NIMMAGADDA, SPOORTHY, M.S. Multiple Bridge Secret Delivery in Wireless Sensor Networks. (2010)
Directed by Dr. Jing Deng. 46 pp.

Achieving security in wireless sensor network is a challenging problem due to the inherent resource and computing constraints. Several key distribution techniques have been proposed in the technical literature for efficient distribution of keys to the nodes prior deployment. These techniques establish secure links for some pairs of physically connected nodes but leave other pairs alone. Remaining nodes use multi-hop scheme to form a secured path connecting these links. Using this technique, the secret is disclosed to all the nodes on the path. Therefore, if any of the nodes are compromised by an adversary, secret is disclosed to the adversary. To solve this problem, a scheme called Babel was proposed recently that finds common bridge node to deliver secret link keys to their neighbors. In this scheme regular paths are used to deliver multiple keys with the common bridge node, hence key compromise probability is lowered compared to previous techniques. Our work is based on the Babel scheme and has several advantages. In our work we propose a new scheme that finds multiple bridge nodes to deliver secret link keys to all its physical neighbors. Keys are distributed to multiple bridge nodes instead of one common bridge node to establish secure connections to the disconnected nodes. Hence even if a few of the bridge nodes are compromised, secret will not be disclosed to the adversary.

We present the details of our scheme’s design and investigate the connectivity and security performance of our scheme in this thesis.

MULTIPLE BRIDGE SECRET DELIVERY IN WIRELESS SENSOR NETWORKS

by

Spoorthy Nimmagadda

A Thesis Submitted to
The Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro

2010

Approved by

_____

Committee Chair

To my parents.

APPROVAL PAGE

This thesis has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair _____

Committee Members _____

_____

_____

Date of Acceptance by Committee

_____

Date of Final Oral Examination

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to each and everyone who were responsible for the successful completion of this thesis.

First of all, I would like to offer my sincerest gratitude to my thesis advisor Dr. Jing Deng for his support, suggestions, guidance, endless patience and unstinting encouragement without whom this thesis would not have been possible.

I would like to express my earnest thanks to my thesis committee: Dr. Fereidoon Sadri and Dr. Nancy Green for their valuable comments which enabled me to complete my thesis successfully.

It would be grateful to thank all my group members : Siddhiben Naik, Yuan Kong, Alexey Bogaevski and Yanfen Song for the discussions in the lab which helped me a lot in my thesis.

I owe deepest gratitude to Aparna Meka for her moral support and always being with me whenever needed. I would also thank all my friends who gave their valuable suggestions and support for the completion of my thesis.

Finally, It's my pleasure to thank my father, Nimmagadda Nagendra Babu and my mother Kavitha for their unflagging love and support throughout my life. I am indebted to my sister, Spandana Nimmagadda and my brother-in-law, Ramesh Ghattamaneni who has been my strength and support during the whole course of my thesis.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

I.1 Wireless Networks

Wireless network is associated with telecommunications network where terminals, nodes and links are connected together to enable communications among the users of the terminals without the use of wires. Driven by pressure to ease mobile computing, everyone is plunging into wireless networking [1]. Wireless networking makes the data portable, mobile and accessible. Moving data over wireless networks involve radio signals, data format and network structure. In a wireless network, the network interface adapters in each computer and base station convert digital data to radio signals, which they transmit to other devices on the same network, and they receive and convert incoming radio signals from other network elements back to digital data [2].

Wireless networking proves to be very useful in public places where one might find wireless access to the Internet. Quality of Service (QoS) is not guaranteed in wireless network because if there is any interference with the link the connection may be dropped. Different types of wireless networks available are Wide area networks (WAN), Local area networks (LAN), Personal area networks (PAN), Metropolitan area networks (MAN) and Mobile device networks.

I.1.1 Mobile ad-hoc Networks

Ad-hoc networks are a key in the evolution of wireless networks [3]. Wireless networks are adopted to enable mobility [4]. They are a collection of two or more devices equipped with wireless communications and network capability. Such devices can communicate with another node that is immediately within their radio range or one that is outside their radio range [5]. They eliminate the final limitation of the traditional cellular and mobile networks in sense of infrastructure. These kinds of networks are self organizing and adaptive. They are comprised of equal nodes that communicate over the wireless links without any central control. They inherit traditional problems of both mobile and wireless communications. The highly dynamic nature of a mobile ad-hoc network (MANET) results in rapid and unpredictable change of the topology over time. The routes among the nodes in an ad-hoc network may include multiple hops and hence it is appropriate to call such networks “multi-hop wireless ad-hoc networks” [4]. Transmitting data in MANET is based on the RTS/CTS control sequence used by the popular IEEE 802.11. Short control frame named RTS, is sent from source station to the receiving station to announce the upcoming frame transmission. On receiving the RTS frame the destination station replies by a CTS frame to show that it is ready to receive the data frame. Both the RTS and CTS frames contain the total duration of the transmission that is the overall time needed to transmit the data frame and the related ACK. This information can be read by any station within the transmission range of either the source or the destination station.

Hence a station becomes aware of a transmission from a hidden station and the length of time the channel will be used for transmission [4]. As mobile ad-hoc networks rely on battery power or other exhaustible devices power consumption becomes a critical issue.

I.1.2 Wireless Sensor Networks

Wireless sensor networks (WSNs) are an increasingly attractive means to bridge the gap between the physical and virtual world [6]. They are one form of an ad hoc wireless network. Recently these networks are drawing considerable attention because they are crucial for the digital battlefield. These networks will consist of hundreds or thousands of self-organizing, low-power, low cost wireless nodes deployed to monitor and affect the environment [7]. These sensor nodes are very minute in size which makes them hard to detect and destroy by the enemies.

Sensor nodes are densely deployed in multiple locations and helps in gathering the sensory information required by the smart environments. They use their processing and computational abilities to transmit only the required and processed data instead of sending the raw data to the nodes. They can be easily installed and maintained.

WSNs generally consist of data acquisition network and data distribution network monitored and controlled by a management center [8]. These messages will be received and transmitted over the wireless links. These links can be formed by radio, infrared or optical medium [9].

Quality of service (QoS) which can be specified in terms of message delay, bit error rates, packet loss, economic cost of transmission, transmission power, etc is the

basic issue while designing the network topology for transmission of the messages. One unique feature of sensor nodes are its cooperative effort. Due to its features WSNs are used in wide range of applications such as military applications, habitat monitoring, environmental observations, health care and other commercial applications where nodes can be captured by an adversary.

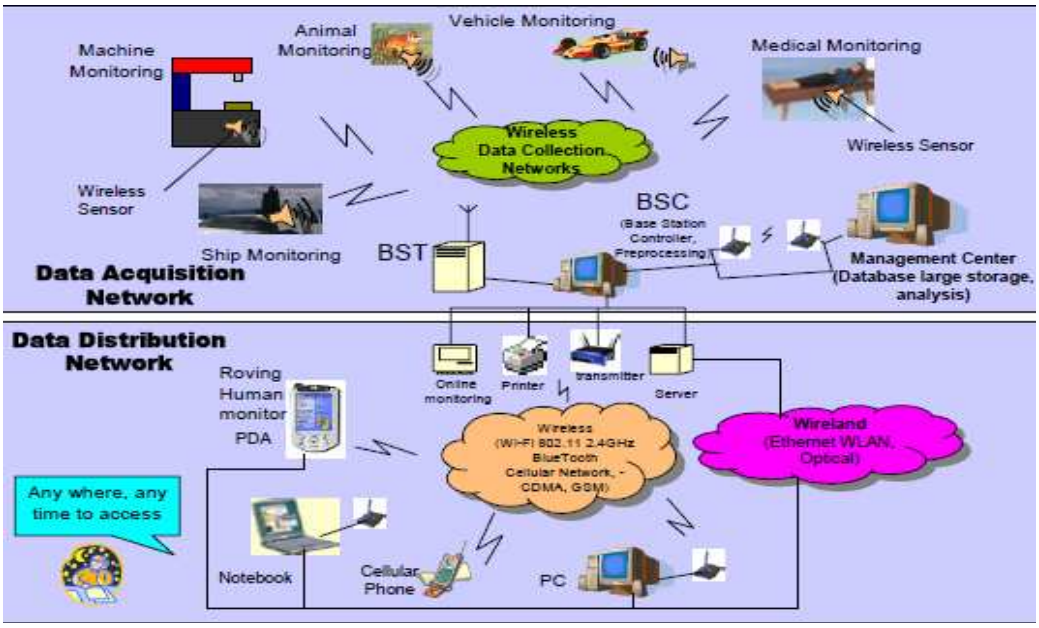

Figure. I.1 Typical wireless sensor network [8]

I.2 Security in Wireless Sensor Networks

Sensor networks are typically characterized by limited power supplies, low bandwidth, small memory size and limited energy which leads to a very demanding

environment to provide security [7]. Also due to its broadcast nature WSNs are vulnerable to security attacks. Immediate concerns for wireless communications are device theft, denial of service, malicious hackers, malicious code, theft of service, industrial and foreign espionage [10]. As they increased popularity demand for effective security mechanisms also increased. Physical attacks by the adversaries are the most prominent security issues in WSNs. There is no fixed infrastructure for the management of the sensor networks hence security became more difficult.

