A Novel Clustering Paradigm for Pre-distribution Key Management for Mobile Homogenous Wireless Sensor Networks

A Thesis

Presented to the

Graduate Faculty of the

University of Louisiana at Lafayette

In Partial Fulfillment of the

Requirements for the Degree

Master of Science

Mohammad Rezaeirad

Fall 2012

A Novel Clustering Paradigm for Pre-distribution Key Management for Mobile Homogenous Wireless Sensor Networks

Mohammad Rezaeirad

APPROVED:

Magdy A. Bayoumi, Chair Professor of Computer Engineering Director of The Center for Advanced Computer Studies	Dmitri Perkins Associate Professor of Computer Science The Center for Advanced Computer Studies
Hong-Yi Wu Professor of Computer Engineering The Center for Advanced Computer Studies	David Breaux Dean of the Graduate School

Dedication

To my dears: Ali and Mansoureh

Acknowledgments

I sincerely thank my supervisor, Professor M. Bayoumi, and my co-supervisor, Professor D. Perkins. I owe my deepest gratitude to my parents, Professor G. A. Rezaeirad and Professor M. Y. Nikravesh. This thesis would never have been possible without Dr. M. Orooji's made a great contribution.

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTORY MATERIAL

1.1 Introcuction to Wireless Sensor Networks

Wireless sensor networks (WSNs) comprise of a base station and number of tiny battery-powered and low cost devices each with sensing, data processing (one or more micro-controllers, CPUs or DSP chips), may contain multiple types of memory (program, data and flash memories), and communication capabilities. The nodes communicate wirelessly and often self-organize after being deployed in an ad-hoc fashion. The sensor node has a actuator unit that allows it to interrelate with the surround environment [1]. This physical characteristic of sensor nodes and various types of WSNs topologies make it possible to project them into an essential part of a wide variety of applications. Example of these applications are:

Military Applications: WSNs are an integral part of military command, control, communications, computing, intelligence, surveillance, and reconnaissance. A number of WSN applications have been found in the military, for instance: battlefield surveillance, nuclear, biological and chemical attack detection and reconnaissance, and battle damage assessment.

Environmental monitoring: WSNs are used in many applications for monitoring the environment. For example, there are applications for monitoring the movements of animals, detection of forest fires, floods detection, and surveillance of the environmental factors that affect agricultural crops and livestock.

Medical and Health Care Applications: WSNs have been used in a number of health care applications including integrated patient monitoring, diagnostics, drug administration in

hospitals, and tracking and monitoring doctors and patients inside a hospital.

Home Applications: As we mentioned, the technology is improving rapidly, and smart sensor nodes and actuators can be found in home appliances, such as vacuum cleaners, micro-wave ovens, and refrigerators. These sensor nodes found inside home devices can interact with each other and with an external network permitting end users to manage home devices locally and remotely.

Other Commercial Applications: Commercial applications of WSNs include monitoring; product quality; constructing smart office spaces; environmental control in office buildings; robot control and guidance in automatic manufacturing environments; interactive toys; detecting and monitoring car thefts; and vehicle tracking and detection.

Among WSNs application, mission critical application such as military needs to put more into consideration. In this type of deployment, a large number of sensor nodes are utilize to monitor a wide area, where the working situations are commonly tough [2, 3]. Since these nodes are typically positioned in distant locations, they should be armed with security appliances to provide information assurance against any unwanted information leakage [4].

1.1.1 Organization of Thesis

In this chapter, we give an introduction about WSNs, its security requirements, limitations, attacks and cryptographic solutions. In chapter 2, we review existing key managements for WSNs as the fundamental building block of any cryptographic solution. We[1] discuses cluster-based key managements and propose our cluster paradigm, as well as developed schemes based on this paradigm, in chapter 3. In chapter 4, we introduce the evaluation metrics for key management and, we base and present our analysis and results upon these metrics. In chapter 5, we conclude our work and we discuss our future work.

[1]This thesis is a part of a research project conducted by Mohammad Rezaeirad, Sahar Mazloom, Dr. Mahdi Orooji, Dr. Miao Jin, Dr. Dmitri Perkins and Dr. Magdy A. Bayoumi.

1.1.2 Security Requirements

A WSN is a distinct type of networks. It also shares some commonalities with a traditional computer network, but also positions exclusive security requirements of its own [4, 5, 6, 2, 3]. The aim of security in a WSN is to shield the information and resources against attacks and misbehavior. Therefore, any security service should include the following perpetrates:

Access control: It avoids unauthorized access to the resources.

Anonymity: It hides the source of a data, this service also can help with data confidentiality and privacy.

Authentication: It ensures that the sender of data is the node that is claimed.

Authorization: It ensures that only authorized parties can provide information to network services.

Availability: It is an ability to sustain the networking functionality without any interruption due to security threats such as DOS attacks.

Confidentiality and privacy: It protects content of a message from unwanted parties. It prevents opponents from obtaining information that may have private content.

Freshness: It ensures that a data is recent and a malicious node does not resend previously captured packets.

Integrity: It ensures that a packet is not altered throughout transmission.

Non-repudiation: It verifies the source of a packet and avoids the source from denying that it sent a packet where in authentication the source only proves its identity.

Resilience (robustness): It sustains the network functionality when a portion of nodes is compromised or destroyed.

Secure Localization: It is an ability of attaining exact location information in order to isolate the location of a fault.

Self-Organization: A wireless sensor network requires every sensor node be autonomous and flexible enough to be self-organizing and self-healing allowing the different situations.

Time Synchronization: The ability to achieve time synchronization in other to serve time synchronized network such as collaborative WSNs.

1.1.3 Limitations of WSNs

A WSN has several limitations [5, 6, 1]. Due to these limitations, it is hard to employ any arbitrary security mechanism on wireless sensor networks. There limitations are:

Memory: Any security mechanism requires a certain amount of resources to satisfy security requirements. A sensor node is a miniscule device with only a slight volume of memory and storage space for security related code. So as to form an effective security mechanism, it is essential to minimize the code size of the security algorithms.

Energy: In the other hand, power is the major constraint toward wireless sensor capabilities. Energy in a sensor node can be consumed by sensor transducer, sensor radio transceiver or sensor microprocessor. A study in [5, 7] shows that communication is the major consumer element in WSNs. Additionally, providing higher security levels in WSNs has a direct impact on energy consumption. When it comes to implementation of a cryptographic protocol within a sensor node, the energy impact of the added security code must be highly considered. The additional power consumed by sensor nodes as a result of applying security, is related to the processing needed for security functions, the energy required to exchange security related data, and the energy required to securely store the security parameters.

Communication: Beyond doubt, unreliable communication is alternative risk towards

of sensor security. The network security highly rest on communication protocols. Generally, WSNs offer unreliable connectionless routing. In the course of data exchange packets, they may possibly get corrupted due to channel errors or dropped right at extremely congested nodes. More importantly, with the absence of the proper error control mechanism it is more likely to lose critical security data. Even though the channel is reliable, the communication might still be unreliable. Due to the broadcast nature of the wireless sensor network packets might collide during course of communication. This will be especially more problematic in overcrowded sensor networks. Besides, in the multi-hop routing, network congestion can result in more interruption. Consequently, it is difficult to attain synchronization between nodes. This issue is serious in some application such as key management system. Additionally, the communication range of low power sensor nodes is technically imperfect. The real range of radio transmission depends on several environmental aspects for instance climate and terrain.

Administration: It is hard to administrate and control a sensor node remotely; this makes it almost infeasible to notice physical attack even worse when physical maintenance has to be done. For instance, sensor nodes might be used for critical missions in remote environments such as a battle field. In such a case, the node may not have any physical interaction with friendly forces once deployed [5, 8]. Besides, there is no central management point. Generally WSNs are distributed type of networks without a central management point. This may intensify the strength of the sensor network. However, it makes the network organization tough, wasteful, and delicate if they do not design properly.

1.2 Security Attacks in WSNs

Security attacks can be classified into two broad categories of passive and active attacks [5, 3]. In passive attacks, opponents do not make any visible attack and are mostly

against data confidentiality. Whereas in active attacks, malicious nodes act against both data confidentiality and data integrity. Active attacks can also target to launch unauthorized access, to consume resources or to interrupt communications. An active opponent makes an emission that can be spotted.

1.2.1 Passive Attacks

In passive attacks opponents are normally unknown and unseen, and intercept the communication to gather sensitive data [9]. For example, in *Passive Eavesdropping*, classified data can be eavesdropped by intercepting communication mediums, and WSNs are easier to tap and WSNs are more vulnerable to passive attacks. In particular, when known standards are used and unencrypted data are sent wirelessly, an opponent can simply obtain and read the data. Essentially, sensor networks are a bit harder to be tapped comparing to other longer range wireless technologies, that is because of the signals are sent over shorter distances. In order to do interception, an opponent requires getting close enough to the target node.

In addition to the content of data, in *Traffic Analysis*, the traffic pattern may possibly expose sufficient information about network to the opponents. Through analyzing the traffic, this sort of valuable information can be derived. In WSNs, the nodes closer to the base station, make more transmissions than the nodes farther from the base station. The nodes close to base station may perhaps be too valuable for opponents because a DOS attack against these nodes or sniffing the data intended for them may have a greater impact.

1.2.2 Active Attacks

This type of attack needs the opponent to be able to transmit data to one or both of the sender and receiver, or obstruct the data stream in one or both directions [10]. It is also possible that the opponent is situated between the communicating parties. Sometimes the opponent tries to stay invisible, aiming to gain unauthorized access to the resources of the

network. An active attack could occur in the form of *physical attack* where an opponent may physically harm hardware to let off the nodes. Physical attacks against hardware may become a serious problem, especially in WSNs. Sensor nodes may be installed unattended in areas available by the opponent. Hence, an opponent can extract sensitive information on the node. The node may also be transformed to a node which the opponent controls and launches attack with. Thus, tamper resilience needs to be considered cautiously in WSNs. When these risks are forthcoming, nodes require being robust against these physical attacks. Also an active attack could be a *Denial of Service (DOS) attack* where the opponent tries to interrupt, destabilize, or extinguish a network. In WSNs vulnerable to DOS attacks may occur in any layer of communication in sensor networks [11]. DoS attacks primarily target the availability of network services. A DOS is defined as any occurrence that weakens a network's capacity to perform its expected function appropriately in a timely manner.

1.2.3 Active Attacks Based on Communication Layers

Active attacked fall into following categories based on communication layers [5, 3]:

1.2.3.1 Physical Layer Attacks

As with any wireless network, there is the opportunity of jamming in WSNs [12, 11, 10]. Also, nodes in WSNs may be positioned in insecure environments where they are accessible to opponents.

Jamming is one physical layer attack, where opponent via transmission of radio signals disrupts communications by decreasing the signal to noise ratio [12, 11, 10]. A jamming source may be powerful enough to disturb the whole of the network. General defenses against jamming involve variations of spread-spectrum communication techniques such as frequency hopping and code spreading. But, the available frequency band is limited;

an opponent may instead jam a wide sector of the frequency band. On the other hand, most of these techniques require greater design complexity. Generally, sensor nodes are design to use the single-frequency due to preserve low cost and low power requirements. Therefore, WSNs are extremely vulnerable to jamming attacks.

1.2.3.2 Data Link Layer Attacks

Collisions: In this type of attack, opponent might deliberately force collisions in particular type of frames such as ACK [11]. A possible consequence of such collisions is the high exponential back-off in certain media access control (MAC) protocols. A usual countermeasure against collisions is to use the error detection and correction protocols. However, these protocols also add extra processing and communication overhead. It is workable to treat an opponent like other participating nodes that may cause collision unintentionally. However, it is always possible to detect these malicious collisions.

Exhaustion: Resource exhaustion attack is a DoS by starving a network resources [11, 10]. In particular, opponent may cause frequent collisions and as a result the energy reserves of the communicating node and the surrounding ones will be significantly depleted. One possible resolution is to control and limits the rate of admission on MAC protocol in which the network can drops unnecessary requests, to do so, we may avoid the energy drain caused by repeated transmissions [13]. A second technique is to use multiplexing techniques such as time-division multiplexing (TDM) in which each node is allowed to transmit during its time slot [14]. However, it is still vulnerable to collisions.

Unfairness: Unfairness can be measured as a weak form of a DoS attack [10]. An opponent might launch this type of attack against a WSN by irregularly using the maintained link-layer type of attacks. Instead of preventing access to a service entirely, an opponent can

reduce it so as to take an advantage such as causing other nodes in a real-time MAC protocol to miss their transmission deadline. The use of small frames reduces the chance of such attacks by minimizing the amount of time an opponent may be able to occupy the transmission channel. However, this technique often reduces efficiency and is vulnerable to other type of unfairness where the opponent, instead of back off randomly, attempts to retransmit swiftly.

1.2.3.3 Network Layer Attacks

Spoofed Routing Information: An opponent may spoof, alter, or replay routing information in order to disrupt traffic in the network [15]. Specifically, a malicious node may broadcast routing or other information with high enough transmission power to convince every other node in the network that it is their neighbor. When the other nodes send their packets to the malicious node, those packets are not received by any node. This type of attack consider as the most direct attack against a routing protocol in any network. It is used to target the routing advertisement while it is being exchanged between nodes. A countermeasure against spoofing is to apply a message authentication code (MAC). Through adding a MAC to the message, the receivers can ensure whether the messages have been spoofed. There are two types of this attack. *(1) Selective Forwarding:* a simple hypothesis made in WSN is that all nodes in the network will truthfully forward and received packets. An opponent may form a set of malicious nodes which selectively forward only certain packets and simply drop the others. *(2) Detour attack:* an opponent may detour traffic to a partition of the network. Various techniques can be used for this. In [23], Hu et al. define a gratuitous detour attack, where a malicious node adds virtual nodes to the routing table such that the updated route becomes more costly compared to another route to which the opponent tries to detour traffic.

Black Hole Attack: a malicious node drops all the packets that it receives for

forwarding [16]. This attack is particularly effective when the black hole node is also a sink hole. Such an attack combination may stop all the data traffic surrounding the black hole. One defense against this attack is to detect the malicious node to isolate it and then seek an alternative route [17].

Sinkhole Attack: In a sinkhole attack, the opponents' goal is to trap almost all the traffic from a specific area over a compromised node, making a figurative sinkhole with the opponent at the center. Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the forge routing algorithm. An opponent might spoof or replay an advertisement for an extremely high quality route to a sink, effectively an attracting route for a sink from nodes several hops away from the compromised node.

Sybil Attack: The Sybil attack is a case where a single node presents multiple identities to other nodes in the network. They pose a significant threat to Protocols such as distributed storage protocol, geographic routing protocol. In particular, a distributed storage protocol depends on existence of replicated of the same data to achieve redundancy. If attacking node displays multiple identities, the protocol assumes that redundancy has been achieved while it has not [18].

