DYNAMIC KEYRING UPDATE MECHANISM FOR MOBILE WIRELESS SENSOR NETWORKS

by

MERVE ŞAHİN

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
August 2013

DYNAMIC KEYRING UPDATE MECHANISM FOR MOBILE WIRELESS SENSOR NETWORKS

APPROVED BY

Assoc. Prof. Dr. Albert Levi ....

(Thesis Supervisor)

Assoc. Prof. Dr. Yücel Saygın ....

Assoc. Prof. Dr. Selim Balcısoy ....

Assoc. Prof. Dr. Cem Güneri ....

Asst. Prof. Dr. Hakan Erdoğan ....

DATE OF APPROVAL 13.08.2013

DYNAMIC KEYRING UPDATE MECHANISM FOR MOBILE WIRELESS SENSOR NETWORKS

Merve Şahin
Computer Science and Engineering, MS Thesis, 2013
Thesis Supervisor: Assoc. Prof. Albert Levi

Keywords: Mobile Wireless Sensor Networks, Security, Keyring Update, Connectivity

Abstract

Wireless Sensor Networks (WSNs) are composed of small, battery-powered devices called sensor nodes. Sensor nodes have sensing, processing and communication capabilities to monitor the environment and gather data. WSNs have various application areas ranging from military surveillance to forest fire detection. Security is an important issue for Wireless Sensor Networks because sensor nodes are deployed in hostile and unattended areas. Nodes are vulnerable to physical capture attacks and the attackers can easily eavesdrop on network communications.

To provide security to WSNs, many key predistribution schemes have been proposed. However, most of these schemes consider the static WSNs and they perform poorly when they are applied to Mobile Wireless Sensor Networks (MWSNs). In this thesis, we propose Dynamic Keyring Update (DKRU) mechanism for MWSNs. The aim of DKRU mechanism is to enable sensor nodes to update their keyrings periodically during movement, by observing the frequent keys in their neighbors. Our mechanism can be used together with different key predistribution schemes and it helps to increase the performance of them.

For performance evaluation reasons, we used our mechanism together with an existing random key predistribution scheme and a location-based key predistribution scheme. For each of these key predistribution schemes, we analyzed our mechanism using two different mobility models. Our results show that DKRU mechanism increases the local and global connectivity when it is applied to MWSNs. Moreover, our mechanism is scalable and it does not cause significant degradation in network resiliency and communication overhead.

MOBİL TELSİZ DUYARGA AĞLARI İÇİN DİNAMİK ANAHTAR HALKASI GÜNCELLEME MEKANİZMASI

Merve Şahin
Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2013
Tez Danışmanı: Doç. Dr. Albert Levi

Anahtar Kelimeler: Mobil Telsiz Duyarga Ağları, Güvenlik, Anahtar Halkası Güncelleme, Bağlanırlık

Özet

Telsiz Duyarga Ağları (TDA), duyarga düğümleri olarak adlandırılan küçük ve pil gücü ile çalışan cihazlardan oluşur. Duyarga düğümleri, algılama, veri işleme ve iletişim yeteneklerini kullanarak çevreyi gözlemler ve veri toplarlar. TDA'ların, askeri taramadan orman yangını tespitine kadar çok çeşitli uygulama alanları bulunmaktadır. Bu uygulamalarda duyarga düğümleri genel olarak gözetimsiz ve kontrolden uzak alanlara bırakılırlar. Bu sebeple, düğümler fiziksel anlamda ele geçirilmeye müsaittirler. Ayrıca ağdaki bağlantılar bir saldırgan tarafından kolaylıkla dinlenebilir. Bu yüzden, TDA'larda ağ güvenliğini sağlamak önemli bir sorun haline gelmiştir.

TDA'larda güvenlik sorununu çözmek için bir çok ön yüklemeli anahtar dağıtım şeması önerilmiştir. Fakat, bu şemaların çoğu duyarga düğümlerinin durağan olduğunu varsayar ve Mobil Telsiz Duyarga Ağlarına (MTDA) uygulandıklarında yetersiz kalırlar. Bu tezde, MTDA'lar için Dinamik Anahtar Halkası Güncelleme (DAHG) mekanizması sunulmaktadır. Bu mekanizmanın amacı, duyarga düğümlerinin hareketleri sırasında komşularında sıklıkla bulunan anahtarları gözlemleyerek, kendi anahtar halkalarını periyodik olarak güncellemeleridir. Mekanizmamız farklı ön yüklemeli anahtar dağıtım şemaları ile birlikte kullanılabilir ve bu şemaların performansının arttırılmasına yardımcı olur.

