www.s2is.org

International Journal on Smart Sensing and Intelligent Systems

ISSN 1178-5608

SECRECY TRANSFER FOR SENSOR NETWORKS: FROM RANDOM GRAPHS TO SECURE RANDOM GEOMETRIC GRAPHS

Zhihong Liu, Jianfeng Ma, and Yong Zeng
School of Computer Science and Technology,
Xidian University, Xi'an, China.
Emails: {liuzhihong, jfma, yzeng}@mail.xidian.edu.cn

Submitted: Oct. 02, 2012 Accepted: Jan. 6, 2013 Published: Feb. 20, 2013

Abstract: Suppose n nodes with n_0 acquaintances per node are randomly deployed in a two-dimensional Euclidean space with the geographic restriction that each pair of nodes can exchange information between them directly only if the distance between them is at most r, the acquaintanceship between nodes form a random graph, while the physical communication links constitute a random geometric graph. To get a fully connected and secure graph, we introduce a secrecy transfer algorithm which combines the random graph and the random geometric graph via an introduction process to produce an acquaintanceship graph $G_{n,n0}$. We find that the maximum component of graph $G_{n,n0}$ transitions rapidly from small components to a giant component when n_0 is larger than a threshold, the threshold is derived, and applications for sensor networks are presented.

Keywords- Random graph, Random geometric graph, Sensor networks, Security.

I. INTRODUCTION

Suppose at a classroom with *n* students, each of whom initially has n_0 acquaintances who are randomly chosen among them. Now mention to one of them a message, but ask that student to share this message only with his or her acquaintances. Assume students are not allowed to leave their sites, and can talk only with adjacent students. At first, students are isolated from each other. If two adjacent students are acquainted with each other, a link forms between them. As a consequence, subtle paths start connecting students who are still strangers to each other. For example, though John has not met Mary yet, they have both met Mike, and so there is a path from John to Mary through Mike. If John and Mary are neighbors, chances are that now they become new acquaintances through the introduction of Mike, and a link forms between them. As time goes on, some small acquaintance groups emerge. By following the links in the group, one can now find a path between any two students in the same group. Further, two stranger students, say Alice and Bob, belonging to different groups may be adjacent, but if there are any two students in the two groups respectively familiar with each other, Alice and Bob may use them as introducers to get common acquaintance and establish a link between them. Then, two groups melt to form a larger group. By repeating this process, the students will be increasingly interwoven by such links, creating a web of acquaintances. We denote this construction as secrecy transfer and the resulting network as an acquaintanceship graph. We are here interested in the question: for which value of n_0 is there likely to be a connected acquaintanceship graph that includes all the students after the secrecy transfer process?

The acquaintanceship graph, denoted as $G_{n,n0}$, combines random graph [1] and random geometric graph [2]. A random geometric graph $G_{n,r}$ is a graph resulting from placing *n* nodes randomly in a plane and connecting each pair of nodes iff their Euclidean distance is at most the radius *r*, whereas a random graph $G_{n,p}$ is a graph with *n* nodes in which each edge (out of the $\binom{n}{2}$ possible edges) is chosen independently at random with a probability *p*. Random graphs and random geometric graphs have been studied extensively, but in a separate way. Random graph and its variations have been used as models of social structure in, for example, epidemiology[3], while random geometric graph is always viewed as a wireless communication network[4][5], such as Ad hoc, Mesh, or sensor network. In fact, random graphs and random geometric graphs have different structural properties. Any two nodes in a random graph can be connected by a link with certain probability regardless of their geographical position. Random key graphs have recently been used by Di Pietro et al.[6] to model the random key predistribution scheme of Eschenauer and Gligor[7]. The random key graph is a random graph obtained as follows. *n* nodes, each assigned a subset of keys, are distributed uniformly at random on the given field. An edge is added if two nodes are within a radius *r* and share at least one common key. Formally, the resulting graph, matching a random graph with identical link probability to a random geometric graph, can be considered as the initial graph of the acquaintanceship graph $G_{n,n0}$. Note that, unlike random key graphs, secrecy transfer is a growth model, and can be considered as a stochastic process.

We are interested in the crucial property, connectivity, of the resulting acquaintanceship graph. Intuitively, we think that there is a threshold value. If n_0 is larger than that value and the underlying graph $G_{n,r}$ is connected physically, the graph $G_{n,n0}$ may be connected. In [8], we use secrecy transfer to enhance the security performance of key infection[9], but do not explore its properties. In this paper, some results are given and complemented by simulations.

II. SECRECY TRANSFER

Let *n* nodes distributed uniformly in a field, each of them has n_0 acquaintances. Assuming nodes A and B are adjacent. At first, A and B are connected if they are acquainted with each other (Fig. 1a). If A and B are connected by a path, then an edge A-B is added (Fig. 1b). As time goes on, the graph $G_{n,n0}$ evolves continuously and gradually consists of components. If node A belongs to component C_A, and B has acquaint with at least one of nodes in C_A, say node C in C_A, we connect A and B by a new edge (Fig. 1c). For the case where A and B belong to different components C_A and C_B, if there exist two familiar nodes C and D in C_A and C_B respectively, we introduce an edge between A and B (Fig. 1d); Otherwise, A and B are disconnected at present stage. Continuing this process, *n* nodes are turned into a graph $G_{n,n0}$.

