This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

Modeling Secure Connectivity of Self-Organized Wireless Ad Hoc Networks

Chi Zhang, Yang Song and Yuguang Fang
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611
Email: {zhangchi@, yangsong@, fang@ece.}ufl.edu

Abstract—**Wireless ad hoc networks (WANETs) offer communications over a shared wireless channel without any pre-existing infrastructure. Forming peer-to-peer security associations in self-organized WANETs is more challenging than in conventional networks due to the lack of central authorities. In this paper, we propose a generic model to evaluate the relationship of connectivity, memory size, communication overhead and security in fully self-organized WANETs. Based on some reasonable assumptions on node deployment and mobility, we show that when the average number of authenticated neighbors of each node is $\Theta(1)$, with respect to the network size n, most of the nodes can be securely connected, forming a connected secure backbone, i.e., the secure network percolates. This connected secure backbone can be utilized to break routing-security dependency loop, and provide enough derived secure links connecting isolated nodes with the secure backbone in a multi-hop fashion, which leads to the secure connectivity of the whole network.**

I. Introduction

By definition, a *wireless ad hoc network* (WANET) (mobile or stationary) is a group of wireless nodes that cooperatively form a network which operates without the support of any pre-established or centralized network management infrastructure [1]–[3]. In the literature, there are two extreme ways to introduce security in WANETs: (1) through a single authority domain, where certificates and/or keys are issued by a single authority, typically, in the system setup phase [4]–[9], or (2) through full self-organization, where security does not rely on any trusted authority or fixed server, not even in the initialization phase [1]–[3], [10]. In this paper, we follow the second approach. Our main motivation comes from observations on some WANET scenarios, which require self-organized network management, e.g.,

- when the WANET merges and partitions in a sporadic way and the network cannot be pre-planned;
- when each user has individual interests and thus is its own authority domain where the full control of the security settings of its own nodes is desired;
- when users prefer to join and leave the network at random without contacting any remote trusted authority;
- when the number of users gets large and thus the key servers, if any, will become the bottleneck or central points of failure.

All these situations highlight the need of a security architecture with *self-organization* property, i.e., the ability of nodes to establish security associations among themselves after network formation without the aid of any form of on-line or off-line *trusted third party* (TTP) [1]. Although not all WANETs are required to be fully self-organized, this property is considered as one of the final objectives in most of the ongoing research projects, such as *Terminodes*, *Spontnet* and *zero-configuration networks* [1]–[3], [10].

Impromptu, self-organized WANETs can be informally visualized as a group of wireless communication devices (called *WANET nodes* in this paper), held by people without any pre-planning, coming together to form a network for a common purpose (e.g., emergency response). Some keying materials for primary security associations (SAs), which we will formally define later, are already pre-configured in communication devices, based on the trust relationships between the people involved. The problem is how to exploit those primary security associations to provide secure communication for arbitrary node pairs when needed.

Neighbor authentication, which provides hop-by-hop security, is the first step for secure communications in all kinds of networks. This is especially crucial for WANETs since every node need to act as the router to forward packets for others. If the node cannot authenticate its physical neighbors, how can it trust all those physical neighbors to handle its packet correctly? Obviously, neighboring nodes with primary security associations can authenticate each other directly with pre-configured keying materials. Since the number and the distribution of primary security associations are determined by the embedded social network (e.g., trust relations) of users, a node may not have primary security associations with any of its physical neighbors. In this case, a *Neighbor Authentication Protocol* (NAP) is required to set up derived security associations with its neighbors on the need basis with the help of already authenticated neighbors.

Although some examples of fully self-organized security architecture for WANETs have been discussed in the literature [1]–[3], [10], many theoretical problems are still remaining: e.g., what is the minimum fraction of primary SAs for securing all the links? What is the communication overhead for NAPs to provide derived SAs? How do the characteristics of the trust graph of users affect the performance of NAPs and the security of WANETs? Is this kind of self-organized

This work was supported in part by the U.S. National Science Foundation under grant CNS-0721744, CNS-0716450 and CNS-0626881.

978-1-4244-2026-1/08/$25.00 © 2008 IEEE

WANETs scalable in terms of the required memory size for keying material in each node or the communication overhead for the NEP, when the network size becomes arbitrarily large?

In this paper, we propose a generic model to evaluate the relations of connectivity, memory size, communication overhead and security in fully self-organized WANETs. Based on some reasonable assumptions on node deployment and mobility, we show that when the average number of authenticated neighbors of each node is $\Theta(1)$, with respect to the network size n, most of the nodes can be securely connected, forming a connected secure backbone, i.e., the secure network percolates. This connected secure backbone can be utilized to break routing-security dependency loop (cf. Section II-B), and provide enough derived secure links connecting isolated nodes with the secure backbone in a multi-hop fashion, which leads to the secure connectivity of whole network.

II. Network Model and Problem Formulation

A. Network Model

1) Physical Graph $G(\mathcal{X}_n, \mathcal{E}_{pl})$: Let $\mathcal{X}_n = \{X_1, X_2, \cdots, X_n\}$ denote the node set of a WANET, with network size $|\mathcal{X}_n| = n$. By slight abuse of notation, we let X_i denote the location, as well as the identity, of a node. The n nodes are distributed uniformly at random on the given geographical area, which is, without loss of generality, a unit disk $\mathcal{D}$ centered at the origin. Two nodes X_i and X_j have a physical wireless link (X_i, X_j) if their Euclidean distance is no greater than r_n, the *communication range*, and X_i and X_j are called *physical neighbor* with respect to each other. Let $\mathcal{E}_{pl}$ denote the physical link set. The graph $G(\mathcal{X}_n, \mathcal{E}_{link})$ with vertex set $\mathcal{X}_n$ and edge set $\mathcal{E}_{link}$ is termed *random geometric graph* (RGG) [11], which is widely used in the literature to model the *physical graph* of a WANET. A *physical path* between two nodes, e.g., X_1 and X_k, is a set of consequential edges in $\mathcal{E}_{pl}$. Two nodes are said to be *physically connected* if there exists a physical path that starts at one and ends at the other. Graph $G(\mathcal{X}_n, \mathcal{E}_{pl})$ is said to be *physically connected* if every pair of nodes in the graph is physically connected. In this paper, we are mainly concerned with events that occur inside the unit disk $\mathcal{D}$ with high probability (*w.h.p.*); that is, with probability tending to one as $n \to \infty$.

For a mobile WANET, we assume that nodes move independently in the unit disk $\mathcal{D}$ according to a *Brownian motion model* (BMM) in [12]. We assume that the initial positions of nodes are *i.i.d.* and uniformly distributed in the disk $\mathcal{D}$. This implies that the positions of the nodes will remain uniform at all times under the BMM. Since we are only interested in the statistical properties of $G(\mathcal{X}_n, \mathcal{E}_{pl})$ and other related graph models, based on the BMM, we can omit the time dimension and treat the mobile WANET as static in a given time point, which will greatly simplify our analysis. Note that the results of this paper also apply to other related mobility models such as the random walk mobility model [13] and the Markovian mobility model [14]. This is because the Brownian motion model can be viewed as a limiting case of these other mobility models [12].

