On Applicability of Random Graphs for Modeling Random Key Predistribution for Wireless Sensor Networks

Tuan Manh Vu, Reihaneh Safavi-Naini and Carey Williamson

University of Calgary, Calgary, AB, Canada

Abstract. We study the applicability of random graph theory in modeling secure connectivity of wireless sensor networks. Specifically, our work focuses on the highly influential random key predistribution scheme by Eschenauer and Gligor to examine the appropriateness of the modeling in finding system parameters for desired connectivity. We use extensive simulation and theoretical results to identify ranges of the parameters where i) random graph theory is not applicable, ii) random graph theory may lead to estimates with excessive errors, and iii) random graph theory gives very accurate results. We also investigate the similarities and dissimilarities in the structure of random graphs and key graphs (i.e., graphs describing key sharing information between sensor nodes). Our results provide insights into research relying on random graph modeling to examine behaviors of key graphs.

1 Introduction

Wireless sensor networks (WSNs) are *ad-hoc* networks that consist of hundreds to thousands of small sensor nodes communicating wirelessly to collect and deliver data to base stations. Generally, sensor networks rely on symmetric key algorithms to avoid the high computation cost of public key crypto-systems such as Diffie-Hellman key exchange [3]. Furthermore, traditional methods of key establishment that use a trusted authority (e.g., Kerberos protocol [16]) are not suitable due to the frequently used unattended deployments of WSNs.

Eschenauer and Gligor (EG) [6] pioneered an innovative randomized approach to key establishment that provides an efficient and self-organising way of constructing pairwise keys for sensor nodes with guaranteed security. In the EG scheme, every sensor receives a random subset of m keys called a *key ring*, from a *key pool* of N keys. Once deployed, sensor nodes broadcast the identifiers of keys in their key rings to discover wireless neighbors with whom they have at least one key in common, and then establish *secure links*.

The *key graph* for a WSN of n sensor nodes is a graph with n vertices in which each node is represented by a vertex, and two vertices are joined by an edge if the two corresponding nodes share at least one key. The important question is how to choose key ring size m and key pool size N so that the key graph is connected with desired probability c. A connected key graph implies that any two nodes can

set up a pairwise key. Eschenauer and Gligor modeled the key graph as an *Erdős-Rényi random graph* $G(n, p)$, a graph of n nodes where each possible edge exists independently with a fixed probability p. This modeling allows to determine m given N and n such that secure connectivity is guaranteed with probability c. The EG scheme has generated a flurry of research on key predistribution for WSNs that employ Erdős-Rényi's theory of random graphs to study connectivity [2, 4, 10, 13–15]. This modeling makes two crucial assumptions:

1. *Asymptotic results provide accurate prediction for all network sizes.* Erdős-Rényi[5] studied asymptotic behavior of random graphs, showing that as $n \to \infty$ there are certain properties that will *almost surely* appear in the graph, once the edge probability exceeds a threshold that depends on the property. Connectivity is one such property; a random graph with edge probability $p = \frac{\ln(n)-\ln(-\ln(c))}{n}$ is connected with probability c. It is however unclear if choosing p as above for all possible values of n will achieve connectivity with probability c.
2. *Edge probability in key graphs is fixed and edge probabilities are independent.* In key graphs, although the probability that an edge between two arbitrarily given nodes exists is fixed, these probabilities are not independent of the existence of other edges in the graph [17]. For example consider a key graph for a network with three nodes X, Y, and Z, and key ring size $m = 2$. Using the results from [6] for sufficiently large key pool size, say $N > 10$, the probability that two nodes share at least one key is less than 50% (see Equation 5). However, if the two pairs (X, Y) and (Y, Z) each share a key, then the probability that X and Z have at least one key in common exceeds 50%, regardless of N. This is because both X and Z have at least one key in a fixed set of two keys (i.e., key ring of Y). This is referred to as the *transitivity* property. There are also certain values of (N, m) for which the key graph is clearly not an Erdős-Rényi random graph. For instance, if each node has only one key (i.e., $m = 1$), then the set of all nodes having the same key forms a complete graph.

The goals of this paper are i) to study the applicability of random graph theory in estimating key ring size (for desired connectivity probability) in key graphs, and ii) to compare structural properties of the two families of graphs. The latter study is motivated by application of random key predistribution in secure group connectivity in sensor and *ad-hoc* networks.

1.1 Our work

We consider the following two questions:

Q1 Can random graphs be always used to model secure connectivity achieved by random key predistribution systems, and when applicable how accurate are the estimated key ring sizes?
Q2 How similar are the structural properties of the two graph families?

We provide two types of results: (i) analytical results and (ii) simulation results. The latter results are obtained by constructing many random instances of key graphs and determining the relative frequency of instances that have the property of interest. For this, we developed a simulator [19] that can efficiently generate key graphs from the prescribed random key assignment process. Using results from statistics we estimate the measurement error and confidence intervals for the obtained values.

Our results for Q1: The simulated key ring sizes for a network of size n differ from the analytical results. The differences can be categorized into four intervals delineated by three threshold values $n_1^{[N,c]}, n_2^{[N,c]}, n_3^{[N,c]}$.