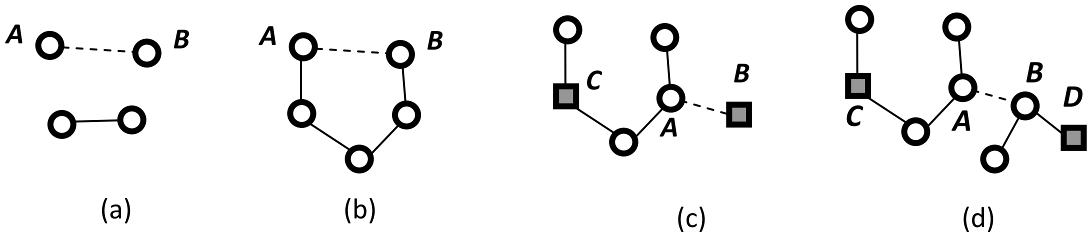

Fig. 1. Secrecy transfer.

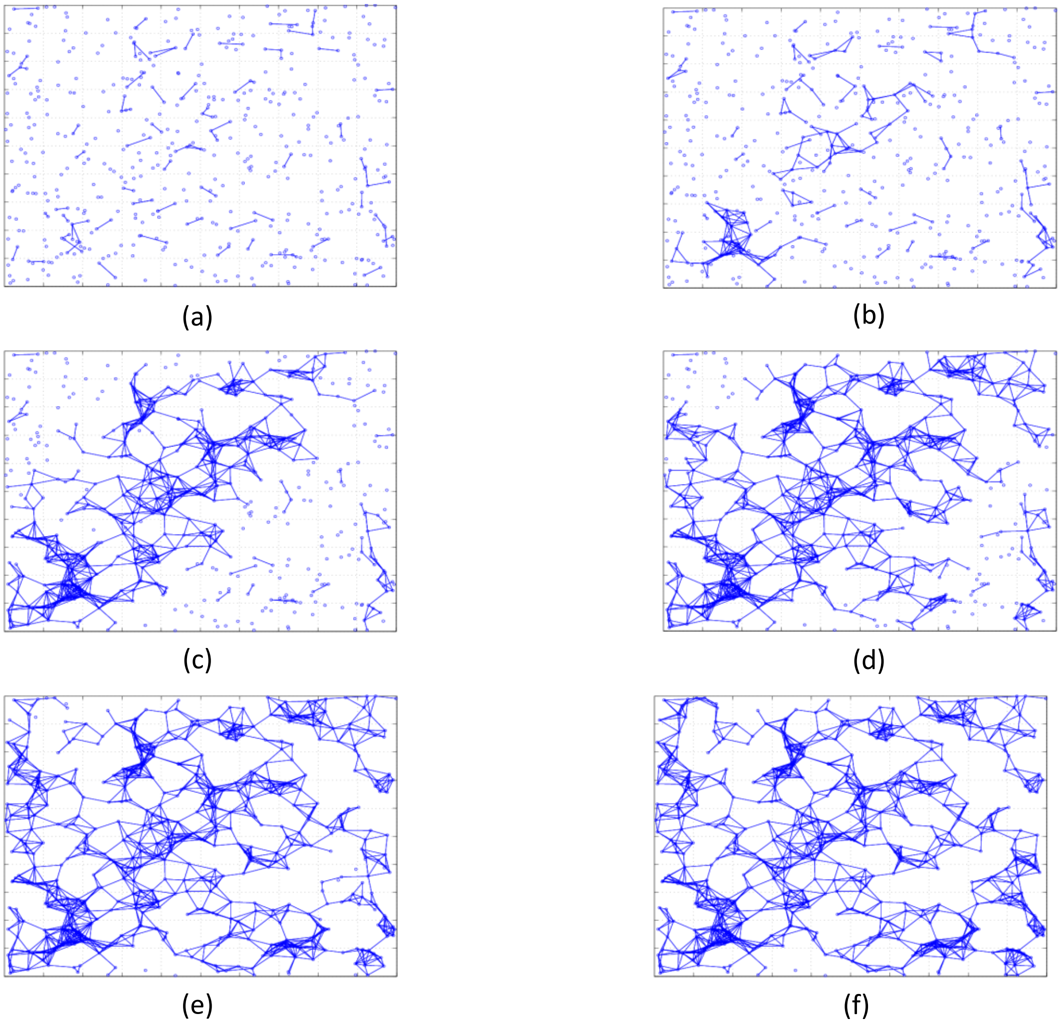

Fig. 2. An example of the secrecy transfer, with *n*=500 nodes randomly distributed over a 500× 500m^2 field, n_0=20, and *r*=35m.

As depicted in Fig. 2, 500 nodes are randomly distributed over a $500 \times 500m^2$ field, $n_0 = 20$, and the radius $r = 35m$. At first, two adjacent nodes connect with probability $p = n_0/n$, and we get the initial graph G_{n,n_0}, as illustrated in Fig. 2a. After repeating secrecy transfer, it gradually turns to be the graph in Fig. 1f, which approximates to random geometric graph $G_{n,r}$.

One of our goals is to design a security mechanism to enable any two adjacent nodes to establish a pairwise key after they are deployed. Suppose every node in the network has been loaded before its deployment with n_0 secret keys, each of which is shared with one of its acquaintances.