Performans değerlendirmelerinde mekanizmamız, bir rastgele ön yüklemeli anahtar dağıtım şeması, bir de konuma dayalı ön yüklemeli anahtar dağıtım şeması olmak üzere iki farklı şemayı temel alacak şekilde kullanılmıştır. Her iki ön yüklemeli anahtar dağıtım şeması için ayrıca iki farklı mobilite modeli ile analizler yapılmıştır. Değerlendirme sonuçlarımız, DAHG mekanizmasının her durumda ağdaki yerel ve genel bağlantı oranlarını arttırdığını göstermiştir. Ayrıca mekanizmamız ölçeklendirilebilir olup, ağ dayanıklılığına zarar vermez ve düşük bir ek iletişim maliyeti gerektirir.

Acknowledgements

I would like to thank my thesis supervisor, Assoc. Prof. Albert Levi, for all his support throughout my undergraduate and graduate education and for guiding me in my studies.

I also thank Assoc. Prof. Yücel Saygın, Assoc. Prof. Selim Balcısoy, Assoc. Prof. Cem Güneri and Asst. Prof. Hakan Erdoğan for devoting their time to join my jury despite their busy schedule.

I specially thank Onur Çatakoğlu and Salim Sarımurat for their great support and help during my graduate studies. I also thank all of my classmates at FENS 2001 Lab for their valuable advices and support in many aspects of my life. I thank my dearest family for their constant support and love, for being there whenever I need them.

I also thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for funding me with BİDEB scholarship through my graduate education and supporting this research under grant 110E180.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

Wireless Sensor Networks (WSNs), consisting of small, autonomous devices called sensor nodes, have increasing range of application areas such as military surveillance, environmental tracking or hazard detection, patient monitoring and smart home applications [1]. All these applications convey sensitive data, so they require a secure communication medium among the sensor nodes and the base station (sink node), where the data is collected. However, sensor nodes have many limitations that make it complicated to develop security protocols for WSNs. Sensor nodes are battery-powered, memory-constrained and they have limited computation and transmission power. Moreover, they are vulnerable to physical capture attacks because making the sensor nodes tamper-proof is too costly [2]. Due to these limitations of sensor nodes, using asymmetric cryptography is not a feasible solution to provide security for WSNs. Using symmetric cryptography with a single network-wide key or using pairwise shared keys are also not applicable solutions considering the physical attack threats, memory limitations and scalability issues [2].

A promising solution on key distribution, which is suitable for most of the requirements and limitations of WSNs, is proposed by Eschenauer and Gligor [3] in 2002. In their scheme, a set of keys are randomly distributed to sensor nodes from a key pool before deployment, so that two nodes can communicate with each other if they share any common keys. This scheme is also referred as the *basic scheme*. There are many studies in the literature which are based on the notion of the predistribution of keying material. These studies include the matrix based, polynomial based, combinatorial design based and location based approaches [4]. All these studies assume the sensor nodes to be static, which means that their location does not change after initial deployment.

The concept of Mobile Wireless Sensor Networks (MWSNs) emerged later than the static WSNs. In MWSNs, sensor nodes and/or other entities in sensor network are mobile, which means that the topology of network dynamically changes. MWSNs has many advantages over static WSNs, however most of the efficient security protocols proposed for static WSNs perform poorly in MWSNs [5].

1.1. Our Motivation and Contribution of the Thesis

Most of the solutions on the key distribution problem in WSNs assume that the sensor nodes are static. However, many application areas of WSNs require the sensor nodes to be mobile, such as battlefield surveillance, vehicle tracking, animal tracking, etc. Our initial analyses show that existing schemes perform poorly in Mobile Wireless Sensor Networks (MWSNs). The random keyring based solutions require high keyring size to achieve an acceptable level of connectivity, which results in an increase in memory overhead and decrease in resiliency. The location based solutions can provide better connectivity for a short time after deployment, however because the sensor nodes are mobile, the initial deployment knowledge becomes useless and network connectivity decreases substantially over time. Although there exist some work in the literature focusing on key distribution in MWSNs, most of these studies support limited node mobility or they introduce expensive protocols for key establishment. Overall, there is room for improvement in connectivity and resiliency of MWSNs, as the mobility feature of WSNs is understood better.