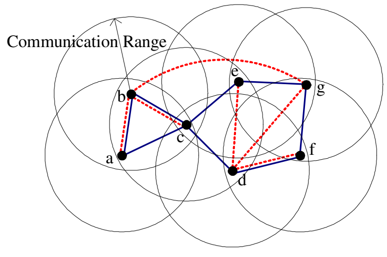

Fig. 1. **An exemplary WANET with primary SAs.** Here solid lines and dashed lines represent physical links and primary SAs, respectively.

2) Trust Graph $G(\mathcal{X}_n, \mathcal{E}_{SA})$: When we say two nodes have primary *security association* (SA), we mean that two nodes trust each other and either a symmetric key are shared between them, or that the nodes know each others' authentic public keys. Two nodes may have exchanged their keys through a side channel (e.g., over an infrared channel at the time of a physical encounter, or just manually set up those keys). We further assume that security associations are always symmetric. In the beginning, this assumption may seem conflicting with the asymmetric nature of certificates on public keys. Due to the fact that node X_i knows the authentic public key of node X_j does not necessarily imply that node X_j also holds the authentic public key of node X_i, and vice verse, relations in *certificate graph* [1], [3], [14] are asymmetric. However, first of all, in practice the statistical analysis of the "Web of Trust" among users of *Pretty Good Privacy* (PGP) [15], the market leader in the world of secure email communications, shows that about $2/3$ of the links in the large strongly connected component are bidirectional [16]. Second, in our scheme, we require two nodes to bidirectionally exchange their keying materials in order to successfully establish a primary SA. Third, besides the requirement of possessing correct keying material, we also require that two nodes trust each other, e.g., they rely on each other to handle their packets, which is the reason that this kind of trust must be bidirectional. Primary SA can be deemed as a *logical link* connecting two nodes in the WANET. We name two nodes with a primary SA as *friends* and denote the set of primary SAs as $\mathcal{E}_{SA}$. The graph $G(\mathcal{X}_n, \mathcal{E}_{SA})$ with vertex set $\mathcal{X}_n$ and edge set $\mathcal{E}_{SA}$ is termed the *trust graph*, and is used to model the trust relationships between users in the WANET.

The characteristics of trust graphs created in WANETs will depend on the existing social relationships between users. In what follows, we just introduce one simplified parameter p_f to quantify the existence of SAs in the WANET.

We consider a homogeneous trust graph model, i.e., the number of friends of each node is on the same order of n. Based on the *i.i.d.* and uniform distribution of initial positions of the nodes and the BMM, whenever two nodes meet as physical neighbors, they will be friends (have a SA) with almost the same probability, denoted as p_f.

3) Secure Graph $G(\mathcal{X}_n, \mathcal{E}_{sl})$ or $G(\mathcal{X}_n, g_{p_f})$: Obviously, when one of X_i's friends, e.g., node X_j, becomes the physical neighbor of X_i, then nodes X_i and X_j can directly authenticate each other. We call X_i the *neighboring friend* of X_j, and there exists a *secure physical link* through wireless com-

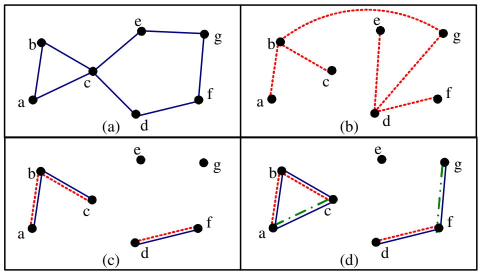

Fig. 2. **Graph models constructed from the original wireless ad hoc network in Fig. 1.** (a) Physical graph; (b) trust graph; (c) secure graph; (d) local augmented secure graph.

munication between X_i and X_j. We denote the set of secure physical links as $\mathcal{E}_{sl}$, and define the graph with vertex set $\mathcal{X}_n$ and edge set $\mathcal{E}_{sl}$ as the *secure graph*. Since $(X_i, X_j) \in \mathcal{E}_{sl}$ *iff* $(X_i, X_j) \in \mathcal{E}_{pl}$ and $(X_i, X_j) \in \mathcal{E}_{SA}$, $G(\mathcal{X}_n, \mathcal{E}_{sl})$ is the coupled graph of $G(\mathcal{X}_n, \mathcal{E}_{pl})$ and $G(\mathcal{X}_n, \mathcal{E}_{SA})$.

Note that some links in set $\mathcal{E}_{pl} \setminus \mathcal{E}_{sl}$ can also be secured with the assistance of local authenticated neighbors. We name all those links *locally augmented secure links*, denoted as $\mathcal{E}_{lasl}$. The graph $G(\mathcal{X}_n, \mathcal{E}'_{sl})$ with $\mathcal{E}'_{sl} = \mathcal{E}_{sl} \cup \mathcal{E}_{lasl}$ is termed *locally augmented secure graph*. The relationship between $G(\mathcal{X}_n, \mathcal{E}_{sl})$ and $G(\mathcal{X}_n, \mathcal{E}'_{sl})$, and the techniques to locally augment secure links will be further investigated in Section III-C. We can define the *secure connectivity* in $G(\mathcal{X}_n, \mathcal{E}_{sl})$ (or $G(\mathcal{X}_n, \mathcal{E}'_{sl})$) in a way similar to physical connectivity in $G(\mathcal{X}_n, \mathcal{E}_{pl})$, i.e., there exists a *secure path* with links in $\mathcal{E}_{sl}$ (or $\mathcal{E}'_{sl}$) for arbitrary node pairs in $\mathcal{X}_n \times \mathcal{X}_n$.

Here we give a generalized formulation to describe the graph models introduced above. Given $\mathcal{X}_n$ as the vertex set, we connect every two nodes X_i and X_j $(i \neq j)$ with probability $g(X_i - X_j)$. The resulting graph is denoted as $G(\mathcal{X}_n, g)$. We term the connected components of $G(\mathcal{X}_n, g)$ as *clusters*, and the number of nodes in each cluster as its *order*. The cluster with the maximal order is called the *giant cluster*. The graph $G(\mathcal{X}_n, g)$ is *connected iff* there exists only one cluster. Therefore, $G(\mathcal{X}_n, \mathcal{E}_{pl})$ can also be denoted as $G(\mathcal{X}_n, g_1)$, where

$$g_1(x) = \begin{cases} 1 & \text{if } |x| \leq r_n, \\ 0 & \text{if } |x| > r_n. \end{cases} \tag{1}$$

Given p_f, $G(\mathcal{X}_n, \mathcal{E}_{sl})$ can also be denoted as $G(\mathcal{X}_n, g_{p_f})$, where

$$g_p(x) = \begin{cases} p_f & \text{if } |x| \leq r_n, \\ 0 & \text{if } |x| > r_n. \end{cases} \tag{2}$$

Based on the property of the Poisson process, the number of physical neighbors and the number of neighboring friends for each node in $\mathcal{X}_n$ are random variables following Poisson-distribution with expected values of $n\pi r_n^2$ and $p_f \cdot n\pi r_n^2$, respectively (for simplicity, here we ignore the border effect, cf. the Appendix in [17]). Using the term of graph theory, these two values are also called the *average node degree* for $G(\mathcal{X}_n, g_1)$ and $G(\mathcal{X}_n, g_{p_f})$, respectively.

B. Neighbor Authentication and Pairwise Key Establishment

For self-organized WANETs, neighbor authentication consists of three phases, namely neighboring-friend discovery, local secure-link and multi-hop secure-link augmentation.