Let nodes A and B be two adjacent nodes, which means their Euclidean distance is at most the radius *r*. If A and B happen to be acquaintances, they must share a key K_{AB} which can be used to protect their communication link. If A and B are not acquaintances, but are connected by an existing path, A can generate a new key K_{AB} and send it to B along the secure path. Then, the key K_{AB} is shared between A and B. As more secure edges are added to the graph G_{n,n_0}, larger components emerge. Suppose A belongs to a component C_A, if node B is familiar with a node $C \in C_A$, which means that nodes B and C have a shared key K_{BC}. In this case, node A randomly generates a key K_{AB}, and sends it along the trusted path in the component C_A to node C. Node C encrypts K_{AB} with the key K_{BC}, $\{K_{AB}\}_{K_{BC}}$, and sends the result back to A. Node A, then, sends $\{K_{AB}\}_{K_{BC}}$ to B via the unsecure channel. Finally, node B gets key K_{AB}, for it has the key K_{BC}. In another case, where nodes A and B belong to different components C_A and C_B, but node $C \in C_A$ is familiar with node $D \in C_B$, as Fig. 1d. Node A first sends a key K_{AB} to node C. Node C encrypts K_{AB} with key K_{CD} which is shared with node D, and sends $\{K_{AB}\}_{K_{CD}}$ to node D via nodes A and B. For node D has K_{CD}, it can decrypt the message $\{K_{AB}\}_{K_{CD}}$ to obtain K_{AB}, as plotted in Fig. 3.

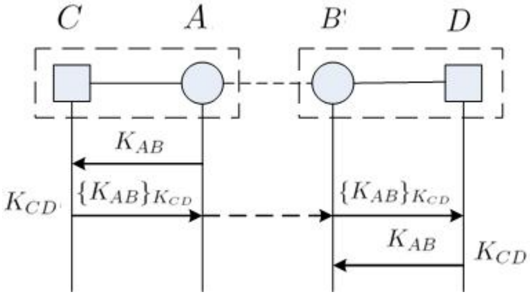

Fig. 3. Secret key establishment.

Given a randomly deployed network with *n* nodes, we can view it as a random geometric graph $G_{n,r}$ with each edge representing a possible communication link. Without the protection of a secret key, an adversary can eavesdrop conversations between two nodes. If each node has several trusted nodes initially, the trust relationship can be considered as a random graph $G_{n,p}$ with each edge connecting a pair of nodes which have established a secret key. However, random graph does not consider the transmission radius of nodes, but simply assumes any two nodes have the same probability *p* to establish a connection. When the distance between two nodes is larger than the radius *r*, they cannot communicate directly. Roughly speaking, $G_{n,p}$ reflects the logical trust relationship between nodes, while $G_{n,r}$ depicts the physical communication structure of nodes in the network. Secrecy transfer constructs a graph G_{n,n_0} from $G_{n,r}$ and $G_{n,p}$, and turns it to a secure random geometric graph approximate to $G_{n,r}$ by adding secure edges to it.

III. CONNECTIVITY THRESHOLD

The component structure of G_{n,n_0} changes gradually as the secrecy transfer is applied. As illustrated in Fig. 2, at first, the greatest component of G_{n,n_0} is tree or cycle of small order. Gradually, a giant component emerges, swallowing the whole network. Suppose two adjacent components, C_A and C_B, have respectively m_1 and m_2 vertices, node A in C_A and B in C_B are adjacent. We first estimate the probability P_{m_1,m_2} that two components C_A, C_B may get connected and melt into a larger component.

Let random variable $\mathbb{X}$ be the total number of nodes with whom the nodes in component C_A are familiar, $\mathbf{X}_i$ be a bernoulli random variable, where $\mathbf{X}_i = 1$ when the cycle of acquaintances of node *i* includes at least one node in the component C_A, $\mathbf{X}_i = 0$ otherwise. Therefore,

$$\mathbb{X} = \mathbf{X}_1 + \mathbf{X}_2 + ... + \mathbf{X}_n.$$

If component C_A consists of m_1 nodes, we have the probability of $\mathbf{X}_i = 1$

$$\mathbb{P}(\mathbf{X}_i = 1) = 1 - (1-P)^{m_1}, \quad \text{where } P = n_0/n.$$

Thus, the expectation of random variable $\mathbf{X}_i$ is $\mathbb{E}(\mathbf{X}_i) = 1 - (1-P)^{m_1}$.

For $\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n$ are mutually independent, the expectation of $\mathbb{X}$ is

$$\mathbb{E}(\mathbb{X}) = \sum_{i=1}^{n} \mathbb{E}(\mathbf{X}_i) = n[1 - (1-P)^{m_1}],$$

For a component of order m_1, the cycle of acquaintances of this component may consist of $n[1-(1-P)^{m_1}]$ nodes on average. Let $a = n[1-(1-P)^{m_1}]$, the probability P_{m_1,m_2} that there is at least one common acquaintance between component C_A and C_B is

$$P_{m_1,m_2} = 1 - \frac{\binom{n-m_1}{m_2}\binom{n-m_1}{a-m_1}}{\binom{n}{m_2}\binom{n}{a-m_1}}.$$

For example, for $n = 10,000$, $m_1 = 200$, and $m_2 = 1$, the probability P_{m_1,m_2} tends to 1 when $P > 0.02$. This provides intuition that, a component of order 200 is attractive and will swallow nodes nearby to form a larger component, a kind of rich get richer phenomenon. For two component of order $m_1 = m_2 = 50$, the probability P_{m_1,m_2} approximates 1 if $P > 0.002$. In general, the larger the components, the more likely they are to be mixed together. Popularity is attractive.

In a random graph $G_{n,N(n)}$ with *n* vertices and $N(n)$ edges, if $N(n) \sim cn$ with $c \geq \frac{1}{2}$, the greatest component has (with probability tending to 1 for $n \to +\infty$) approximately $n^{\frac{2}{3}}$ vertices. As a special case, when $n = 10,000$, $n^{\frac{2}{3}} \approx 464$, such large component in G_{n,n_0} will swallow the whole network **whp**. if the network is connected physically.