Security requirements of a wireless sensor networks are data confidentiality, data integrity, data freshness, availability, self organization, time synchronization, secure localization and authentication [11]. To protect the sensitive data during the communication between the nodes security keys are distributed to each node. Lack of trusted servers nearby (for public/private key schemes), secret key schemes may be more viable to protect such communications [12].

Several mechanisms were proposed for security in wireless sensor networks. However there is no technique everyone accepts with regardless of the network. Few of them were applied in some environments and research is still going on actively for future solutions.

I.3 Related Work

As physical topology of WSNs is unknown before deployment the only option for distribution of keys to sensor nodes are key pre-distribution. Keys must be installed in sensor nodes before deployment for secure connectivity among nodes.

Traditional key pre-distribution offers two solutions 1) single mission key 2) set of separate n-1 keys. Both solutions are inadequate because in single mission key same key is installed in all the nodes and hence if one of the sensor node is compromised the entire network will be compromised and in set of separate n-1 keys each sensor node must be installed with n-1 keys each being pair-wise privately shared with another node. This solution is impractical in large networks because memory dedicated for storing n-1 keys for each sensor network may not be sufficient and addition or deletion of the sensor nodes would become more expensive and complex.

In 2002, Eschenauer and Gligor proposed a simple key pre-distribution scheme that requires memory storage for only few tens to a couple of hundred keys, and yet has similar security and superior operational properties when compared to those of the pair-wise private key-sharing scheme [13]. In this scheme ring of keys are distributed to each sensor node. As the key ring is chosen randomly from large pool of keys some pair of nodes may not have a shared key. Such nodes deliver secrets using multi-hop path scheme where the secret is disclosed to all the nodes on the path. Hence if one of node on the path is compromised, the secret is disclosed to the adversary.

In 2007, Jing Deng and Yunghsiang S.Han proposed Babel Scheme that finds a common bridge node to deliver secret link keys to establish secure communication with the nodes which are not connected. The common bridge node will be the only node other than the source and the receiving nodes

knowing the secrets [12]. Hence the chance of secret disclosure is small compared to previous techniques but if the common bridge node is compromised the secrets of all the nodes which do not share keys with the source node will be disclosed to the adversary.

I.4 Our Approach

Our scheme is to deliver multiple keys with the use of multiple common bridge nodes. In the Babel scheme [12] there is only one common bridge node which shares keys with all the disconnected nodes but in this scheme there are multiple bridge nodes to share keys with nodes those are not connected. This scheme lowers the compromise probability compared to other techniques as this does not disclose secrets of all the nodes on its path to the adversary. Thus this technique can be used in the networks where the communication with all the physical neighbors need to be done securely.

I.5 Problem Formulation

The objective of the proposed work is to improve security aspect in wireless sensor networks using the technique of multiple bridge nodes. The chapters in this paper is organized in the following manner.

- Chapter I introduces the background of wireless networks and importance for its security.
- Chapter II explains various techniques proposed for the security in WSNs.

- Chapter III describes the multiple bridge nodes secret delivery technique in detail.
- Chapter IV shows the performance evaluation of our scheme with the simulation results.
- Chapter V ends with the conclusion and future work.
- Appendix A provides the code of our simulations.

CHAPTER II

VARIOUS SECURITY MECHANISMS

II.1 Key Pre-distribution in Wireless Sensor Networks

Key management is one of the fundamental building blocks of security services and is also a challenging problem in sensor networks. To solve this problem several key pre-distribution schemes have been proposed [14].

Eschenauer and Gligor first proposed a random key pre-distribution scheme in wireless sensor networks [13] which is known as the basic scheme. Let *P* be the large pool of keys generated and *k* be the number of keys randomly chosen from *P* keys forming a key ring. This scheme has the three different phases. In the first phase knew as key pre-distribution, before sensor nodes are deployed *k* keys are stored into the sensor memory for each node. Each pair of nodes establishes a secure connection if they share at least one common key with a chosen probability.

After the sensor nodes are deployed shared-key discovery phase is performed. In this phase nodes find out which of its neighbors share a key. When the nodes discover that they share a key with its neighbor then that key establishes a direct link between two nodes. After shared-key discovery phase is complete, a connected graph with secured links is formed in path-key establishment phase. Some pair of nodes may not share a key then these

nodes set up path keys by it's securely connected neighbors. In this way key can be sent through the path from the source node to the targeted node securely.

Chan, Perrig and Song reviewed the approach in basic scheme and proposed three new mechanisms [15]. First in q-composite random key pre-distribution scheme *q* common keys are needed from their key rings instead of single common key as in [13] to establish a secure link. As the amount of required keys increases it becomes harder for an attacker to break the link. Next in multi-path key reinforcement scheme security of an established link is strengthened through multiple paths. The basic idea of this scheme was explored by Anderson and Perrig [16]. For suppose we have a secure link from *A* to *B* after key-setup, their approach is to find out multiple paths from *A* to *B* where each path may have *h* hops or less. Suppose *j* be the number of disjoint paths from *A* to *B*. *A* then generates *j* random values which have same length as the encryption/decryption key. *A* then routes each random value along a different path to *B*. When *B* received all *j* keys then new link key can be computed by both *A* and *B.* In this way much more security is provided for the links.

Finally in the last scheme random pairwise keys, a modification to basic pairwise keys scheme is done where not all *n-1* keys need to be stored in the node's key ring to have a connected graph with high probability. To achieve high probability *p* in a network with *n* nodes each node need to store a

random set of *np* pairwise keys instead of *n-1* keys. This scheme is beneficial over purely random keys chosen from a given pool because this gives node-node authentication properties where each nodes hold some key *k*, also stores the identity (ID) of other node which also holds *k*. Hence both nodes will be certain of the identity of one another when *k* is used to create a secure link with another node since no other nodes can hold *k*. This provides improved security, since any captured node reveals no information about links in which it is not directly involved [14].

Du, Deng, Han and Varshney proposed a new key pre-distribution scheme [17] which substantially improves the resilience of the network compared to the existing schemes [13, 15]. This scheme is built on Bloom's key pre-distribution scheme [18] and combines with the random key pre-distribution method. In [18], Bloom proposed a key pre-distribution scheme which uses only *λ+1* memory spaces to find a secret pairwise key between any pair of nodes. Compared to *N* in *(N-1)* pairwise key pre-distribution scheme *λ* is much smaller. While an adversary compromises less than or equal to *λ* nodes, uncompromised nodes are secure and when adversary compromises more than *λ* nodes then all pairwise keys of the entire network is compromised.

Blom's scheme uses one key space for all the nodes in which any pair can compute its pairwise key in this key space whereas the new scheme uses multiple key spaces. Du et. al. construct *ω* spaces using Blom's scheme and each sensor node carries key information from Γ ($2 \leq \Gamma \leq \omega$) randomly selected

key spaces. Pairwise key can be computed if two nodes carry key information from a common space and when two nodes do not carry key information from a common space they can compute their pairwise key via other nodes which share keys with them. This scheme is more resilient than Blom's scheme and other key pre-distribution schemes because it uses same amount of memory. Liu and Ning [19] also developed similar method based on polynomial-based key pre-distribution [20].