The *neighboring-friend discovery* phase takes place when a node discovers that a new neighboring node appears in its one-hop neighborhood. The simplest way for any two nodes to discover if they are friends is that each node encodes its node ID and relevant identification information in clear text into its beacon message (also called HELLO message) which is periodically broadcast to its physical neighbors for neighbor discovery and channel usage coordination in MAC layer. Alternate methods exist which hide identity information from an adversary thereby establishing private neighboring-friend discovery. When one node finds another as its friend, they mutually authenticate the identity of the other party as it claims, using any challenge-response protocols. When two neighboring nodes are mutually authenticated, a secure link is established, and their security association is *realized*, or *activated*. After the first phase, each node X_i knows all its physical neighbors and neighboring friends, which are denoted by sets $N(X_i)$ and $NF(X_i)$, respectively. The sets authenticated and unauthenticated neighbors of node X_i are represented by $NA(X_i)$ and $UA(X_i)$, respectively. After the first phase, $NA(X_i) = NF(X_i)$, $UA(X_i) = N(X_i) \setminus NF(X_i)$ and the nodes with secure links are modeled by the trust graph $G(\mathcal{X}_n, \mathcal{E}_{sl})$.

Due to the limitation of the trust graph, nodes can only share primary SAs with a subset of the neighboring nodes. Therefore, the *local secure-link augmentation* phase is needed to establish derived SAs with the remaining neighbors with the help of authenticated neighbors. The techniques and quantitative analysis of the second phase will be detailed in Section III-C. Here, we just emphasize that in this phase, each user only makes use of its one-hop neighbors' information, which can be collected in the MAC layer. Once a neighbor in set $UA(X_i)$ is authenticated and the pairwise key is established in the second phase, it is included in set $NA(X_i)$ and deleted from set $UA(X_i)$, and the corresponding locally augmented secure links will be added in the set $\mathcal{E}_{lasl}$. After this phase, nodes with secure links are modeled by the graph $G(\mathcal{X}_n, \mathcal{E}'_{sl})$, where $\mathcal{E}'_{sl} = \mathcal{E}_{sl} \cup \mathcal{E}_{lasl}$.

Note that the key difference between WANETs and other distributed systems is that, each node in the WANET is required to act as a router to forward packets for other nodes. This unique feature introduces the well-known *routing-security dependency loop* [18], [19]: acyclic dependency arises between security services and routing services since multi-hop security services require routing layer security themselves. This loop implies two consequences especially for node authentication. On the one hand, the primary SA between two remote nodes cannot be utilized if there is no secure path between them. Even though the packets can be end-to-end encrypted or authenticated, they are at risk to be sent through false routes, or simply dropped without the secure path (routing layer security). On the other hand, derived SAs between neighboring nodes cannot be established over multiple hops if the routing protocol does not operate securely. Here, we rely on the first two phases to break this loop, since they only need the information collected on the MAC layer and are independent of both secure routing and other security services. Upon completion of these two phases, some multi-hop secure paths with hop-by-hop security will emerge, based on which secure routing can be implemented. In general, there still exists unauthenticated physical neighbors after the second phase. In the following *multi-hop secure-link augmentation* phase, multi-hop secure paths will be used to authenticate and establish pairwise keys with the remaining neighbors in set $UA(X_i)$. The criterion that the neighbor authentication completes depends on the application requirement of a particular WANET. For example, if the main purpose of a WANET is to facilitate the cooperation between the neighboring nodes/users, the main traffic patten will be local communications where a node needs to authenticate as many neighbors as possible, or stops until the fraction of authenticated neighbors $\frac{|NA(X_i)|}{|N(X_i)|}$ is greater than a system parameter, say c. If the main purpose of a WANET is to provide an infrastructure to support communications with remote nodes, the main traffic pattern will be multi-hop communications. If every source node is able to find a secure path to the target destination, no augmented secure links are required. The authentication with the remaining neighbors can be activated in an on-demand fashion, i.e., to authenticate each other only when they need to communicate. Once a neighbor in set $UA(X_i)$ is authenticated and the pairwise key is established, this neighbor will be included in set $NA(X_i)$ and deleted from set $UA(X_i)$, and the corresponding *globally augmented secure links* will be added in the set $\mathcal{E}_{gasl}$. At the end of this phase, nodes with secure links are modeled by the graph $G(\mathcal{X}_n, \mathcal{E}''_{sl})$, where $\mathcal{E}''_{sl} = \mathcal{E}'_{sl} \cup \mathcal{E}_{gasl}$.

C. Problem Formulation

Now, we can formally define the objective of secure WANETs as follows:

Objective: *Constructing a secure path between an arbitrary pair of nodes in $\mathcal{X}_n \times \mathcal{X}_n$ w.h.p.*

Constraints: (i) *Physical graph $G(\mathcal{X}_n, \mathcal{E}_{pl})$ is connected*, and (ii) *trust graph $G(\mathcal{X}_n, \mathcal{E}_{SA})$ is connected.*

Mathematically, the objective is equivalent to the situation that the globally augmented secure graph $G(\mathcal{X}_n, \mathcal{E}''_{sl})$ is connected *w.h.p.* Here, we need to explain these two constraints in detail. In fact, these two constraints are the necessary conditions to achieve the objective. First of all, when the physical graph is disconnected, it is impossible to provide multi-hop communications for disconnected nodes with traditional routing schemes (reactive or proactive) designed for WANETs. Although the mobility of nodes can be utilized, and some routing schemes designed for *delay tolerant networks* (DTNs) can be used to provide communications between some disconnected nodes, no protocols can guarantee the delivery of packets between them so far even in the *w.h.p.* sense. However, it is interesting to examine in what degree this constraint can be relaxed by adopting the DTN routing protocols, which will be our future work. Second, a disconnected trust graph implies that users can be divided into several isolated trusted groups, and there is no trust relationship between those groups. In this situation, the objective of constructing secure path between two nodes that belong to different trust groups is meaningless. Therefore, when we mention the objective of providing a secure path between an arbitrary pair of nodes, we implicitly include these two constraints. Previous research on Erdös-Rényi random graph [20], [21] and random geometric graph [11], [17] show that the necessary conditions for the connectivity of the trust graph and physical graph are $p_f = \Omega(\frac{\log n}{n})$ and $r_n = \Omega(\sqrt{\frac{\log n}{n}})$, respectively.

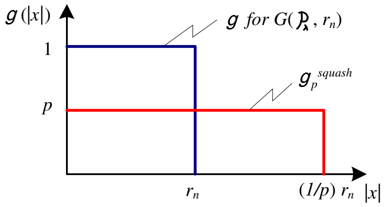

Fig. 3. **Squishing and Squashing.** The function g is squished and squashed to give the function g_p^{squash}.

Under this objective, we list the problems that need to be addressed as follows: (1) What are the requirements on the physical graph $G(\mathcal{X}_n, \mathcal{E}_{pl})$ and trust graph $G(\mathcal{X}_n, \mathcal{E}_{SA})$ so that it is possible for $G(\mathcal{X}_n, \mathcal{E}''_{sl})$ to be connected? Or more precisely, what should r_n and p_f be? (2) What is the communication overhead incurred due to the neighbor authentication? (3) What is the effect of the secure link distribution on the performance of secure routing?

III. Properties of Secure Graphs

A. Background of Continuum Percolation

In this section, we make use of the results from *continuum percolation*, which was first introduced by Gilbert [22] and later analyzed by Meester and Roy [23], Penrose [11], [24].