1. Interval I: $n \leq n_1^{[N,c]}$. Random graph theory cannot be used as it results in $p > 1$ which is an invalid edge probability. The value $n_1^{(N,c)}$ is independent of key pool size N and is determined only by connectivity c.
2. Interval II: $n_1^{[N,c]} < n < n_2^{[N,c]}$. Random graph theory can be used in this interval, however key ring sizes predicted by random graph modeling are slight over-estimation. We define over-estimation as the relative error $\geq 5\%$ *and* the absolute error > 2 (see Section 3.2 for more details). When the network size n and connectivity c are fixed, the value of $n_2^{[N,c]}$ increases as N grows. Nevertheless, the empirical simulation results suggest that for wide ranges of parameters (see Table 2), $n_2^{[N,c]}$ never exceeds 100.
3. Interval III: $n_2^{[N,c]} \leq n < n_3^{[N,c]}$. In this interval, random graph modeling provides very good estimates for m (i.e., compared to key ring sizes obtained by simulations, either the relative error $< 5\%$ or the absolute error ≤ 2), and the predicted key ring size ≥ 2. The value $n_3^{[N,c]}$ depends on connectivity c and key pool size N.
4. Interval IV: $n \geq n_3^{[N,c]}$. In this interval, random graph modeling always gives $m = 1$. This however does not provide connectivity except for the trivial case where all nodes have the same key. Simulation results suggest that m should be 2 instead.

The results suggest that random graph theory, when used within appropriate parameter ranges, provides very good estimates of key predistribution parameters for achieving desired secure connectivity of WSNs.

Our results for Q2: We consider the following structural properties:

1. Global clustering coefficient.
2. The size of the maximal clique.
3. The number of cliques with respect to clique sizes.

These properties are important in studying key predistribution as well as routing in WSN. Global clustering is a measure of transitivity that also enables us to explain behavior of the other two properties. Cliques are used in many *ad-hoc* algorithms for constructing group-wise keys, detecting intrusion and choosing

group leaders (i.e., clusterheads in WSNs), as well as algorithms for finding capacity, quality of service, and routing [8, 11, 12, 21]. Our study shows that:

1. Global clustering coefficient of key graphs deviates significantly from that of random graphs for smaller key ring sizes, but starts to converge as the key ring size increases.
2. Key graphs contain many more cliques than do random graphs of the same size.
3. The maximal clique size observed in a key graph is much larger than in random graphs of the same size.

The rest of this paper is organized as follows. In Section 2, we give background preliminaries and definitions. Section 3 explains our methodology and results for estimating key ring size. Structural properties of the two graph families are compared in Section 4 and finally, Section 5 provides concluding remarks and directions for future work.

2 Preliminaries

2.1 Random graph

An Erdős-Rényi random graph, denoted by $G_r(n, p)$, is a graph of size n generated through the following random process. First, we start with n vertices and no edges. Next, each of $\frac{n\cdot(n-1)}{2}$ possible edges between vertex pairs is added to the graph with probability p, determined by a biased coin flip. A graph obtained through this random and independent edge generation process is an instance from a family of random graphs.

Connectivity
The probability of $G_r(n, p)$ being connected is the probability that the random and independent edge generation process with parameters n and p results in a connected graph. Erdős and Rény showed that,

$$\lim_{n\to\infty} \mathtt{Pr}\left[G_r\left(n,p\right) \text{ is connected}\right] = c, \text{ where } p = \frac{\mathtt{ln}(n) - \mathtt{ln}(-\mathtt{ln}(c))}{n} \quad (1)$$

Global clustering coefficient
Holland and Leinhardt [9] introduced the notion of global clustering coefficient C. This is an important measurement in studying social and real-world networks to examine the property, "a friend of my friend is likely to be my friend as well". In other words, C implies transitivity relation between pairs of vertices: if there exists an edge between vertices (X, Y) and an edge between vertices (Y, Z), then there is a high probability that there is an edge between vertices (X, Z). Global clustering coefficient is defined as a metric for a particular graph. In the family of random graphs, since each edge occurs independently, we have

$$\mathtt{E}[C] = p. \quad (2)$$

Number of cliques
A clique in an undirected graph G is a subset S of vertices such that every two vertices in S are connected by an edge. Let the random variable $\Gamma_k(G)$ denote the number of cliques of size k in G. Given a subset S of k vertices in $G_r(n,p)$, the number of pairs of vertices is $\frac{k\cdot(k-1)}{2}$, and thus the probability of S being a clique is $p^{k\cdot(k-1)/2}$ [1] and

$$\mathtt{E}[\Gamma_k(G_r(n,p))] = \binom{n}{k} \cdot p^{k\cdot(k-1)/2}. \tag{3}$$

Maximal clique size
Let the random variable $\Upsilon(G)$ denote the maximal size of a clique in an undirected graph G. Specifically, $\Upsilon(G) = \mathtt{max}\{k : \Gamma_k(G) > 0\}$. Grimmett and McDiarmid [7] studied asymptotic behavior of the maximal clique size in random graphs, showing that

$$\lim_{n\to\infty} \frac{\Upsilon(G_r(n,p))}{\mathtt{ln}(n)} = \frac{2}{\mathtt{ln}(1/p)}. \tag{4}$$

2.2 Key graph

A key graph $G_k(n,N,m)$ describing key sharing information between nodes is constructed through the following random key assignment process. We start with n nodes, each with a randomly chosen key ring of size m from a key pool of size N. For every two nodes X and Y that share at least one key, we add the edge (X,Y). All instances of graphs corresponding to all possible key assignments with parameters (n,N,m) define a family of key graphs.