To determine the value *n₀* which will guarantee the connectivity of G_{n,n_0}, we employ the well-known algorithm [10] to generate random graphs $G_{n,p}$ with *n* nodes and n_0 links per node where $n_0 = np$, then deploy the nodes into a square region to obtain a random topology. For $n = 500$, $r = 35m$, and n_0 varying, we repeat our simulations 100 times to yield an acceptable confidence of results. For each simulation, we measure empirical values for the maximum component and the second component for each trial, averaged over 50 random topologies. In Fig.4, an interesting phenomenon observed is a phase transition as n_0 increases. There is a critical value of n_0, above which the graph will almost surely be connected. The maximum component transitions rapidly from a component of small size to a giant component when $n_0 > 10$. Similarly, the size of the second component decreases as $n_0 > 10$.

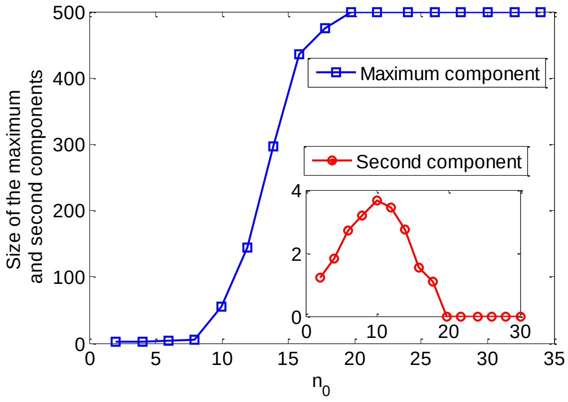

Fig 4. Size of the maximum and second components in graph G_{n,n_0} for $n = 500$,$r = 35m$.

Within this context, the question is, under what conditions is the graph G_{n,n_0} be connected? More specifically, how can we choose n_0 such that with high probability, the graph G_{n,n_0} constructed by secrecy transfer will be connected.

Consider an arbitrary pair of adjacent nodes A and B in graph G_{n,n_0} which have not established secret key between them. If $G_{n,p}$ is connected, there is at least one path in graph $G_{n,p}$, say

$P_{AB} = Ax_0x_1...x_kB$ between nodes A and B. Given any adjacent nodes in path P_{AB}, say x_i and x_{i+1}, there must exist a path $P' = x_iy_1y_2...y_tx_{i+1}$ from x_i to x_{i+1} in graph $G_{n,r}$, if graph $G_{n,r}$ is connected. Thus, arbitrary pair of nodes can establish a secret link in graph G_{n,n_0} by secrecy transfer. Hence, by the addition of secure links, G_{n,n_0} can be eventually turned into a secure and connected graph which is approximate to $G_{n,r}$.

Thus, to get a fully connected graph G_{n,n_0}, two conditions must be satisfied. First, the graph $G_{n,r}$ must be connected, which means that, given the value *n* and a deployment region, the value *r* should be large enough to guarantee a connected graph $G_{n,r}$. Assume *n* nodes are uniformly deployed in a unit square $[0,1]^2$, the well-known connectivity threshold $r_c = \sqrt{\frac{\log n \pm O(1)}{\pi n}}$ [5]. In this paper, we set the radius *r* to be above this critical value to ensure the random geometric graph $G_{n,r}$ be fully connected. Second, the value n_0 must be large enough to get the random graph $G_{n,p}$ fully connected. For a random graph $G_{n,p}$, when *p* is zero, the graph does not have any edge, whereas when *p* is one, the graph is fully connected. Erdös and Rényi [1] showed that, for monotone properties, there exists a value of *p* such that the property moves from nonexistent to certainly true in a very large random graph. The function defining *p* is called the threshold function of a property. Given a desired probability P_c for graph connectivity, the threshold function *p* is defined by

$$P_c = \lim_{n\to\infty} P_r[G_{n,p}\ is\ connected] = e^{e^{-c}},$$

where $p = \frac{\ln(n)}{n} + \frac{c}{n}$ and *c* is any real constant.

Therefore, given *n* we can find *p* for which the resulting graph $G_{n,p}$ is connected with desired probability P_c. Thus, the connectivity threshold of n_0 is

$$n_0 = p \times (n-1) = \frac{n-1}{n}\left[\ln(n) - \ln(-\ln(P_c))\right].$$

IV. CONVERGENCE ANALYSIS

Consider a graph $G_{n,r}$ of *n* nodes with n_0 acquaintances per node randomly selected among the nodes in the graph, we are also interested in the time needed for secrecy transfer to reach a stable state. The speed of the convergence of secrecy transfer depends on the values of n_0, *r* for given *n*. To gain insight, we first consider the value *r* and perform a simulation-based study of it.

Employing a uniform random generator, we position $n = 500$ nodes in a square planar region of $500 \times 500m^2$, following our deployment from Section 3. For each random topology, we estimate the speed of the convergence of secrecy transfer as the number of rounds that it needs to perform to reach a stable state. At each round, each pair of adjacent nodes in the graph G_{n,n_0} employ secrecy transfer to try to get connected. If there is no new edge is added in this round, secrecy transfer terminates. We observe from Fig. 5 that, as the value *r* increases, the stable state is reached with a speed that is faster, and for value n_0, the number of rounds reaches its peak when n_0 approximates to its connectivity threshold.

Conventionally, a wireless network consists of some nodes as supernodes, those using a communication radius greater than used by normal nodes. The use of these supernodes lead to important characteristics of complex networks [11]: a small average shortest path length between all nodes, and a high cluster coefficient, which help us saving network resources, avoiding excessive communication, and reducing the time to data delivery. Fig. 6 depicts plots of a secrecy transfer with $n = 500$ nodes deployed over a $500 \times 500m^2$ field, $n_0 = 20$, $r = 35m$, among them there are 25 supernodes with a larger communication radius $R = 150m$.