In continuum percolation theory, nodes are assumed to be distributed as a homogeneous Poisson process $\mathcal{P}_\lambda$ on $\mathbb{R}^2$ with density λ. However, our original network model concerning a fixed number of nodes $\mathcal{X}_n$ randomly and uniformly distributed in the unit area disc $\mathcal{D}$. In most of the proofs in this paper, it is useful to first consider, instead of $\mathcal{X}_n$, a coupled Poisson process $\mathcal{P}_\lambda$ with density λ close to n, then deduce the results of $\mathcal{X}_n$ from the results of $\mathcal{P}_\lambda$, using the so-called *Poissonization technique* [11, Chapter 1.7]. We first define the network model based on node distribution $\mathcal{P}_\lambda$. Let $\{X_1, X_2, X_3, \cdots\}$ be the set of nodes according to $\mathcal{P}_\lambda$ in $\mathbb{R}^2$, so that the expected number of nodes in any region is equal to the area of the region multiplied by λ. Let $G(\mathcal{P}_\lambda, g)$ denote the following random geometric graph: we connect any two points X_i and X_j $(i \neq j)$ in $\mathcal{P}_\lambda$ with probability $g(X_i - X_j)$, independent of any other pairs of nodes. We term the connected components of $G(\mathcal{P}_\lambda, g)$ as *clusters*, and the number of nodes in each cluster as its *order*. We first show the existence of a non-trivial critical density at which *percolation* occurs (that is, an infinite-order cluster forms) in $G(\mathcal{P}_\lambda, g)$.

Lemma 1: **[Percolation Properties of $G(\mathcal{P}_\lambda, g)$]** Consider a graph $G(\mathcal{P}_\lambda, g)$ for some given measurable function g :

$\mathbb{R}^2 \to [0,1]$, satisfying:

$$g(x) = g(|x|), x \in \mathbb{R}^2, \tag{3}$$

$$0 < \int_{\mathbb{R}^2} g(x)dx < \infty, \tag{4}$$

we have the following results:

(1) There exist a *critical value* (continuum percolation threshold) λ_p such that

$$0 < \lambda_p(g) = \inf\{\lambda : \exists \text{ infinite-order cluster } \textit{w.h.p.}\} < \infty.$$

When $\lambda > \lambda_p$ we say that the $G(\mathcal{P}_\lambda, g)$ *percolates* [24, Theorem 1].

(2) If $g(x) \leq g(y)$ whenever $|x| \geq |y|$, for a $G(\mathcal{P}_\lambda, g)$, there is at most one infinite-order cluster *w.h.p.* [23, Theorem 6.3, pp.172].

(3) Given a measurable function g and $0 < p < 1$, define the *squashed function* g_p^{squash} of g as $g_p^{squash}(x) = p \cdot g(\sqrt{p}x)$, we have

$$\lambda_p(g) \geq \lambda_p(g_p^{squash}), \tag{5}$$

see Fig. 3 for an example [25, Theorem 2.1].

(4) When $g(x) = g_1(x)$ as defined in (1), we have a special RGG denoted by $G(\mathcal{P}_\lambda, g_1)$. The exact value λ_p for this case is not known. Simulation studies indicate that $\lambda_p \approx \frac{4.5}{\pi r_n^2}$ [11, pp.189] while rigorous bounds $\frac{2.187}{\pi r_n^2} \leq \lambda_p \leq \frac{10.593}{\pi r_n^2}$ are given in Meester and Roy [23, Chapter 3.9]. A recent result of Balister and et al. [26] shows that with 99.99% confidence, the critical value λ_p lies between $\frac{4.508}{\pi r_n^2}$ and $\frac{4.515}{\pi r_n^2}$, i.e.,

$$4.508 \leq \pi r_n^2 \lambda_p \leq 4.515. \tag{6}$$

Gupta and Kumar [17] and Penrose [11] show that, $G(\mathcal{P}_\lambda, g_1)$ is equivalent to $G(\mathcal{X}_n, g_1)$ when $\lambda = n$. Therefore, in this paper we assume $\pi r_n^2 \lambda_p = 4.5$ for $G(\mathcal{P}_\lambda, g_1)$ or $G(\mathcal{X}_n, g_1)$.

We place a node X_0 at the origin. Then the resulting Poisson point process $\mathcal{P}_\lambda \cup X_0$ is "conditioned to have a point at 0, in the sense of Palm measures" and X_0 is assumed to be an "arbitrary point of the Poisson process" [24]. Let $C(0)$ be the "cluster at the origin", the set of nodes having a path to X_0 in $G(\mathcal{P}_\lambda \cup X_0, g)$. Let $p_k(\lambda)$ denote the probability that $C(0)$ has k points. The *percolation probability*, i.e., $p_\infty(\lambda)$, is the probability that 0 lies in an infinite cluster when $\lambda \to \infty$.

We have the following result:

Lemma 2: **[Isolated Nodes in $G(\mathcal{P}_\lambda \cup X_0, g)$]** Assume a measurable function g satisfying equation (3) and inequality (4). If g also encloses zero (essentially, g is symmetric, has bounded support, and is bounded away from zero in some open neighborhood of the surface, see [24]; note that all the functions g considered in this paper encloses zero), then

$$\lim_{\lambda \to \infty} \frac{\sum_{k=1}^{\infty} p_k(\lambda)}{p_1(\lambda)} = 1, \tag{7}$$

which implies that for large λ, the origin lies in either an infinite-order cluster or an order-one cluster (i.e., it is isolated) *w.h.p.* [24, Theorem 3].

B. Theoretical Results on the Secure Graph

Based on Lemma 2 and Lemma 1(2), the following lemma can be easily derived.

Lemma 3: **[Isolation and Connectivity of $G(\mathcal{P}_\lambda \cup X_0, g)$]** Assume a measurable function g satisfying equation (3) and inequality (4), g also encloses zero, and $g(x) \leq g(y)$ whenever $|x| \geq |y|$. As $\lambda \to \infty$, the probability that the graph $G(\mathcal{P}_\lambda \cup X_0, g)$ is connected is asymptotically equal to the probability that the graph $G(\mathcal{P}_\lambda \cup X_0, g)$ has no isolated nodes, i.e.,

$$\begin{aligned} & \lim_{\lambda \to \infty} \mathbf{Pr}[G(\mathcal{P}_\lambda \cup X_0, g) \text{ is connected}] \\ = \; & \lim_{\lambda \to \infty} \mathbf{Pr}[G(\mathcal{P}_\lambda \cup X_0, g) \text{ has no isolated nodes}]. \end{aligned}$$

Now we can investigate the connectivity property of the secure graph $G(\mathcal{X}_n, g_p)$.

Theorem 1: **[Connectivity of $G(\mathcal{X}_n, g_p)$]** Assume that p is any constant in $[0,1]$. Let $p \cdot n \cdot \pi r_n^2 = \log(n) + c(n)$. Then, the graph $G(\mathcal{X}_n, g_p)$ is connected *w.h.p. iff* $c(n) \to \infty$ and is disconnected *w.h.p. iff* $c(n) \to -\infty$.

