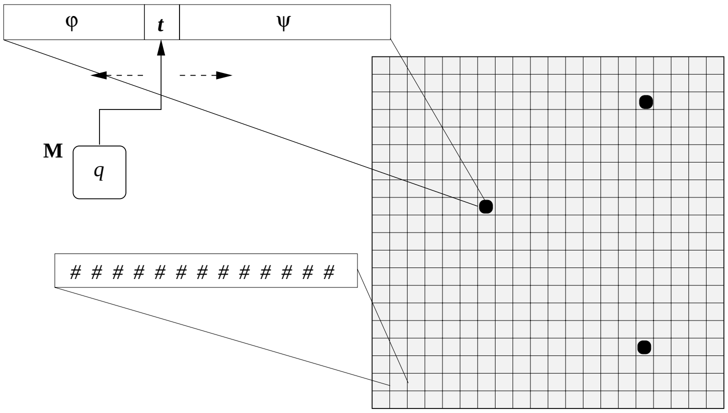

Figure 3: A $\mathcal{GVG}$ Probabilistic Turing Machine ($\mathcal{GVG}$ PTM) to model mobile nodes in a finite square area with a large number of node "positions". The figure shows that $N = 3$ of η ($N \ll \eta < O(poly(N))$) such "positions" have been taken by $N = 3$ mobile nodes. Each empty "position" 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 total 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). 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

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

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

DEFINITION 6. ***($\mathcal{RP}$ n-runs class):*** *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$[M\ accepts\ x] \geq 1 - \frac{1}{poly(n)}$ for every polynomial $poly(n)$.*
- *for every $x{\notin}L$ it holds that Pr$[M\ accepts\ x] = 0$.*

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

The procedure to obtain $\mathcal{RP}$ n-runs class is also called $\mathcal{RP}$ amplification. Similarly, $\mathcal{BPP}$ can also be amplified. In below, we will show that the probability of security failure decreases exponentially toward 0 when the corresponding network metrics increase linearly. This conforms to the $\mathcal{RP}$ n-runs class. 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 cryptography, the longer the key length is, the more asymptotically secure a cryptosystem is; In our model, the larger the network scale is, the more asymptotically secure the network is.

2.2.3 MODELING MOBILE NETWORKS: A PTM APPROACH WITH A $\mathcal{GVG}$ ORACLE

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* ($\mathcal{GVG}$) oracle to handle the PTM's control states, while each mobile node is only treated as a tape carrier.

As depicted in Figure 3, 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. We model that the $\mathcal{GVG}$ can make decisions for all mobile nodes and emulate all the decisions globally within the granularity τ (e.g., 1 nano-second).

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 Turing Machine 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 total 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 of $\mathcal{GVG}$ PTM We formally define $\mathcal{GVG}$ Probabilistic Turing Machine and $\mathcal{GVG}$ polynomial-time protocols in below.

DEFINITION 7. *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 a global virtual god 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 1-tape $\mathcal{GVG}$-PPTM, δ is*

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

while for k-tape $\mathcal{GVG}$-PPTM, 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.

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 $Pr[M\ accepts\ x] = 0$.*

$\mathcal{GVG} - \mathcal{RP}$ (n-runs) 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 $\mathcal{GVG} - \mathcal{RP}$[2], 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 $\mathcal{GVG} - \mathcal{BPP}$. Besides the difference in input to negligibility (key length n vs. network scale N), this $\mathcal{BPP}$ vs. $\mathcal{RP}$ difference is another one between cryptography and our $\mathcal{GVG}$ network-centric security model.

EXAMPLE 1. ***(Snapshot ad hoc routing)*** *In a* snapshot *of a mobile*[3] *network running AODV or DSR routing, nodes can be viewed as proxies 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 $2 \cdot N < O(poly(N))$ stops. When the corresponding RREP symbols come back to the source node, $\mathcal{GVG}$ enters a final acceptance state to finish*[4] *the on-demand 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. For a secure routing protocol in $\mathcal{GVG} - \mathcal{RP}$, the probability of route discovery success $Pr[RREP\ received\ by\ source]$ must be $1 - \epsilon(N)$, while the probability of route discovery failure $Pr[RREP\ not\ received\ by\ source]$ must be $\epsilon(N)$.*

However, as we illustrate later, 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 (Theorem 2).* □

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

[2] In this paper, $\mathcal{GVG}$-$\mathcal{RP}$ denotes $\mathcal{GVG}$-$\mathcal{RP}$ n-runs class for the ease of presentation, since $\mathcal{GVG}$-$\mathcal{RP}$ 1-run is impractical in mobile networks.

[3] 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 a node who forwarded RREQ previously can forward the coming back RREP.

[4] As demonstrated in [17], the source and the destination can then employ periodic *probes* to maintain the route during mobile data delivery. These proactive probes effectively replace the on-demand floods in mobile scenarios, but such optimizations are beyond the scope of this paper.

centralized infrastructure systems from the self-organizing infrastructureless networks studied here. In the former case, a centralized server can beat a polynomially increasing 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 one node doesn't crash the networked function.