2.3 Modeling key graphs using Erdős-Rényi random graph theory

Two arbitrary nodes are joined by an edge if their assigned key rings intersect. As shown by Eschenauer and Gligor [6], this occurs with probability

$$p_{\mathtt{key\ sharing}} = 1 - \frac{((N-m)!)^2}{N!\cdot(N-2\cdot m)!}. \tag{5}$$

Assuming $G_k(n,N,m)$ is $G_r(n,p_{\mathtt{key\ sharing}})$, Equation 1 suggests that to achieve connectivity c, $p_{\mathtt{key\ sharing}}$ should be at least $\frac{\mathtt{ln}(n)-\mathtt{ln}(-\mathtt{ln}(c))}{n}$. This allows key ring size m to be estimated as a function of n, N, and c.

3 Applicability of Random Graph Theory in Estimating Key Ring Size

In the following, we give our theoretical results and explain how simulation data are obtained, and then discuss our observations.

3.1 Framework

Using random graph theory: As explained in Section 2.3, for network size n, key pool size N, and desired connectivity c, key ring size m is estimated as the smallest integer that satisfies the following

$$1 - \frac{((N-m)!)^2}{N! \cdot (N-2 \cdot m)!} \geq \frac{\ln(n) - \ln(-\ln(c))}{n}. \tag{6}$$

Using simulation: To find true connectivity probability c for key graphs with parameters (n, N, m), we need to generate all key graphs with these parameters and find the ratio of the connected graphs to the total number. The number of key assignments, however, grows exponentially with n, N, and m, which makes this calculation infeasible. We therefore use random sampling of the set of key graphs to estimate connectivity. By selecting a sufficiently large sample size, the simulation results give, with high confidence, an accurate estimate of c.

We use two algorithms. Algorithm 1 takes inputs n, N, m, and the size S of the sample set, generates S random instances of $G_k(n, N, m)$, examines their connectivity, and calculates the ratio of connected key graphs to S. This is the estimation of connectivity $\hat{c}$ of $G_k(n, N, m)$. Algorithm 2 performs binary search on the given interval (i.e., the algorithm inputs) to determine the smallest m such that $\hat{c} \geq c$. This method works correctly since when network size n and key pool size N are fixed, connectivity probability c monotonically increases as key ring size m grows. The pseudocode for both algorithms are given in the appendix.

Error and confidence interval: In the experiments, we use a sample size of $S = 10,000$ to achieve a reasonable statistical behavior. In particular, if $\hat{c}$ is the estimated connectivity of $G_k(n, N, m)$ obtained by Algorithm 1, the standard deviation of $\hat{c}$ is $\sigma = \sqrt{\frac{\hat{c} \cdot (1-\hat{c})}{S}} = \frac{\sqrt{\hat{c} \cdot (1-\hat{c})}}{100}$. With the 99% confidence level, the true connectivity c falls within $z^* \cdot \sigma$ from $\hat{c}$, where z^* is the critical value corresponding to the desired confidence level. For the 99% confidence level, we have $z^* = 2.58$. Table 1 summarizes the confidence intervals for different values

Table 1. Accuracy of graph connectivity simulation results with 99% confidence level

Connectivity $\hat{c}$	Std error σ	Margin of error $(z^* \cdot \sigma = 2.58 \cdot \sigma)$	Confidence interval
0.500	0.0050	0.0129	0.500 ± 0.0129
0.700	0.0046	0.0119	0.700 ± 0.0119
0.900	0.0030	0.0077	0.900 ± 0.0077
0.999	0.0003	0.0008	0.999 ± 0.0008

of $\hat{c}$, showing that 10,000 random samples give a very good estimate of graph

connectivity. For instance, if $G_k(n, N, m)$ is connected with probability $\hat{c} = 0.8$ in the experiment, then the true connectivity c lies in the interval (0.8 ± 0.0103) 99% of the time.

Data sets: We obtain key ring size m theoretically and using simulation over wide ranges of n, N, and c as indicated in Table 2. Our observations are discussed in the next section.

Table 2. Data sets

Parameter	Range
Network size n	$\{3..100,\ i \cdot 10^j \mid i = 2..10, j = 2..3\}$
Key pool size N	$\{10,\ \frac{1}{4} \cdot 10^i,\ \frac{1}{2} \cdot 10^i,\ \frac{3}{4} \cdot 10^i,\ 10^i \mid i = 2..5\}$
Desired connectivity c	$\{0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 0.99,\ 0.999\}$

3.2 The results

We first compare m obtained theoretically and by simulation when both N and c are fixed, and discuss how the value of n affects the applicability and accuracy of random graph modeling. We then extend these results to all n, N, and $c \geq 50\%$.