Proof: This result can be proved from Lemma 2 and 3 using Poissonization technique. The details of the proof has been omitted due to space constraints. ■

Note that in Theorem 1, if p is the probability that two neighboring nodes can establish the SA, then $k = p \cdot d = p \cdot n\pi r_n^2$ denotes the average number of authenticated neighbors. Theorem 1 shows that for $G(\mathcal{X}_n, g_p)$, there exists a *phase transition phenomenon*, i.e., there is a critical threshold $k_c = \log(n)$ for k, corresponding to a minimum number of authenticated neighbors for individual nodes, above which a desirable global property (e.g., connectivity) exists with high probability. When k is below the threshold k_c, the desired global property exists with a low probability. This phase transition is typically seen to become sharper as the number of nodes n in the network increases. The following theorem will reveal another phase transition phenomenon for $G(\mathcal{X}_n, g_p)$.

Theorem 2: **[Percolation Properties of $G(\mathcal{X}_n, g_p)$]** Define the critical value (percolation threshold) of the average node degree for $G(\mathcal{X}_n, g_p)$ as

$$\begin{aligned} k_p \;=\; & \inf\{k : \exists \text{ infinite-order cluster } \textit{w.h.p.} \\ & \text{in } G(\mathcal{X}_n, g_p) \text{ as } n \to \infty\}. \end{aligned}$$

(1) $0 < k_p \leq 4.5$. **(2)** When $k = p_f \cdot n\pi r_n^2 > k_p$, we say that $G(\mathcal{X}_n, g_p)$ percolates, and there exists only one infinite-order cluster *w.h.p.* **(3)** When the graph $G(\mathcal{X}_n, g_p)$ percolates, all of the finite-order clusters are one-order clusters *w.h.p.*

Proof: The existence of a non-trivial critical value of the average node degree for $G(\mathcal{P}_\lambda, g_p)$ comes from Lemma 1(1). We define $G(\mathcal{P}_\lambda, g_1')$ as the graph with vertex set $\mathcal{P}_\lambda$, and

$$g(x) = g_1'(x) \equiv \begin{cases} 1 & \text{if } |x| \leq r_n' \\ 0 & \text{if } |x| > r_n', \end{cases}$$

where $r_n' = \sqrt{p_f} r_n$. Obviously, g_p is the squashed function of g_1'. According to Lemma 1(3) and (4), we have

$$0 < \lambda_p(g_p) \leq \lambda_p(g_1') = \frac{4.5}{\pi {r_n'}^2}. \tag{8}$$

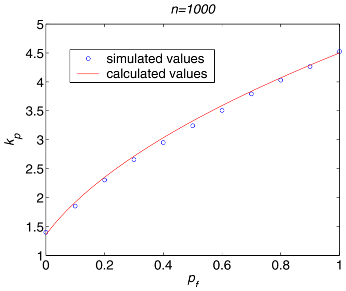

Fig. 4. **Percolation thresholds of different p_f values.** Calculated values follow the equation $k_p = 4.5 \cdot \sqrt{\frac{p_f+0.1}{1+0.1}}$ in (11).

The average node degree k_p for $G(\mathcal{P}_\lambda, g_p)$ is given by

$$k_p = p_f \cdot \lambda_p(g_p) \cdot \pi r_n^2 \leq \lambda_p(g_1') \cdot \pi {r_n'}^2. \tag{9}$$

From (8) and (9), we obtain $0 < k_p \leq 4.5$. Let $\lambda = n$, we can get the same result for $G(\mathcal{X}_n, g_p)$.

The second part of the theorem follows from Lemma 1(2). For g_p, since $g_p(x) \leq g_p(y)$ whenever $|x| \geq |y|$, there exists only one infinite-order cluster *w.h.p.* for $G(\mathcal{P}_\lambda, g_p)$. Using Poissonization technique, we can prove that it also holds for $G(\mathcal{X}_n, g_p)$. The third part of the theorem follows form Lemma 2, using Poissonization technique. ■

Let S be the giant cluster's size fraction, i.e., the number of nodes in the giant cluster divided by n. For Erdös-Rényi random graphs [20], [21], it is well known that S is a function of the average node degree k, and is the non-zero solution to the following equation

$$S = 1 - e^{-k \cdot S} \text{ when } n \to \infty. \tag{10}$$

For RGGs, simulation results show that, $S - k$ relationship also follows the same form but appears to be shifted along the average node degree axis (cf. Fig. 7). Since the proof procedure of Theorem 1 indicates that the probability of isolated nodes for $G(\mathcal{X}_n, g_p)$ is upper and lower bounded by the probabilities of isolated nodes for Erdös-Rényi random graph model $G(\mathcal{X}_n, g_0)$ and RGG model $G(\mathcal{X}_n, g_1')$, respectively, it is reasonable to assume that $S - k$ relationship for $G(\mathcal{X}_n, g_p)$ bears the same form with some shift along the average node degree axis. Therefore, we have the following conjecture:

Conjecture 1: **[Size of the Largest Cluster of $G(\mathcal{X}_n, g_p)$]** The size of the largest cluster of $G(\mathcal{X}_n, g_p)$, say S, is the non-zero solution to the following equation *w.h.p.*:

$$S = 1 - e^{-(k - k_p + 1.45) \cdot S}, \text{ where } k_p = 4.5 \cdot \sqrt{\frac{p_f + 0.1}{1 + 0.1}}. \tag{11}$$

Note that (11) is merely a shifted version of (10). We need to estimate this shift. Obviously it depends on k_p. However, Theorem 2(1) only gives the range of k_p. Given p_f, the exact value of k_p is still an open problem. Simulated values of k_p for different p_f values are given in Fig. 4. We have tried several function forms to estimate the $p_f - k_p$ relationship. A good approximation is found to be $k_p = 4.5 \cdot \sqrt{\frac{p_f+0.1}{1+0.1}}$. Fig. 4 illustrates the calculated k_p resulting from this approximation, which shows a good match between the simulated and calculated values of k_p. Two important practical consequences follow from Conjecture 1: (i) Given p_f, we can calculate the critical threshold of k for percolation; (ii) Given k, we can calculate the size of the giant cluster in $G(\mathcal{X}_n, g_p)$. We further support Conjecture 1 with simulations performed using different values of n and p_f (see Section III-D).

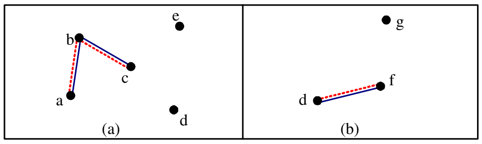

Fig. 5. **Local trust graphs for (a) node *a* and (b) node *f* in Fig. 1.** Note that in the local trust graph, two nodes are connected iff they are physical neighbors and they have a primary SA.

C. Properties of the Graph $G(\mathcal{X}_n, \mathcal{E}_{sl}')$

In this subsection, we want to study the following problem: given p_f, what is the probability that two neighboring nodes can establish a derived SA after the local secure-link augmentation phase? Recall that after the neighboring-friend discovery phase (cf. Section II-B), every node X_i knows its physical neighbors $N(X_i)$ and neighboring friends $NF(X_i)$. In the local secure-link augmentation phase, every node X_i first locally broadcasts its sets $N(X_i)$ and $NF(X_i)$. Using the sets received from neighbors, a node can build a local trust graph based on the friend relations among neighbors. The *local trust graph* maintained by node X_i is defined as $G_i(V_i, E_i)$, where the vertex set $V_i = \{X_j | X_j \in N(X_i) \vee j = i\}$ and the edge set $E_i = \{(X_j, X_k) | X_j, X_k \in N(X_i) \wedge X_k \in N(X_j) \wedge X_j \in N(X_k) \wedge (X_k, X_j) \in \mathcal{E}_{SA}\}$. Fig. 5 illustrates local trust graphs maintained by individual nodes.