II.2 Key Distribution Using Deployment Knowledge

In all the previous key pre-distribution schemes no deployment knowledge is available. Although they proposed viable solutions they have not exploited information that significantly improves their performance. This new scheme proposed by Du, Deng, Han, Chen, Varshney [21] shows that the knowledge regarding the actual non-uniform sensor deployment can help us improve the performance of a key pre-distribution scheme. In wireless sensor networks secure communications are done only between the neighboring nodes hence the knowledge about the nodes that are likely to be the neighbors of each node benefits the key pre-distribution scheme. Due to the randomness of the deployment knowing the exact set of neighbors of each node becomes unrealistic but we can know the set of possible or likely neighbors for each node. This scheme uses random key pre-distribution in [13] and exploits the deployment knowledge. Deployment knowledge can be modeled using probability density functions (pdfs). Due to the deployment knowledge the first

phase in basic key pre-distribution differs and the last two phases remain the same.

We presume that the sensor nodes are evenly divided into *t * n* groups *Gi,j,* for *i=1,…,t,* and *j=1,…,n*. Presume that the global key pool is *S* with size |*S*| and also presume that the deployment points are arranged in a grid. Each node carries *m* keys. In first phase called key pre-distribution, before deploying the sensor nodes the key pool *S* is divided into *t*n* key pools *Si,j (for i=1,…,t and j=1,…,n),* with *Si,j* corresponding with the deployment group *Gi,j*. Setting up key pool *Si,j* will allow the nearby key pools to share more keys and far away from one another share less keys. After key pool set up each sensor node in deployment group *Gi,j,* randomly selects m keys from its corresponding key pool *Si,j,* and load those keys into the memory of the node [21]. The secure links between nodes *i* and *j* can be found using flooding [22]. Key- sharing graph *G* may have isolated components which do not have secure links. Hence global connectivity of the graph G is measured. Global connectivity can be estimated using Erdos random graph theorem [23] when node distribution and key sharing are uniform. Since node distribution and key sharing is not uniform this theorem will not be a good estimation. Recently, Shakkottai and et. al. have determined the connectivity of a wireless sensor grid network with unreliable nodes [24]. Hence using deployment knowledge each node needs to carry only a fraction of keys compared to other key pre-distribution schemes but achieves same level of connectivity. This scheme reduces the memory requirement and

substantially improves network resilience against node capture.

II.3 Key Distribution for Mobile Computing

Several other key distribution schemes have been introduced for mobile computing. Tatebayashi, Matsuzaki, and New-man proposed a key distribution protocol suitable for digital mobile communications [25]. In this scheme a public key cryptosystem is employed for uplink channels (from an user terminal to a network center) making the mobile communication free from key management problems. High speed performance is enabled at hardware-limited terminals by employing secret key cryptosystem for downlink channels (from a network center to the user terminal). This scheme introduces a structure in the transmitted data and a mechanism checking the replay attack to avoid a protocol failure based on multiplicative property of the RSA cryptography [26].

This work is further improved by Park et al. in [27] and proposed an encryption algorithm of an attack based on the algebraic properties. Although it suggests an improved protocol *S* has a shared secret with all parties in repaired protocols hence public key cryptography is not justified here. Various other key agreement and authentication protocols specifically designed for the use in mobile applications have been proposed in [28, 29, 30, 31]. A survey was made on most prominent security protocols proposed for mobile applications in [32] based on security, suitability and optimization. Zhou and Hass [33] proposed a secure key management service in an ad hoc networking environment. This scheme proposed threshold cryptography to distribute trust between a set of

servers. Kong et al. proposed a certificate based authentication approach based on asymmetric cryptographic defacto standard RSA [34]. They proposed localized public-key infrastructure mechanisms, based on secret sharing schemes.

II.4 Path Key Establishment

Path key establishment has been introduced in [35]. In the key pre-distribution schemes communications among end nodes are exposed to intermediate nodes along the path. In [35] multiple node-disjoint secure paths are used to establish the path key which decreases the risk of path key being revealed. Further Li et al. [36] proposed multiple one-hop paths instead of node disjoint paths to enhance the security of path key establishment. Traynor et al. proposed to use a few more powerful sensors to achieve key establishment [37]. Deng and Han proposed a scheme [38] to address the problem of compromised sensors modifying and eavesdropping the information passing through such multi-hop paths. They use MDS codes to develop the IRT scheme to provide protection for information delivery. Another multi-path pairwise key establishment scheme was proposed to counter Byzantine [39] attacks due to packet dropping and cheating [40]. This scheme can tolerates upto t faulty paths among the communication pairs.

In all the previous techniques there are still some local links which are not connected securely. This is because an extremely high local connectivity (on the security plane) would mean higher vulnerability and lower network resilience.

Hence Deng and Han [12] proposed a new scheme called Babel which focuses on delivering secret link keys from a source to multiple neighbors. This new technique called Babel is used to find a common bridge node to deliver secret link keys to these nodes which are disconnected. This scheme uses regular paths and delivers multiple keys using the common bridge nodes. As the delivered keys are disclosed only to one node the common bridge node unlike key pre-distribution scheme which discloses keys to all the nodes on the path, key compromise capability is lower compared to other delivery techniques.

CHAPTER III

MULTIPLE-BRIDGE SECRET DELIVERY TECHNIQUE

III.1 Design of the Proposed Scheme

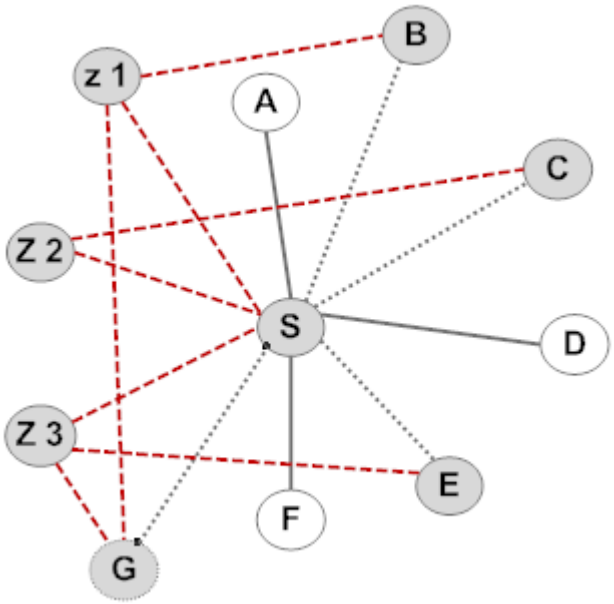

Figure III.1 Illustration of secure connectivity around node S.

The problem of delivering multiple secret link keys to the neighbor nodes are illustrated in the Fig. III.1. Here node S is the source node having several physical neighbors A, B, C, D, E, F and G clockwise. The solid lines in Fig. III.1

represents secure connectivity between the nodes as they share common keys. Hence links S-A, S-D and S-F are connected on security plane. Dotted lines represent physical connectivity as they do not share any common keys. Hence links S-B, S-C,S-E and S-G are disconnected on security plane. These nodes B, C,E and G are termed “to-be-connected neighbors”, N_{tbc}. Nodes Z1, Z2 and Z3 are bridge nodes relatively far from the neighborhood but share keys with the source node S and some of the to-be-connected neighbors.

We use the following notations and variables in our scheme.

TTL: Predefined number of hops for the request message to travel;

P: Large pool of Keys;

m: Number of keys carried by each node;

N: Total number of nodes in the network;

S: Source node;

BN: Bridge Nodes;

Z_i: Multiple bridge nodes where 1≤i<N;

N_{tbc}: Set of to-be-connected neighbors of S;

$K_{i,t}$: Keys on node i, $i \in \{S\}$ â $N_{tbc,}$ $1 \leq t \leq m;$

Xc: Compromise Capability of the nodes;

Xp: Probability of to-be-connected neighbors being compromised.