Wormholes Attack: In this type of attack, an opponent tunnels data received in one part of the network over a low latency link and forwards and replies them in a different part. This link may be established by a single node situated between two adjacent nodes forwarding data between the two of them. Conversely, wormhole attacks more frequently involve with non-neighboring nodes where two distant malicious nodes planning to minimize their distance from each other by relaying packets along an out-of-bound channel available only to the opponent. In the other scenario which is closely related to the sinkhole attack, an opponent

located nearby to a sink can completely interrupt routing by making a well-placed wormhole. An opponent could interest nodes who would normally be multiple hops away from a sink that they are just one or two hops away via the wormhole. This can fashion a sinkhole: since the opponent placed on the other side of the wormhole can exaggeratedly deliver a high quality route to the sink. In [19] authors introduced an original method called packet leashes for discovering and defending against wormhole attacks.

Hello Flood Attacks: Given the fact that many protocols use HELLO packets to advertise to receiving node that the sender is within radio range and therefore is a neighbor [16]. A malicious node may broadcast poison routing advertisement with high enough transmission power to trick every possible node in the network that it is their neighbor. When the other nodes send their packets to the malicious node, those packets are not received by any node

Acknowledgment Spoofing: Despite the fact that most of routing protocol in WSN are connectionless and also unreliable, but still there are routing algorithms which used in sensor networks sometimes require Acknowledgments to be used. An attacking node can spoof the Acknowledgments of overheard packets destined for neighboring nodes in order to provide false information to those neighboring nodes [20]. An example of such false information is claiming that a node is alive when in fact it is dead.

1.2.3.4 Transport Layer Attacks

Flooding: it is also known as connection request spoofing such that a malicious node can send many connection requests to a node, using up its resources such that it cannot accept any more connection request. In specific, TCP protocol is become vulnerable in the course of closing the connection [11, 5]. An opponent may frequently make new connection requests

until the resources required by each connection are drained. In this case, all the new requests will be dropped. One suggested solution to this problem is to oblige every single connecting node to prove its commitment to the connection by solving a puzzle [21, 11]. The idea is that a connecting client will not surplus its resources by creating unnecessary connections. Assuming that an opponent does not likely have infinite resources, it will be dreadful for him/her to create new connections fast enough to let resource starvation on the targeting node. However, this technique introduces processing overhead.

De-synchronization: De-synchronization denotes the interruption of an existing connection [11]. An opponent frequently spoofs control segment to a node, make the source node believe that the segment was never delivered successfully. If timed correctly, an opponent may prevent the ability of the end nodes to effectively exchange data, so then causing them to drain energy by trying to recover from errors which never really happened.

Acknowledgment Spoofing: is a phony acknowledgment or acknowledgment with large receiver sliding windows size may force the source node generate more segments than the network can handle, causing congestion and degrading the network capacity [11, 5, 3].

1.3 Cryptographic Solutions

The cryptographic methods can be seen as satisfactory mechanisms to meet security needs of WSN. Table Table 1.1, shows a correlation between security services and security requirement of WSN. Among these services, public key cryptographic methods are not possible to be used in WSN [22, 23, 24]. These types of services require higher hardware capacity in terms of processor and energy while private key cryptographic methods are friendlier to existing WSN platforms. With a longer key size, a private key system can offer a good level of security as compare to a public key system. Therefore, applying any cryptographic method as a security solution to WSN, demands corresponding hardware

Table 1.1: Cryptographic mechanisms.

Cryptographic Method	**Security Service**	**Remark**	**Examples**
Public Key (asymmetric) Encryption	Confidentiality and privacy	Computational expensive, Higher security, Higher scalability	RSA [25], ECC [26]
Private Key (symmetric) Encryption	Confidentiality and privacy	Low computation overhead, secure with large key length	IDEA [27], AES [28]
Digital Signature	Source Authentication, Data Integrity and Non-repudiation	Computational expensive, Higher security, Higher scalability	ElGemal [29]
Hash Function	Data Integrity, Verification, Key Generation	Low computation overhead, secure with large key length	MD5 [30], SHA-1 [31]
Message Authentication Code (MAC)	Message Authentication and Data Integrity	Low computation overhead, secure with large key length	MD5, SHA-1

resources.

Chapter 2

TAXONOMY OF KEY MANAGEMENT SCHEME FOR WIRELESS SENSOR NETWORKS

Wireless sensor networks (WSNs) target a wide range of applications: from civilization to military tasks. While in a WSN, nodes are resource limited and rely on intermittent wireless communication, WSNs are designated as a good candidate for mission critical applications (such as espionage assignment) where human intermission is not possible and used devices should be small enough not to be easily detected. In such applications, sensor nodes are deployed in hostile areas where communication might be intercepted, sensor nodes are subject to capture and impersonation by an adversary and maintenance after node installation is impossible [3]. In these situations, security services become tremendously essential to safe guard the communication between sensors network elements. Among these security services, key management is known as the most vital element.

Due to the importance of key management, a variety of key management schemes have been proposed. Considering the power and storage constraints of sensor nodes in WSNs, a key management should have a lightweight design when satisfying the security requirements. Traditional key management techniques using public key infrastructure (PKI) may not be appropriate for a WSN. It is due to excessive energy consumption and hardware requirements of the PKI. Centralized key management techniques also may not be suitable for WSNs. Centralized architecture limits the scalability of network and leaves a single point of compromising for entire of the network.

The well accepted energy-efficient approach for key establishment in WSNs is based

on the idea of the pre-distribution of the symmetric keys among the sensors before deployment of a network. A very first approach using key pre-distribution scheme (KPS) is to employ a single mission key for the entire network nodes to secure their communication traffic. The main drawback to this approach is that, compromising of only one key in the network leads to compromising of the network. The second approach is to generate a unique key for each node in the network of size n and load each node with $n-1$ keys belong to other sensor nodes. Obviously, this scheme is not memory efficient and the re-keying mechanism will be very hard to apply to this scheme. Therefore, Eschenauer *et al.* [32] proposed the first random key pre-distribution scheme (aka. Basic scheme) that relies on probability of key sharing among the nodes of a random graph. In this method, each sensor node receives a set of keys from a large key pool before deployment. Two nodes can initiate a secure links if they find a key in common. Chan *et al.* [33] improved [32] by proposing a pre-distribution key management (aka. q-composite scheme) that requires two nodes to share at least q number of keys to setup a secure communication. Their scheme improves the security of basic approach by providing more resilience against node capture attack. However both the basic and q-composite schemes assume no deployment information prior to node placement (e.g. node location), and nodes are randomly positioned in network.

Considering security issue and scalability limitation imposed by key pre-distribution [34], an alternative approach is to pre-load sensors with a key generation method introducing limited computational overhead instead of the actual keys
[35, 36, 34, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46]. As such schemes, [35] relies on pre-distribution of polynomial shares of a randomly generated and symmetric bivariate $t-$degree polynomial. The major limitation of this scheme is the lack of scalability because the security of a network will be jeopardized after the capture of $t+1$ number of nodes [47].

Works proposed in [44, 48, 49] utilize a set of perturbation polynomials to solve the scalability problem of Blundo's scheme by increasing the resiliency threshold while maintaining efficiency. Unfortunately, a comprehensive security argument given in [50] indicates that these modified schemes can be broken easily by attackers.

Consequently, some works utilized deployment knowledge to improve the security. Such schemes [51, 43, 42, 52, 53, 45, 54, 55, 46] improve sensor network memory efficiency, and resilience against node capture attack by avoiding unnecessary key assignments to nodes. Unfortunately, pre-deployment information is not guaranteed to be available in some scenarios. In particular, Liu and Ning [37] apply deployment knowledge to enhance the scalability and connectivity of q-composite scheme [33], and combine with Blundo's scheme to achieve higher connectivity and less communication overhead.

Unfortunately, scalability is still an unsolved issue of the discussed pre-distribution schemes. A wireless sensor network can be divided to clusters. Cluster-based pre-distribution schemes improve network security, scalability, and flexibility. Specifically, cluster keying offers more resilience based on the fact that compromising a node impacts only residing cluster rather than the entire of a network. Node addition and revocation become more flexible for a large network and this makes the design of a scalable pre-distribution scheme possible. However, most of previous cluster-based pre-distribution schemes consider heterogeneous wireless sensor networks only and assume cluster-heads with stronger hardware capacity (better computation power, memory storage, and radio coverage). Cluster-heads have better control against security attacks, and sometimes they are even assumed to be totally secured. These assumptions are not practical for many mission critical applications such as military espionage operations to detect moving targets or the attendance of micro agent listeners [56]. However, only a few previous works consider a cluster-based key establishment scheme on

homogenous wireless sensor networks in the last few years [51, 54, 40, 45, 55, 53]. They assume deployment knowledge available and consider only static sensor nodes.

Chapter 3

CLUSTER-BASED KEY MANAGEMENT

3.1 Overview of Cluster-based Key Managements

Key management of WSNs has been discussed extensively in previous works [4, 5, 6, 2, 8]. We only give a brief review of cluster-based key management which is most related to ours. For a complete list of recent cluster-based key management schemes, we refer readers to [56].

Liu and Ning [34] extended work at [35] and adopted key pre-distribution schemes [32, 33]. Unfortunately, this approach still suffers from impact of node capture attack to a certain threshold. Similarly, Du *et al.* [43] improved the resiliency of [36] adopting scheme at [32]. Beside the better resilience of this design as compared to [36], there still exists performance issues such as higher storage requirement, communication overhead, and lower local connectivity for this scheme.

Some schemes improve efficiency by applying deployment knowledge. Liu and Ning [37] enhance the scalability and connectivity of [33] by adding location information. In the same work, authors propose to combine their approach with [35] that results in higher connectivity and less communication overhead.

Clustering is another approach to improve security, scalability, and flexibility by dividing network to clusters [56]. Du *et al.* [51] adapted the basic scheme [32] by combining it with deployment knowledge to propose a group-based deployment scheme. This scheme is more memory efficient and offer less communication overhead as well as a better resilience. Since it uses deployment knowledge as an enhancement mechanism, is not flexible for verities

of applications. With the similar approach, the hexagonal group-based key management [54] combined [35] with deployment knowledge, and later this work improved at [40]. In [45], Liu *et al.* introduced a group-based deployment scheme. This general framework is compatible with most of mentioned schemes as its underlying solution and improves their security and performance. This framework allows efficient scalability for those key management schemes that do not scale well. Also, it avoids unnecessary communication overhead.

There are cluster based key managements where the cluster-heads are considered to be more powerful than the other nodes (aka. heterogeneous key management). In these heterogeneous schemes, occasionally cluster-heads carry out special duties in key establishments and therefore they need to be provided with stronger hardware spec. Lu *et al.* [42] proposed a unified framework for distributed key management schemes in heterogeneous wireless sensor networks. This framework highly relies on employing heterogeneous nodes as the main element to improve security and performance. Proposing a solution to improve security capacity of [32, 33] has been an interesting research topic in key management. Chan *et al.* [52] proposed a random key pre-distribution scheme based on [32, 33] and using knowledge of deployment of sensor nodes for different regions of network. Their scheme clusters the network to different regions to provide better scalability and security. This approach suggests good performance while increasing resilience to the attacks for the sensor cluster, compared to those prior schemes. The main disadvantage with heterogeneous schemes is that, they are not flexible with most of network topologies and applications [5, 8, 56]. It is hard to suppose, cluster-heads are fully connected and also secure against attacker, as it is assumed in some heterogeneous schemes [57, 58, 59].

3.2 Outline of Our Proposed Clustering Paradigm

Proposing a solution to improve security capacity of [32, 33, 36, 34] has been an interesting research topic in key management for last few years. The proposed solutions are based on utilizing of pre-deployment knowledge and heterogeneous nodes. Unfortunately, none of these solutions can be applied for a situation where there is no deployment knowledge, and heterogeneous nodes are not available. Therefore, we propose a novel clustering method to develop a random pre-distribution schemes for HWSNs based on [32, 33, 36, 34]. In our design, sensor nodes are randomly distributed in every location of network and clustered into groups. Size and number of groups (clusters) are customized by user application before node deployment. For each one of the clusters a distinct key pool is generated in such a way that there is no overlap between key pools of any two clusters. Cluster formation is only based on the type of pre-loaded keys in sensor nodes. Cluster formation and key establishment are accomplished with zero knowledge about node deployment. We assume all nodes have identical physical characteristics (HWSN). It means that sensor nodes have the same capacity of processing and communication, but they might have different data aggregation or sensing target. On top of all these clusters, we build another cluster that is called universal cluster, as it is shown in Fig. Figure 3.1. Members of this cluster are called consular nodes. These nodes are randomly chosen from all the other clusters by user application, in an offline process. Therefore, a consular node is belonged to two clusters. This universal cluster provides a way for two clusters to exchange some limited information such as security control messages. The type and amount of information exchange can be defined under the application policies. Clusters isolate security threats from within and deal with the security problem locally. For each cluster (including universal cluster), we run a totally separated key distribution. Each cluster can be granted with different security privileges which provide a hierarchical security

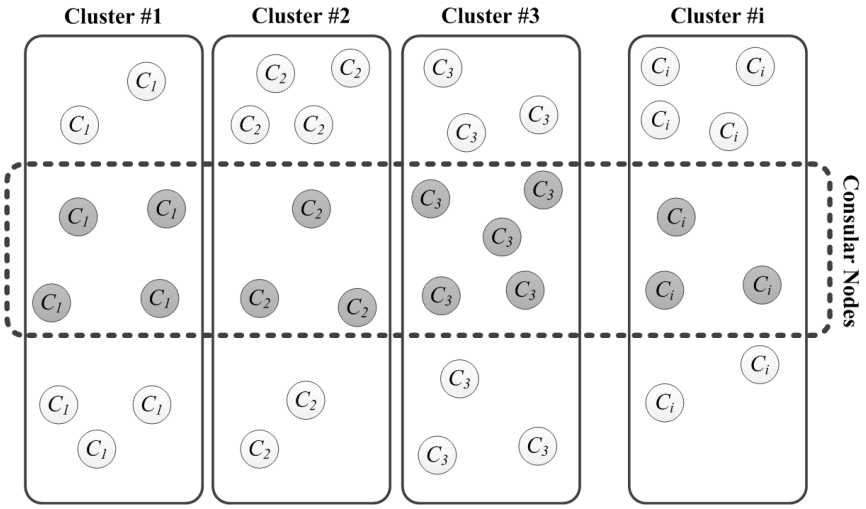


Figure 3.1: Conceptual schematic of proposed paradigm

mechanism. The sensor node can operate in every region of the network (regardless of their geographic location) and communicate with those that are from its own cluster. Of course two communicating nodes should be within the communication range of each other, therefore, a certain node density should be provided to avoid node isolation. To address this issue [32, 33] have made an assumption in which the average number of neighboring nodes is given. We make a slightly similar assumption for the network where for each cluster this number is provided. As it is shown in Fig. Figure 3.2 as long as two adjacent nodes belongs to a similar cluster they may establish an secure intera-cluster communication, and similarly any two neighbor consular nodes may establish a secure inter-cluster links.