There are two possible cases to locally augment secure links.

In the first case, two nodes do not share a primary SA (not directly connected in the trust graph), but have a common physical neighbor that shares a primary SA with each of them. In Fig. 5 (a), for instance, nodes a and c do not have a primary SA, but have a common physical neighbor node b that has a primary SA with each of them. Secure communications between nodes a and c can now be achieved via the help of node b. We denote the probability that this event occurs as p_1. When the symmetric key based cryptography is used, the pairwise key generated by node a is sent to node c through the secure path in the local trust graph, say secure links (a, b) and (b, c) in Fig. 5 (a). The pairwise key is encrypted/decrypted in each hop till it reaches the destination. When the asymmetric key based cryptograph is used, node a can obtain node c's valid public key from node b, and generates the pairwise key which is sent to node c directly, encrypted by node c's public key. In general, if a node finds a secure path in its local trust graph from itself to one of its unauthenticated neighbors, this node can adopt the method described above to authenticate each other and establish direct secure link between them.

In the second case, two nodes do not share a primary SA (not directly connected in the trust graph), and cannot find a neighbor satisfying the first case. However, there exists a node that shares a primary SA with each of the two nodes, and is a physical neighbor of only one of them. In Fig. 5 (b), nodes

g and f do not have a primary SA, but node d has a primary SA with each of them, and node d is the physical neighbor of only node f. Secure communications between nodes g and f can be achieved via the help of node d. We denote the probability that this event occurs as p_2. In this situation, node d will first find the possibility to augment secure link (f, g), and take the responsibility to initiate this procedure. When the symmetric key based cryptography is used, the pairwise key generated by node a will be sent to node d, and then this key will be sent back to f which is encrypted by nodes' d and g's pairwise key in their primary SA. Node f then relays the encrypted key to node g. When the asymmetric key based cryptograph is used, node f can obtain node g's valid public key from node d, and the pairwise key generated by node f is sent to node g directly, which is encrypted by node g's public key. In general, if a node X_i finds a secure path from itself to one of its authenticated neighbors, say X_j, and node X_j does not have a primary SA with one of node X_i's 2-hop away neighbors, say X_k, which is one of node X_i's friends, then node X_i can use this secure path to help these two nodes authenticate each other and establish a direct secure link.

Probabilities p_1 and p_2 can be calculated as the following:

$$p_1 = (1-p_f) \cdot \sum_{k=1}^{0.5865d} \left(\binom{0.5865d}{k} (p_f)^k (1-p_f)^{0.5865d-k} \cdot \left(1-(1-p_f)^k\right) \right). \quad (12)$$

$$p_2 = (1-p_f) \cdot (1-p_1) \cdot \sum_{k=1}^{0.8270d} \left(\binom{0.8270d}{k} (p_f)^k (1-p_f)^{0.8270d-k} \cdot \left(1-(1-p_f)^k\right) \right). \quad (13)$$

Note that p_1 and p_2 are functions of d and p_f, and $d = n\pi r_n^2$ is determined by n and r_n. In general, r_n and p_f are also given as a function of n. Therefore, p_1 and p_2 are functions of n only. After the local secure-link augmentation phase, the probability that there exists a secure link between two neighboring nodes, denoted by p'_f, is given by,

$$p'_f(n) = p_f(n) + p_1(n) + p_2(n). \quad (14)$$

Then the graph $G(\mathcal{X}_n, \mathcal{E}'_{sl})$ is equivalent to $G(\mathcal{X}_n, g_{p'_f})$, where

$$g_{p'_f}(x) = \begin{cases} p'_f & \text{if } |x| \le r_n \\ 0 & \text{if } |x| > r_n. \end{cases}$$

A simple examination on equations (12) and (13) shows that $p_1(n) \sim p_2(n) \sim \Theta(p_f(n))$, which implies $p'_f(n) \sim \Theta(p_f(n))$ from equation (14). As a result, we have the following theorem:

Theorem 3: **[Effect of Local Secure-Link Augmentation]** The local secure-link augmentation schemes will not alter the order of the probability for two neighboring nodes to establish a secure link, i.e. $p'_f(n) \sim \Theta(p_f(n))$.

Since we are only interested in the asymptotic properties of the WANET with n tending to infinity, Theorem 3 indicates that $G(\mathcal{X}_n, \mathcal{E}'_{sl})$ (i.e., $G(\mathcal{X}_n, g_{p'_f})$) is equivalent to $G(\mathcal{X}_n, \mathcal{E}_{sl})$ (i.e., $G(\mathcal{X}_n, g_p)$) since $p'_f(n) \sim \Theta(p_f(n))$.

D. Summary of the Properties of Secure Graphs

Results in Section III-B show that the secure graph $G(\mathcal{X}_n, g_f)$ undergoes two phase transitions with the varying of k, the average node degree of $G(\mathcal{X}_n, g_f)$.

The two critical values of k, corresponding to two phase transitions, separate the value of k into three intervals:

(1) $k \le k_p \approx 4.5$: secure graph $G(\mathcal{X}_n, g_f)$ is in the *subcritical phase*, indicating that $G(\mathcal{X}_n, g_f)$ does not percolate. The entire network consists of $O(n)$ small clusters, and the number of nodes in any cluster is not greater than $O(\log(n))$.

(2) $k_p < k \le k_c = \Theta(\log(n))$: secure graph $G(\mathcal{X}_n, g_f)$ is in the *supercritical phase*, indicating that $G(\mathcal{X}_n, g_f)$ percolates and is not connected *w.h.p.* In this phase, $G(\mathcal{X}_n, g_f)$ consists of one infinite-order cluster and some isolated nodes. Each node lies in either the infinite-order cluster or an one-order cluster (i.e., it is isolated) *w.h.p.*

(3) $k > k_c$: the secure graph $G(\mathcal{X}_n, g_f)$ is connected *w.h.p.*, i.e., there exists only one cluster.

Note that $k = p_f \cdot d = p_f \cdot \pi r_n^2$, which indicates that k is determined by two parameters: p_f from the trust graph and r_n from the physical graph. Now, we can proceed to investigate the trade-off between memory size, communication overhead and secure connectivity in $G(\mathcal{X}_n, g_f)$.

Connected Phase Previous works [27]–[29] suggest that in order to achieve the objective, at least $G(\mathcal{X}_n, \mathcal{E}_{sl})$ should be connected. Therefore, after the neighboring-friend discovery phase, the trust graph $G(\mathcal{X}_n, g_p)$ is created and all the nodes are connected with secure links *w.h.p.* A secure path between an arbitrary node pair can be established with any routing schemes operated on those secure links.

Next, we consider the communication overhead in this phase. The secure graph $G(\mathcal{X}_n, g_p)$ must have a lower average node degree compared to the corresponding physical graph $G(\mathcal{X}_n, g_1)$, in that a lower node degree will increase the path length which will affect the communication overhead. Obviously, this overhead can be characterized by the average path length. Previous works [17] show that, in order to keep the physical graph connected, the average node degree for $G(\mathcal{X}_n, \mathcal{E}_{pl})$, i.e., $d = \pi r_n^2$, should be on the order of $\log(n)$ minimally, and the average path length is $O(\sqrt{n})$. The following theorem indicates that, when $G(\mathcal{X}_n, g_p)$ is connected, the average path length is on the same order, which implies that the communication overhead introduced by the secure operations is asymptotically negligible.

