Analysis of a New Random Key Pre-distribution Scheme for WSN Based on Random Graph Theory and Kryptograph

Seema Verma and Prachi

Abstract— Wireless Sensor Networks (WSNs) vast myriad of futuristic applications makes it matter of incessant research interest. Key management is crucial for WSN due to their high security requirements and resource constrained nature. Randomized key pre-distribution seems to be best suited solution for WSN due to scarceness of resources. However, most of the earlier proposed schemes are based on random graph theory model which is not that suitable for WSN. In this paper we present and implement a new randomized key pre-distribution scheme on TinyOS. Later on, we perform a rigorous mathematical analysis of our scheme under random graph theory on which most of the earlier proposed schemes are based and recently introduced kryptograph model. Our results prove that kryptograph model is more vital for secure WSNs.

Index Terms— Random key pre-distribution, random graph theory, kryptograph.

I. INTRODUCTION

WSN compromises of huge number of ultra-small autonomous sensors also known as motes capable of sensing physical and environmental conditions. Sensor networks work in collaborated peer-to-peer environment to aggregate data from various autonomous sensors deployed in ad-hoc manner not in a predefined architecture. WSN can be widely used in number of diversified applications, for instance, unattended surveillance, disaster management (railways, volcanoes), nature reserves (water reservoir), in seismically threatened structures and for communication in infrastructureless wireless environment. As sensor’s power supply is based on battery, best optimization of communication and computation becomes mandatory. Secure communication acts as pre-requisite for secure routing. Further, when the thrust is miniaturization, provision of security features becomes challenging [1]. Especially, in hostile environment where security and privacy of data is very crucial and can be snooped very easily, implementation of security features becomes most crucial for WSN survival. To embark upon security problem of WSN, viable and efficient cryptographic mechanisms are imperative.

As cryptographic primitive need secret keys for operation, this escalate need for a secure and appropriate key management algorithm which is efficient in all aspects like memory, energy etc. 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 [2]. In last few years a number of light-weight key management algorithms have been presented.

In this paper, we like to present a new static random key pre-distribution approach as this type of scheme removes the overhead of dynamic key generation as well as dependency on base station. Our scheme is specifically designed for homogenous networks (all nodes are alike) because they are most general to use in WSNs. Our proposal can be easily embedded in existing key pre-distribution scheme to enhance their performance.

The remainder of this paper is organized as follows: Section 2 give literature review of some of the previously presented random key pre-distribution schemes for homogenous networks. Our new scheme is presented in section 3. Section 4 elaborates mathematical analysis for both random graph theory and kryptograph model. Results for two previous models are followed by section 5. Finally, section 6 draws conclusion.

II. LITERATURE REVIEW

Here, our idea is to discuss few probabilistic approaches proposed earlier for WSN where each node receives only a share of total number of keys to ensure that whole network is connected with certain probability. It addresses the problem of memory overhead of full pairwise scheme and low security of single network wide key.

Eschenauer et al. in [3] pioneered first innovative randomized approach for key pre-distribution scheme of WSNs also known as basic scheme. A randomly selected subset of key from large size key pool is assigned to each node before deployment. Once deployed node exchange keys with neighbors. If two nodes share common key they are securely linked and use this key to encrypt their communication in future. Otherwise, they need to establish path key by finding intermediate node which can be securely linked to both of them.

Based on [3] Chan et al. [4] further extended the idea and proposed q-composite key pre-distribution scheme to increase
network resilience at the cost of processing overhead. In q-composite, pairwise key is computed from at least q (q=1) shared key between two nodes.

Hashed random key pre-distribution, slight modification to basic scheme was proposed by T. Shan et al. [5]. Here, keys of key pool are hashed different number of times for distinct nodes. Only first node get original key from key pool, jth node receives (j-1) times hashed version of original key. When nodes exchange keys they also exchange how many times their key is hashed. This scheme also increase network resilience as when a node get captured only its hashed key compromised, not original key and other hashed versions of that key.

To avoid path establishment, two key distribution methods were proposed by Law et al. [6] key redistribution scheme, forward and reverse key distribution. Suppose A and B wants to communicate but don’t share any key whereas A share key k1 with C and B share key k2 with C. Now, A will ask C to send k2 encrypted with k1 and delete it from its own memory. This is known as forward key distribution. If in case this key is not available, A will choose an unused key from its key pool say k3 and send it to C, then C compute k3, Hash(k3) and send E(k3) to A. A decrypt E(k3) with k1 to get k3 and forward E(k3) to B.

S. Zhu et al. introduced Pairwise Key Establishment protocol [7] to avoid communication overhead involved in the Shared-Key Discovery phase. Instead of randomly distributing subset of keys out of key pool pseudo random function uses node’s ID as seed. As a result, any node can determine which keys another node possesses only by knowing its node’s ID. This scheme is more communication efficient as compared to earlier suggested approaches since broadcast of list of key IDs is not required.