III.2 Operational Details of the Proposed Scheme

Main idea of our scheme is to find multiple bridge nodes that share keys with the source node and some of the to-be-connected neighbors. Its purpose is to deliver multiple secret link keys to the to-be-connected neighbors with lower compromise probability. Fig. III.1 is an example of Multiple Bridge secret delivery technique. Suppose S needs to send secret to one of the to-be-connected neighbors (B, C, E or G) it will collect Message Authentication Codes (MACs) of a challenge message based on each of the keys in K_S, K_B, $K_{C,}$ K_E and K_G. This information is broadcasted over the network which is control-flooded. Each node compares the MACs of the message based on the carried keys and responds if it shares a key with the source node and some of the to-be-connected neighbors. The reply will be the response to all the challenges with the shared key. For suppose, node Z1, Z2 and Z3 satisfies the above condition then they will respond and serve as bridge nodes. In our example we a have three bridge nodes which share keys with the source node S and atleast one of the to-be-connected neighbors. The shared keys with N_{tbc} could be same or different. Each to-be-connected neighbor may have more than one bridge node. So it may have different paths to share the secret link keys. Hence if one path is compromised the secret will not be disclosed to the adversary. For example node S wants to share secret with G, it has the two bridge nodes Z1 and Z3. If Z1 gets compromised still the secret link key can be sent through Z3. Also as Z2 does not

carry keys of all the to-be-connected neighbors the entire network will not be compromised if one of the bridge node is hacked. This way the message transmission can be done more securely.

Nodes which do not share keys forward the message with their ID attached at the end of the message. The message may travel only upto certain number of hops (TTL) and will be discarded when it has been forwarded TTL times. When node S receives response from Z_i it sends the message to the to-be-connected neighbors. Each to-be-connected neighbor validates the keys sent by S and ensures if Z_i share key with itself. After that, node S sends secret link keys for the nodes B, C, E and G to their corresponding Z_i. In our example we can see that Z1 shares keys with node B and G. Later the bridge node says Z1 encrypts the secret link key of nodes B and G and sends it back to node S. Then S sends the encrypted key to their corresponding to-be-connected neighbors which decrypt the secret link keys. As the messages are sent using shared keys the chances for the secret to be disclosed is very low. We will investigate the effect of this scheme in Chapter IV.

CHAPTER IV

PERFORMANCE EVALUATION

We performed simulations in Matlab to investigate our scheme. Our simulations mainly focused on finding the multiple bridge nodes and probability of secrets being disclosed. We used simplified circular connectivity model and focused on key sharing among the nodes hence other simulators such as ns2 or OPNET are unnecessary at this stage. N number of nodes are randomly deployed in a network size 1000 meters by 1000 meters. Radio transmission range is assumed to be 200 meters. Different m (from 2 to 40)number of keys are randomly chosen from a pool of P=2000 keys and distributed to each sensor node. A source node will look for bridge nodes within a time-to-live (TTL) hop and TTL=2. These system parameters remain the same throughout this work unless mentioned otherwise.

In our evaluation, we first study the availability of multiple bridge nodes in different network set ups. We also look for the neighbors and to-be-connected neighbors for the source node S. Then we investigate the probability of to-be-connected neighbors compromised for security analysis. Here we define BN as the total number of bridge nodes.

IV.1 Availability of Bridge Nodes

We investigate the availability of Multiple Bridge nodes in this section. First we run simulation for different N values. We assumed source node to be the N-th node (note that all nodes are randomly placed in the network).

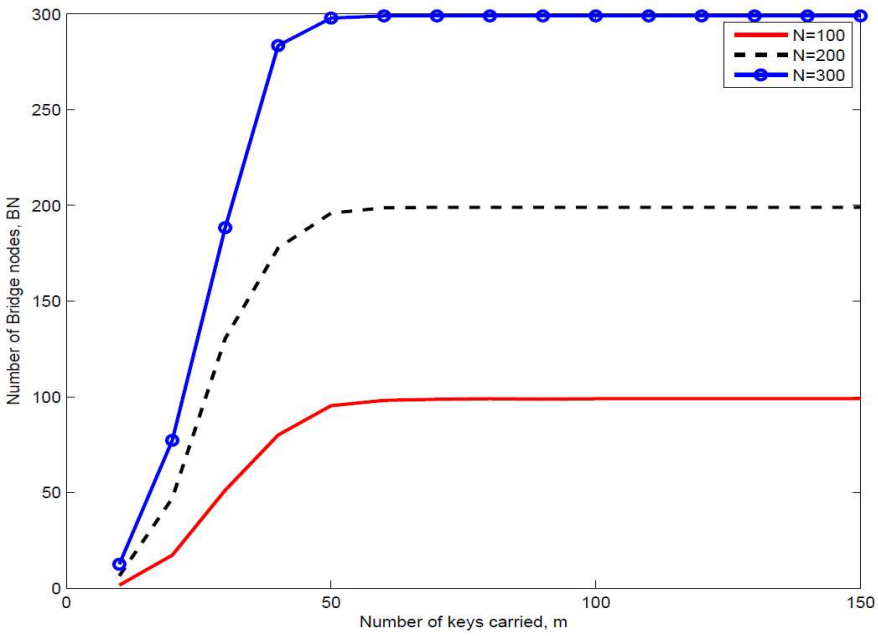

Figure IV.1: Number of bridge nodes as a function of memory size comparing different node densities.

From the figure IV.1 we can see that, as the number of nodes *(N)* increases, the total number of common bridge nodes *(BN)* also increases. This is because the

number of to-be-connected neighbors increases with N so probability of finding the bridge nodes also increases. We can see that number of bridge nodes remained constant at nearly m=40. This is because number of bridge nodes reached the maximum value. From figure IV.2 we can observe that at m=2-14 we can find only less bridge nodes and from figure IV.1 we can see that at m>30 bridge nodes reached maximum value. From both these figures we can conclude that m=15-20 is enough to find the bridge nodes

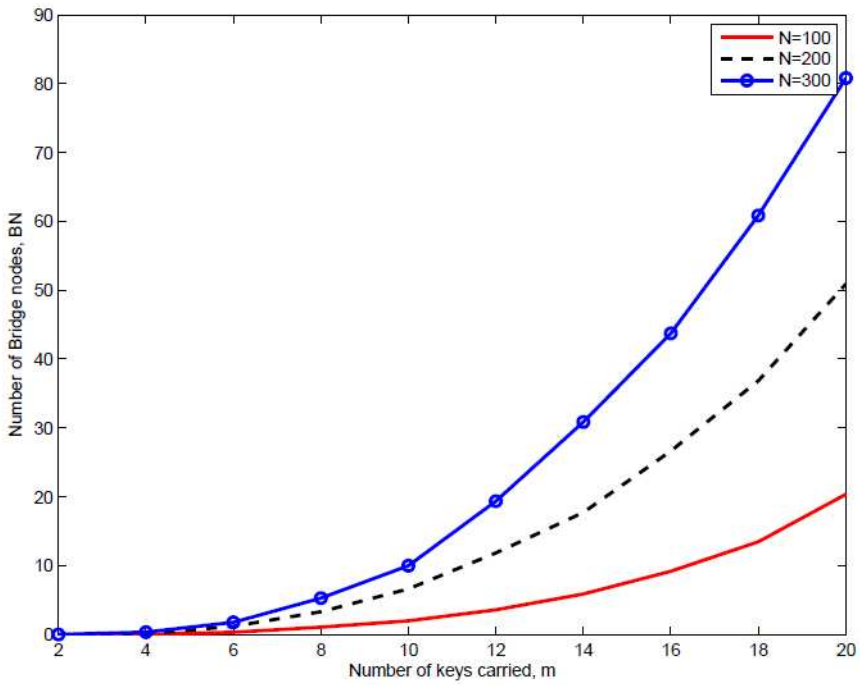

Figure IV.2: Number of bridge nodes as a function of memory size comparing different node densities for lower m.