3.2.1 Network Elements

The proposed paradigm employs a large number of static (or mobile) sensor nodes with identical physical characteristic (homogenous). We assume nodes are uniformly deployed in the operation zone. Also, a sensor node has enough neighbors, thus, there would be no isolated nodes. In this decentralized network, a base station could be mobile and operates in a location where it is connected to sensor nodes.

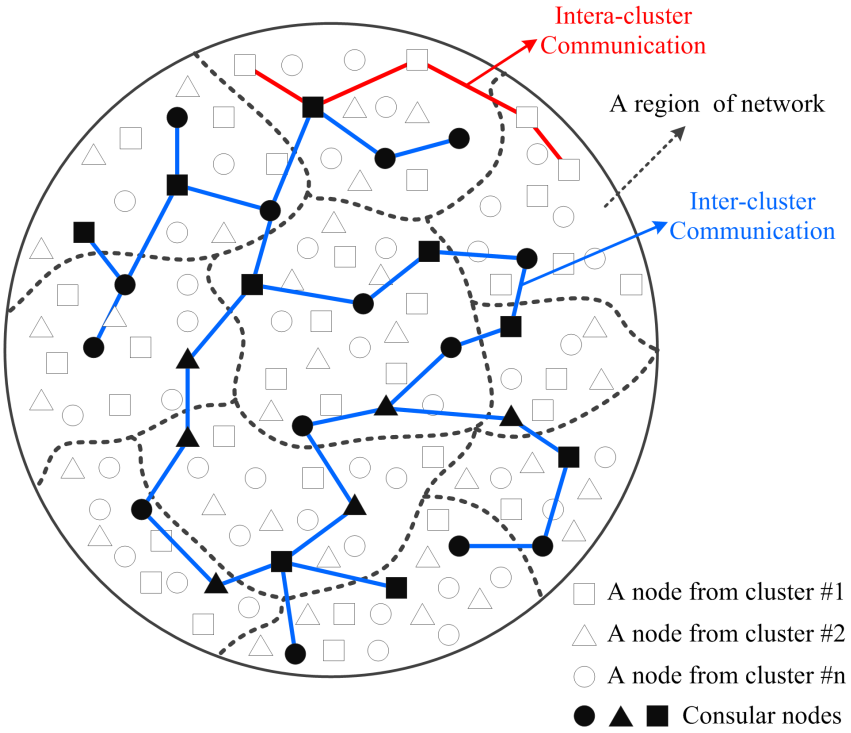


Figure 3.2: A network schematic of proposed scheme

3.2.2 Attacker Properties

We assume an attacker has a very powerful hardware platform. An attacker can eavesdrop any encrypted or unencrypted conversation between nodes instantly (but needs keys to interpret them), and physically capture nodes and immediately discover their containing information. An attacker tries to capture more nodes from network intangibly, although the attacker has limited time to utilize captured information before victimized node is revoked.

3.2.3 Cluster Model

Clusters are formed abstractly; nodes group into clusters based on the roles in their assignment of tasks instead of deployment information and geographic information of nodes. Key materials are uniquely designed for each cluster; therefore clusters provide boundaries for information flow such that there would be no information leakage from one cluster to another. A node cannot leave its current cluster and join another one freely, but the paradigm provides mechanisms for both node addition and revocation based on adopted underlying schemes. Consular nodes are randomly chosen from nodes belonging to the cluster; this confirms that, we do not suggest a heterogeneous design. Fig. Figure 3.1 illustrates the cluster formation. Communication of consular nodes (intra-cluster communication) is the only way for two clusters to exchange critical information (such as control messages) in a secure fashion. In some scenario consular nodes may exchange news of an attack. We assume the amount of inter-cluster communication is much less than intra-cluster. Additionally, two nodes from two different clusters may not have privilege (required keys) to establish a secure connection via consular nodes.

3.2.4 Contributions

The main contributions of this paper are summarized as following:

1. We propose a novel clustering paradigm to be used in static and mobile HWSNs, and

develop three pre-distribution key management schemes based on this paradigm. The paradigm can embrace general pre-distribution schemes as its key establishment protocol.

2. We illustrate that resulting schemes based on our paradigm significantly improves resilience against node capture, and scalability of a HWSN.

3.3 Overview of the Developed Schemes

In this section, we describe our developed key management schemes based on the proposed clustering paradigm. These schemes rely on probabilistic key (or key generator) sharing among the cluster nodes (pre-loaded nodes of clusters) of a random graph. Table Table 3.1 explains some of the notations used in this paper.

3.3.1 Scheme 1: Clustered q-Composite (CqC)

3.3.1.1 Key distribution

Similar to [32, 33], key distribution in our scheme consists of three phases that are: key pre-distribution, shared-key discovery, and path-key establishment.

Key pre-distribution phase: Prior to deployment of sensor nodes in operation field, a key pre-distribution phase is performed. The key pre-distribution phase includes following steps:

- Generation of large key pools S_i, and associated key identifiers for every of ith clusters.
- Random selection of R_i keys out of corresponding key pool S_i, without replacement to form the key ring of each sensor node belonging to i^{th} clusters.
- Generation of a large universal key pool $\mathcal{S}$ and associated key identifiers.
- Random selection of y_i nodes from each cluster members of ith cluster to be assigned as the consular nodes.

Table 3.1: Table of selected notation.

Symbol	**Explanation**
n_i, $\acute{n_i}$, n	Number of nodes, Average number of neighboring nodes of a nodes in i^{th} cluster, Number of nodes in the network
Y, $\acute{Y}$, y_i	Number of consular nodes, Average number of neighboring nodes of a consular node, Number of consular nodes in i^{th} cluster
$\mathcal{C}$	Total number of clusters
R, R_i, $\mathcal{R}$	Ring of key materials, Ring of key materials of a node from i^{th} cluster, Ring of key materials that is assigned to a consular node
S, S_i, $\mathcal{S}$	Pool of key materials, Pool of key materials for a node from i^{th} cluster, Key pool for the universal cluster
q, $\acute{q}$	Least number of required common keys for two nodes in order to establish a pair-wise key, Number of common keys for two nodes
λ, $\acute{\lambda}$	Least number of required common keys for two consular nodes in order to establish a pair-wise key, Number of common keys for two consular nodes
K_{c_i}, $\acute{K_{c_i}}$	A cluster key, A new cluster key for i^{th} cluster
$K_\sigma^{n_i}$, $K_\sigma^{y_i}$	A master key for a node, A master key for a consular node
K_{pw}	A pair-wise key of two nodes
$P_{\mathcal{G}}$, $P_g^{(i)}$, $\mathcal{P}_g$	Global connectivity for the entire of the network, Global connectivity for i^{th} cluster
P_l, $P_l^{(i)}$, $\mathcal{P}_l$	Local connectivity (the probability of existence of a link between two nodes), Local connectivity for i^{th} cluster, Local connectivity for the universal cluster
X, x_i, x_u	Number of captured nodes, Number of captured nodes in i^{th} cluster, Number of captured nodes in universal cluster
P_b, $P_b^{(i)}$, $\mathcal{P}_b$	Probability of breaking a polynomial, after capturing of X nodes by attacker for, the network, i^{th} cluster, universal cluster
P_f, $P_f^{(i)}$, $\mathcal{P}_f$	Probability of cracking a link between two un-compromised nodes after capturing of X nodes by attacker, in network, i^{th} cluster and universal cluster
$\mathcal{P}_r$	Resiliency (Robustness) of the network against X captured nodes
RC, $RC^{(i)}$, $\mathcal{RC}$	Resilient-Connectivity for, network, i^{th} and universal cluster accordingly

- Random selection of $\mathcal{R}$ keys out of key pool $\mathcal{S}$ without replacement to form the key ring of each consular nodes belong to every cluster. The number of keys in the key pools S_i and $\mathcal{S}$, is chosen such that each two random subsets of size R_i in S_i and $\mathcal{R}$ in $\mathcal{S}$ will share at least one key with some probability.

- Generation of a unique cluster-key K_{c_i} for each clusters (including universal cluster).

- Loading related cluster keys and the key rings into the memory of each sensor node; saving of the key identifiers of key rings and associated sensor identifier on the base station; and loading each node with a unique master key that computed as $K_{\sigma}^{n_i} = Enc\,\langle (bi); K_x \rangle$ (where K_x denotes the XORed of all the stored keys in a node $K_x = [K_1 \bigoplus K_2 \bigoplus \ldots K_m \bigoplus \ldots \bigoplus K_l]$, bi denotes the base station's identification and $Enc\,\langle A; B \rangle$ is encryption of data A via key B.).

Shared-key discovery phase: In the shared-key discovery phase, each node tries to discover all common keys with each of its immediate neighbors. Basically, any two neighboring nodes must find out if they are from the same cluster. This can be accomplished by exchanging an encrypted $hello$ message via cluster-key between any two sensor nodes. Similarly, any consular node uses its universal cluster-key to locate other ones. If any two nodes acknowledge its adjacent node, then they can continue to send the list of key identifiers associated with their key rings. An encrypted catalog of sensor's stored keys $Enc\,\langle \alpha; K_{c_i} \rangle$[1] is broadcasted to all the immediate cluster nodes. Then, each node examines received identifier list with its own catalog to find the common keys. If the number of common keys equals to $\acute{q}$ where $q < \acute{q}$ (or $\acute{\lambda}$ for any two consular nodes, where $\lambda < \acute{\lambda}$), then they hash all the $\acute{q}$ (or $\acute{\lambda}$) keys to compute their pair-wise key as such $K_{pw} = H[K_1 \parallel K_2 \parallel K_3 \parallel \ldots \parallel K_{\acute{q}}]$. Therefore,

[1] α is the catalog of stored keys in the sensor node.

every time any two sensor nodes want to establish a secure link, they encrypt their traffic via their pair-wise shared key.

Path-key establishment phase: After shared-key discovery phase, in each cluster there are some nodes that could not establish a pair-wise key and they are still within the communication range of each other. In this situation, intra-cluster connection would not be ideal for some applications that need high network connectivity. Therefore by establishing path-keys, we can enforce the network graph to provide more secure links, and consequently more network connectivity will be delivered. To establish a path-key for any two adjacent sensor nodes, no new keys will be generated via sensor nodes. Simply, sensor nodes utilize their unassigned keys from their key ring. Note that for any two nodes that desire to establish a path-key, there should be a path between the source and destination nodes via some intermediate nodes to let the path-key procedure take place. Path-key establishment can be done via more secure mechanism that is introduced by Ling and Znati [60]; however, applying this scheme incurs extra routing overhead and not well suited for highly mobile sensor nodes.

3.3.1.2 Key revocation

Whenever a sensor node is compromised, it is essential to be able to revoke the entire key ring of that node. Response of our scheme to a compromised node is varied based on type of node participation since the base station knows the key ring and cluster of revoking node.

Intra-cluster revocation: Where the compromised node is not a consular node, the base station identifies the corresponding consular nodes of the revoking node. Then, it sends an encrypted message to each of consular nodes using related consular nodes master key $K_{\sigma}^{y_i}$. This message contains a list of all key identifiers LS, for the key ring to be revoked, and a new

cluster key appended with message authentication code (MAC) that is produced via $K_{\acute{c}_i}$. Meanwhile, base station sends a unicast encrypted message (with corresponding node master key) contains of the $K_{\acute{c}_i}$ to every single cluster member of revoking node (excluding the compromised node) tagged with correct MAC. Once a consular node verifies the revocation message, sends encrypted acknowledge to the base station via $K_{\acute{c}_i}$, immediately after deletes all its keys that match with revoked key ring, and then it generates and broadcasts an encrypted message for cluster members (via $K_{\acute{c}_i}$) to revoke the compromised node. Cluster members leave no chance to establish a secure connection with the revoking node by deleting their common keys with LS. Intra-cluster revocation mechanism assures a list of key identifiers of the compromised node and a new cluster key is securely delivered to each node, absolutely hidden to compromised node. Nodes can also authenticate the revocation message coming from consular node since this message is encrypted with a new cluster key that only shared by base station, consular node and each node.

Consular node revocation: When a consular node is compromised, the base station has to work with all the other consular nodes to complete the revocation process. The base station has to make sure both the universal cluster and targeting cluster members (which its consular node was compromised) delete the possible revealed keys from their key rings. So that, the base station sends an encrypted message using related $K_{\sigma}^{y_i}$ to all the members of the universal cluster. This message contains a LS, and a $K_{\acute{c}_i}$ appended with message authentication code (MAC) that is produced via $K_{\acute{c}_i}$. Consular nodes remove the matched keys with the revoked key ring, and then acknowledge the base station security (with $K_{\acute{c}_i}$). In this situation, there are $Y - 1$ consular nodes remained for the targeting cluster (that its consular node was just revoked within the universal cluster). Thus, the base station and these consular nodes execute

an intra-cluster revocation procedure accordingly.

By the end of revocation process, all the possible secure links to compromised node are removed. In this process, some of the possible security links (those links that are established or might possibly be established) are destroyed. Therefore, involved nodes need to reestablish the shared-key discovery and, possibly path-key establishment, to recover those failed links.

3.3.1.3 Re-Keying

Key-freshness is one of the major requirements for a key management system. Key-freshness defines a lifetime for a key, and if the keys are expired then protocol should update them accordingly. The base station enforces a mechanism to establish re-keying procedures once a key needs to be updated. As it was mentioned in previous section, the revocation process also demands having a re-keying mechanism. The re-keying mechanism could be either event-driven or periodical. Since our scheme is a pre-distribution one, it uses an event-driven key update (path-key establishment technique) to provide new secure links between disjoint nodes (recall that, once a key is revoked some links might go off-line). The only online re-keying mechanism is to provide cluster keys to nodes (and possibly related consular nodes). Our scheme does not provide any periodical key update for sensor nodes. Therefore, if any two securely connected nodes want to update their expired keys, they simply run a random selection of a key K_r from their common keys, and then hash their pair-wise key with that preferred key to produce a fresh pair-wise key ($H[K_r \parallel K_{pw}]$).

3.3.2 Scheme 2: Basic Polynomial-based Instantiation (BPI)

In this instantiation, Blundo's key establishment scheme [35] is employed. Blundo's scheme utilizes a symmetric bivariate t-degree polynomial $f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$ which is

randomly generated over a finite field $\mathbb{F}_Q$, where Q is a pre-determined prime number that is large enough. Prior to deployment, the key setup server computes a polynomial share for every sensor node using its assigned unique identity. To establish a pairwise key between two sensor nodes A_a[2] and B_b, node A_a needs to evaluate its polynomial share $f(a, y)$[3] at node B_b, and similarly, node B_b evaluates its polynomial share $f(b, y)$ using identity of node A_a. Due to the desirable nature of symmetric polynomials, each pair of sensor nodes can compute a common pair-wise key to establish a secure communication, $K_{ab} = K_{ba} = f(a, b) = f(b, a)$.