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. 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.

Finally, any network member only virtually occupies a network area of sufficiently small size. *Exponential capability* in our negligibility-based model means the capability to overwhelm any *sub-area* with a constant size in the network. This is beyond the capability of capturing a subset of legitimate network members. To make the model work, a compromised sub-area of a constant size must be excluded from the network area. Then our analytic results are applied to the remaining network area, which is treated as the whole network area A. The mathematical reasons are described below.

2.2.4 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 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* [7] 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 ρ_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

$$\mathit{Pr}[x = k] \; = \; \frac{(N \cdot \rho_1 \cdot \mathcal{A}')^k}{k!} \cdot e^{-N \cdot \rho_1 \cdot \mathcal{A}'}. \qquad (1)$$

- (*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

$$\mathit{Pr}[x = k] \; = \; \iint_{\mathcal{A}'} \left(\frac{(N \cdot \rho_1)^k}{k!} \cdot e^{-N \cdot \rho_1} \right) \mathrm{d}\mathcal{A}. \qquad (2)$$

Discussion on mobility PDF ρ_1: Our study is based on the mobility probability distribution function 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

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

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

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

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 [4], we can use the analytical expression $\rho_1 \approx \frac{36}{a^6} \left(x^2 - \frac{a^2}{4} \right) \left(y^2 - \frac{a^2}{4} \right)$ 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.

2.3 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 per-hop forwarding success* is negligible, thus the *probability of multi-hop routing success* 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 attackers and $(1 - \theta) \cdot N$ honest members. The random variable y denotes the number of honest nodes in an arbitrary area $\mathcal{A}'$. The probability that there are k honest nodes in the area $\mathcal{A}'$ is

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

Let z denote the random variable of number of dishonest attackers in the same area $\mathcal{A}'$. The probability that there are k dishonest attackers in the area $\mathcal{A}'$ is

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

In a regular on-demand routing scheme, the per-hop RREP forwarding success ratio, namely the per-hop route discovery success

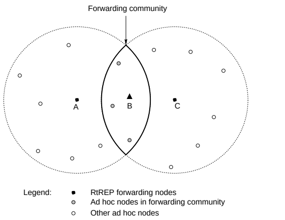

Figure 4: A forwarding community between a 2-hop source and destination pair

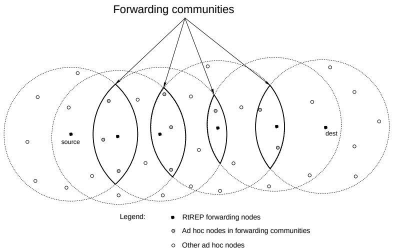

Figure 5: Forwarding communities along a multi-hop path

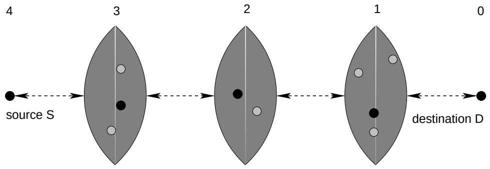

Figure 6: Forwarding communities as "big" virtual areas

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

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

where $\mathcal{A}'$ 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.

The mobility PDF ρ is arbitrary in our study as long as it is continuous in the area A, 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 or differentials 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 orders $O(e^N)$ and polynomial orders $O(poly(N))$ are unchanged in magnitude through these integrals or differentials. And this concludes that the last step $= \epsilon(N)$ holds. Rushing attack is a severe routing attack that can reduce the success ratio of regular on-demand routing schemes to negligible.

2.4 Negligibility-based Network Security

In order to ensure negligibility-based secure routing, at first we must ensure that security failure probability is negligible (with respect to network scale N) per hop. One candidate solution is "localized coordination community" 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 4 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 5 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 6).

$\mathcal{GVG}$ **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. In a secure routing scheme that ideally implements forwarding communities, the per hop/step probability of failure is*

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

As previously shown, the mobility PDF ρ is arbitrary in our study as long as it is continuous in a network area A'. Exponential orders $O(e^N)$ and polynomial orders $O(poly(N))$ are unchanged in magnitude through integrals or differentials. This concludes that last step holds and the probability of security failure per hop/step $P_{stepfail}$ is negligible with respect to the network scale N. □

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 given $\frac{1}{p(N)\cdot q(N)}$, where $q(N)$ is a positive polynomial and so $p(N)\cdot q(N)$ is also a positive polynomial. In other words, there exists a positive integer $N_c > 0$, such that $P_{stepfail} < \frac{1}{p(N)\cdot q(N)}$ for all $x > N_c$. Then we have

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

since $(1-\frac{1}{x})^x > e^{-1}$ 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,

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

Therefore, by Definition 7, such a protocol implementation with *ideally* implemented forwarding communities is in $\mathcal{GVG} - \mathcal{RP}$.