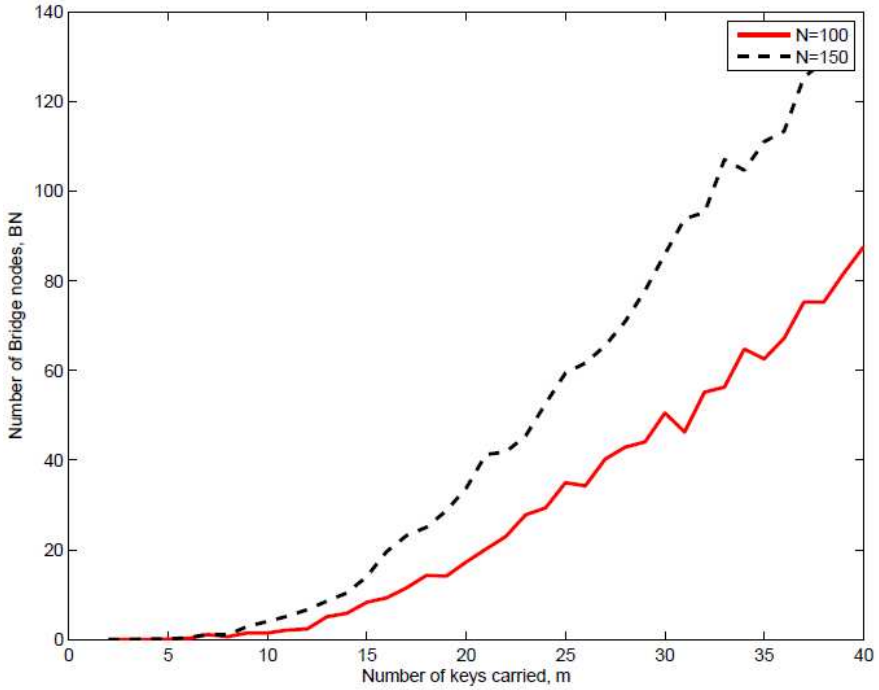

Figure IV.3: Number of bridge nodes for TTL=3 as a function of memory size comparing different node densities.

Figure IV.3 shows the simulation results for TTL=3. From the figure we can see that as m value increased BN value also increased because keys carried by each node increases. Thus it increases the probability of finding the bridge node which share keys with the source node and some of the to-be-connected neighbors. We observed at nearly m=40 BN reached maximum value for TTL=3. But we do not need so many bridge nodes hence when m is close to 40 we only

need to look for TTL=1 or 2.

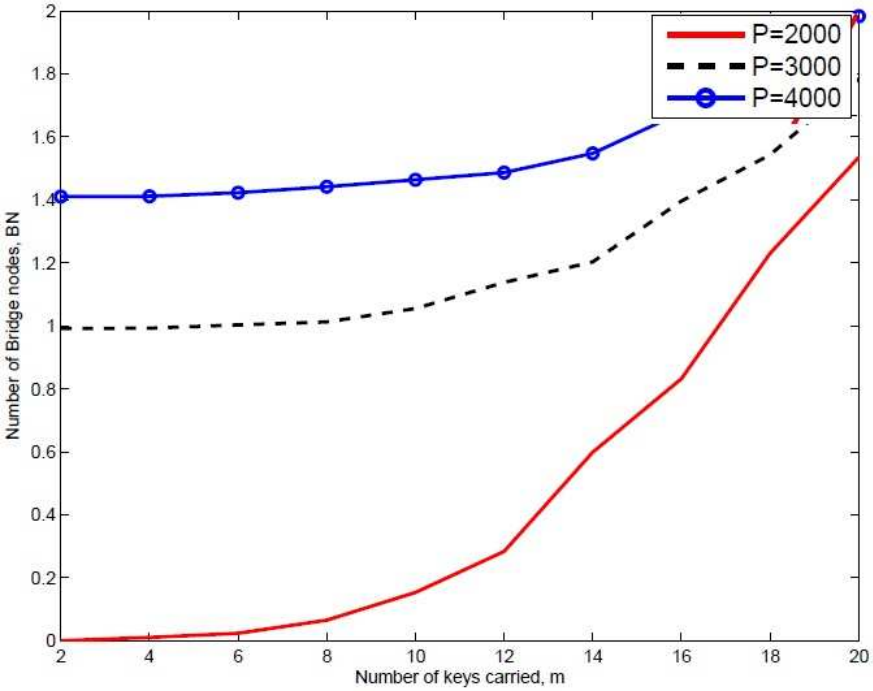

Figure IV.4: Number of Bridge nodes for N=100 as a function of memory size comparing different pool of keys.

From the figure IV.4 we can see that as P value increased the number of bridge nodes also increased. We can observe that when P=2000 number of bridge nodes raised slowly from zero but when P=4000 rise of BN is very small and is almost constant with increase of m value. Hence we can take large P and small m value for better performance but large P value decreases the physical

connectivity of the network. Therefore optimum P value should be small.

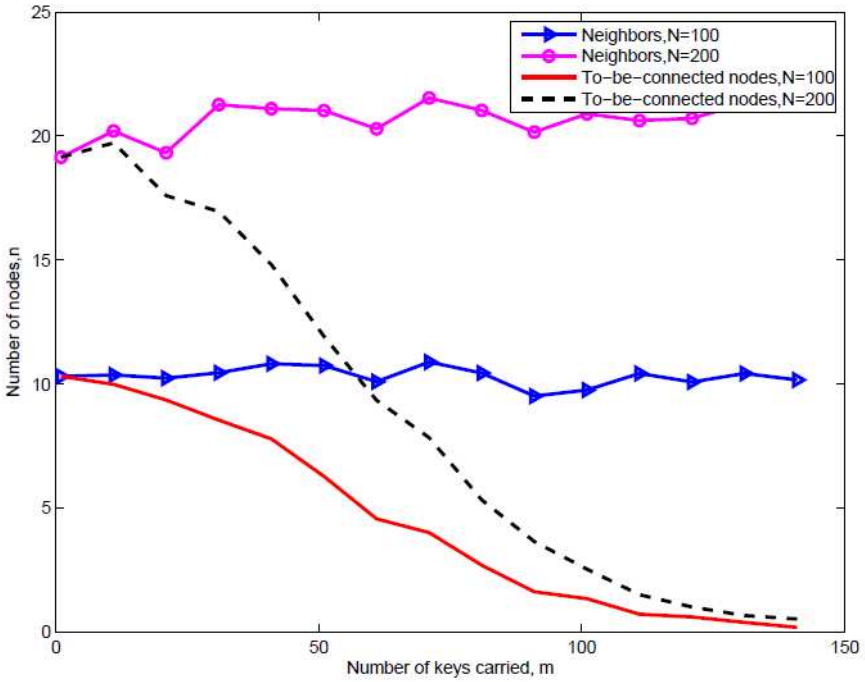

Figure IV.5: Number of nodes for P=2000 as a function of memory size comparing different node densities.

To find the availability of bridge nodes it's also important to know how many neighbors and to-be-connected neighbors a source node has. This can be shown in the figure IV.5. Here we can see that as number of nodes *N* doubled total number of neighbors and to-be-connected neighbors also doubled. We can

also see the decrease in the to-be-connected neighbors with the increase in the m value. It becomes zero for large m. This is because if a node carries more keys then probability of finding the shared keys increases. This increases the physical connectivity between the nodes. Hence there will be none to-be-connected neighbors and finding bridge nodes will be unnecessary at this point.. Number of neighbors remained almost constant with increase in m value even when the network topology is changed for each simulation run. From this we can understand that for any kind of network finding the neighbors remain same for same N and S.

It's also interesting to know the number of to-be-connected neighbors of a source. Here M=i shows that the bridge nodes share keys with source node S and i number of the to-be-connected neighbors. From the figure IV.6 we can see that when M=1 and M=2 number of bridge nodes increased after certain m value but when M=3 and M=4 they are almost zero for lower m. This is because when memory size is small number of shared keys among the nodes are low. Thus it decreases the probability of finding bridge nodes sharing keys with more number of to-be-connected neighbors.