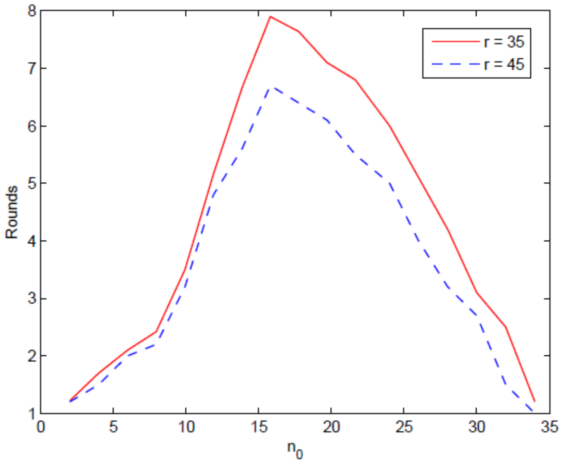

Fig 5. Value *n₀* vs. number of rounds of secrecy transfer for various values of the radius *r*.

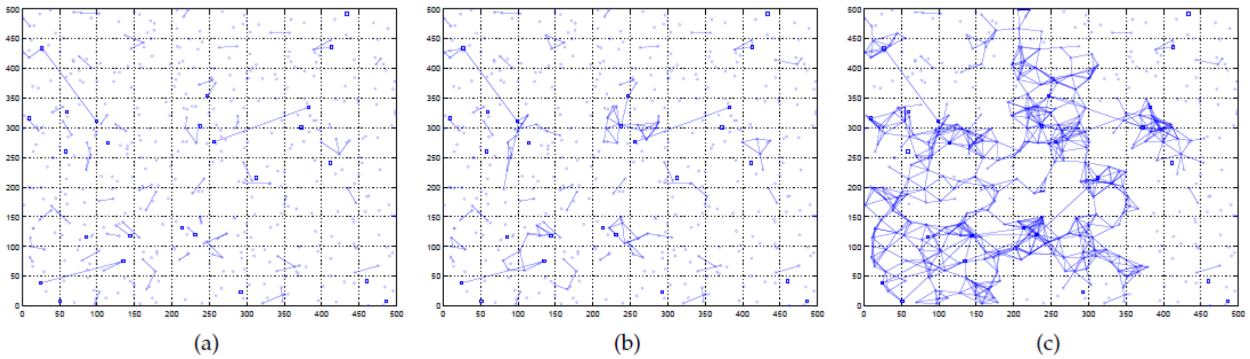

Fig 6. Secrecy transfer process in a heterogeneous network.

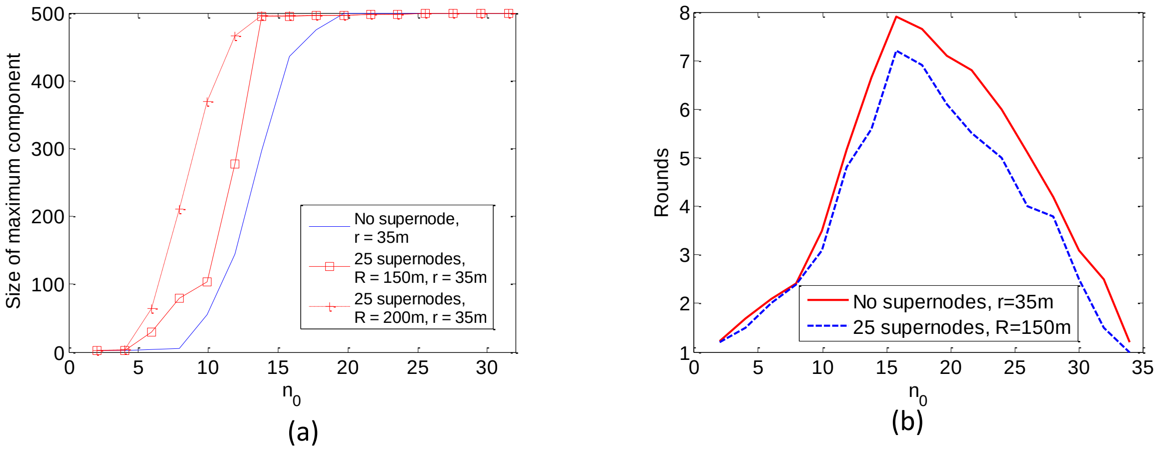

Fig 7. The maximum component and rounds of secrecy transfer process in heterogeneous networks.

From the simulation results illustrated in Fig. 7, we conclude that, compared to the homogeneous network case, for a heterogeneous network with supernodes, as the radius of supernodes R grows, the value of n_0 required to maintain connectivity of graph G_{n,n_0} decreases, the speed of the convergence of secrecy transfer accelerates. Hierarchically, the supernodes can form a higher layer, while the normal nodes constitute a lower layer of the network. An implication of a heterogeneous network is that it has better performance with regard to improving energy, power and topology control, scalability, and fault-tolerance and routing efficiency.

V. IMPLEMENTATION OF SECRECY TRANSFER

In this section, we elaborate the implementation method of secrecy transfer. The method contains three phases: the initialization phase, the secrecy transfer phase, and the update phase. To implement secrecy transfer efficiently, we use Bloom Filter [15] for membership queries.

Bloom Filter: A Bloom Filter is a popular data structure used for membership queries. It represents a set $S = [s_1, ..., s_n]$ using k independent hash functions $h_1, ..., h_k$ and a string of m bits, each of which is initially set to 0. For each $s \in S$, we hash it with all the k hash functions and obtain their values $h_i(s)$ $(1 \leq i \leq k)$. The bits corresponding to these values are then set to 1 in the string. To determine whether an item s' is in S, bits $h_i(s')$ are checked. If all these bits are 1s, s' is considered to be in S.