Our aim in this thesis is to turn the node mobility into advantage by providing a smart keyring update mechanism for sensor nodes. Using this mechanism, sensor nodes can re-organize their keyrings with the help of the base stations in the area. This mechanism can be used together with different key predistribution schemes. Regardless of the initial key predistribution scheme, our mechanism increases the local and global connectivity values, without an important decrease in resiliency. Moreover, it does not require an increase in the keyring size and it causes only a small amount of communication overhead. We evaluated the performance of our dynamic keyring update (DKRU) mechanism, by applying it to two different random key predistribution schemes, which are the basic scheme [3] and a deployment knowledge based scheme proposed in [6]. Then, we measure the global connectivity, local connectivity, resiliency and communication overhead of the network via simulations. According to our simulation results, DKRU mechanism provides almost perfect global connectivity and increases the local connectivity by almost 40%, without a significant change in resiliency and communication overhead. Moreover, we show that our mechanism is scalable over different network sizes.

1.2. Organization of the Thesis

The rest of the thesis is organized as follows. In Section 2, background information is given about Wireless Sensor Networks, their constraints, security requirements and mobility features. This section also includes the related work about key distribution schemes in WSNs and MWSNs. The proposed mechanism is explained in Section 3. Section 4 describes the performance metrics and incorporation of proposed mechanism with existing schemes. Performance evaluation of the proposed mechanism is presented comparatively in Section 4. Section 5 summarizes the results and concludes the thesis.

2. BACKGROUND

In this section, we give more detailed information about Wireless Sensor Networks (WSNs), their security requirements and limitations. We also explain the necessary cryptography background and summarize the related work on WSN security. Finally, we focus on the need for mobility in WSNs and introduce the proposed mobility models.

2.1. Wireless Sensor Networks (WSNs)

Wireless sensor networks (WSNs) are composed of small sensor nodes, which are low-powered, low-cost and multifunctional devices using micro-electro-mechanical systems (MEMS) technology and wireless communication [1]. The duty of sensor nodes is to gather data by sensing the environment, process this data and transmit it to a nearby base station. Base stations collect data from sensor nodes and send this data to a remote system via their direct connection to external network. Base stations also perform costly operations and manage the network. Hence, base stations have more resources compared to sensor nodes.

WSNs can be categorized as hierarchical and distributed sensor networks. In hierarchical WSNs, sensor nodes are divided into clusters and nodes in a cluster communicate with the cluster head. Cluster heads relay data between cluster members, other cluster heads and base station. In this hierarchical architecture, failure of a cluster head causes lack of communication with the nodes in that cluster. On the other hand, in distributed WSNs there is no fixed infrastructure. Sensor nodes are deployed to an area and after deployment they form a self organizing, multi-hop wireless network. Hence, failure of a node does not affect a large proportion of network. In this thesis, we work on distributed WSNs.

Sensor nodes are able to monitor a wide range of environmental conditions such as temperature, humidity, lightning, noise, vehicular movement, speed of an object etc.

[1]. Some of the important application areas of WSNs are military applications (e.g. battlefield surveillance, vehicle tracking), environmental applications (e.g. frost fire detection, animal tracking), health applications (e.g. monitoring patient data) and home applications (e.g. smart homes) [1]. According to the requirements of applications, sensor nodes can be static or mobile. Most of the studies in literature propose solutions for static sensor networks. Mobile sensor networks (MWSNs), on the other hand, introduce different challenges such as dynamic network topology, high power consumption and localization problems. There are studies in the literature that focus on these challenges, investigate the impact of mobility on sensor network performance and propose network architectures and realistic mobility models for MWSNs. In this thesis, we address the tradeoff between security and connectivity in MWSNs. Our aim is to increase the secure network connectivity of MWSNs, without deteriorating the resiliency of network against node capture attacks.