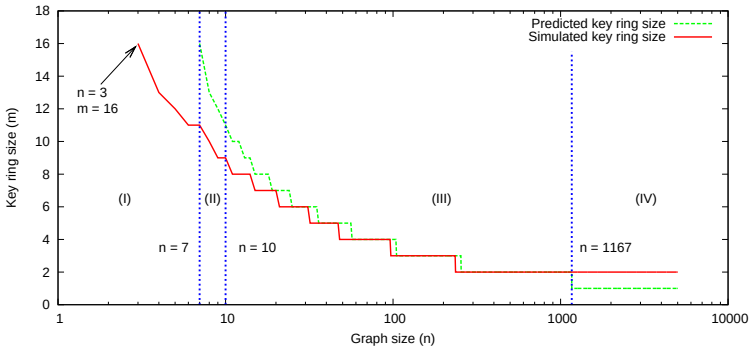

Fig. 1. Key ring size with respect to network size when key pool size $N = 100$ and desired connectivity $c = 99\%$

Figure 1 plots key ring sizes with respect to network size n when $N = 100$ and $c = 99\%$. In Figure 1, we can identify four distinct intervals.

Interval I, $n \leq n_1^{[N,c]}$ - Small graphs: Figure 1 does not have the plot for theoretical key ring size for $n \leq 6$. This is because, for $n \leq 6$, $N = 100$, and

$c = 99\%$, random graph theory estimates edge probability $p > 1$ (see Equation 1), which cannot be achieved.

Finding $n_1^{[N,c]}$*:* Random graph theory suggests that $p \geq \frac{\ln(n)-\ln(-\ln(c))}{n}$ where connectivity $c \in (0,1)$. Additionally, $n \in [3,\infty)$ as only networks with at least three nodes are of interest. We want to determine for which values of c and n in their respective domains, $\frac{\ln(n)-\ln(-\ln(c))}{n}$ becomes an invalid probability. Due to the limited space, we do not go into the details but summarize the results instead. Basically, we examine the variation of $p(n) = \frac{\ln(n)-\ln(-\ln(c))}{n}$, and find when $p(n)$ is equal to 0 and 1, and reaches the maximum. From that, we make the following observations.

1. When c is close to 0, there is a bound $n_0^{[N,c]}$ of n such that $\forall n \in [3, n_0^{[N,c]}] : p(n) < 0$. Explicitly, let $n_0^{[N,c]}$ be the solution to $p(n) = 0$, $n_0^{[N,c]} = -\ln(c)$. It follows that, $n_0^{[N,c]} \geq 3$ (i.e., $-\ln(c) \geq 3$) if and only if $c < c^\dagger$ where $c^\dagger = e^{-3} \simeq 0.05$. In short, $\forall c < c^\dagger, \forall n \in [3, -\ln(c)] : p(n) < 0$. In such cases, random graph theory suggests that choosing edge probability 0 (i.e., key ring size $m = 0$) will achieve connectivity c. However, if $m = 0$, the key graph is surely disconnected, and c is 0. Nevertheless, since c is too small (i.e., $c < 0.05$) to be considered in an actual WSN deployment, we are not interested in this case.
2. When c is close to 1, there is a bound $n_1^{[N,c]}$ of n such that $\forall n \in [3, n_1^{[N,c]}] : p(n) > 1$. That means the edge probability estimated by random graph theory is larger than 1, which cannot be achieved. This range is noted as (I) in Figure 1 for the particular parameters $N = 100$ and $c = 0.99$. In such cases, random graph theory is not applicable. We define $n_1^{[N,c]}$ as

$$n_1^{[N,c]} = \texttt{max}\{n : p(n) > 1 \texttt{ and } n \geq 3\}.$$

The value of $n_1^{[N,c]}$ depends on c only, and grows as c increases. Let c^* be the solution to $\frac{\ln(3)-\ln(-\ln(x))}{3} = 1$, we have $c^* = e^{-e^{\ln(3)-3}} \simeq .86$, and $\forall c < c^*, \forall n \geq 3 : p(n) < 1$ (i.e., $n_1^{[N,c]}$ does not exist). Some examples of $n_1^{[N,c]}$ are presented in Table 3. One cannot use random graph theory to estimate m given n and c if $c \geq c^*$ and $n \in [3, n_1^{[N,c]}]$.

Table 3. Lower bound of network size n with respect to connectivity c

Desired connectivity c	$n_1^{[N,c]}$	Desired connectivity c	$n_1^{[N,c]}$
0.8	n/a	0.9	3
0.99	6	0.999	9

Interval II - Over-estimation: We compare the theoretical and simulated key ring sizes. We define *over-estimation* when the relative error $\geq 5\%$ and the

absolute error > 2. Both conditions are required since considering only one kind of error may be misleading. For example, if theoretical and simulated values of m are 5 and 4, respectively, then the random graph modeling estimate is off by only one key and we consider it as a good estimation despite the relative error being 25%. On the other hand, if the two values are 111 and 107, respectively, then the relative error is less than 4% while the absolute error is 4 keys.

For $n \in [7, 10)$, $N = 100$, and $c = 99\%$, random graph theory results in over-estimation. For wide ranges of parameters where N is up to 10^5, n is up to 10^4, and c is from 50% to 99.9%, the comparison results suggest that there is a small interval of n in which using random graph theory gives over-estimation. The left side of this interval is $n_1^{[N,c]} + 1$ if $c \geq c^*$, or 3 otherwise.