Since multiple hash values may map to the same bit, Bloom Filter may yield false positives. That is, an element is not in S but its bits $h_i(s)$ are collectively marked by elements in S. If the hash is uniformly random over *m* values, the probability that a bit is 0 after all the *n* elements are hashed and their bits marked is $(1 - \frac{1}{m})^{kn} \approx e^{-\frac{kn}{m}}$. Therefore, the probability for a false positive is $(1 - (1 - \frac{1}{m})^{kn})^k \approx (1 - e^{-\frac{kn}{m}})^k$. The right hand side is minimized when $k = \frac{m}{n} \ln 2$ in which case it becomes $(\frac{1}{2})^k = (0.6185)^{m/n}$.

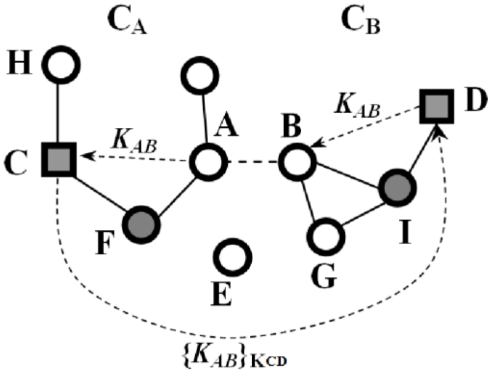

Fig 8. Secrecy transfer phase.

Initialization phase:

We first generate a random graph with *n* nodes and n_0 links per node. For each link a secret key is assigned to it. Each node stores the ID of its neighbors and the corresponding secret key between them. For instance, if node *i* has n_0 neighbors $i_1, ..., i_{n_0}$, it constructs an *acquaintanceship set* $A_i = \{(i_1, K_{i,i_1}), ..., (i_{n_0}, K_{i,i_{n_0}})\}$, where K_{i,i_1} is the assigned secret key between node *i* and its neighbor i_1. After that nodes are deployed randomly over a field.

Secrecy transfer phase:

Suppose two adjacent components, C_A and C_B, have respectively m_1 and m_2 nodes, nodes $A \in C_A$ and $B \in C_B$ are adjacent. For component C_A, a *component head* is selected[1]. He stores all the ID of nodes belonging to the component C_A in a *component member set*

$$CM_{C_A} = \{a_1, ..., a_{m_1}\}, \text{ where } a_i \in C_A.$$

Each node stores a Bloom Filter BF_{C_A} which contains all the nodes in the acquaintance circle of C_A. i.e., the nodes in C_A and the acquaintances of node *i* for all $i \in C_A$. If an adjacent node *k* is added to C_A, the Bloom Filter BF_k of node *k* is inserted into BF_{C_A}, i.e., a new Bloom Filter $BF_{C'_A}$ for the new component $C'_A = C_A + k$ is created, i.e., $BF_{C'_A} = BF_{C_A} + BF_k$.

If two components C_A and C_B get connected and melt into a larger component C_{AB}, a new Bloom Filter of component C_{AB} , $BF_{C_{AB}} = BF_{C_A} + BF_{C_B}$, is created and stored in nodes of C_{AB}. To further improve the performance, not all nodes in C_A or C_B need to update its BF_{C_A} or BF_{C_B} to $BF_{C_{AB}}$, only nodes whose neighbors are not all connected to them need to store the updated Bloom Filter $BF_{C_{AB}}$ of the new component C_{AB}. As depicted in Fig. 8, C_A and C_B melt into a larger component C_{AB}, an isolated node E is adjacent to nodes A, B, F, and *G*. After C_A and C_B get connected, only nodes *A*, *B*, *F*, and *G* in C_{AB} have unconnected neighbor. Therefore, they need to store the new $BF_{C_{AB}}$ and will broadcast it later.

Next, we give an overview of the operations of secrecy transfer. In general, the operation of secrecy transfer is initiated by a new created component. Let C_A be a new component that has ``swallowed" node *H*, nodes *A* and *F* have already updated their BF_{C_A} (to insert the ID of node *H* into it), and let C_B be an adjacent component of C_A. After that, nodes *A* and *F* broadcast BF_{C_A} to their adjacent nodes *B* and *E*. On receiving the BF_{C_A} from component C_A, node *B* sends a query message containing BF_{C_A} to the component head of C_B, say node *I*, where the component member set of C_B, $CM_{C_B} = \{b_1, ..., b_{m_2}\}$, is stored. The component head *I* then

[1] At first after initialization phase, each node is a component head of its own since all nodes are isolated. After several rounds of secrecy transfer process, some large components emerge. To reduce the communication cost, a node is selected to be a component head according to its centrality in the component. To simply the procedure, the node with the highest degree is chosen to be the component head

determines whether the nodes in set CM_{C_B} are in the Bloom Filter BF_{C_A}. If a node, say $D \in CM_{C_B}$, is found in the Bloom Filter BF_{C_A}, node *I* answers node *B* by sending *D* to it. Node *B* then tells *A* that there is a node $D \in C_B$ belonging to the acquaintance circle of component C_A.

After that, nodes *A* broadcasts a query message with the ID of node *D* in component C_A. Each node in C_A verifies whether node *D* belongs to its acquaintanceship set. As illustrated in Fig. 8, if the acquaintanceship set of node $C \in C_A$ contains node *D*, i.e., $A_C = \{..., (D, K_{CD}), ...\}$, the node *C* transmits a response message (*C*, *D*) to node *A*. After obtaining the acquaintance node pair (*C*, *D*) from *C*, node *A* knows that nodes $C \in C_A$ and $D \in C_B$ are acquaint with each other (they have a shared key K_{CD}). Now nodes *A* and *B* can establish a secret key K_{AB}.