3.3.2.1 Pre-distribution

- For the ith cluster, a unique symmetric bivariate t_i-degree polynomial $f_i(x, y)$[4] is randomly generated.
- The jth sensor node belonging to the ith cluster is loaded with a polynomial share $f_i(j, y)$ computed based on the node's unique identity and associated cluster polynomial.
- A unique symmetric bivariate t_i-degree polynomial $h(x, y)$ is randomly generated, and the kth consular node is loaded with an additional polynomial share $h(k, y)$.

3.3.2.2 Pair discovery and pair-wise key establishment

Any two adjacent mobile sensor nodes $A_{a,u}$ and $B_{b,v}$ transmit their cluster identifiers, where $u, v \in i$. If they belong to the same cluster ($u = v$), then they exchange their unique identities, where $a, b \in j$ or k. Then, both sensor nodes evaluate their pre-loaded polynomial share at the partner's unique identity $f_u(a, b) = f_v(b, a)$, where the resulting value is their pair-wise key K_{ab}[5] $= K_{ba}$. Since every node belonging to each cluster stores one unique

[2] $\boldsymbol{A_{a,u}}$: Representation of node A, where a and u are the unique identity of node and cluster, respectively.
[3] $\boldsymbol{f_{u,p}(a, y)}$: A polynomial share for node $A_{a,u}$
[4] $\boldsymbol{f_{i,p}(x, y)}$: A bivariate polynomial with index number of p for i^{th} cluster.
[5] $\boldsymbol{K_{ab}}$: A pair-wise key for nodes $A_{a,u}$ and $B_{b,v}$, where $u = v$.

polynomial share , any pair of adjacent nodes from the same cluster may communicate directly and they do not need to establish a path-key.

3.3.3 Scheme 3: Pool-based Polynomial Instantiation (PPI)

The idea of using multiple polynomials originated by Liu *et al.* [34], forms the basis of our second instantiation. In Liu's method, a set of multiple bivariat t-degree polynomials are randomly generated. Afterward a subset of polynomials are picked, and used to computed polynomial shares to be allocated for each sensor node. Then, the sensor nodes try to find at least a common polynomial share with other sensor nodes in order to establish a pair-wise key using the polynomial-based key pre-distribution scheme discussed in [35].

3.3.3.1 Pre-distribution phase

- For the ith cluster, a pool of multiple symmetric bivariate t_i-degree polynomials $S_i = \bigcup f_{i,p}(x, y)$ is randomly generated (where p is the polynomial identifier).

- For the jth sensor node, a fixed size subset of polynomials $R_i = \bigcup f_{i,p}(j, y)$ is randomly chosen from the ith cluster polynomial pool, where $R_j \subset S_i$.

- A pool of multiple symmetric bivariate t_i-degree polynomials $\mathcal{S} = \bigcup h_p(x, y)$ is randomly generated, and the kth consular node is loaded with additional polynomial shares $\mathcal{R} = \bigcup h_p(k, y)$ where $\mathcal{R} \subset \mathcal{S}$.

3.3.3.2 Pair discovery and pair-wise key establishment

Any two adjacent mobile sensor nodes $A_{a,u}$ and $B_{b,v}$ transmit their cluster identifiers, where $u, v \in i$ or k. If they belong to the same cluster ($u = v$), then they exchange a list of their polynomial identifiers α_j (α_a and α_b), as well as their unique identities, where $a, b \in j$. Both sensor nodes examine the received list of polynomial share identifiers. If they find at

least one polynomial share on p^{th} polynomial, the pair-wise key is computed as $K_{ab} = K_{ba} = f_{u,p}(a, y) = f_{v,p}(b, y)$.

3.3.3.3 Path-key establishment

If two adjacent sensor nodes $A_{a,u}$ and $B_{b,v}$, belonging to the ith cluster could not find at least one common polynomial share from p^{th} polynomial, then, they try to find an intermediate adjacent node $C_{c,w}$ (where $c \in j$ or k, and $w \in i$) from their cluster which both of them have established a pair-wise key with him. Assuming there exist $K_{a,c}$ and $K_{c,b}$, then resulting path-key may be computed at node $C_{c,w}$ as, $K_{a,b} = H[K_{a,c}||K_{c,b}]$ (Hash value of $K_{a,c}$ and $K_{c,b}$). This Path-key is encrypted via relative pair-wise key ($K_{a,c}$ or $K_{c,b}$) and sent to $A_{a,u}$ and $B_{b,v}$, then $C_{c,w}$ removes $K_{a,b}$ from its memory. Note that this process can be communication intensive and might introduce security risk since the actual key is computed at intermediate node. In [34], there are two suggested methods for path-key establishment that can be also applied here.

Chapter 4

ANALYSIS, EVALUATION AND RESULTS

4.1 Analysis and Performance Evaluation of the Developed Schemes

In this section, we develop analytical models to evaluate the performance of the key establishment schemes within the proposed paradigm. Specifically, the connectivity of the paradigm under different scenarios is analyzed in section 4.1.1. Resiliency against node attacks is discussed in Section 4.1.4. Scalability and the maximum size of a network are defined in Section 4.1.5.

4.1.1 Connectivity

Irregular connection characteristic of wireless sensor networks is an inevitable issue that impacts the performance of a key establishment scheme. Therefore, connectivity is one of the performance metrics that must be highly considered in evaluation of a key establishment scheme [8]. BPI provides full (100%) global connectivity within each cluster, therefore this section focuses on the local and global connectivity of CqC and PPI.

4.1.1.1 Connected Random Graph

Intermittent connection in wireless sensor networks is one issue that impacts the overall performance of key management protocol. Any two remote nodes establish a secure link if all the intermediate nodes establish required secure links. Any two nodes can establish a direct secure link when they are within the range of each other and share adequate number of keys based on the protocol requirement. Thus, before explaining the probability of having a connected network graph, we have to discuss the probability of sharing the right number of keys between two nodes. As it is described in the fundamental random pair-wise key schemes,

Erdös and Renyi's random graph theory [61] explains the relationship between the local connectivity and the global connectivity. More specifically, considering asymptotic statistical properties for a very large random graph G_n of n nodes, the probability of desire global connectivity is given by $P_{\mathcal{G}} = \lim_{n\to+\infty}[P(G_n)] = e^{-e^{-2c}}$, where $P(G_n)$ denotes the probability of G_n being completely connected and c is an arbitrary fixed real number.

Thus, the degree of a node d in a random graph of n node (the number of established links to other nodes) can be verified as, $d = P_l \times (n-1)$ and $d = \frac{2N_c}{n}$. Therefore, we have $N_c = \frac{n}{2}\log n + cn$, where N_c is the total number of links (connections) between nodes in a random graph.

This implies,

$$P_l = \left(\frac{1}{n-1}\right)\ln\left(\frac{n}{\ln\frac{1}{P_{\mathcal{G}}}}\right). \tag{4.1}$$

(4.1) and Fig. Figure 4.1 illustrate the relationship between P_l and $P_{\mathcal{G}}$. Therefore, for a given n and a desired $P_{\mathcal{G}}$, the required P_l can be computed. For instance, given desired $P_{\mathcal{G}} = 0.99999$ and network size of $n = 600$ nodes, we need to provide $P_l = 0.0299$. Also, the curves in Fig. Figure 4.1 are almost flat when n is large, which indicates that the size of the network has little impact on the required P_l to let the network graph stays connected.

Global network connectivity $P_{\mathcal{G}}$, is influenced by the $P_g^{(i)}$ and $\mathcal{P}_g$. Therefore, for a network consists of $\mathcal{C}$ clusters, $P_{\mathcal{G}} = \mathcal{P}_g \prod_{i=1}^{\mathcal{C}} P_g^{(i)}$.

4.1.1.2 $K-$connected Graph:

A graph said to be $K-$connected, if any $K-1$ nodes of network is failed (or revoked), the graph is guaranteed to be still connected. Form [62], for a $K-$connected graph and any arbitrary node distribution, $P_g^{(i)}$ is given by,

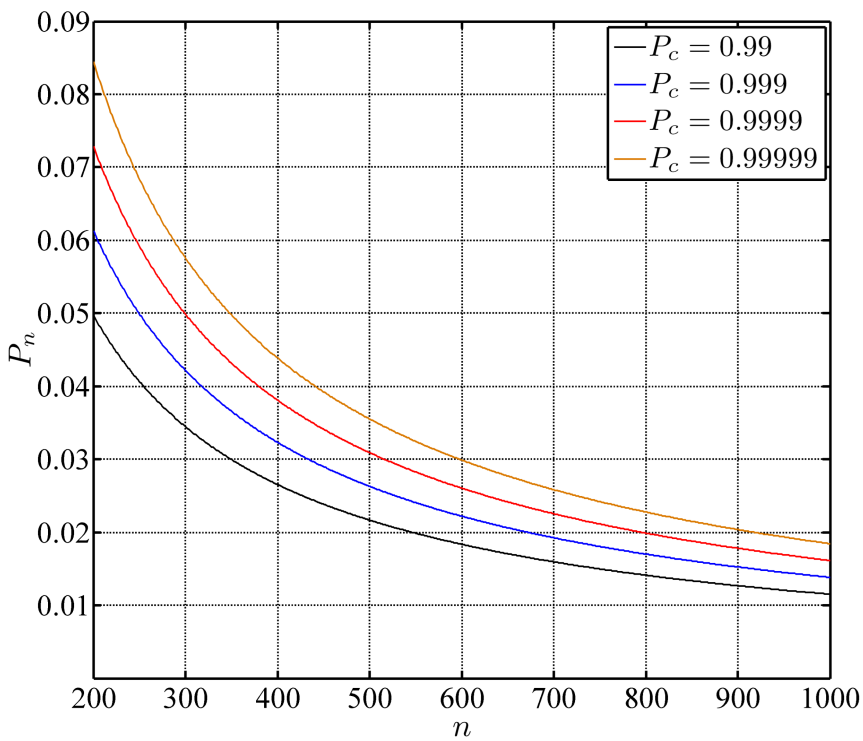


Figure 4.1: Relationship between desire global connectivity $P_{\mathfrak{g}}$ and local connectivity P_l with regard to n.

$$P_g^{(i)} = \exp\left(\frac{-n_i \Gamma(K, n_i P_l^{(i)})}{(K-1)!}\right) \tag{4.2}$$

where $\Gamma(a, b)$ denotes an incomplete Gamma function, defined by

$\Gamma(a, b) = (a-1)! e^{-b} \sum_{i=0}^{a-1} (b^i / i!)$ [63]. From (4.2), when $K = 2$ we have,

$$P_l^{(i)} = \left(1 - W_{-1}\left(\frac{-e \ln P_g^{(i)}}{n_i}\right)\right) \tag{4.3}$$

where $W_{-1}(.)$ signifies the real-valued, non-principal branch of the Lambert function, $W(x)$ [64]. For real valued x, $e^{W(x)} W(x) = x$ has two answers of $W_0 \geq -1$ and $W_{-1} \leq -1$, denoted by principal branch and non-principal branch, respectively.

4.1.1.3 Connected Graph for Mobile Nodes:

In [62], the author defined a general formula to describe the global connectivity when graph includes large number of nodes ($n_i \gg 1$) as,

$$P_g^{(i)} = e^{-n_i e^{-\mu_0}}. \tag{4.4}$$

In (4.4), μ_0 is interpreted as average degree of a node in a graph, that is defined for mobile nodes by

$$\mu_0 \approx \frac{n_i \hat{r}_0^2}{3} \left(\left(4 - 2P_p + P_p^2\right) - \frac{4}{\pi} P_p^2 \hat{r}_0 - 3\left(1 - P_p\right) \hat{r}_0^2 \right) \tag{4.5}$$

where $\hat{r}_0^2 = \dfrac{r_0^2}{a^2} = P_l^{(i)}$ [1] and P_p states the probability that a node pauses at a given time instantaneous. (4.5) holds when $a \gg r_0$ and mobile nodes travels autonomously of other nodes affording to the random way point (RWP) model [65]. If one assumes no pausing node in the network ($P_p = 0$), by substituting $\hat{r}_0^2$ for $P_l^{(i)}$ in (4.5), we have,

$$\mu_0 \approx \frac{-n_i P_l^{(i)}}{3} \left(4 - 3P_l^{(i)}\right). \tag{4.6}$$

Thus, from (4.4) and (4.5) we drive the value of $P_l^{(i)}$ for a desired value of $P_g^{(i)}$, when nodes are mobile, as

$$P_g^{(i)} = e^{-n_i e^{\frac{-n_i P_l^{(i)}}{3}\left(4 - 3P_l^{(i)}\right)}}. \tag{4.7}$$

[1] The area in which the nodes are distributed is a disk of radius a. r_0 is the transmission range of a node, i.e., two nodes can establish a link if the distance between them is less than or equal to r_0.

4.1.1.4 Global Connectivity of the Proposed Paradigm:

$P_{\mathcal{G}}$ is inclined by the global connectivity of every clusters in the network. Therefore, for a network consists of $\mathcal{C}$ clusters, we have

$$P_{\mathcal{G}} = \mathcal{P}_g \prod_{i=1}^{\mathcal{C}} P_g^{(i)}. \tag{4.8}$$

Let us define $\acute{P}_g^{(i)} \triangleq 1 - P_g^{(i)}$ and $\acute{\mathcal{P}}_g \triangleq 1 - \mathcal{P}_g$. for small values of $\acute{P}_g^{(i)}$ and $\acute{\mathcal{P}}_g$ with a good approximation we have,

$$\begin{aligned} P_{\mathcal{G}} &= \left(1 - \acute{\mathcal{P}}_g\right) \prod_{i=1}^{\mathcal{C}} 1 - \acute{P}_g^{(i)} \\ &\approx 1 - \left(\acute{\mathcal{P}}_g + \sum_{i=1}^{\mathcal{C}} \acute{P}_g^{(i)}\right). \end{aligned} \tag{4.9}$$

4.1.2 Determining $|S|$ for CqC

To produce a key pool per cluster and key ring for each node, we should find the right values for size of them. We can calculate these values, from the average number of neighborhood for cluster nodes $\acute{n}$ (as one of the connectivity constraints imposed by wireless communication) and (4.1). Indeed, the number of neighborhoods limited to $\acute{n} \ll n$ nodes, and obviously the probability of direct connection between two nodes in neighborhood will be $P_{\acute{n}} \gg P_l$.