Finding $n_2^{[N,c]}$*:* For fixed c, the right hand side of the interval increases as N grows but over the ranges of parameters summarized in Table 3, it never exceeds 100. Thus, when $n < 100$, our simulation results show that an excessive over-estimation *may* occur. [1] One can always use the simulator [19] to obtain key ring size m by simulations for better estimate.

Interval III - Good estimation: Figure 1 shows that when $N = 100$ and $c = 99\%$, random graph theory gives accurate estimates for $n \in [10, 1167)$. Specifically, the theoretical key ring sizes perfectly match the simulated ones in most cases. In other cases, random graph theory over-estimates by only one key. We did not observe any under-estimation. When $n \geq 1167$, the theoretical results show $m = 1$, while the simulation experiments yield $m = 2$. This phenomenon as well as at which values of n it occurs (e.g., $n = 1167$ for $N = 100$ and $c = 99\%$) is further discussed later in this section.

Let us assume that $n_3^{[N,c]}$, which is a function of c and N, is a lower bound for n such that random graph modeling estimates $m = 1, \forall n \geq n_3^{[N,c]}$. *We call the interval* $[100, n_3^{[N,c]})$ *safe for using random graph theory for estimating* m. Table 5 presents selected comparison results of key ring sizes for $c = 99.9\%$ and different values of n and N. Aside from the correct estimates and a few over-estimates with low relative errors, the theoretical key ring size is 1 when $N = 100$ for some cases. In these cases, n is outside the safe interval (i.e., $n \geq n_3^{[100,0.999]}$). Generally, when the key pool size is large and network size is small, over-estimation may occur, but the relative error is less than 5%. For small key pool sizes, if there is an over-estimation, the absolute error is only one or two keys.

Tables 6a-d in Appendix show the difference between theoretical and simulated key ring sizes with respect to n, N, and c. A light-gray cell represents an under-estimation while dark-gray cell indicates an over-estimation. Again, we can see that there is a pattern of under-estimating key ring sizes when n is large

[1] Over-estimation leads to extra keys in the key rings. In the case of an eavesdropping adversary, this only results in less efficient (larger key ring size) systems. In the case of a node capturing adversary, larger key rings result in higher probability of edge compromise.

and N is small in all four tables. In those cases, m is estimated by random graph theory as 1, and n is outside the safe interval. There are rare situations in which random graph theory under-estimates m when $n \in [100, n_3^{[N,c]})$. One such case is when $n = 1000$, $N = 50000$, and $c = 70\%$ as shown in Table 6b. We believe this is due to statistical error.

As noted earlier, over-estimation occurs when N is very large and n is small; the relative error in such cases, however, is less than 5%. In general, when n lies in the safe interval, the data in Tables 6a-d supports the claim that, *the estimate for key ring size based on random graph theory is very precise for* $n \in [100, n_3^{[N,c]})$.

Interval IV - Large graphs: For a fixed N, the key sharing probability for $m = 1$ is $\frac{1}{N}$. As the network size n increases, the edge probability p that is required for connectivity c given by $p = \frac{\ln(n)-\ln(-\ln(c))}{n}$ decreases. For sufficiently large n, we will have $\frac{\ln(n)-\ln(-\ln(c))}{n} \leq \frac{1}{N}$. Therefore, according to random graph theory, connectivity c can be achieved with $m = 1$. In this case, nodes that have the same key can be grouped together and so the graph can be decomposed into disjoint cliques. The key graph is connected only if all n nodes have the same key and this happens with probability $\frac{1}{N^{n-1}}$, which could be much lower than desired connectivity c. We define $n_3^{[N,c]}$ as follows

$$n_3^{[N,c]} = \min\{n : \frac{\ln(n) - \ln(-\ln(c))}{n} \leq \frac{1}{N}\}.$$

Given desired connectivity c and key pool N, random graph theory always gives incorrect estimate for key ring size (i.e., m is estimated as 1) when $n \geq n_3^{[N,c]}$. Figure 2 plots $n_3^{[N,c]}$ with respect to key pool size N and connectivity c.

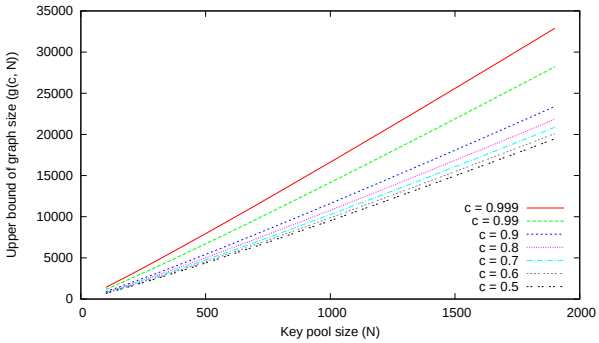

Fig. 2. Upper bounds of network size n with respect to connectivity c and key pool size N

Summary Table 4 describes four intervals of network size n for given connectivity c and key pool N. These intervals provide insights into the applicability of random graph theory in estimating key ring size to achieve connectivity c when $c \geq 50\%$. For $c < 50\%$, one would expect similar interval structure.