Despite much research effort key agreement still remains an open problem in WSNs.

### III. OUR NEW SCHEME

Our scheme introduces some changes to earlier suggested very popular basic scheme to make it more secure, attack resistant and communication efficient. We implemented this scheme using TinyOS mote simulator (TOSSIM) [8] to provide a high fidelity simulation for WSN. TOSSIM is specifically designed for TinyOS WSN and de-facto for testing, debugging, and analyzing TinyOS applications. Applications developed using TinyOS can be readily downloaded onto motes. We also use TinyViz (GUI tool for TOSSIM) for visualization of our new scheme and exchange of messages between sensors. Here, sensors are randomly distributed over a geographical area. In key pre-distribution phase, we generate pool K of keys from pseudo random number generator along-with their IDs. Afterwards, we assign a random subset of k keys (where k<<K) drawn from key pool to each node with their IDs. In shared key discovery phase, instead of broadcasting list of key IDs to all nodes we send key IDs only to nodes with which we actually wants to communicate. Instead of broadcasting key IDs to every node in probabilistic approach, establishing connection and communication with only desired neighbors not only make our scheme communication efficient but also save loads of resources for resource constrained WSN. Suppose A and B are within communication range and A want to communicate with B then A sends its list of key IDs to B, if they share any common key then B send key id of shared key to A. Key IDs doesn’t give attack opportunity to adversary for traffic analysis attack. Now, A and B are securely linked and communicate by encrypting message through their shared key. If A and B do not share a key then B send its list of key IDs to A. Now, A broadcast a message containing list of key IDs for both A and B. During path key establishment phase, intermediate node determine if it share keys with both A and B or not. If not, it further broadcast message until such a node is found. Suppose we find an intermediate node say, C has common key k1, k2 with A and B respectively. Now, instead of issuing an unused key from C’s own key ring, new key is selected from key pool corresponding to key ID generated using pseudo random number generator. In basic scheme if C gets captured communication between A and B also gets compromised. However in our case, even if C gets captured communication between A and B remains unaffected. Further, after getting new random key value say k3 instead of sending plain key C encrypt k3 with keys k1 and k2 to get E(k3) and E(k3) respectively to enhance security. For encryption we use RC5 algorithm. Now, these encrypted values with key identifiers of shared keys are sent to node A. Now, firstly node A determine which of its key from k1…k4 is common with node C using key ID send by node C and finally decrypt E(k3) to get k3 and forward E(k3) to node B. Now, node B find shared key with the help of key identifier and decrypt E(k3) to get key k3 which from now on act as shared key between A and B. In our scheme, path using C is used only once, because as soon as key is issued by intermediary both A and B can communicate directly through k3. Therefore, effect of traversing hops to set up path key is negligible. Figure 1 shows the screenshot of the simulation using TinyViz. Here the blue circle indicates broadcasting a message if no shared key between nodes and the magenta arrows show one-to-one communication between nodes sharing common key. In this example, node 0 and node 1, node 2 and node 3, node 6 and node 7 want to communicate and share key with each other, while node 4 and node 5 use node 1 to establish path key whereas node 8 and node 9 don’t want to communicate with any of the nodes.
IV. ANALYSIS OF OUR SCHEME BASED ON RANDOM GRAPH THEORY AND KRYPTOGRAPH

Now, in this section of paper we perform mathematical analysis of our proposed scheme based on random graph theory proposed by Erdos and Renyi and kryptograph model.

Erdos-Renyi (ER) model [9] was implemented in most of the previously discussed key management schemes. It was considered as quite a simplified model for evaluation of communication networks. In this model,

\[ p = \frac{\ln(N)}{N} + \frac{c}{N} \]  

where \( p \) is probability for connectivity, \( N \) is number of nodes in network and \( c \) is any real constant.

\[ p = 1 - \Pr[\text{two nodes do not share any key}] \]  

As a result key pool size \( K \) is computed as,

\[ p = 1 - \frac{(K-k)!^2}{((K-2k)!P!)} \]  

However, we want to emphasize that ER model is not that compatible and relevant for secure WSNs. Kryptograph model is more realistic and powerful as compared to ER model on which most of the random key pre-distribution schemes are based, here we discuss two major reasons why it is so:

Firstly, ER model based WSN assume that any two nodes can be directly linked, irrespective of their geographical location. Whereas, in kryptograph based WSN if two nodes are not within communication range(r) they cannot be directly linked. In kryptograph, edges exist if two nodes share a common key + they are within communication range. So, they perfectly suited to WSN.

Secondly, ER model based WSN assume that edges exist independently, whenever a new edge required to be inserted a coin is flipped each time regardless of previous one. However, to make WSN graph connected with some probability, whether a new edge required to be inserted or not is dependent on connectivity provided by previously inserted edges in the graph.