Since n_i and $\acute{n}_i$ are given, and $P_{\acute{n}_i} = P_{n_i}(\frac{n_i}{\acute{n}_i}) \gg P_{n_i}$, the probability of secure direct connection for any two nodes belonging to the ith cluster is calculated as,

$$P_{\acute{n}_i} = 1 - \sum_{j=0}^{q_i-1} \frac{\binom{R_i}{j}\binom{S_i-R_i}{R_i-j}}{\binom{S_i}{R_i}} \tag{4.10}$$

Depending on hardware platform of sensor nodes, the maximum allowable size of the key ring R_i can be defined, and this value is an input to find the minimum value for key pool

size S_i. Combination of (4.1) and (4.10) leads to,

$$\frac{n_i}{\acute{n}_i(n_i-1)} \ln \frac{n_i}{\ln(\frac{1}{P_g^{(i)}})} = 1 - \sum_{j=0}^{q_i-1} \frac{\binom{R_i}{j}\binom{S_i-R_i}{R_i-j}}{\binom{S_i}{R_i}} \tag{4.11}$$

since we have the knowledge of $\acute{Y}$ neighboring node density and Y number universal cluster members (all the consular nodes), and $\mathcal{P}_g$ desired network connectivity of the universal cluster. Then, with the same analysis we can compute the key ring $\mathcal{R}$ and pool $\mathcal{S}$ for the universal cluster by,

$$\frac{Y}{\acute{Y}(Y-1)} \ln \frac{Y}{\ln(\frac{1}{\mathcal{P}_g})} = 1 - \sum_{j=0}^{\lambda-1} \frac{\binom{\mathcal{R}}{j}\binom{\mathcal{S}-\mathcal{R}}{\mathcal{R}-j}}{\binom{\mathcal{S}}{\mathcal{R}}} \tag{4.12}$$

4.1.3 Determining $|S|$ for PPI

Local connectivity in context of key establishment is the probability for two sensor nodes to establish a pair-wise key. The value of local connectivity is directly associated with value of S and R. Thus, we should know how much S should be given for a fixed value of R to hold a required local connectivity. $P_l^{(i)}$ in terms of $|S_i|$ (or $|\mathcal{S}|$), $|R_i|$ (or $|\mathcal{R}|$) and q[2] is given by,

$$P_l^{(i)} = 1 - \sum_{j=0}^{q_i-1} \frac{\binom{R_i}{j}\binom{S_i-R_i}{R_i-j}}{\binom{S_i}{R_i}} \tag{4.13}$$

(4.13) does not consider the communication range of a sensor node. To resolve that, we assume the number of neighbor nodes $\acute{n}$ within communication range of a sensor node is given. Thus, we have $\acute{P}_l^{(i)} = \frac{n}{\acute{n}} P_l^{(i)}$.

Depending on different connectivity requirements (which is defined by an application based on different connectivity models), value of S_i (or $\mathcal{S}$) is varied. Thus, to satisfy a required local connectivity $P_l^{(i)}$ (or corresponding $P_g^{(i)}$), we need to replace $P_l^{(i)}$ in (4.1), (4.3), (4.7) with $\acute{P}_l^{(i)}$ for connected, $K-$connected and mobile connected, respectively.

[2] Assume, these two sensors need to find at least q common number of polynomial shares.

4.1.4 Resiliency Against the Node Capture Attack

In this section, the resilience of our developed key management schemes against node capture attack is evaluated. Once a node is captured, the attacker tries to recover all the keys in the captured sensor node to decrypt the possible encrypted traffic among the other nodes. It is possible that some links can be tapped via the same recovered keys. Therefore, we want to know what fraction of secure connections can be compromised if X number of nodes are captured.

4.1.4.1 Resiliency of CqC

We start our analysis by answering the following question. When $\gamma - 1$ nodes are captured and b keys have been discovered by the attacker, if one more node is captured, what is the probability of a keys becoming known to the attacker, where $a \geq b$. Lemma 1 answers this question.

Lemma 1. *Lets R_i denote the set of key rings of all the ith captured nodes where $2 \leq i \leq X$, and $T_\gamma = R_1 \cup R_1 \cup R_2 \cup \ldots \cup R_\gamma = \bigcup_{n=1}^{\gamma} R_n$, where $2 \leq \gamma \leq X$. Then we have,*

$$\Pr(|T_\gamma| = a \mid |T_{\gamma-1}| = b) = \frac{\binom{S_i - b}{a-b}\binom{b}{R_i - a + b}}{\binom{S_i}{R_i}} \tag{4.14}$$

T_γ represents the set of keys discovered to attacker by successfully capturing γ nodes. We consider the worst case scenario where the entire key ring of a node discovered when a node is captured. Since $|T_{\gamma-1}| = |\bigcup_{n=1}^{\gamma-1} R_n| = b$ thus, $|T_\gamma| = |R_\gamma \cup T_{\gamma-1}| = a$. Note that, $a - b$ keys are chosen from $S_i - b$ remaining keys that were not in key rings of $\gamma - 1$ captured nodes, there are also $R_i - a + b$ keys are common among R_γ and $T_{\gamma-1}$. Therefore (4.14) is a valid.

Suppose x_i and x_u represent the number of capture nodes in i^{th} cluster and the number of capture nodes in universal cluster, respectively. Note that, when γ nodes are captured a

could not be greater than $\min(S_i, \gamma R_i)$, and b could not be greater than a. On the other hand, where $\gamma \leq R_i + 1$ then a could be as small as $R_i + 1$; however, if $\gamma > R_i + 1$ then a may be as small as γ. The lower bound for b can be derived the same as the lower bound for a by considering the fact that $|T_1| = R_i$. Consequently, let us define upper and lower bound values for a and b as $a_{\min} \triangleq \max(R_i + 1, \gamma)$, $a_{\max} \triangleq \min(S_i, \gamma R_i)$, $b_{\min} \triangleq \max(R_i + \min(\gamma - 2, 1), \gamma - 1, a - R_i)$, and $b_{\max} \triangleq a$. Then from (4.14) we drive probability of a keys are discovered by attacker when $\gamma \geq 2$ nodes are captured as

$$\Pr(|T_\gamma| = a) = \sum_{b=b_{\min}}^{b_{\max}} \Pr(|T_\gamma| = a \mid |T_{\gamma-1}| = b) \Pr(|T_{\gamma-1}| = b) \tag{4.15}$$

for $a_{\min} \leq a \leq a_{\max}$, otherwise $\Pr(|T_\gamma| = a) = 0$ for any other values of a.

From (4.14) and (4.15) we have

$$\Pr(|T_\gamma| = a) = \sum_{b=b_{\min}}^{b_{\max}} \frac{\binom{S_i - b}{a-b}\binom{b}{R_i - a + b}}{\binom{S_i}{R_i}} \Pr(|T_{\gamma-1}| = b). \tag{4.16}$$

Let us define $\zeta_i \triangleq \min(X, n_i)$, $\xi_i \triangleq \max(X, n_i)$, and $n = \sum_{i=1}^{\mathcal{C}} n_i$. Then, P_{x_i} denotes the expected fraction of total keys compromised for the ith cluster, that is

$$P_{x_i} = \frac{1}{S_i} \left(\frac{R_i \binom{\xi_i}{1}\binom{n-\xi_i}{\zeta_i - 1}}{\binom{n}{\zeta_i}} + \sum_{x_i=2}^{\zeta_i} \frac{\binom{\xi_i}{x_i}\binom{n-\xi_i}{\zeta_i - x_i}}{\binom{n}{\zeta_i}} \sum_{a=\max(R_i+1, x_i)}^{\min(R_i x_i, S_i)} a \Pr(|T_{x_i}| = a) \right). \tag{4.17}$$

Similarly, we define $\zeta_u \triangleq \min(X, y)$ and $\xi_u \triangleq \max(X, y)$. Then P_{x_u} denotes the expected fraction of total keys compromised for universal cluster, that is

$$P_{x_u} = \frac{1}{\mathcal{S}} \left(\frac{\mathcal{R} \binom{\xi_u}{1}\binom{n-\xi_u}{\zeta_u - 1}}{\binom{n}{\zeta_u}} + \sum_{x_u=2}^{\zeta_u} \frac{\binom{\xi_u}{x_u}\binom{n-\xi_u}{\zeta_u - x_u}}{\binom{n}{\zeta_u}} \sum_{a=\max(\mathcal{R}+1, x_u)}^{\min(\mathcal{R} x_u, \mathcal{S})} a \Pr(|T_{x_u}| = a) \right). \tag{4.18}$$

As we discussed in 3.3.1.1, any two nodes may produce a unique pair-wise key to form a secure link when they share $\acute{q} \geq q$ (or $\acute{\lambda} \geq \lambda$) keys. Therefore, this communication link becomes compromised when exactly $\acute{q}$ (or $\acute{\lambda}$) keys found among all the keys revealed from X captured nodes by adversary. Hence, when X nodes are captured, the probability that a link between any two nodes within i^{th} cluster have been compromised is

$$P_f^{(i)} = \frac{\sum_{j=q}^{R_i} \frac{\binom{R_i}{j}\binom{S_i-R_i}{R_i-j}}{\binom{S_i}{R_i}} (P_{x_i})^j}{\sum_{j=q}^{R_i} \frac{\binom{R_i}{j}\binom{S_i-R_i}{R_i-j}}{\binom{S_i}{R_i}}}. \tag{4.19}$$

Probability of link compromise for universal cluster $\mathcal{P}_f^{(u)}$ can be calculated by replacing x_i with x_u, $\acute{q}$ with $\acute{\lambda}$, and R_i with $\mathcal{R}$ in (4.19) accordingly. Consider, $\binom{n_i}{2}$ the total number of possible secure links for any two nodes for i^{th} cluster, and $y_i y_j$ total number of secure links for any two consular nodes from i^{th} and j^{th} cluster (where $i \neq j$), therefore, from (4.19), the probability of successful cracking of any secure link in the entire network by attacker when X nodes are captured is given by,

$$P_f = \frac{\sum_{i=1}^{\mathcal{C}} \binom{n_i}{2} P_f^{(i)} + \frac{1}{2} \sum_{i=1}^{\mathcal{C}} \sum_{j=1, j\neq i}^{\mathcal{C}} y_i y_j \mathcal{P}_f}{\sum_{i=1}^{\mathcal{C}} \binom{n_i}{2} + \frac{1}{2} \sum_{i=1}^{\mathcal{C}} \sum_{j=1, j\neq i}^{\mathcal{C}} y_i y_j} \tag{4.20}$$

We define the resiliency $R_{\mathcal{L}}^{(i)}$ for i^{th} cluster as, $R_{\mathcal{L}}^{(i)} = 1 - P_f^{(i)}$.

This probability represents the durability of a link in a cluster against node capture attack after x_i nodes were compromised out of n_i cluster nodes. In other words, it is the fraction of compromised links over the total number of links for a cluster when x_i nodes are captured. The resiliency for any two nodes (from the same cluster) in the entire network will be, $\mathcal{P}_r = 1 - P_f$.

4.1.4.2 Resiliency of BPI and PPI

Resilience represents the compliment fraction of connections that adversary can compromise as a result of recovering key materials from captured nodes. We go on our analysis by answering the following questions (a,b and c):

For a network including a single cluster, what is P_b, the probability that a polynomial is being compromised after X number of nodes are captured? When we assume a single cluster network for our analysis, similar to [34], for $n \leq t$, the probability $P_b = 0$. Here, we have considered more accurate value for P_b when $n > t$ than the study in [34]. Therefore, we drive,

$$P_b = \begin{cases} 0 & \text{if } X \leq t \\ 1 - \sum_{z=0}^{min(X,t)} \binom{X}{z} \left(\frac{R}{S}\right)^z \left(1 - \frac{R}{S}\right)^{X-z} & \text{Otherwise} \end{cases} \tag{4.21}$$

Substituting 1 for R and S in (4.21), results in similar security analysis in [35] which is,

$$P_b = \begin{cases} 0 & \text{if } X \leq t \\ 1 & \text{Otherwise} \end{cases} \tag{4.22}$$

What would be the probability of compromising a connection link P_f in a single cluster network?

$$\begin{aligned} P_f &= P(\text{A link is broken} \mid \text{There exists a link}) \\ &= \frac{P(\text{A link is broken } \& \text{ There exists a link})}{P(\text{There exists a link})} \end{aligned} \tag{4.23}$$

Thus,

$$P_f = \frac{\sum_{i=q}^{R} \frac{\binom{R}{i}\binom{S-R}{R-i}}{\binom{S}{R}} (P_b)^i}{\sum_{i=q}^{R} \frac{\binom{R}{i}\binom{S-R}{R_i}}{\binom{S}{R}}} \tag{4.24}$$

In order to crack a secure link, an attacker needs to extract the polynomials by achieving polynomial shares from captured nodes. Thus, the attacker must capture enough nodes to attain sufficient polynomial shares. The probability of compromising a secure link P_f is directly dependent on the probability of the discovering the polynomial P_b.

For the proposed paradigm (when we have multiple clusters), what is the $P_b^{(i)}$ for i^{th} cluster?

$$\begin{aligned} P_b^{(i)} &= P(f_i \text{ is broken } |X) \\ &= \sum_{x_i=0}^{X} P(f_i \text{ is broken } |x_i, X).P(x_i|X) \end{aligned} \tag{4.25}$$

It is clear that if $X \leq t_i \Rightarrow P_b^{(i)} = 0$. On the other hand, x_i could not be greater than n_i and X, so upper bound of x_i is $\min(n_i, X)$. So, we have:

$$P_b^{(i)} = \sum_{x_i=t_i}^{min(n_i,X)} P(f_i|x_i, X).P(x_i|X) \tag{4.26}$$

and since $x_i \geq t_i$, so $\min(x_i, t_i) = t_i$, from (4.21) we have,

$$\begin{aligned} P(f_i|x_i, X) &= P(f_i|x_i) \\ &= 1 - \sum_{z=0}^{t_i} \binom{x_i}{z} \left(\frac{R_i}{S_i}\right)^z \left(1 - \frac{R_i}{S_i}\right)^{x_i - z} \end{aligned} \tag{4.27}$$

On the other hand,

$$P(x_i|X) = \frac{\binom{\max(X,n_i)}{x_i}\binom{n-\max(X,n_i)}{\min(X,n_i)-x_i}}{\binom{n}{\min(X,n_i)}} \tag{4.28}$$

Let us define $M_i \triangleq \max(X, n_i)$ and $R_i \triangleq \min(X, n_i)$, by putting (4.27) and (4.28) into (4.26), we have $P_b^{(i)}$.