Table 4. Four intervals of network size n with respect to key pool size N and desired connectivity c

		$0.5 \leq c < c^*$	$c \geq c^*$
n	Interval I	n/a	$[3,\ n_1^{[N,c]}]$
	Interval II	$[3,\ 100)$	$(n_1^{[N,c]},\ 100)$
	Interval III	$[100,\ n_3^{[N,c]})$	
	Interval IV	$[n_3^{[N,c]},\ +\infty)$	

4 Structural Properties

We examine and compare structural properties of random graphs and key graphs for given n and p, that is the same network size and the same edge probability. We use the following approach:

1. For each set of parameters (n, N, m), we calculate the edge probability p according to Equation 5.
2. We then generate α random instances of $G_k(n, N, m)$ and measure the average value of property X.
3. Finally, we compare the simulation result with the theoretical values obtained for $G_r(n, p)$. The parameters n, N, and m are chosen to achieve 'reasonable' edge probability p.

The number of random instances of key graphs in each set of simulation experiments is $\alpha = 1,000$. In Figures 3-5, the simulation results are plotted with the error bars indicating the 99% confidence intervals for the true mean.

4.1 Global clustering coefficient

In the first set of simulation experiments, we measure the global clustering coefficient C in the two graph families. Recall that in a random graph, each edge occurs independently, and thus C is always equal to the edge probability. However, this value in key graphs gives the probability that two nodes share keys if they both share keys with some common node. In the experiments, we choose $n = 100$, $N = 1000$, and observe C as m varies. Figure 3 provides our comparison results. It can be seen that for very small values of m, C in key graphs is much higher than that in random graphs. This is because if nodes X and Y share key k_1, nodes Y and Z share key k_2, and m is small, then it is likely that k_1 is k_2, which means X and Z have at least one key in common. Nevertheless, when m is large enough, C in key graphs converges to C in random graphs. In the case of $n = 100$ and $N = 1000$ in our experiments, when the key ring size is 25 or more, there is hardly any difference between the global clustering coefficients C in the two graph families.

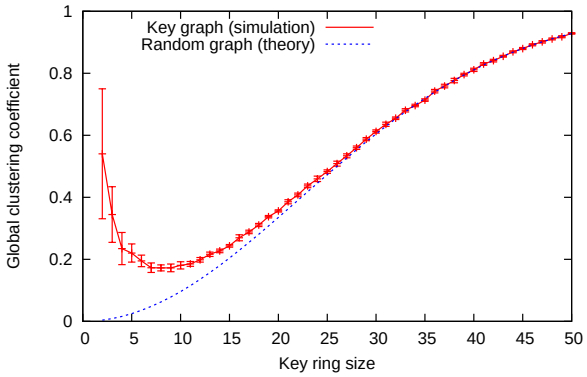

Fig. 3. Global clustering coefficient with respect to key ring size ($n = 100$, $N = 1000$)

4.2 Size of the maximal clique

In the second set of simulation experiments, we study the maximal clique in key graphs. The simulations assume that N is 1000 and m is 11 (i.e., edge probability $p \simeq 0.1$). The comparison results summarized in Figure 4 show that the average size of the maximal clique in key graphs increases linearly with the network size n. On the other hand, the expected size of the maximal clique in random graphs grows very slowly as n increases. In the following, we give a lower bound of the

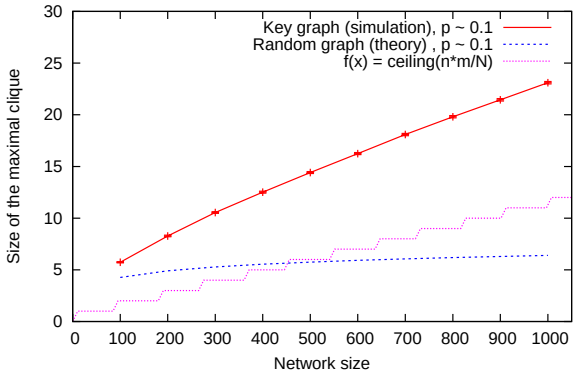

Fig. 4. Size of the maximal clique with respect to network size ($N = 1000$, $m = 11$)

size of the maximal clique in a key graph, to explain the linear behavior.

Proposition 1. *The maximal clique size a in key graph is at least* $\lceil \frac{n \cdot m}{N} \rceil$.

Proof: Each of n nodes has m keys, and the total number of keys including duplicated ones is $n \cdot m$ keys. Since the number of distinct keys is at most N, by

pigeon hole principle, there exists some key k duplicated at least $\lceil \frac{n \cdot m}{N} \rceil$ times. In other words, at least $\lceil \frac{n \cdot m}{N} \rceil$ nodes have the same key k, hence they form a clique. Since keys are distributed randomly, the frequencies of occurrence for each key may vary in a given random key assignment. That is, the number of nodes having the same key can well exceed $\lceil \frac{n \cdot m}{N} \rceil$. Furthermore, it is not required that all the nodes in a clique must have the same key. Thus, $\lceil \frac{n \cdot m}{N} \rceil$ is a loose lower bound for the size of the maximal clique in a key graph. When N and m are fixed, this lower bound increases linearly as n increases. That means the actual size of the maximal clique grows at least linearly with the network size. □