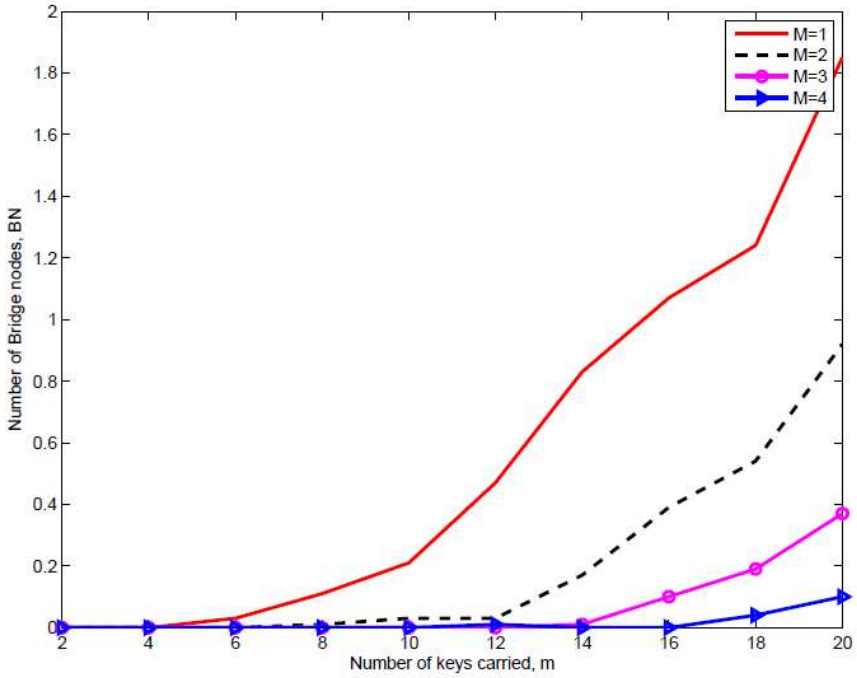

Figure IV.6: Number of bridge nodes for N=50 as a function of memory size comparing different M values.

Comparing figure IV.7 with figure IV.6 we can see that number of to-be-connected neighbors sharing keys with the bridge nodes increased when N=100. We are not interested at m>40 so if we observe both the figures for m=20, we can see the BN value is almost same for M=3 and M=4 in both the figures but it increased for M=1 and M=2 in figure IV.7. From these figures we can say that at optimum m value BN has less number of to-be-connected neighbors.

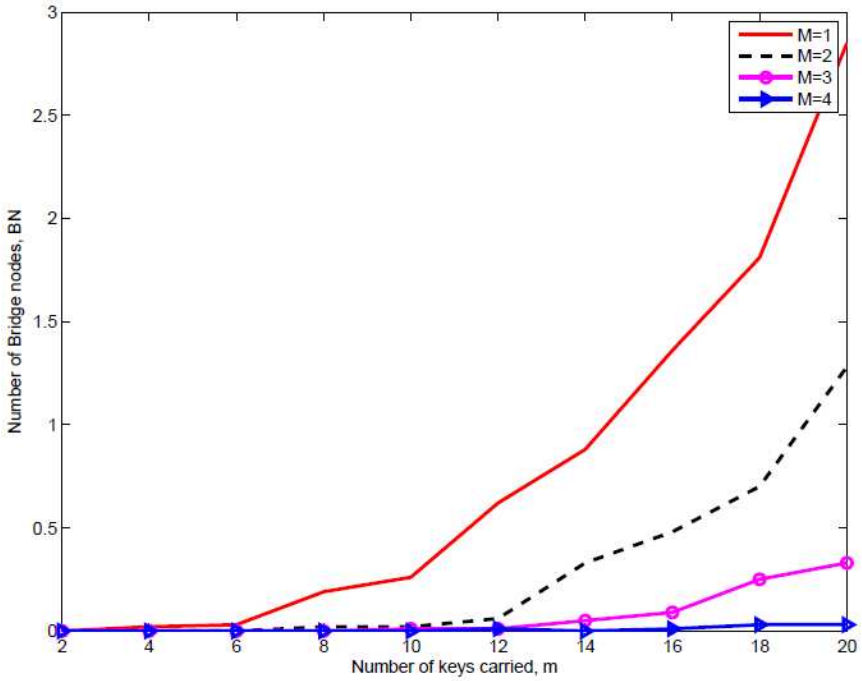

Figure IV.7: Number of bridge nodes for N=100 as a function of memory size comparing different M values.

Figure IV.8 compares number of bridge nodes for different hop count. When TTL=2,3,4 and 5 rise in m value increases the number of bridge nodes. When m=20 we can observe that BN value differs only a little for TTL=2,3,4 and 5. From this we can understand that for lower m value hop count does not matter for finding the bridge nodes. But if we increase the m value then we need to reduce the hop count to find the bridge nodes.

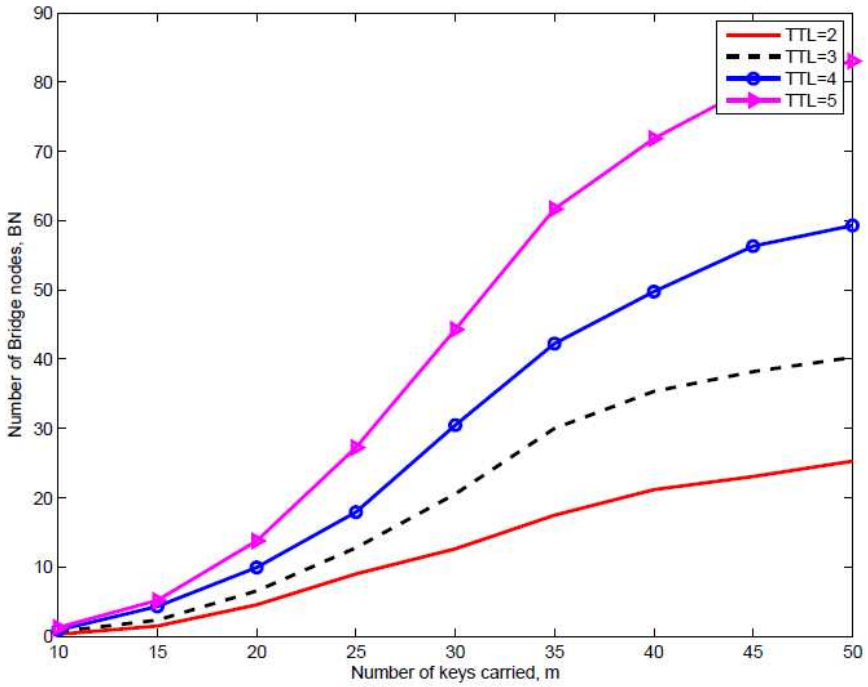

Figure IV.8: Number of bridge nodes for N=100 as a function of memory size comparing different hops.

IV.2 Security Analysis

In the previous section we found the availability of bridge nodes for sharing secret link keys in different network topologies. In this section we investigate the number of to-be-connected nodes compromised in the entire network as some of the nodes are randomly chosen as compromise. We also analyze percentage of the to-be-connected nodes compromised for different compromise capability. From this analysis we can find out the secure level of our

scheme. Here we define x_p as the ratio of number of the bridge nodes compromised to the total number of bridge nodes for each to-be-connected neighbors. x_c is assumed to be the compromise capability of the node. m=20, P=2000 and TTL=2 in this section.

Figure IV.9 compares the number of to-be-connected neighbors compromised for different N and x_c values. From the figure we can see that number of to-be-connected neighbors compromised increases for N=200 than in N=100. Here x_p>0.3 indicates that if more than 30% of the bridge nodes sharing keys with particular to-be-connected neighbor are compromised then that to-be-connected neighbor is said to be compromised. We can see that at N=100 for different x_p values the to-be-connected neighbors compromised varied slightly but at N=200 it shows much difference. From these two observations we can say that the network is more secure at lower N value when the absolute numbers of compromised bridge nodes are concerned.