Then, similar to (4.24), we derive the probability of compromising a link in i^{th} cluster, by

$$P_f^{(i)} = \frac{\sum_{z=q}^{R_i} \frac{\binom{R_i}{z}\binom{S_i-R_i}{R_i-z}}{\binom{S_i}{R_i}} \left(P_b^{(i)}\right)^z}{\sum_{z=q}^{R_i} \frac{\binom{R_i}{z}\binom{S_i-R_i}{R_i-z}}{\binom{S_i}{R_i}}} \tag{4.29}$$

For universal cluster, similar to (4.26), (4.27) and (4.28), the probability of recovering one polynomial is given by

$$\begin{aligned} \mathcal{P}_b = & \sum_{\mathcal{X}=\mathcal{T}}^{\min(Y,X)} \frac{\binom{\alpha}{\mathcal{X}}\binom{n-\alpha}{\beta-\mathcal{X}}}{\binom{n}{\beta}} \\ & \times \left(1 - \sum_{z=0}^{\mathcal{T}} \binom{\mathcal{X}}{z} \left(\frac{\mathcal{R}}{\mathcal{S}}\right)^z \left(1 - \frac{\mathcal{R}}{\mathcal{S}}\right)^{(\mathcal{X}-z)}\right) \end{aligned} \tag{4.30}$$

where $\beta \triangleq \min(X, Y)$, $\alpha \triangleq \max(X, Y)$, we see

$$\mathcal{P}_f = \frac{\sum_{z=q}^{\mathcal{R}} \frac{\binom{\mathcal{R}}{z}\binom{\mathcal{S}-\mathcal{R}}{\mathcal{R}-z}}{\binom{\mathcal{S}}{\mathcal{R}}} (\mathcal{P}_b)^z}{\sum_{z=q}^{\mathcal{R}} \frac{\binom{\mathcal{R}}{z}\binom{\mathcal{S}-\mathcal{R}}{\mathcal{R}-z}}{\binom{\mathcal{S}}{\mathcal{R}}}} \tag{4.31}$$

And finally, probability of breaking a link in entire network is given by

$$P_f = \frac{\sum_{z=1}^{\mathcal{C}} \binom{n_z}{2} P_f^{(z)} + \frac{1}{2}\sum_{i=1}^{\mathcal{C}} \sum_{z=1, z\neq i}^{\mathcal{C}} y_i y_z \mathcal{P}_f}{\sum_{z=1}^{\mathcal{C}} \binom{n_z}{2} + \frac{1}{2}\sum_{i=1}^{\mathcal{C}} \sum_{z=1, z\neq i}^{\mathcal{C}} y_i y_z} \tag{4.32}$$

P_f in (4.32) represents the probability of a successful node capture attack when X nodes are captured. Therefore, the resiliency (robustness) of a network $\mathcal{P}_r$ is given by $\mathcal{P}_r = 1 - P_f$.

$P_f^{(i)}$ and $\mathcal{P}_f$ in (4.29) and (4.31), considered a more sophisticated key establishment than PPI where two nodes need to find q number of common polynomial shares in order to establish a pair-wise key. Therefore, to find P_f for PPI, $q = 1$ in (4.29) and (4.31).

In BPI, only one polynomial share is loaded into a sensor node memory. Thus, values for $R_i, S_i, \mathcal{R}, \mathcal{S}, q$ equals to 1 in (4.27), (4.29), (4.30), (4.31) , consequently,

$$P_f^{(i)} = P_b^{(i)} = \sum_{x_i=t_i}^{min(n_i,X)} \frac{\binom{\max(X,n_i)}{x_i}\binom{n-\max(X,n_i)}{\min(X,n_i)-x_i}}{\binom{n}{\min(X,n_i)}} \tag{4.33}$$

and,

$$\mathcal{P}_f = \mathcal{P}_b = \sum_{\mathcal{X}=\mathcal{T}}^{\min(Y,X)} \frac{\binom{\alpha}{\mathcal{X}}\binom{n-\alpha}{\beta-\mathcal{X}}}{\binom{n}{\beta}} \tag{4.34}$$

4.1.5 Scalability and the Maximum Network Size of BPI and PPI

A key establishment scheme must support admission of a large number of sensor nodes in the network without loss of security, efficiency and flexibility. Many works were done to address the scalability of a key establishment scheme in term of efficiency and security [4, 5, 6, 2, 8, 56]. In one hand, connectivity (local and global connectivity) is one essential efficiency metrics. On the other hand, resiliency is a security metric that defines the robustness of scheme against node capture attacks. Unfortunately, these two metrics are contrary to each other [66]. Specifically, desiring a global connectivity $P_{\mathcal{G}}$ (or $P_g^{(i)}$) only is satisfied with providing a sufficient local connectivity $P_l^{(i)}$ and therefore for a fixed size of R (or $\mathcal{R}$), S_i (or $\mathcal{S}$) should be minimized as much as possible. Minimizing S_i (or $\mathcal{S}$) decreases the $\mathcal{P}_r$. After all, when the number of nodes n is large in the network, $P_l^{(i)}$ could be smaller to satisfy an anticipated $P_{\mathcal{G}}$ (or $P_g^{(i)}$), consequently, smaller S_i (or $\mathcal{S}$) would be needed, and indeed the value of $\mathcal{P}_r$ is reduced. Scalability can be seen in different perspective, where node density or the scale size of network defines the scalability. In fact, when n is increased in the static area of network then the node density (or $\frac{n'_i}{n_i}$) is increased. Hence, $P_l^{(i)}$ is improved for the fixed S_i (or $\mathcal{S}$) and R (or $\mathcal{R}$). With a good intuition, when $\frac{n'_i}{n_i}$ is large therefore attacker may find a better chance to monitor more communication and capture more nodes in low scale node capture attacks.

Table 4.1: Maximum size of network.

Scheme	$\boldsymbol{n_{\max}}$	**Remark**
Blundo [35]	t	$\mathcal{C} = 1$
Liu and Ning [34]	$\frac{(t+1)S}{R}$	$\mathcal{C} = 1$
BPI	$\sum_{i=1}^{\mathcal{C}} t_i$	$Y \leq \mathcal{T}$
PPI	$\sum_{i=1}^{\mathcal{C}} (t_i + 1)\frac{S}{R}$	$Y \leq (\mathcal{T} + 1)\frac{\mathcal{S}}{\mathcal{R}}$

In [67], the authors defined a metric that consider both the resiliency and connectivity that is called Resilient-Connectivity (RC). Therefore, for our proposed paradigm we have,

$$RC = \frac{\sum_{i=1}^{\mathcal{C}} n_i RC^{(i)} + Y\mathcal{RC}}{\sum_{i=1}^{\mathcal{C}} n_i + Y} \tag{4.35}$$

where $RC^{(i)} = P_g^{(i)} \times (1 - P_f^{(i)})$ and $\mathcal{RC} = \mathcal{P}_g \times (1 - \mathcal{P}_f)$.

Some scenarios of interest is to obtain the maximum possible security ($\mathcal{P}_r = 1$). Let us denote $n_{\max}$ the maximum network size which satisfies $\mathcal{P}_r = 1$. Table Table 4.1 shows the $n_{\max}$ for discussed instantiations in contrast with their adopted schemes. It clearly can be seen PPI and BPI are highly scalable when ultimate security is demanded as opposed to [34] and [35] respectively.

4.2 Computational Results

In this section, we conduct experiments and plot our results to evaluate a security aspect of our developed scheme in term of resilience against node capture attack.

Suppose that the attacker randomly or selectively captures a node and tries to recover all the keys from the node (we assume all the keys were successfully discovered by attacker) and then tries to use these keys to intercept the secure communication between any two randomly chosen communicating nodes in the network. A successful attack is when the attacker manages to crack the intercepted data from un-captured nodes via discovered keys.

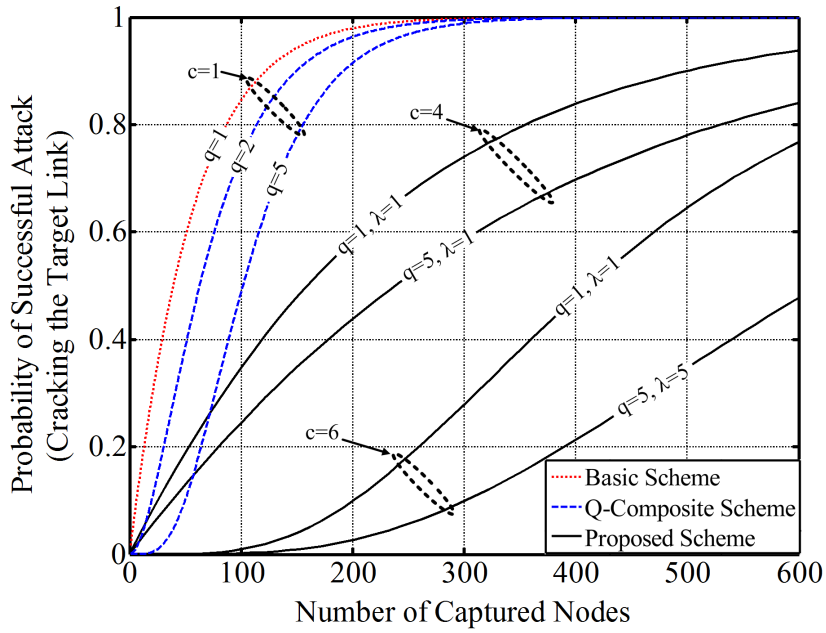


Figure 4.2: Comparison between basic, Q-Composite, and proposed schemes based on probability of successful attack when X nodes are captured with different $\mathcal{C}$, q and λ

Fig. Figure 4.2 demonstrates the probability of successful node capture attack versus the number of capture nodes for a network of $n = 600$, $Y = 60$ and $\mathcal{C} = 1, 4, 6$ (see Table Table 4.2). As it is shown in the figure, our scheme is more sustainable when there is a node capture attack as compared to [32, 33]. The reason is, in [32, 33] when an adversary tries to capture nodes from a region of the network, all nodes are associated with one key pool whereas in our scheme nodes may belong to different key pools. It can be observed that increasing the values of q and λ increases the resilience slightly. On the other hand, increasing the number of clusters lead to a better resilience. Fig. Figure 4.2 and a comparison of the local and global connectivity given in Table Table 4.2 show a tradeoff between the global connectivity and the resilience. For instance, when $\mathcal{C} = 6$ (we partition the network into 6 clusters), $q = 1$ and $\lambda = 1$ the resiliency is improved as compare to when $\mathcal{C} = 1$, and local and global connectivity level drop (local connectivity is decreased from 0.9999 to 0.092066). The

Table 4.2: System security parameters for Fig. Figure 4.2 and a comparison of local and global connectivity for each experiment

$\mathcal{C}$	R_i	$\mathcal{R}$	q	λ	$P_l^{(i)}$	$P_{\mathcal{G}}$	S_i	$\mathcal{S}$
1	20	N/A	1	N/A	0.33501	0.9999	1000	N/A
1	20	N/A	2	N/A	0.058219	0.9999	1000	N/A
1	20	N/A	5	N/A	2.4087×10^{-5}	1.4317×10^{-257}	1000	N/A
4	20	10	1	1	0.083753	0.99762	1000	400
4	20	10	5	1	6.0216×10^{-6}	4.539×10^{-261}	1000	400
6	20	10	1	1	0.055835	0.092066	1000	400
6	20	10	5	5	4.0144×10^{-6}	2.9532×10^{-287}	1000	400

reason is, $\acute{n}_i$ is decreased for each cluster when the number of cluster is increased.

It is possible that a portion of key pool never be assigned to any current node in the network. It is likely when the size of key pool is relatively large in compare to the number of nodes in the network. In this situation, even if all the nodes are captured, it is not guaranteed that all the keys (key pool) have become available to an adversary. Consequently, future deployments of nodes might be able to establish secure links using unassigned keys from key pool. Hence, for $\mathcal{C} = 4, 6$ when $X = n_i$, the probability of successful attack is not equal to 1.

In some scenarios a certain level of global connectivity is desired, therefore a certain level of required local connectivity should be satisfied as it is discussed in Section 4.1.2. Fig. Figure 4.3 compares the probability of successful node capture attack under different scenarios that the related system parameters can be found at Table Table 4.3. For example, when we increase number of nodes from $n = 600$ to $n = 2400$ we need to increase the key pool from 1000 to 4000 to increase the resiliency. Unfortunately, increasing the key pool may result in lower connectivity. More specifically, moving from (a-1) to (a-2) the local connectivity will drop significantly, if we use [32]. Instead, our scheme increases the resiliency while maintaining the similar level of local connectivity, (in Fig. Figure 4.3 compare (a-2) with (d-1)). Additionally, using our scheme makes it possible to have almost

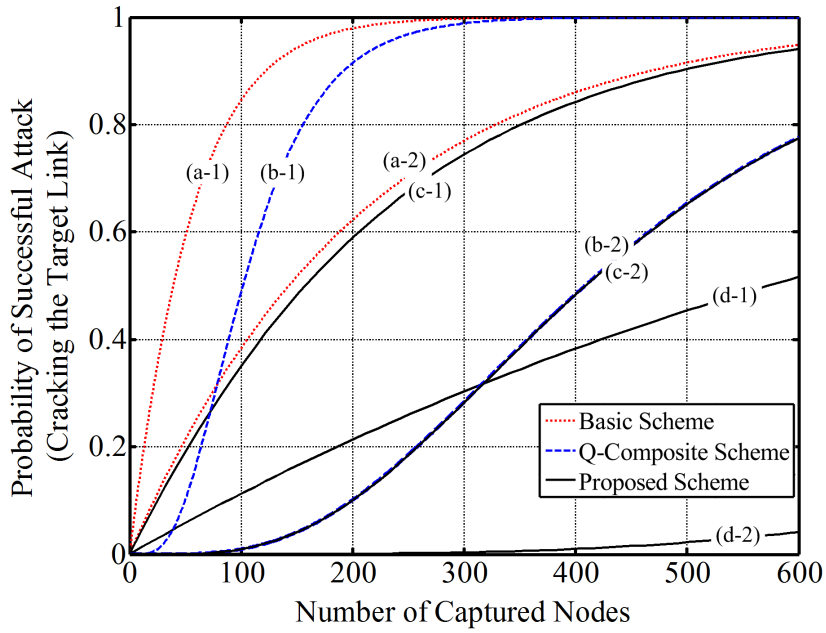


Figure 4.3: The probability of successful attack against number of captured nodes X for different configurations (see Table Table 4.3 for system parameters used)

similar (or even slightly better) resilience for (a-2) with smaller key pool per cluster. In such situation, the local connectivity for each cluster is enhanced (in Fig. Figure 4.3 and Table Table 4.3 compare (a-2) with (c-1)).

The type and requirement of an assignment require a guaranteed global connectivity. To achieve a desire global connectivity, a local connectivity should be provided accordingly.