Theorem 4: **[Average Path length of $G(\mathcal{X}_n, g_p)$]** When $G(\mathcal{X}_n, g_p)$ is connected, the average path length over all node pairs (or hop-counts) is $O(\sqrt{n})$.

Proof: Consider the graph $G(\mathcal{X}_n, g'_1)$ with

$$g'_1(x) = \begin{cases} 1 & \text{if } |x| \le \sqrt{p} r_n \\ 0 & \text{if } |x| > \sqrt{p} r_n. \end{cases}$$

$G(\mathcal{X}_n, g'_1)$ has the same average node degree as $G(\mathcal{X}_n, g_p)$. Since g_p is the squashed function of g'_1, the average path length

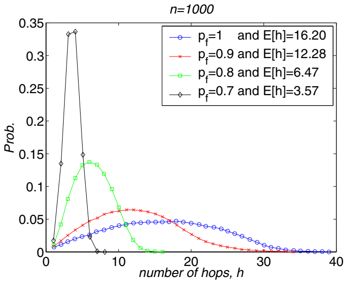

Fig. 6. **Effect of p_f on the hop-count distribution in the secure graph.**

for $G(\mathcal{X}_n, g_p)$ is smaller than that for $G(\mathcal{X}_n, g_1')$. $G(\mathcal{X}_n, g_1')$ is also a RGG, and has the same order of average path length as $G(\mathcal{X}_n, g_1)$. Therefore, the average path length for $G(\mathcal{X}_n, g_p)$ is on the same order as that for $G(\mathcal{X}_n, g_1)$, i.e., is $O(\sqrt{n})$. ■

Fig. 6 displays the simulated values of hop-count distribution for different p_f's given that the average node degrees remain the same. It can be observed that, when p_f becomes smaller, the average path length reduces significantly. The reason is that, to keep the average node degree on the order of $\log(n)$, r_n should be correspondingly increased. Therefore, the probability of having a direct link between two nodes at a longer distance increases as well, which will effectively decrease the average path length.

Theorem 4 indicates that when the network is operated in the situation that $G(\mathcal{X}_n, g_p)$ is connected, the secure connectivity objective can be achieved with negligible communication overhead. The problem, however, is that it is not scalable, which is demonstrated by the following "back-of-the-envelope" calculation.

In order to keep the physical graph connected, the average node degree for $G(\mathcal{X}_n, \mathcal{E}_{pl})$, i.e., $d = \pi r_n^2$ is at least on the order of $\log(n)$ [17]. Therefore, if we want secure graph $G(\mathcal{X}_n, \mathcal{E}_{sl})$ to be connected, p_f is at least on the order of $O(1)$, which means the average number of friends for every node, i.e., $p_f \cdot n$, is in the order of $O(n)$. Obviously, it is unrealistic for the trust graph as well as the required memory size in each node for primary SAs.

Last, we consider the effect of local secure-link augmentation phase. Theorem 3 shows that, after this phase, the probability that two neighboring nodes share a SA (primary or derived) is $p_f'(n)$, which is on the same order of $p_f(n)$. Therefore, given $p_f(n)$, which cannot achieve the connectivity of secure graph $G(\mathcal{X}_n, g_{p_f})$, although local secure-link augmentation techniques can achieve a p_f' greater than p_f, the resulting secure graph $G(\mathcal{X}_n, g_{p_f'})$ is still disconnected. Put it in another way, since $p_f'(n) \sim \Theta(p_f(n))$, it is impossible to rely on local secure-link augmentation techniques to reduce the requirement on the order of $p_f(n)$ for the secure connectivity.

Supercritical Phase Theorem 1 shows that in order to keep $G(\mathcal{X}_n, g_p)$ connected, the average node degree must grow approximately like $\log(n)$, when the number of nodes in the network increases, which is expensive and results in poor scalability. We show, however, that it is no longer the case if we only slightly loosen the connectivity requirement, by just imposing that $G(\mathcal{X}_n, g_p)$ is in the supercritical phase. More precisely, we want a giant cluster to appear in $G(\mathcal{X}_n, g_p)$, that contains a vast majority of the nodes, but we can leave a small number of nodes out of it (in other words, we want the securely connected nodes to percolate). This apparently benign change gives a much more optimistic perspective on the scalability of our scheme.

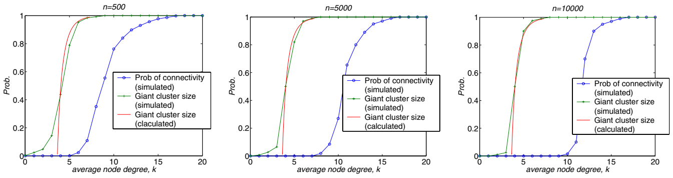

Fig. 7. **Simulated and calculated values for the giant cluster size and probability of connectivity in RGGs ($p_f = 1$) for different values of n.**

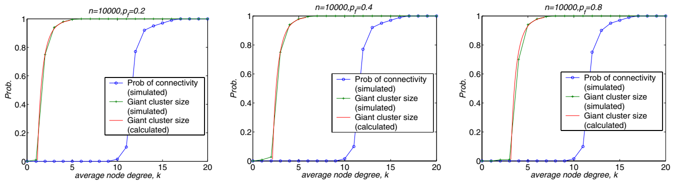

Fig. 8. **Simulated and calculated values for the giant cluster size and probability of connectivity in trust graphs for different values of p_f.**

Fig. 7 and 8 show the calculated and simulated results about the size of the giant cluster and the probability of connectivity for different p_f and n values. The first conclusion we can draw after analyzing simulation data is that Conjecture 1 gives a quite accurate estimation of the giant cluster size when n is large enough. Secondly, in all simulated cases we see that the giant cluster size is growing steeply towards 1 for those values of the mean degree that the probability of 1-connectivity is very low. For a relatively large span of the mean degree values the giant cluster is already covering most of the network but 1-connectivity is not achieved yet. This is due to only a few isolated nodes or small node clusters outside the giant cluster.

Note that the giant cluster of $G(\mathcal{X}_n, g_p)$ forms a connected secure backbone, i.e., all nodes in the giant cluster are connected with secure links or paths. It is easy to show that the average secure path length of node pairs in the giant cluster is also on the order of $O(\sqrt{n})$, which means the communication overhead (or routing stretch) is asymptotically negligible. Therefore, we can reduce the memory size or enhance the scalability of the network by a factor of $\log(n)$ with the trade-off of isolation of an arbitrary small fraction of nodes. Here, isolation only means that there is no secure links connecting isolated nodes with the secure backbone. We can still guarantee the physical connectivity by setting that $d = \pi r_n^2$ is at least on the order of $\log(n)$ [17].