Update phase:

After the secret key K_{AB} between nodes *A* and *B* is established, two components become a larger component C_{AB}, we then should update the acquaintance circle of C_{AB} for nodes who have unconnected neighbors. A new component head is also need to be selected according to the degree distribution of nodes in C_{AB}. As to the network in Fig. 8, if node *I* is the new component head of C_{AB}, the component member set is updated to be

$$CM_{C_{AB}} = CM_{C_A} + CM_{C_B} = \{a_1, ..., a_{m_1}, b_1, ..., b_{m_2}\}.$$

Finally, if nodes have updated their Bloom Filter $BF_{C_{AB}}$, they broadcast the new $BF_{C_{AB}}$ to their neighbors to find chances for new links. Recursively, this procedure is applied until there is no node has updated its Bloom Filter.

Security analysis

As discussed in Section II, secrecy transfer is robust against eavesdrop attack, for each edge is added via the existing trustiness between nodes. In this subsection, we study the resilience of secrecy transfer against the node compromise attack. Let n_c denote the number of nodes that have been captured. Suppose the compromised nodes are independently and random distributed among the entire deployment region.

Theoretically, as depicted in Fig.8, if any node in the paths *A-F-C* and *D-I-B* is compromised, the key K_{AB} between nodes *A* and *B* is not secure. Suppose the length of two paths are l_1 and l_2, respectively. It is easy to estimate the probability that a new established key K_{AB} is compromised as following

$$P\{K_{AB} \text{ is compromised}\} = 1 - \frac{\binom{l_1+l_2}{n-n_c}}{\binom{l_1+l_2}{n}},$$

where *n* is the number of node in the network.

Unfortunately, even if all nodes in two paths are not compromised, the key K_{AB} may be unsecure. For instance, let a path from *A* to *C* be $A - H_1 - H_2 - H_3 - H_4 - C$, and all nodes in the path have not been compromised. Node *A* sends K_{AB} to H_1 by sending $\{K_{AB}\}_{K_{AH_1}}$, H_1 then transmits $\{K_{AB}\}_{K_{H_1H_2}}$ to node H_2 until K_{AB} reaches the last node *C*. If $K_{AH_1}, K_{H_1H_2}, ..., K_{H_4C}$ are not compromised, K_{AB} is still secure after it is transmitted across the path. However, if a key, such as $K_{H_1H_2}$, is compromised, an adversary may eavesdrop on the communication flows between nodes H_1 and H_2 to obtain $\{K_{AB}\}_{K_{H_1H_2}}$, thus K_{AB} is leaked.

In general, if there are compromised nodes in the network, any key established by secrecy transfer between two neighbors H_1 and H_2 may be unsecure unless nodes H_1 and H_2 are acquaint with each other initially. For any pair of acquaintance nodes, the secret key between them is preloaded before the network is deployed and is considered unbreakable (unless the node is compromised). As to any key established by secrecy transfer, compromised nodes may degrade its security since lots of nodes are involved in the process of the negotiation of a new link key.

In order to set up a more secure channel between nodes *A* and *C*, it is reasonable to use the acquaintanceship set of nodes. Suppose in a path $A - H_1 - H_2 - H_3 - H_4 - C$, (*A*,$H_3$), ($H_1$, H_3), and (H_1, *C*) are three pair of acquaintances. To send a secret key K_{AB} to *C*, node *A* can send $\{K_{AB}\}_{K_{AH_3}}$ to H_3, H_3 then sends $\{K_{AB}\}_{K_{H_1H_3}}$ to H_1. At last, node *C* can get $\{K_{AB}\}_{K_{H_1C}}$ from H_1. The advantage of this method is that all communications are encrypted with pre-distributed keys. If nodes *A*, *C*, H_1, and H_3 are not compromised, the key K_{AB} is secure after the transmission. However, such a secure logical path in a set of nodes may not exist. For a path of *l* nodes, their initial acquaintanceship can be viewed as a random graph $\hat{G}_{n,p}$, where $n = l$ and $p = \frac{n_0}{n}$. If $\hat{G}_{n,p}$ is connected, a logical path exists.

If an adversary is not present at the network before secrecy transfer has completed, or it takes more time than a secure interval to compromise nodes, the communication links established by secrecy transfer are secure; otherwise, undetected malicious nodes may degrade the security of secrecy transfer and jeopardize the network. In [16], authors investigated the potentially disastrous threat of node compromise spreading (via communication and pre-established mutual trust) in wireless sensor networks, and proposed an epidemiological model to investigate the probability of a breakout. This model can be adapted to analyze the spread of malicious behavior of compromised nodes in the process of secrecy transfer. But how to design efficient countermeasures is still unknown.

V. APPLICATIONS OF SECRECY TRANSFER

Sensor networks have been envisioned to consist of groups of lightweight sensor nodes that may be randomly and densely deployed to observe data within a physical region of interest. The nodes form an ad hoc multihop network, communicating readings to base stations. The connectivity of these self-organizing networks is critical for reliable sensing and inference capabilities [12]. Conventionally, sensor network is modeled as a random geometric graph $G_{n,r}$, two nodes A and B establish a bidirectional link if they are within a radius *r*. To protect the sensitive data in hostile environments, secret keys should be established to achieve data confidentially, integrity and authentication between communicating parties [13]. The first practical key predistribution scheme for sensor network is random key predistribution scheme introduced by Eschenauer and Gligor [7]. For a pool size $S = 100,000$ keys, 250 keys need to be stored in a node's memory to have the probability that they share a key in their key sets to be $p = 0.5$. A major advantage of this scheme is the exclusion of the base station in key management. Disadvantages of it are that it is not suitable for sparse deployed networks where the number of adjacent nodes of any node is small, and the storage overhead is still high for lightweight nodes (many keys in node's key set are not used finally). As mentioned previously, secrecy transfer can turn a random graph to a secure random geometric graph. If the secrecy transfer is applied with random key predistribution scheme, the storage overhead of nodes is lower.