Table 4.3: System security parameters for Fig. Figure 4.3 and a comparison of local and global connectivity for each experiment ($n = n_i \times \mathcal{C}$ and $R_i = 20$ for all the experiments)

Label	n	$\mathcal{C}$	$\mathcal{R}$	q	λ	$P_l^{(i)}$	$P_{\mathcal{G}}$	S_i	$\mathcal{S}$
(a-1)	600	1	N/A	1	N/A	0.33501	0.9999	1000	N/A
(a-2)	2400	1	N/A	1	N/A	0.095606	0.9999	4000	N/A
(b-1)	600	1	N/A	5	N/A	2.4087×10^{-5}	1.4317×10^{-257}	1000	N/A
(b-2)	2400	1	N/A	5	N/A	2.6942×10^{-8}	1.2325×10^{-260}	4000	N/A
(c-1)	2400	4	10	1	1	0.33501	0.9999	1000	400
(c-2)	2400	4	10	5	1	2.4087×10^{-5}	1.4317×10^{-257}	1000	400
(d-1)	2400	4	10	1	1	0.095606	0.9999	4000	400
(d-2)	2400	4	10	5	1	2.6942×10^{-8}	1.2325×10^{-260}	4000	400

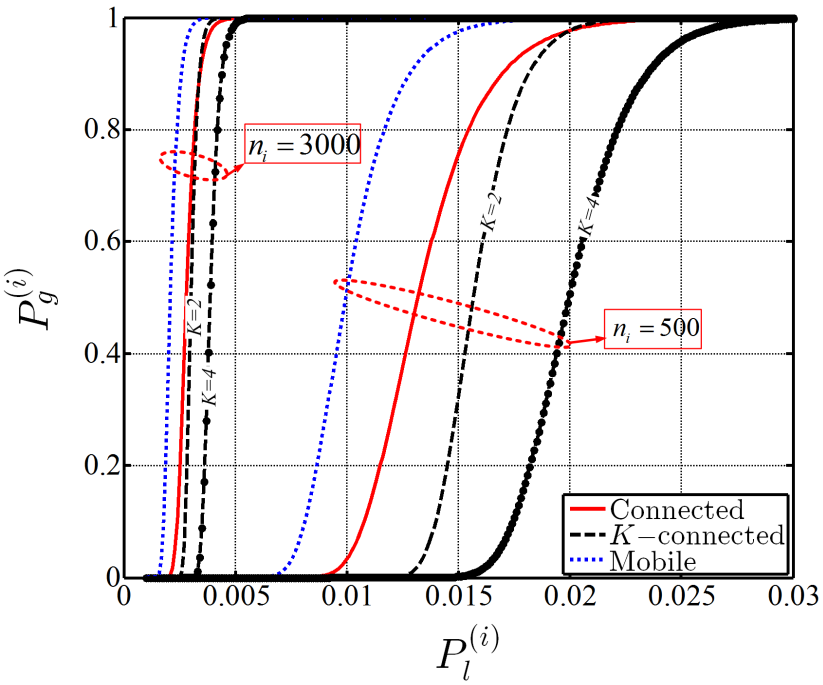


Figure 4.4: Correlation of desire global connectivity $P_g^{(i)}$ and required local connectivity $P_l^{(i)}$ based on different connectivity model, for a network with n_i nodes

Fig. Figure 4.4 illustrates the desire global connectivity $P_g^{(i)}$ versus required local connectivity $P_l^{(i)}$ for cluster sizes of $n_i = 500$ and 3000 respectively, under three connectivity models of connected graph, $K-$connected graph and mobile. As it is inferred from graphs in Fig. Figure 4.4, when n_i increases the chance of having more $P_l^{(i)}$ is better. $K-$connected graph requires higher local connectivity as opposed to a connected graph whereas in mobile model, less local connectivity is needed to satisfy a defined global connectivity. The reason is, when in mobile network an isolated node needs to communicate, it must wait until it moves or takes place into another node's radio range where it comes with a considerable communication delays. On the other hand, a high needed local connectivity $P_l^{(i)}$ may result in a resistance against movement of a node. Therefore an optimum value for local connectivity should be considered [62] which is less than $P_l^{(i)}$ for both the connected and $K-$connected graph models. Works in [68, 69] studied the optimum value for $P_l^{(i)}$ based on different assumptions.

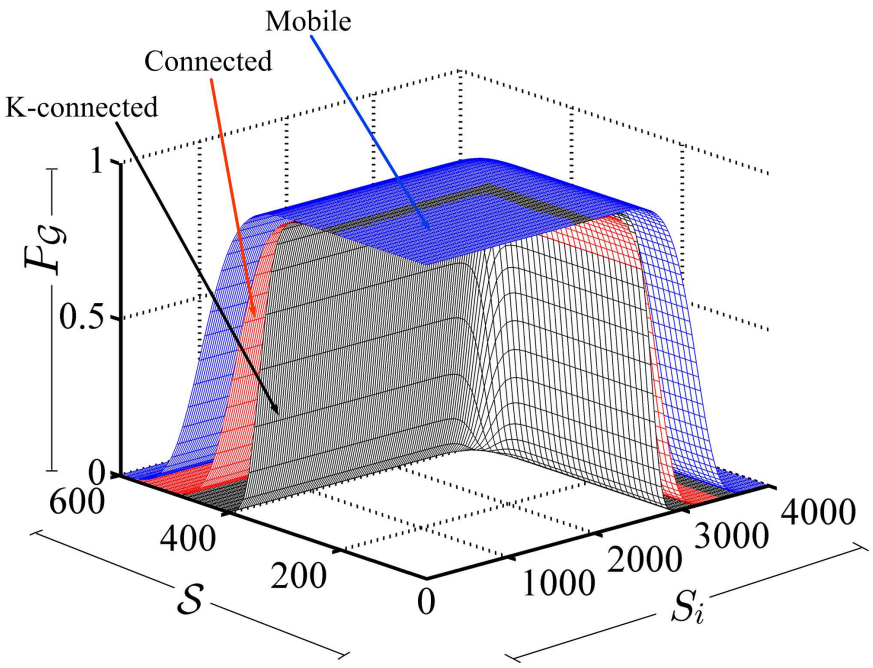
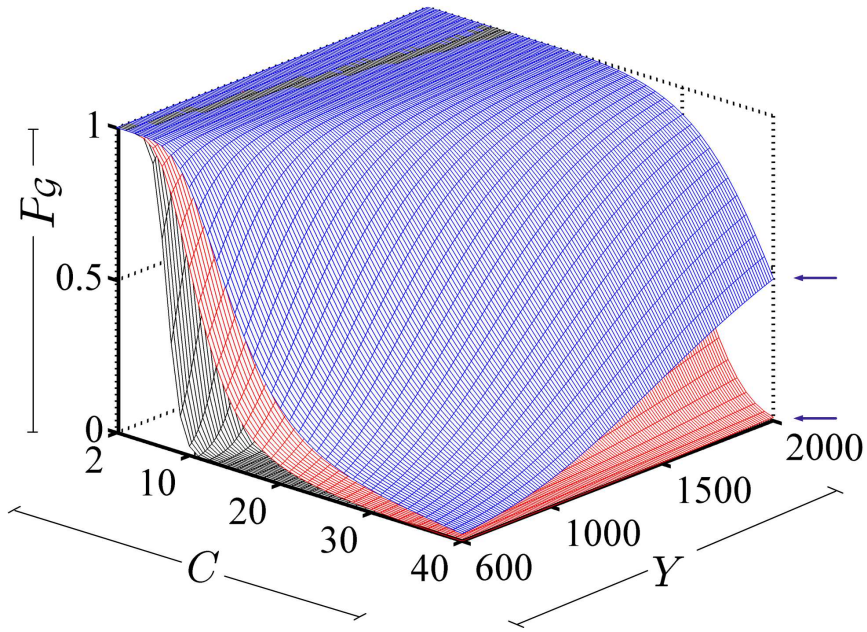


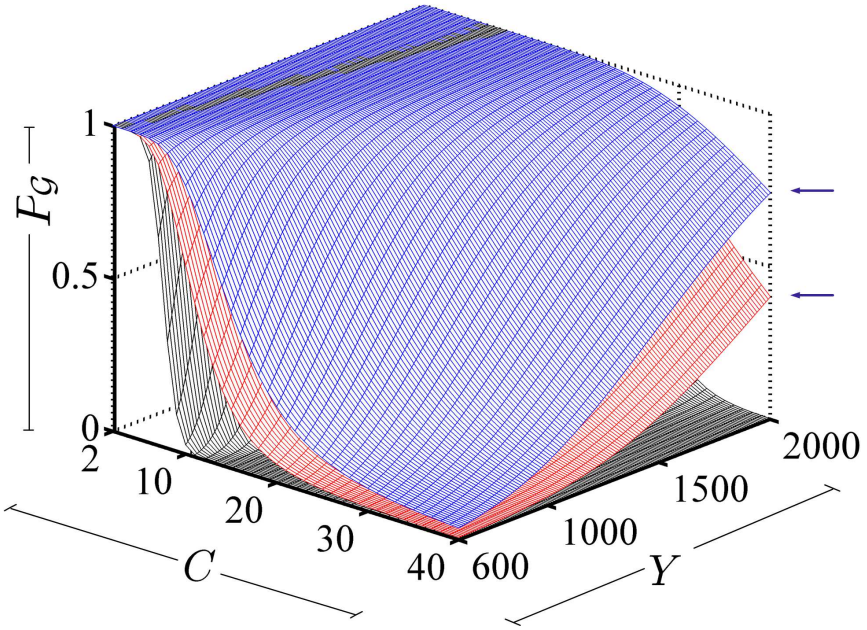
Figure 4.5: Impact of $|S_i|$ and $|\mathcal{S}|$ on network global connectivity $P_{\mathcal{G}}$, for $n_i = 1000$, $y_i = 50$, $\mathcal{C} = 5$, $R_i = 20$, $\mathcal{R} = 10$ and $K = 2$ (for $K-$connected)

One can see that, the maximum $|R_i|$ (or $|\mathcal{R}|$) is restricted by the hardware platform of a sensor node. Despite the fact that $|S_i|$ and $|\mathcal{S}|$ are not limited by the hardware constraints, the satisfactory value should be set for them to deliver an adequate level of connectivity. Fig. Figure 4.5 makes evident $P_{\mathcal{G}}$ versus $|S_i|$ and $|\mathcal{S}|$ for different connectivity models. $P_{\mathcal{G}}$ will be decreased if a large $|S_i|$ and $|\mathcal{S}|$ is chosen. From Fig. Figure 4.5, first, the impact of $|\mathcal{S}|$ is more than $|S_i|$, due to the inter-connectivity that is provided by consular nodes. In other words, a large $|\mathcal{S}|$ reduces the $\mathcal{P}_g$ and consequently $P_{\mathcal{G}}$ will be decreased. Second, mobile connectivity level is higher, and less sensitive to $|S_i|$ and $|\mathcal{S}|$ changes than both the other connectivity models.

Fig. Figure 4.6 demonstrates the changes of $P_{\mathcal{G}}$ under various connectivity scenarios as the function of $\mathcal{C}$ and Y. As it is seen $P_{\mathcal{G}}$ is declined when the number of clusters increases; and is enhanced when Y is increasing. A Comparison between graphs in Fig. Figure 4.6



(a) $n = 5000$



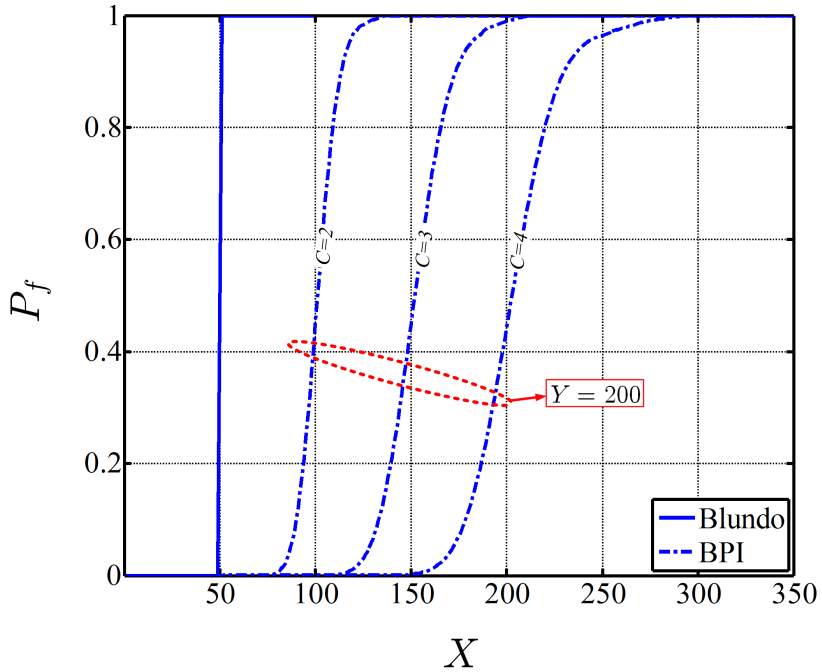
(b) $n = 10000$

Figure 4.6: Effects of increasing number of cluster $\mathcal{C}$ and consular node Y on the $P_{\mathcal{G}}$

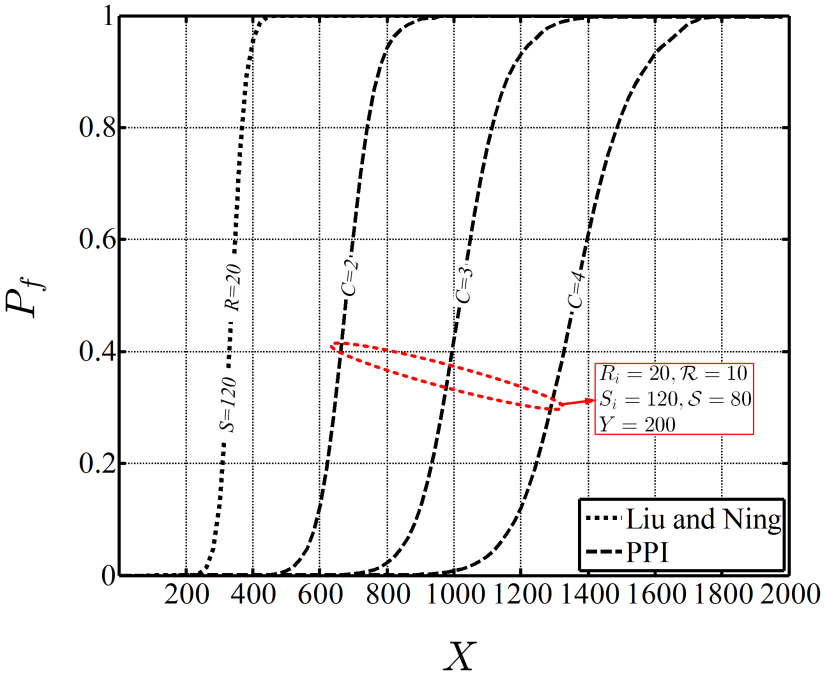
shows the increase of n, takes less from reduction in network connectivity level.

Fig. Figure 4.7 shows how the proposed paradigm can improve the resiliency of a network against node capture attacks. P_f decreases significantly with the increase of $\mathcal{C}$, while there is only 200 consular nodes are employed. Increasing the consular nodes does not improve security, as we tried the same experiment with more consular nodes ranging from 300 to 400. Having consular nodes imposes a nominal storage overhead. Which is more acceptable than the computation overheads as a consequences of increasing t especially at [35]. Fig. Figure 4.7(a) verifies that security of [35] can be improved by suffering a bit of memory overhead (while keeping connectivity for almost 100%). Similar inference is applied to Liu and Ning scheme [34] after employing the proposed paradigm, as it is seen at Fig. Figure 4.7(b). Additionally, comparing graphs at Fig. Figure 4.7, PPI is far securer than BPI. This comes from security improvement characteristic of [34].