Discussion: from ideal to practical Although Theorem 2 proves that an asymptotic security invariant based on negligibility does exist in ad hoc networks, it is an open challenge to realize a practical implementation. We believe that *data origin authentication*, *secure neighbor detection*, *distance bounding*, *anti-jamming* and *short-term mobility constraints* are prerequisites, though the list is not meant to be complete. (1) All packet transmissions (including control, data packets and their ACKs) are protected by data origin authentication service. For those uncompromised 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 [23][9]. Therefore, the adversary cannot forge packet transmissions from uncompromised nodes, and cannot launch Sybil attack [8] by faking uncompromised honest nodes' identities; (2) Secure neighbor detection protocols [11][20] 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 at link layer 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 [5]. 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. (5) This paper relies on a stochastic model to characterize long-term mobility patterns. In particular for short-term cases in on-demand routing, the mobility speed must be constrained in geometry so that nodes approximately stay in the same region when the needed RREP comes back. A practical approximation of forwarding communities is studied in "community-based security" [17], but it does not implement the ideal model because all the four pre-conditions have not been studied in [17].

3. 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 [25] or symmetric key based protocols (e.g., TESLA [23])[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 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. [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 a honest member. Our forwarding community design adopts a different approach. We implement a faster RREQ phase, then in the RREP phase we explore the presence of honest network members to heal a damaged route on the fly. Such greedy coordination 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 [10] and secure distance bounding [27][5].

Multi-path routing [24][19] and route fix using local recovery query [26] 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. [3] 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 routing protocol.

The above schemes are not based on formal models. Recently Ács, Buttyán and Vajda [2] have accomplished the earliest work to apply probabilistic complexity theory in ad hoc routing research. Nevertheless, [2]'s approach is based on the $\mathcal{BPP}$ class which measures indistinguishability between truly randomness and cryptographically strong pseudorandomness with respect to key length n. In contrast, our model is based on the $\mathcal{RP}$ class which measures success probability of a protocol execution with respect to network scale N. These are two different components of the same problem. Although both of them are probabilistic polynomial-time algorithms, $\mathcal{BPP}$ indistinguishability is computation-centric against computational cryptanalysis while our $\mathcal{GVG}-\mathcal{RP}$ protocol execution is network-centric against malicious node behaviors.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have formally specified network security following a probabilistic complexity theoretic approach. We study the behavior of rushing attack, which is a bottomline security threat to deplete network resource and reduce the routing performance to minimum. Like formal cryptography, we apply the same asymptotic bounds defined by "negligibility" in ad hoc network security research, but this time the input parameter to negligibility is not in regard to key length n but rather to network scale N. We propose an $\mathcal{RP}$ (n-runs) complexity class with a global virtual god oracle ($\mathcal{GVG}$) to model a general class of network protocols. In $\mathcal{GVG}$-$\mathcal{RP}$ (n-runs) class, the network scale (i.e., number of network members) N replaces the role of key length n in cryptography. Using the formal model, we show that rushing attack can unfortunately drive the success probability of an unprotected ad hoc routing protocol to negligible. This means the bottomline rushing attack is a severe security threat.

Fortunately, a security invariant does exist in $\mathcal{GVG}$-$\mathcal{RP}$. By ensuring negligible probability of security failure at per-hop (per-step), the overall probability of security failure of polynomial-many hops/steps stays as negligible in a sufficiently-large network. In ideal scenarios, such security guarantee is what can be achieved by provably secure algorithms in cryptography in regard to sufficiently-large key length, and is also what can be achieved in network security in regard to sufficiently large network size. Nevertheless, like the case we do not know whether ideal one-way functions do exist, it is unknown that this ideal model can be realized in wireless network. We listed a few pre-conditions to implement the ideal model, but the list is not proved to be complete and the current empirical implementations have not fully realized the pre-conditions. The purpose of this paper is to formally connect network security with network scalability, but *not* designing practical protocols to realize the ideal model. Although our efforts have shown that practical protocols[17] [18][15] can be devised to approximate the formal approach, they are unfortunately not the ideal implementation. This unsolved puzzle will be further inspected in the future work.

Acknowledgement An early version of this paper is an IACR ePrint Report [14]. The authors thank all anonymous reviewers for their helpful comments to convert it into this paper.

5. 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] G. Ács, L. Buttyán, and I. Vajda. Provable Security of On-Demand Distance Vector Routing in Wireless Ad Hoc Networks. In *European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS)*, 2005.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

[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] J. Kong. GVG-RP: A Net-centric Negligibility-based Security Model for Self-organizing Networks. Technical Report 2006/140, IACR Cryptology ePrint Archive, April 2006.

[15] J. Kong and X. Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. In *ACM MOBIHOC'03*, pages 291–302, 2003.

[16] 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.

[17] 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.

[18] J. Kong, D. Wu, X. Hong, and M. Gerla. Mobile Traffic Sensor Network versus Motion-MIX: Tracing and Protecting Mobile Wireless Nodes. In *ACM SASN'05 in conjunction with CCS'06*, pages 97–106, 2005.

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

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

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

[25] 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.

[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.