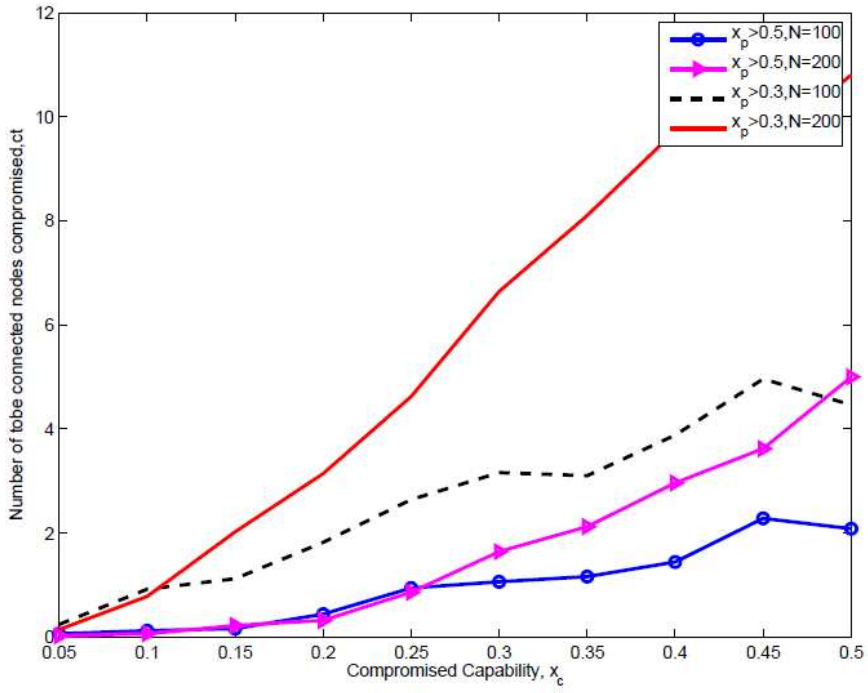

Figure IV.9: Number of to-be-connected neighbors compromised as a function of x_c.

Figure IV.10 shows the percentage of to-be-connected neighbors compromised for different node compromise capability. This figure shows that at $x_p > 0.3$, percentage of to-be-connected neighbors compromised is more than $x_p > 0.5$. This is because finding compromised to-be-connected neighbors is less when we say a node is compromised only when the percentage of bridge nodes compromised is more. Hence for the network to be more secure x_p should be high but it cannot have very large value. Note that x_p value is a system parameter

to be determined by how the key is encoded.

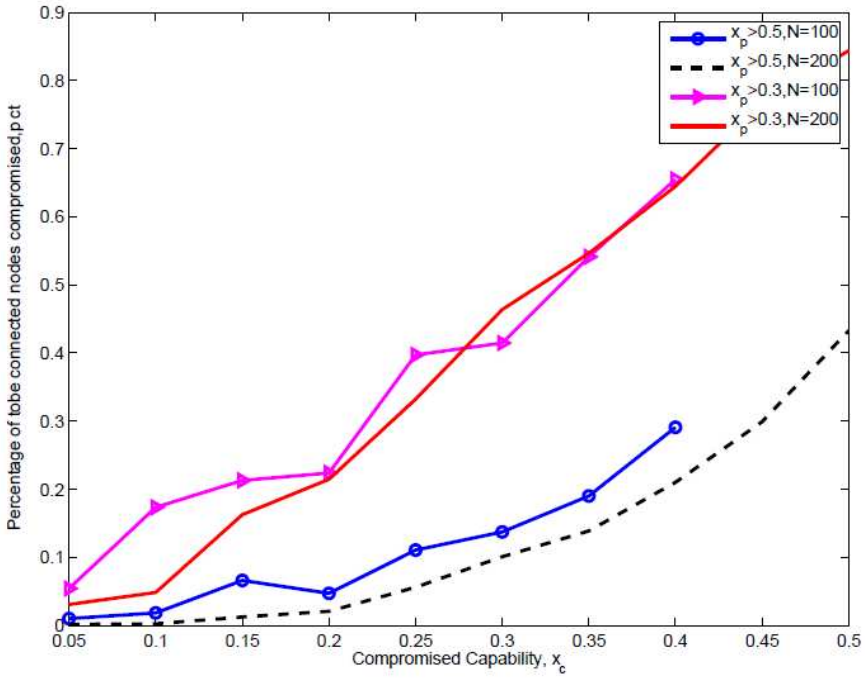

Figure IV.10: Percentage of to-be-connected neighbors compromised as a function of x_c.

CHAPTER V

CONCLUSION

Security is an important aspect in wireless sensor networks as they are crucial for the digital battlefield. Hence keys are used for encryption and authentication purposes between the communicating nodes. Many key agreement schemes have been investigated but unsuitable for wireless sensor networks. Lately several key pre-distribution schemes were proposed but they focused only on the connectivity of physical neighbors.

To overcome this problem we proposed “Multiple Bridge secret delivery scheme” to deliver secret link keys to the to-be-connected neighbors in wireless sensor networks. As the source node does not share keys with all its neighbors this scheme uses multiple bridge nodes to share keys with the source node and some of the to-be-connected neighbors. We designed this scheme for secure communication between the nodes.

In our performance evaluation we have observed that probability of finding bridge nodes increased with increased number of nodes. On the other hand it decreases the percentage of to-be-connected neighbors being compromised. Hence for a secure network, number of nodes compromised must be low and the capacity of the to-be-connected neighbor being compromised must be high when

number of bridge nodes compromised for each to-be-connected neighbor increased. We also observed that, distributing large number of keys to each node is unnecessary because the number of bridge nodes remained the same but this requires larger memory space, a precious resource. We also observed that number of to-be-connected neighbors sharing keys with the bridge nodes increased with increased N and m value. Network can be more secure when N_{tbc} is high for the bridge nodes because it increases the paths to deliver secret link keys. So if one of the paths is compromised we will still have other paths to send secrets.

In our future work, we will investigate the proposed scheme under more realistic network environments. For example, nodes might be compromised in a region instead of randomly throughout the network. The performance of our scheme may be affected by such a compromise model. Furthermore, we have not studied the effect of node mobility. Moving nodes may change the connectivity and security connection as well.

REFERENCES

[1] J. Allen, J. Wilson, “Securing a Wireless Network”, User Services Conference, ACM Press, New York, 2002, pp. 213-215.

[2] John Ross, A Painless Guide to Wi-Fi and Broadband Wireless, January 2008, 336 pp.

[3] S. Giordano, Mobile ad hoc networks, in: Wireless Networks and Mobile Computing Handbook, ed. I. Stojmenovic (Wiley, 2002), to appear.

[4] Geetha Jayakumar, G. Gopinath, “Ad hoc Mobile Wireless Networks Routing Protocols- A Review”, Journal of Computer science, Vol.3, pp. 574-582, 2007.

[5] C.-K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall PTR, Englewood Cliffs, NJ, 2002.

[6] Jeremy Elson and Kay Romer, “Wireless Sensor Networks: A New Regime for Time Synchronization”, Technical report, UCLA Technical Report, July 2002. http://lecs.cs.ucla.edu/Publications.

[7] Mayank Saraogi, “Security in Wireless Sensor Networks”, Department of Computer Science, University of Tennessee, Knoxville.

[8] F. L. Lewis, Technologies, Protocols and Applications: Smart Environments, editors D.J. Cook and S. K. Das, (New York, NY: John Wiley, 2004).

[9] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, vol. 40, no. 8, pp. 102--114, August 2002.

[10] T. Karygiannis and L. Owens, Wireless Network Security-802.11, Bluetooth and Handheld Devices. National Institute of Standards and Technology. Technology Administration, U.S Department of Commerce, *Special Publication* 800-848, 2002.

[11] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary, Wireless Sensor Network Security: A Survey, Security in Distributed, Grid, and Pervasive Computing Yang Xiao,(Eds.) 2006, Auerbach Publications, CRC Press.

[12] Jing Deng and Yunghsiang S. Han, “Babel: Using a Common Bridge Node to Deliver Multiple Keys in Wireless Sensor Networks”, in Proceedings of the Global Communications Conference, 2007, GLOBECOM 07, Washington, DC, USA, 26-30 November 2007, pp.161-165, IEEE, 2007.

[13] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in Proceedings of the 9th ACM conference on Computer and communications security, Washington, DC, USA, November 18-22 2002, pp. 41–47.

[14] Hwang, J. and Kim, Y. 2004, “Revisiting random key pre-distribution for sensor networks”. In ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 04).

[15] H. Chan, A. Perrig and D. Song, “Random key predistribution schemes for sensor networks.” In *IEEE Symposium on Security and Privacy*, pages 197–213, Berkeley, California, May 11-14 2003.

[16] Ross Anderson and Adrian Perrig. Key infection: Smart trust for smart dust. Unpublished Manuscript, November 2001.

[17] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks.” In ACM Conference on Computer and Communications Security (CCS), 2003.