In order to increase $N_{\max}$, involving security parameters in Table Table 4.1 can be modified accordingly. $N_{\max}$ for [35] cannot be increased unusually when it is only reliant on t. t (similarly $\mathcal{T}$) cannot be a very large number as it imposes an unwanted computation overhead for a sensor node. Also, increasing of $N_{\max}$ in [34] is limited by increasing S and t, and decreasing R. Besides, Values for S and R (or R_i, $\mathcal{R}$, S_i and $\mathcal{R}$) cannot be any arbitrary values as they can aggravate the $P_{\mathcal{G}}$. For instance, an observation on Fig. Figure 4.5 says that S_i and S should be carefully chosen to avoid any degradation for $P_{\mathcal{G}}$. To solve the above issue, our proposed paradigm makes it possible to increase $N_{\max}$ by increasing value of $\mathcal{C}$ and without need of modifying other security parameters.



(a)



(b)

Figure 4.7: Effects of increasing number of cluster $\mathcal{C}$ on the probability of P_f after X node captures, when $n = 2000$

Chapter 5

CONCLUSION

In this work, we have proposed a clustering paradigm to be used in any pre-distribution schemes for homogeneous wireless sensor networks. We developed three key pre-distribution schemes based on this paradigm. The resulting schemes isolate the effect of node compromise into one specific cluster in which resiliency of the entire network significantly improved against node capture attack. Also, our schemes provide scalability for both node and cluster addition. Our design enables inter-cluster communication (under a specific assumption) by applying universal cluster that is composed of randomly chosen consular nodes from each cluster, and that comes with nominal memory overhead. In this work, we draw an inclusive security analysis that can lead to better understanding about previous works. Our computational results prove a considerable security improvement as opposed to similar prior schemes.

Bibliography

[1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer networks*, 2008.

[2] Y. Wang, G. Attebury, and B. Ramamurthy, “A survey of security issues in wireless sensor networks,” *Communications Surveys Tutorials, IEEE*, 2006.

[3] X. Chen, K. Makki, K. Yen, and N. Pissinou, “Sensor network security: A survey,” *Communications Surveys & Tutorials, IEEE*, 2009.

[4] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, “A survey of key management schemes in wireless sensor networks,” *Comput. Commun.*, 2007.

[5] J. Zhang and V. Varadharajan, “Review: Wireless sensor network key management survey and taxonomy,” *J. Netw. Comput. Appl.*, vol. 33, 2010.

[6] S. A. Camtepe and B. Yener, “Key distribution mechanisms for wireless sensor networks: a survey,” Tech. Rep., 2005.

[7] D. D. Hwang, B. cheng Charles Lai, and I. Verbauwhede, “Energy-memory-security tradeoffs in distributed sensor networks,” in *In ADHOC-NOW*, 2004, pp. 70–81.

[8] M. A. Simplício, Jr., P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, “A survey on key management mechanisms for distributed wireless sensor networks,” *Comput. Netw.*, 2010.

[9] A. Becher, Z. Benenson, and M. Dornseif, “Tampering with motes: Real-world physical attacks on wireless sensor networks,” *Security in Pervasive Computing*, pp. 104–118, 2006.

[10] K. Sharma and M. Ghose, “Wireless sensor networks: An overview on its security threats,” *International Journal of Computer Applications IJCA*, no. 1, pp. 42–45, 2010.

[11] D. Raymond and S. Midkiff, “Denial-of-service in wireless sensor networks: Attacks and defenses,” *Pervasive Computing, IEEE*, vol. 7, no. 1, pp. 74–81, 2008.

[12] R. Muraleedharan and L. Osadciw, “Jamming attack detection and countermeasures in wireless sensor network using ant system,” *Wireless Sensing and Processing, proceedings of the SPIE*, vol. 6248, p. 62480G, 2006.

[13] M. Brownfield, Y. Gupta, and N. Davis, “Wireless sensor network denial of sleep attack,” in *Information Assurance Workshop, 2005. IAW’05. Proceedings from the Sixth Annual IEEE SMC*. IEEE, 2005, pp. 356–364.

[14] W. Yadong, W. Qin, Z. Xiaotong, X. Jinwu, G. Lei, L. Lei, and W. Ruofei, “A hybrid tdm-fdm mac protocol for wireless sensor network using timestamp self-adjusting synchronization mechanism,” in *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*. IEEE, 2007, pp. 2705–2709.

[15] J. Deng, R. Han, and S. Mishra, “Insens: Intrusion-tolerant routing for wireless sensor networks,” *Computer Communications*, vol. 29, no. 2, pp. 216–230, 2006.

[16] M. Tiwari, K. Arya, R. Choudhari, and K. Choudhary, “Designing intrusion detection to detect black hole and selective forwarding attack in wsn based on local information,” in *Computer Sciences and Convergence Information Technology, 2009. ICCIT’09. Fourth International Conference on*. IEEE, 2009, pp. 824–828.

[17] Z. Karakehayov, “Using reward to detect team black-hole attacks in wireless sensor networks,” *Wksp. Real-World Wireless Sensor Networks*, pp. 20–21, 2005.

[18] M. Demirbas and Y. Song, “An rssi-based scheme for sybil attack detection in wireless sensor networks,” in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*. IEEE Computer Society, 2006, pp. 564–570.

[19] W. Pires Jr, T. de Paula Figueiredo, H. Wong, and A. Loureiro, “Malicious node detection in wireless sensor networks,” in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. IEEE, 2004, p. 24.

[20] H. Kalita and A. Kar, “Wireless sensor network security analysis,” *International Journal of Next-Generation Networks (IJNGN)*, vol. 1, no. 1, pp. 1–10, 2009.

[21] S. Hong and S. Lim, “Analysis of attack models via unified modeling language in wireless sensor networks: A survey study,” in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*. IEEE, 2010, pp. 692–696.

[22] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*. IEEE, 2005, pp. 324–328.

[23] K. Piotrowski, P. Langendoerfer, and S. Peter, “How public key cryptography influences wireless sensor node lifetime,” in *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*. ACM, 2006, pp. 169–176.

[24] G. De Meulenaer, F. Gosset, F. Standaert, and O. Pereira, “On the energy cost of communication and cryptography in wireless sensor networks,” in *Networking and Communications, 2008. WIMOB’08. IEEE International Conference on Wireless and Mobile Computing,*. IEEE, 2008, pp. 580–585.

[25] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, no. 1, 1983.

[26] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[27] X. Lai, J. Massey, and S. Murphy, “Markov ciphers and differential cryptanalysis,” in *Advances in Cryptology EUROCRYPT 91*. Springer, 1991, pp. 17–38.

[28] N. Courtois and J. Pieprzyk, “Cryptanalysis of block ciphers with overdefined systems of equations,” *Advances in Cryptology ASIACRYPT 2002*, pp. 267–287, 2002.

[29] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” in *Advances in Cryptology*. Springer, 1985, pp. 10–18.

[30] R. Rivest, “The md5 message-digest algorithm,” 1992.

[31] V. Rijmen and E. Oswald, “Update on sha-1,” *Topics in Cryptology–CT-RSA 2005*, pp. 58–71, 2005.

[32] 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*, ser. CCS ’02. ACM, 2002.

[33] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, 2003.

[34] D. Liu, P. Ning, and R. Li, “Establishing pairwise keys in distributed sensor networks,” *ACM Trans. Inf. Syst. Secur.*, vol. 8, 2005.

[35] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly-secure key distribution for dynamic conferences,” 1995.

[36] R. Blom, “An optimal class of symmetric key generation systems,” in *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*. Springer-Verlag New York, Inc., 1985.

[37] D. Liu and P. Ning, “Improving key predistribution with deployment knowledge in static sensor networks,” *ACM Trans. Sen. Netw.*, 2005.

[38] F. Delgosha and F. Fekri, “Key pre-distribution in wireless sensor networks using multivariate polynomials,” in *Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on*, 2005.

[39] ——, “A multivariate key-establishment scheme for wireless sensor networks,” *Wireless Communications, IEEE Transactions on*, 2009.

[40] N. T. Canh, P. T. H. Truc, T. H. Hai, L. X. Hung, Y.-K. Lee, and S. Lee, “Enhanced group-based key management scheme for wireless sensor networks using deployment knowledge,” in *Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*, ser. CCNC’09. IEEE Press, 2009.

[41] S. Lee, M. S. Jeong, H. Jeong, H. J. Lee, and E. Y. Choi, “A practical group based key management scheme for ubiquitous sensor networks,” *Information Networking, International Conference on*, 2012.

[42] K. Lu, Y. Qian, M. Guizani, and H.-H. Chen, “A framework for a distributed key management scheme in heterogeneous wireless sensor networks,” *Trans. Wireless. Comm.*, 2008.

[43] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks,” *ACM Trans. Inf. Syst. Secur.*, vol. 8, 2005.

[44] W. Zhang, M. Tran, S. Zhu, and G. Cao, “A random perturbation-based scheme for pairwise key establishment in sensor networks,” in *MobiHoc*, 2007.

[45] D. Liu, P. Ning, and W. Du, “Group-based key predistribution for wireless sensor networks,” *ACM Trans. Sen. Netw.*, 2008.

[46] Y. Zhou, Y. Zhang, and Y. Fang, “Llk: a link-layer key establishment scheme for wireless sensor networks,” in *Wireless Communications and Networking Conference, 2005 IEEE*, 2005.

[47] W. Bechkit, Y. Challal, A. Bouabdallah *et al.*, “A new scalable key pre-distribution scheme for wsn,” in *International Conference on Computer Communication Networks*, 2012.

[48] W. Zhang, N. Subramanian, and G. Wang, “Lightweight and compromise-resilient message authentication in sensor networks,” in *INFOCOM*, 2008, pp. 1418–1426.

[49] N. Subramanian, C. Yang, and W. Zhang, “Securing distributed data storage and retrieval in sensor networks,” *Pervasive Mob. Comput.*, 2007.

[50] M. Albrecht, C. Gentry, S. Halevi, and J. Katz, “Attacking cryptographic schemes based on "perturbation polynomials",” Cryptology ePrint Archive, Report 2009/098, 2009.

[51] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, “A key management scheme for wireless sensor networks using deployment knowledge,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2004.

[52] S.-P. Chan, R. Poovendran, and M.-T. Sun, “A key management scheme in distributed sensor networks using attack probabilities,” in *Global Telecommunications Conference, 2005. GLOBECOM ’05. IEEE*, 2005.

[53] Z. Yu and Y. Guan, “A key management scheme using deployment knowledge for wireless sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, 2008.

[54] N. T. Canh, Y.-K. Lee, and S. Lee, “Hgkm: A group-based key management scheme for sensor networks using deployment knowledge,” in *Proceedings of the Communication Networks and Services Research Conference*, ser. CNSR ’08. IEEE Computer Society, 2008.

[55] K. M. Martin, M. B. Paterson, and D. R. Stinson, “Key predistribution for homogeneous wireless sensor networks with group deployment of nodes,” *ACM Trans. Sen. Netw.*, 2010.

[56] E. Klaoudatou, E. Konstantinou, G. Kambourakis, and S. Gritzalis, “A survey on cluster-based group key agreement protocols for wsns,” *Communications Surveys Tutorials, IEEE*, 2011.

[57] M. Younis, K. Ghumman, and M. Eltoweissy, “Location-aware combinatorial key management scheme for clustered sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, 2006.

[58] M. Eltoweissy, M. Moharrum, and R. Mukkamala, “Dynamic key management in sensor networks,” *Communications Magazine, IEEE*, 2006.

[59] J. Zhang and V. Varadharajan, “A new security scheme for wireless sensor networks,” in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008.

[60] H. Ling and T. Znati, “End-to-end pairwise key establishment using multi-path in wireless sensor network,” in *Global Telecommunications Conference, 2005. GLOBECOM ’05. IEEE*, 2005.

[61] P. Erdös and A. Renyi, “On random graphs,” in *Publication of the mathematical institute of the Hungarian academy of sciences*, 1959, pp. 290–297.

[62] C. Bettstetter, “On the connectivity of ad hoc networks,” *The Computer Journal*, 2004.

[63] M. Abramowitz and I. A. Stegun, “Handbook of mathematical functions: With formulas, graphs, and mathematical tables,” Dover Publications, 1965.

[64] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth, “On the lambert w function,” 1996.

[65] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*. Kluwer Academic Publishers, 1996.

[66] K. Chan and F. Fekri, “A resiliency-connectivity metric in wireless sensor networks with key predistribution schemes and node compromise attacks,” *Physical Communication*, 2008.

[67] W. Gu, S. Chellappan, X. Bai, and H. Wang, “Scaling laws of key predistribution protocols in wireless sensor networks,” *IEEE Transactions on Information Forensics and Security*, 2011.

[68] M. Zavlanos, M. Egerstedt, and G. Pappas, “Graph-theoretic connectivity control of mobile robot networks,” *Proceedings of the IEEE*, 2011.

[69] A. Friend, V. Manshadi, and A. Saberi, “Distributed node placement algorithms for constructing well-connected sensor networks,” in *INFOCOM, 2012 Proceedings IEEE*, 2012.

Rezaeirad, Mohammad. Bachelor of Information Technology, Multimedia University, Spring, 2010; Master of Science, University of Louisiana at Lafayette, Fall 2012
Major: Computer Engineering
Title of Thesis: A Novel Clustering Paradigm for Pre-distribution Key Management for Mobile Homogenous Wireless Sensor Networks
Thesis Director: Dr. Magdy A. Bayoumi
Pages in Thesis: 75; Words in Abstract: 156

ABSTRACT

Mission critical applications on static and mobile homogenous wireless sensor networks (HWSNs) mandate new sets of security appliances to be friendly with existing resource constrained hardware platforms. To deliver a promising security, particularly in military deployments, mechanisms have to build upon an efficient key management that compensates HWSNs constraints. Cluster-based key establishment is being the prime focus among the recent works in key establishment due to its significant improvement on network efficiency, security, scalability and flexibility. Therefore, we propose a clustering paradigm to support pre-distribution key establishment schemes for HWSNs. The proposed paradigm is compatible with most of pre-distribution schemes, and three instantiations (developed schemes based on the proposed paradigm) are provided in this work to support our claim that the proposed paradigm improves security and scalability of the adopted schemes. We develop analytical models and conduct extensive simulations to evaluate the security and performance of the proposed framework, and the network connectivity under different scenarios.

Biographical Sketch

Mohammad Rezaeirad received his Bachelor of Information Technology (Honours) in 2010 in Security Technology from the Multimedia University in Melaka, Malaysia. After working in industry and relocating to Kuala Lumpur, new opportunities became available and, Mohammad Rezaeirad relocated to Louisiana. Mohammad Rezaeirad completed the requirements for this Master of Science in Computer Engineering from the University of Louisiana at Lafayette in Fall 2012.