2.1.1. Security Requirements of WSNs

Application areas of sensor networks bring out different security requirements to the data carried by sensor nodes. Sensor nodes are usually deployed in hostile and unattended areas. Hence, it may be impossible to provide continuous surveillance after deployment. Moreover, wireless communication can be easily monitored by attackers. While the security requirements may change with respect to the application type, the most significant security needs of WSNs can be listed as follows [7, 8, 9].

- *Confidentiality,* assures that the data transmitted between sensor nodes cannot be accessed by unauthorized parties.
- *Integrity,* guarantees that a message is not modified by an attacker or malignant node during its transmission from one node to another.
- *Authenticity,* ensures that a malicious node cannot masquerade as a trusted network node.
- *Availability,* ensures that the desired network services are available whenever they are needed.

2.1.2. Constraints of WSNs

Security services for WSNs could be maintained via cryptographic protocols, just like other types of networks. However, sensor nodes have various limitations, which makes it impractical to use the traditional methods to provide security. Main constraints of WSNs can be listed as follows [7, 8, 9].

- *Power constraints*: Power requirements of sensor nodes include the computation, communication and sensing capabilities. Because sensor nodes are battery-powered and they cannot be re-charged frequently, it is important to minimize the energy consumption of nodes to increase their life-span. Unfortunately, many of the cryptographic algorithms are complex and require high amount of computation, so they are unsuitable for WSNs.
- *Memory and storage limitations:* Sensor nodes have small amount of memory and storage space. This space is used for application program, computation results and sensor data. Due to the limited space, usually there is not enough memory to run complicated cryptograhic algorithms.
- *Unreliable communication:* Sensor networks are inherently unreliable due to their connectionless, broadcast nature. During transmission, packets may get corrupted or get lost due to high congestion.
- *Unattended operation:* Sensor nodes may be left unattended for long time periods. During this period, nodes can be exposed to physical capture attacks or other environmental hazards. Moreover, managing the network remotely makes it impossible to detect physical tampering and making the sensor nodes tamper-proof is not so feasible due to its high cost.

2.2. Overview of Cryptographic Primitives used in this Thesis

Cryptographic algorithms are used to achieve various aspects of information security in computer systems. One of them is for confidentiality, which is needed for the mechanism proposed in this thesis. Cryptographic algorithms that provide confidentiality can be categorized as (i) asymmetric key cryptography and (ii) symmetric key cryptography. Although some research has been done to facilitate the use of asymmetric cryptography [10, 11, 12, 13], symmetric key cryptosystems are far more efficient and preferable than asymmetric cryptosystems in WSNs.

Symmetric key cryptography involves encryption methods, where the sender and receiver use the same key for encryption and decryption operations. As shown in Figure 2.1, a single key is generated by a key distribution mechanism and it is distributed to sender and receiver sides. This key represents a shared secret between sender and receiver parties that is used to maintain a secure communication channel between them. The encryption and decryption algorithms are publicly known, however one needs to know the secret key to be able to decrypt the ciphertext. For secure implementation of symmetric key cryptography, the symmetric encryption algorithm should be strong and the shared key should be known only by the sender and receiver.