As a result, above discussed points clearly state that ER model is not applicable for WSN. Further, we explore how two crucial properties: connectivity via secure links and resiliency against malicious attacks can be achieved simultaneously in context of WSN with the help of kryptograph. Also, our network appreciates another robust security property where adversary cannot partition the network into two linear size components, compromising all the links between them, unless it captures linearly many nodes [10]. As a result, our network is more faults tolerant to node failures.

First of all, to ensure network connectivity we determine crucial relation between \( k \) and \( K \). If \( k \) is very small to \( K \), each key will be used by lesser nodes which in turn good for security but network tends to be disconnected. Also, longer path increase delay and communication cost. If we let \( k \) grow with fixed \( K \), it will result in escalation of network but diminution in security, capturing few nodes discloses entire key pool. Security and connectivity of network always remain in conflict. Choice between \( k \) and \( K \) that guarantees both at an optimal level will be given by probability that link exists between pair of nodes:

\[ \Pr[\text{link exists}] = 1 - (1 - k/K)^k \approx k^2/K \]  

To calculate \( k^2/K \) we determine \( K \) and \( k \) in relation to \( N \). To begin with \( K \), if \( K \) is very large in comparison to \( N \) then network tends to be disconnected. However, small \( K \) increase probability of sharing but compromising only a few nodes can easily compromise entire network. \( K \)'s optimal value should be:

\[ K \approx N \log N \]  

To share a choice of \( k \) is very crucial in respect to \( N \), if \( k \) is very small as compared to \( N \) then chance of sharing would be less and if we choose \( k \) large then security became concern. Balance between \( k \) and \( N \) will be given by the following equation:

\[ k \approx \log N \]  

Using aforementioned equations \( k^2/K \) will be calculated as follows:

\[ k^2/K \approx \log N/N \]  

If \( K >= N \), \( k >= 5 \) and satisfies equation (7) then network is connected with high probability and we ensure that for our graph.

Further, in ER model whenever a node, say \( x \) is compromised not only all links incident on that node gets compromised but all links which are sharing same keys as of
node x also compromised. This dependency is very harmful for security. A network is redoubtable if capturing a node compromises keys of only that node. An adversary needs to capture a large number of nodes to compromise the whole network. Ratio of $k/K$ determines probability of sharing a key between a pair of node, if $k$ increases or $K$ decreases probability of key sharing increases and vice versa, can be easily calculated by dividing equation (4) and (5):

$$\frac{k}{K} = 1/N$$

If our network satisfies this equation we can say our network is redoubtable.

As a result, if $K = N\log N$ and $k$ satisfy equation (7) then network is not only connected but redoubtable. Moreover, a network cannot be partitioned into two parts even if we compromise all links between them, until we capture a linear fraction of nodes. Such networks are known as unsplittable networks.

If we assume full visibility case we reduce our kryptograph from $G_{N,k,K}$ to $G_{N,k,K}$[where each node is assumed within communication range]

$$\frac{k^2}{K} = c\log N/N, c=17 \text{ and } K = N\log N.$$ A graph that satisfies all the above mentioned conditions is a graph that is unsplittable, connected and resilient at the same time and we ensure all of that for our graph.

V. SIMULATION RESULTS

Lastly, we carried out simulation to support our theoretical results. Based on above equations below we show graphs of connectivity for our new scheme based on ER model (Fig. 2.) and kryptograph model (Fig. 3.). In both graphs, we modify value of $k$ for various fixed size pool $K$ to check effect of varying value of $k$ on $p_{\text{key sharing}}$. As expected when we increase value of $k$ for a fixed $K$ then value of $p_{\text{key sharing}}$ increases. Also, when we increase value of $K$ corresponding to $N$ ($K=N$) then connectivity decreases as nodes have to find shared key from a large sized key pool whose probability decrease with increase in network size. However, we can clearly see that probability of connectivity for our scheme based on kryptograph is much better than connectivity of scheme based on ER model. Also, scheme based on kryptograph is more secure and fault tolerant as our network is redoubtable and unsplittable.

VI. CONCLUSION

In this paper, we implemented a new random key pre-distribution scheme for homogenous WSNs on TinyOS. Our scheme is more secure and communication efficient as compared to previously presented key pre-distribution schemes. Further, we discussed random graph theory and kryptograph model in context of WSN. Finally, we performed mathematical analysis of our scheme based on ER and kryptograph model. Simulation results conclude that scheme based on kryptograph has more probability of connectivity and has better resilience and fault tolerance as compared to scheme based on ER model.

Future work involves studying the effects of mobile sensor networks and extension of our scheme to a two dimensional layout, which is more intuitive and also implementation of our scheme on mica motes.

REFERENCES