[18] R. Blom. An optimal class of symmetric key generation systems. In EUROCRYPT 84, 1985.

[19] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, Washington, DC, USA, October 27-31 2003, pp. 52–61.

[20] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. *Lecture Notes in Computer Science*, 740:471–486, 1993.

[21] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. Technical Report, Syracuse University, July 2003. Available from http://www.cis.syr.edu/□wedu/Research/paper/ddhcv03.pdf.

[22] C. E. Perkins, Ed., *Ad Hoc Networking*. Addison-Wesley, 2001.

[23] Erd˝os and R´enyi, “On random graphs I,” *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[24] S. Shakkottai, R. Srikant, and N. Shroff, “Unreliable sensor grids: coverage, connectivity and diameter,” in *Proceedings of the IEEE INFOCOM*, 2003, pp. 1073–1083.

[25] M. Tatebayashi, N. Matsuzaki, and D. B. Newman, “Key distribution protocol for digital mobile communication systems,” *Advances in Cryptology - CRYPTO’89*, pp. 324–334, 1989, INCS Volume 435, Springerverlag.

[26] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public key cryptosystems,” *Commum. ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[27] C. Park, K. Kurosawa, T. Okamoto, and S. Tsujii, “On key distribution and authentication in mobile radio networks,” *Advances in Cryptology-EuroCrypt’93*, pp. 461–465, 1993, INCS Volume 765, Springer-verlag.

[28] AZIZ, A., ANDDIFFIE, W. Privacy and Authentication for Wireless LocalArea Networks. In *IEEE PersonalCommunications* (First Quarter 1994), IEEE, pp. 25–31.

[29] M. J. Beller, L.-F. Chang, and Y. Yacobi. Privacy and authentication on a portable communications system. IEEE Journal of Selected Areas in Communi-cations, 11(6):821-829, 1993.

[30] Carlsen, U., 1994. Optimal privacy and authentication on a portable communications system. Operating Systems Review 28 (3), 16–23.

[31] C.J. Mitchell, “Security in Future Mobile Networks”, *Second International Workshop on Mobile Multi-Media Communications (MoMuC-2)*, Bristol, April 1995. Also available online at http://isg.rhbnc.ac.uk/cjm/SIFMW.ZIP.

[32] C. Boyd and A. Mathuria, “Key establishment protocols for secure mobile communications: A selective survey,” *Lecture Notes in Computer Science*, vol. 1438, pp. 344–355, 1998.

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

[34] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, “Providing robust and ubiquitous security support for mobile ad-hoc networks,” in *International Conference on Network Protocols (ICNP)*, 2001, pp. 251–260.

[35] Hui Ling and Taieb Znati. End-to-End Pairwise Key Establishment using Multi-path in Wireless Sensor Network. *Proceedings of the 2005 IEEE Global Communications Conference (GLOBECOM 2005) (To appear)*, December 2005.

[36] G. Li, H. Ling, and T. Znati, "Path key establishment using multiple secured paths in wireless sensor networks," in *Proc. of CoNEXT '05*, 2005, pp. 43–49.

[37] P. Traynor, H. Choi, G. Cao, S. Zhu, and T. LaPorta, "Establishing pair-wise keys in heterogeneous sensor networks," in *Proc. of the 25th Conference of the IEEE Communications Society (Infocom '06)*, Barcelona, Spain, April 23-29 2006.

[38] J. Deng and Y. S. Han, "Using MDS codes for the key establishment of wireless sensor networks," in *Proc. of the International Conference on Mobile Ad-hoc and Sensor Networks (MSN '05)*, X. Jia, J. Wu, and Y. He, Eds., Wuhan, P. R. China, December 13-15 2005, vol. 3784 of *Lecture Notes in Computer Science (LNCS)*, pp. 732–744, Springer- Verlag.

[39] L. Lamport, R. Shostak, andM. Pease. The Byzantine generals problem. *ACMTransactions on Programming Languages and Systems,* 4(3):382–401, 1982.

[40] D. Huang and D. Medhi, "A byzantine resilient multi-path key establishment scheme and its robustness analysis for sensor networks," in *Proc. of 19th IEEE International Parallel and Distributed Processing Symposium*, Colorado, USA, April 4-8 2005, pp. 240b–240b.

APPENDIX A

SIMULATION CODE

A.1 Network Setup

```
N=100; % Number of nodes

R=200; % Transmission range

P=2000; % Pool of keys

M=50 % Number of keys

        % ----------Randomly distribute the nodes----------%

X=1000*rand ( 1, N );

Y=1000*rand ( 1, N );

        % --------------------Neighbor matrix----------------------%

for i=1:N

for j=1:N

D ( i, j )=sqrt ( ( x(j) - x(i) )^2+( y (j)- y(i) ) ^2 ); % Distance between the nodes

If ( D ( i, j ) > 0 && D ( i ,j ) < R)

NB ( i, j )=1;

else

NB ( i, j )=0;

end
```

```
end
end
        % --------------------Neighbors of Source Node----------------------%
NB_S=zeros (1, N );
for i=N % Source Node
for j=1:N-1
if (NB ( i ,j ) == 1 )
NB_S ( j )=1;
end
end
end
        % --------------------TTL----------------------%
for TTL=1:3
for i=1:N-1
if(NB_S ( i )==TTL)
for j=1:N-1
if(NB ( i ,j )==1 && NB_S ( j )==0)
NB_S ( j )=TTL+1;
end
end
end
end
```

```
end
```

A.2 Key Distribution

```
Keys = zeros ( N, M );
for I = 1:N
temp = randperm(P); % Randomly distribute keys from Pool of keys
for j=1:M
Keys ( i, j ) = temp ( j );
end
end
        % -------------------- Shared Neighbors----------------------%
k1=1;
for i=1:N-1
if (NB_S(i) ==1)
temp1=intersect ( Keys ( i, : ), Keys ( N, :) );
t( k1) = numel ( temp1 );
if (t ( k1 ) > 0)
SN ( i ) = 1;
else
SN ( i ) = 0;
end
k1 = k1+1;
end
```

```
end
        % -------------------- To-be-Connected Neighbors----------------------%
for i=1:N-1
if ( NB_S (i) == 1 && SN (i) == 0 )
TN ( i ) =1;
elseif (NB_S (i)==1 && SN (i)==1)
TN ( i )=0;
else
TN ( i )=0;
end
end
```

A.3 Multiple-Bridge Nodes

```
k2=1;
k3=1;
Index=find ( TN ( 1, : ) );
N_TN=numel ( Index ); % Number of to-be-connected neighbors
for i=1:N-1
for j=1:N_TN
if ( NB_S ( i )==TTL)
temp2 = intersect ( Keys ( i, : ), Keys ( N, : ) );
s ( k2 ) = numel ( temp2 );
```

```
if ( s (k2) > 0 )
temp3 = intersect ( keys ( Index (j), : ), Keys (i , :) );
g ( k3 ) = numel ( temp3 );
if ( g ( k3 ) > 0 )
BN ( i , j )=1; % Multiple bridge nodes
else
BN ( i , j )=0;
end
k3 = k3+1;
end
k2 = k2+1;
end
end
end
```

A.4 Network Security

```
xc = 0.4   % Compromise capability
temp4 = rand ( 1, N );
compromised = zeros ( 1, N );
for i=1:N
if ( temp4 (i) < xc )
compromised ( i ) = 1; % Nodes compromised
```

```
else
compromised ( i ) = 0;
end
end
total_BN=zeros ( 1, N_TN );
comp_BN=zeros ( 1, N_TN);
for i=1:N-1
for j=1:N_TN
if ( BN ( i , j )==1 && compromised ( i )==1 )
comp_BN ( j )=comp_BN ( j )+1; % Compromised BridgeNodes
end
end
end
for i=1:N-1
for j=1:N_TN
if ( BN ( i , j ) ==1 )
total_BN ( j )=total_BN ( j )+1; % Total number of BrigeNodes
end
end
end
xp= 0.5     % Compromise probability
C_TN=0;
```

```
for i=1:N_TN
if (comp_BN ( i ) / total_BN ( i ) > xp )
C_TN=C_TN+1;  % Total number of to-be-connected neighbors compromised
end
end
```