In [14], an asymmetric key predistribution scheme AKPS for sensor network is proposed. In AKPS, each node only stores two secret values initially, a large amount of storage is shifted to keying material servers (KMS). If AKPS needs to provide public keying material for any pair of nodes, a KMS should store $\binom{n}{2}$ public keying material for a network of *n* nodes. Roughly speaking, AKPS is not viable for arbitrary large network. We find that, if secrecy transfer is used, a KMS does not need to be preloaded with $\binom{n}{2}$ public keying material. Specially, suppose n^* out of $\binom{n}{2}$ public keying material are randomly picked, the initial probability that two arbitrary sensors can establish a secret key is $p = \frac{2n^*}{n(n-1)}$, which means that, any nodes has $n_0 = n \times p$ acquaintances on average. As before, if n_0 is larger than the connectivity threshold in graph $G_{n,p}$, we can repeat the construction process of secrecy transfer to get a connected graph G_{n,n_0} which will guarantee that any pair of adjacent nodes can establish secret keys.

VI. CONCLUSIONS

This work presented a secrecy transfer algorithm which is directly based on the idea that networks form primarily by people introducing pairs of their acquaintances to one another. The resulting network, showing both properties of random graph and random geometric graph, may not only model the introduction process in social networks, but also be used to protect the network. In fact, secrecy transfer, a localized algorithm which does require global knowledge of the network, can achieve the desired global behavior.

If an adversary is not present at the network before secrecy transfer has completed, or it takes more time than a secure interval to compromise nodes, the communication links established by secrecy transfer are secure; otherwise, undetected malicious nodes may degrade the security of secrecy transfer and jeopardize the network. How to build a distribution model for the spread of the malicious behavior of compromised nodes and design efficient countermeasures against such active attack are parts of our future researches.

VII. ACKNOWLEGEMENT

This work is partly supported by Program for Changjiang Scholars and Innovative Research Team in University (IRT1078), the Key Program of NSFC-Guangdong Union Foundation (U1135002), Major national S&T program (2011ZX03005-002), National Natural Science Foundation of China (61173135, 61100230, 61100233, 61202389, and 61202390), and the Natural Science Basic Research Plan in Shaanxi Province of China (2009JQ8005, 2011JM8004, 2012JM8025, and 2012JQ8043).

REFERENCES

[1] Bollobás, B.. Random Graphs, 2nd ed. Cambridge University Press, (2001).

[2] Penrose, M.. Random Geometric Graphs. Oxford studies in probability. Oxford University Press, Oxford, (2003).

[3] Anderson, R. M. and May, R. M. Susceptible-infectious-recovered epidemic model with dynamic partnerships. Journal of Mathematical Biology, vol. 33, pp. 661–675, (1995).

[4] Bettstetter, C. On the minimum node degree and connectivity of a wireless multihop network. Proceedings of the third ACM International Symposium on Mobile ad hoc Networking and Computing (MobiHoc 02). ACM, (2002), pp. 80–91.

[5] Gupta, P. and Kumar, P. The capacity of wireless networks. IEEE Transactions on Information Theory, vol. 46, no. 2, pp. 388–404, March (2000).

[6] Pietro, R. D., Mancini, L., Mei, A., Panconesi, A. and Radhakrishnan, J. Redoubtable sensor networks. ACM Transactions on Information Systems Security, vol. 11, no. 3, pp. 13–22, (2008).

[7] Eschenauer, L. and Gligor, V. A key management scheme for distributed sensor networks. Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS’02, (2002), pp. 41–47.

[8] Liu, Z., Ma, J., Pei, Q., Pang, L., and Park, Y. H. Key infection, secrecy transfer, and key evolution for sensor networks. IEEE Transactions on Wireless Communications, vol. 9, no. 8, pp. 2643–2653, August (2010).

[9] Anderson, R., Chan, H., and Perrig, A. Key infection: Smart trust for smart dust, ICNP’04, Berlin, Germany, (2004), pp. 24–31.

[10] Bollobás, B. A probability proof of an asymptotic formula for the number of labelled regular graphs. European Journal of Combinatorics, vol. 1(1980), pp. 311–316, (1980).

[11] Newman, M. E. J. The structure and function of complex networks. SIAM Review, vol. 45, pp. 167–256, (2003).

[12] Freris, N. M., Kowshik, H., and Kumar, P. R. Fundamentals of large sensor networks: Connectivity, capacity, clocks, and computation. Proceedings of the IEEE, vol. 98, no. 11, pp. 1828–1846, November (2010).

[13] Giruka, V. C., Singhal, M., Royalty, J., and Varansi, S. Security in wireless sensor networks. Journal of Wireless Communications and Mobile Computing, vol. 8, pp. 1–24, (2008).

[14] Liu, Z., Ma, J., Huang, Q., and Moon, S. J. Asymmetric key predistribution scheme for sensor networks. IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1366–1372, March (2009).

[15] Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. Communications of ACM, vol. 13, no. 7, pp. 422–426, 1970.

[16] De, P., Liu, Y., and Das, S. K. Deployment-aware modeling of node compromise spread in wireless sensor networks using epidemic theory. ACM Transactions on Sensor Networks., vol. 5, no. 3, pp. 23:1–23:33, 2009.