4.3 Number of cliques with respect to clique sizes

Figure 5 plots the number of cliques in key graphs and random graphs. In these experiments, we choose $n = 100$ and $N = 1000$. We use the two values of $m = 11$ and $m = 15$ so that edge probability is approximately 0.1 and 0.2, respectively. The comparison results suggest that given the same number of nodes and the

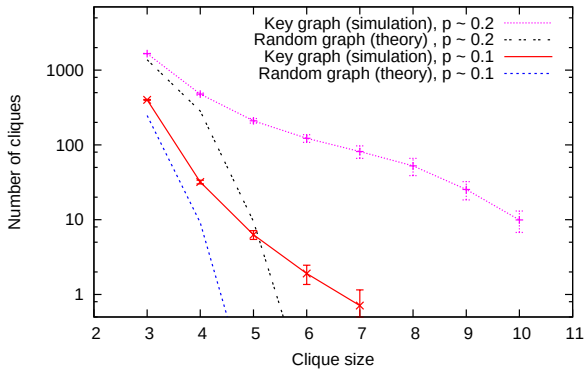

Fig. 5. Number of cliques in a key graph and a random graph for different edge probabilities when $n = 100$ and $N = 1000$.

same edge probability, cliques tend to be larger and more plentiful in key graphs than in random graphs. Specifically, $G_k(100, 1000, 15)$ contains more than 200 cliques of size 5 on average, while there are fewer than 10 cliques of that size in a random graph with the same number of nodes and the same edge probability. Moreover, cliques of size up to 10 can be observed in $G_k(100, 1000, 15)$, as opposed to the random graphs where the expected number of cliques of size 6 (or larger) is negligible.

Explanation of the larger number of cliques in key graphs: Let S_k be the set of all nodes that have the key k. In key graphs, any set $S_k \neq \emptyset$, forms a clique regardless of edge probability. In random graphs, however the expected number of cliques is a function of n and p (see Equation 3). Additionally, because of

i) the maximal clique in key graphs is much larger than that in random graphs (see Figure 4), and ii) any non-empty subset of nodes in a clique is also a clique itself, one expects more cliques in key graphs.

We can also use the global clustering coefficient to explain the formation of cliques in key graphs. As noted earlier, when m is small, if two nodes have a common neighbor in the key graph, there is a higher chance that they are connected by an edge. In general, the more common neighbors that two nodes X and Y have, the higher the probability that the edge XY exists. Thus, given a set of nodes such that many pairs of nodes are connected by an edge, the transitivity property implies there would be even more pairs that are directly connected, and the probability of this set of nodes being a clique increases. In contrast, a set of nodes S in random graphs forms a clique only when all the independent edge formation events between every pair of nodes in S occur at the same time.

5 Conclusions and Future Work

We study the applicability of random graph theory in modeling secure connectivity of wireless sensor networks. We identify ranges of parameters for which random graph modeling is not applicable and suggest how one can estimate key predistribution parameters for such cases. Besides, we determine ranges of parameters for which random graph theory may give estimates with excessive error, as well as other ranges of parameters where random graph theory provides very accurate results. We also study various structural properties in two graph families, observing and discussing the similarities and differences in the structure of random graphs and key graphs. In future work, we may extend the study of applicability of random graph modeling when the wireless connectivity is taken into account. Finally, there are other structural properties that we may investigate as well.

References

1. B. Bollobás and P. Erdős, "Cliques in Random Graphs", *Mathematical Proceedings of the Cambridge Philosophical Society*, 80(3):419–427, 1976.
2. H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks", *Proceedings of IEEE Security and Privacy Symposium*, pp. 197–213, May 2003.
3. W. Diffie and M. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.
4. W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili, "A Pairwise Key predistribution Scheme for Wireless Sensor Networks", *Proceedings of 10th ACM Conference on Computer and Communications Security*, pp. 42–51, October 2003.
5. P. Erdős and A. Rényi, "On Random Graphs", *Publicationes Mathematicae*, 6:290–297, 1959.
6. L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks", *Proceedings of the 9th ACM Conference on Computer and Communication Security*, pp. 41–47, November 2002.