Handling Isolated Nodes Although a small fraction of isolated nodes is a reasonable trade-off, in certain environments we might need to connect these isolated nodes to the network. To connect isolated nodes to the network, the isolated nodes need to detect it is isolated. Existing network partition detection algorithms may be used for this purpose. But according to Theorem 2, when the graph gets into the supercritical phase, except for the giant cluster, the size of the remaining clusters is very small, usually 1 in our simulation results. Thus the cost for the partition detection algorithm is not expensive, or when the cluster size is 1, it can recognize itself as isolated without any partition detection algorithm. The isolated nodes can be securely connected to the giant cluster (secure backbone) by finding one of the nodes in the giant cluster which shares a primary SA with it. To do this it should either increase transmission range or move around or broadcast the message to find a friend with 2 or more hops. The routing-security dependency loop can be avoided since that each isolated node can communicate with the secure backbone (the giant cluster in $G(\mathcal{X}_n, g_p)$) with only one-hop *w.h.p.* and the communication overhead can be ignored since the number of isolated nodes can be made arbitrarily small.

Subcritical Phase It is interesting to ask whether we can further reduce the required number of authenticated neighbors, i.e., k, such that $k < 4.5$ and the secure graph $G(\mathcal{X}_n, g_p)$ is in the subcritical phase. The problem is that in this phase, the entire network consists of $O(n)$ small isolated trusted groups (clusters), and the number of nodes in any trusted group is not greater than $O(\log(n))$. Then, when we need to establish a path between two nodes, this path will go through $O(\sqrt{n})$ isolated trusted groups *w.h.p.*, which is almost on the same order of the path length. Therefore, when $G(\mathcal{X}_n, g_p)$ is in the subcritical phase, we will meet routing-security dependency loop again, and local secure-link augmentation is not enough to break this loop. Our conclusion is that, the secure graph $G(\mathcal{X}_n, g_p)$ is at least in the supercritical phase, and k cannot be further reduced.

IV. Conclusion

In this paper, we propose a generic model to evaluate the relationship of connectivity, memory size, communication overhead and security in fully self-organized WANETs. Based on some reasonable assumptions on node deployment and mobility, we show that we can achieve secure connectivity when the average number of authenticated neighbors of each node is at least $\Omega(1)$. We utilize continuum percolation theory to construct the secure backbone, and connect the isolated nodes to the secure backbone with multi-hop secure link augmentation scheme. The communication overhead incurred due to the neighbor authentication and the routing stretch are asymptotically negligible.

References

[1] J. Hubaux, L. Buttyan, and S. Capkun, “The quest for security in mobile ad hoc networks,” in *MobiHoc 2001*, Long Beach, CA, Oct. 2001.

[2] L. Feeney, B. Ahlgren, and A. Westerlund, “Spontaneous networking: an application-oriented approach to ad hoc networking,” *IEEE Communications Magazine*, vol. 39, no. 6, pp. 176–181, June 2001.

[3] S. Capkun, L. Buttyan, and J. Hubaux, “Self-organized public-key management for mobile ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52–64, Jan. 2003.

[4] L. Zhou and Z. Haas, “Securing ad hoc networks,” *IEEE Network Magazine*, vol. 13, no. 6, pp. 24–30, Nov. 1999.

[5] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, “Providing robust and ubiquitous security support for mobile ad hoc networks,” in *Proc. of IEEE ICNP 2001*, Riverside, CA, Nov. 2001.

[6] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, “Self-securing ad hoc wireless networks,” in *Proc. of IEEE Symposium on Computers and Communications*, Italy, July 2002.

[7] S. Yi and R. Kravets, “Moca: Mobile certificate authority for wireless ad hoc networks,” in *Proc. of the 2nd Annual PKI Research Workshop (PKI 2003)*, Gaithersburg, MD, April 2003.

[8] Y. Zhang, W. Liu, W. Lou, and Y. Fang, “Securing mobile ad hoc networks with certificateless public keys,” *IEEE Transactions on Dependable and Secure Computing,*, vol. 3, no. 4, pp. 386–399, Oct.-Dec. 2006.

[9] K. Ren, W. Lou, and Y. Zhang, “Leds: Providing location-aware end-to-end data security in wireless sensor networks,” in *Proc. of IEEE INFOCOM’06*, Barcelona, Spain, Apr. 2006.

[10] J. Merwe, D. Dawoud, and S. McDonald, “Fully self-organized peer-to-peer key management for mobile ad hoc networks,” in *Proc. of WiSE 2005*, Cologne, Germany, Sep. 2005.

[11] M. Penrose, *Random Geometric Graphs*. Oxford: Oxford University Press, 2003.

[12] X. Lin, G. Sharma, R. Mazumdar, and N. Shroff, “Degenerate delay-capacity trade-offs in ad hoc networks with brownian mobility,” *IEEE/ACM Transactions on Networking*, vol. 52, no. 6, pp. 2777–2784, June 2006.

[13] M. Neely and E. Modiano, “Capacity and delay tradeoffs for ad-hoc mobile networks,” *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1917–1937, June 2005.

[14] S. Capkun, L. Buttyan, and J. Hubaux, “Small worlds in security systems: an analysis of the pgp certificate graph,” in *New Security Paradigms Workshop 2002*, Norfolk, VA, Sep. 2002.

[15] S. Garfinkel, *PGP: Pretty Good Privacy*. Cambridge, MA: O’Reilly & Associates, 1994.

[16] M. Streib, “Keyanalyze—analysis of a large OpenPGP ring.” http://dtype.org/keyanalyze/, Date are from October 3, 2004.

[17] P. Gupta and P. R. Kumar, *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*. Boston: Birkhauser, 1998, ch. Critical Power for Asymptotic Connectivity in Wireless Networks.

[18] Y.-C. Hu, A. Perrig, and D. Johnson, “Ariadne: A secure on-demand routing protocol for ad hoc networks,” in *Proc. of ACM MobiCom 2002*, Atlanta, GA, September 2002.

[19] R. Bobba, L. Eschenauer, V. Gligor, and W. Arbaugh, “Bootstrapping security associations for routing in mobile ad-hoc networks,” in *Proc. of GLOBECOM 2003*, San Francisco, CA, Dec. 2003.

[20] B. Bollobás, *Random Graphs*. Orlando, FL: Academic Press, 1985.

[21] S. Janson, T. Luczak, and A. Rucinski, *Random Graphs*. New York: John Wiley & Sons, 2000.

[22] E. Gilbert, “Random plane networks,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 4, pp. 533–543, Dec. 1961.

[23] R. Meester and R. Roy, *Continuum Percolation*. Cambridge: Cambridge University Press, 1996.

[24] M. Penrose, “On a continuum percolation model,” *Advances in Applied Probability*, vol. 23, no. 3, pp. 536–556, Sep. 1991.

[25] M. Franceschetti, L. Booth, M. Cook, J. Bruck, and R. Meester, “Continuum percolation with unreliable and spread out connections,” *Journal of Statistical Physics*, vol. 118, no. 3/4, pp. 721–734, Feb. 2005.

[26] P. Balister, Bollobás, and M. Walters, “Continuum percolation with steps in the square or the disc,” *Random Structures and Algorithms*, vol. 26, no. 4, pp. 392–403, April 2005.

[27] L. Eschenauer and V. Gligor, “A key-management scheme for distributed sensor networks,” in *Proc. of ACM CCS’02*, Washingtion, DC, Nov. 2002.

[28] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key predistribution scheme for wireless sensor networks,” in *Proc. of ACM CCS’03*, Washingtion, DC, Oct. 2003.

[29] D. Huang, M. Mehta, A. van de Liefvoort, and D. Medhi, “Modeling pairwise key establishment for random key predistribution in large-scale sensor networks,” *IEEE/ACM Trans. on Networking*, vol. 15, no. 5, pp. 1204–1215, Oct. 2007.