7. G. Grimmett and C. McDiarmid, "On Coloring Random Graphs", *Mathematical Proceedings of the Cambridge Philosophical Society*, 77:313–324, 1975.
8. R. Gupta, J. Musacchio, and J. Walrand, "Sufficient rate constraints for QoS flows in ad-hoc networks", *Ad-hoc Network*, 5(4):429–443, May 2007.
9. P. Holland and S. Leinhardt, "Transitivity in Structural Models of Small Groups", *Comparative Group Studies*, 2: 107–124, 1971.
10. D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks", *Proceedings of the 2nd ACM Workshop on Security of Ad-hoc and Sensor Networks*, pp. 29–42, October 2004.
11. X. Huang and B. Bensaou, "On Max-min Fairness and Scheduling in Wireless Ad Hoc Networks: Analytical Framework and Implementation", *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 221–231, October 2001.
12. Y. Huang and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks". *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 135–147, 2003
13. J. Hwang and Y. Kim, "Revisiting random key predistribution schemes for wireless sensor networks", *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 43–52, October 2004.
14. D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks", *Proceedings of the 1st ACM Workshop on Security of Ad-hoc and Sensor Networks*, pp. 72–82, October 2003.
15. D. Liu, P. Ning, and R. Liu, "Establishing Pairwise Keys in Distributed Sensor Networks", *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 52–61, October 2003.
16. Massachusetts Institute of Technology, "Kerberos: The Network Authentication Protocol". `http://web.mit.edu/kerberos/`
17. R. Pietro, L. Mancini, A. Mei, A. Panconesi, and J. Radhakrishnan, "Redoubtable Sensor Networks", *ACM Transactions on Information and System Security*, 11(3):1–22, 2008.
18. J. Spencer, "The Strange Logic of Random Graphs", *Algorithms and Combinatorics*, Vol. 22, Springer-Verlag, ISBN 3-540-41654-4, 2000.
19. T. Vu, C. Williamson, and R. Safavi-Naini, "Simulation Modeling of Secure Wireless Sensor Networks", *Proceedings of ValueTools '09*, Pisa, Italy, October 2009.
20. D. Watts and S. Strogatz, "Collective Dynamics of 'Small-world' Networks", *Nature*, 393:440–442, 1998.
21. Y. Xue, B. Li, and K. Nahrstedt, "Optimal Resource Allocation in Wireless Ad Hoc Networks: A Price-Based Approach", *IEEE Transactions on Mobile Computing*, 5(4):347–364, 2006

Appendix

The attached appendix contains algorithmic details and additional tabular results for the paper.

Input:
- n: network size
- N: key pool size
- m: key ring size
- S: sample size

Output: connectivity c

$counter \leftarrow 0$
for $i \leftarrow 1$ **to** S **do**
  construct a key graph $G_k(n, N, m)$
  if $G_k(n, N, m)$ *is connected* **then**
    $counter{+}{+}$
  end
end
return $\frac{counter}{S}$

Algorithm 1: Determining connectivity

Input:
- n: network size
- N: key pool size
- $lowerBound$: lower bound on key ring size
- $upperBound$: upper bound on key ring size
- S: sample size in each simulation experiment
- c: desired connectivity

Output: Key ring size m

$lBound \leftarrow lowerBound$
$uBound \leftarrow upperBound$
while $uBound - lBound > 1$ **do**
  $mid \leftarrow (uBound + lBound)/2$
  if $connectivity(n, N, mid, S) \geq c$ **then**
    $uBound \leftarrow mid$
  else
    $lBound \leftarrow mid$
  end
end
return $uBound$

Algorithm 2: Binary search for key ring size

Table 5. Theoretical and simulated key ring sizes to achieve connectivity 99.9%

Network size n	Key pool size N = 100	500	1,000	5,000	10,000	50,000	100,000	
100	4	8	11	25	35	77	107	Simulation
	4	8	12	25	35	79	111	Theory
500	2	4	5	12	17	37	51	Simulation
	2	4	6	12	17	37	52	Theory
1,000	2	3	4	9	12	26	37	Simulation
	2	3	4	9	12	27	38	Theory
5,000	2	2	2	4	6	13	18	Simulation
	1	2	2	4	6	13	18	Theory
10,000	2	2	2	3	5	9	13	Simulation
	1	1	2	3	5	9	13	Theory

Table 6. Difference between theoretical and simulated key ring sizes

Network size n	Key pool size N = 10^2	$\frac{10^3}{2}$	10^3	$\frac{10^4}{2}$	10^4	$\frac{10^5}{2}$	10^5
10^2	0	0	0	0	0	0	1
$\frac{10^3}{2}$	0	0	0	0	0	0	0
10^3	-1	0	0	0	0	0	0
$\frac{10^4}{2}$	-1	-1	0	0	0	0	0
10^4	-1	-1	-1	0	0	0	0

(a) Desired connectivity $c = 50\%$

Network size n	Key pool size N = 10^2	$\frac{10^3}{2}$	10^3	$\frac{10^4}{2}$	10^4	$\frac{10^5}{2}$	10^5
10^2	0	0	0	1	1	0	1
$\frac{10^3}{2}$	0	0	0	0	0	1	1
10^3	-1	-1	0	0	0	-1	0
$\frac{10^4}{2}$	-1	-1	0	0	0	0	0
10^4	-1	-1	0	0	0	0	0

(b) Desired connectivity $c = 70\%$

Network size n	Key pool size N = 10^2	$\frac{10^3}{2}$	10^3	$\frac{10^4}{2}$	10^4	$\frac{10^5}{2}$	10^5
10^2	0	0	0	0	0	1	1
$\frac{10^3}{2}$	0	0	0	0	1	0	0
10^3	-1	0	0	0	0	0	0
$\frac{10^4}{2}$	-1	0	0	0	0	0	0
10^4	-1	-1	0	0	0	0	0

(c) Desired connectivity $c = 90\%$

Network size n	Key pool size N = 10^2	$\frac{10^3}{2}$	10^3	$\frac{10^4}{2}$	10^4	$\frac{10^5}{2}$	10^5
10^2	0	0	0	0	1	2	2
$\frac{10^3}{2}$	0	0	0	0	0	1	0
10^3	0	0	0	0	0	0	1
$\frac{10^4}{2}$	-1	0	0	0	0	0	0
10^4	-1	-1	0	0	0	0	0

(d) Desired connectivity $c = 99\%$