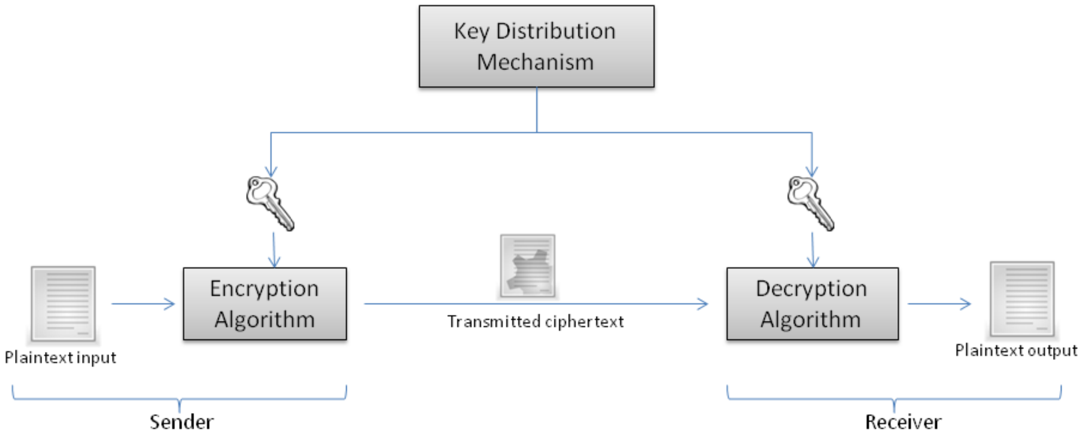


Figure 2.1 Symmetric key cryptography

Some of the best known symmetric key algorithms can be listed as AES, RC4, DES [14] etc. On top of these, lightweight algorithms with small block size and key size can be preferred. Many studies in literature [15, 16, 17] focus on these lightweight, energy-efficient algorithms and their implementations in WSNs.

The main problem of symmetric key cryptography is distribution of the keying material to sender and receiver sides over an unreliable network. Many studies have been conducted to provide robust and reliable key distribution mechanisms for WSNs. These studies will be addressed in detail in the next subsection.

2.3. Literature Survey of Key Distribution in WSNs

Key distribution problem is studied broadly in WSNs. There are good surveys that categorize the existing key management schemes and analyze their performance such as [4, 7, 18, 19, 20, 21, 22]. We also explain the main approaches to key distribution problem below.

2.3.1. Using single network-wide key

In this approach, a single key is loaded to all sensor nodes in the network. The advantage of this approach is that all node pairs can communicate with each other using this single key. Hence, it provides perfect connectivity, which means that each pair of neighboring nodes can form a direct secure link. Moreover, each node keeps only one key, so memory requirement is minimal. BROSK [23] is an example of this approach, which distributes a single master key to all sensor nodes. When two nodes want to communicate, they create a session key using the master key and some other randomly generated information. The problem with network-wide key approach is its vulnerability to physical node capture attacks. When a node is captured by an attacker, master key can be found easily and this key can be used to compromise all the communication links in network. Hence, this approach can only be used if sensor nodes are tamper-proof, which is very costly for WSNs.

2.3.2. Using pairwise keys

In this approach, if there are n nodes in network, each node is loaded with n-1 keys, to communicate with every other node. In this way, each node pair shares a unique pairwise key. This approach provides perfect connectivity and also perfect resilience against node capture attacks because attackers cannot compromise the links between non-captured nodes. However, this solution brings a huge memory overhead to sensor nodes. Considering that the WSNs are usually composed of large number of sensor nodes, using pairwise keys is an infeasible solution due to the memory and storage limitations of sensor nodes.

2.3.3. Probabilistic schemes

Probabilistic schemes aim to balance the tradeoff between network connectivity, resiliency and memory overhead. They provide better resiliency than using single network-wide key, and their memory overhead is much less compared to pairwise schemes. However, they cannot provide perfect connectivity because they cannot guarantee that two nodes in the network will be able to communicate after deployment.

The key predistribution scheme proposed in [3], also known as the basic scheme, is one of the first probabilistic key management schemes. Basic scheme is composed of three simple phases:

1. *Key predistribution phase:* In this phase, firstly a large global key pool is generated. Then, each node is loaded with a subset of keys, chosen randomly from the global key pool without replacement. These keys are loaded to the memory of each sensor node, together with the key identifiers (IDs). These keys form the keyring of the node.

2. *Shared key discovery phase:* After the nodes are deployed to the environment, shared key discovery phase begins. In this phase, sensor nodes broadcast their key identifiers in clear text. If two nodes are in communication range of each other and if they share at least one common key in their keyrings, then these two nodes can communicate securely using this common key, with symmetric encryption. In this case, there is a direct secure link between these nodes. However, shared key discovery phase does not ensure direct secure communication for all node pairs in wireless communication range.

3. *Path key establishment phase:* If a pair of neighboring nodes does not share a common key, they cannot form a direct secure link after the shared key discovery phase. Path key establishment phase aims to assign keys to

these node pairs by using the help of secure links formed in previous phase. If nodes *A* and *B* need to establish a secure communication, they find an intermediary node *C*, that has direct secure links with both *A* and *B*. Then, node *C* helps *A* and *B* to establish a key securely. However, this process causes extra communication cost, so it is important to have as much direct secure links as possible at the end of the shared key discovery phase.

Basic scheme also has a tradeoff between connectivity and security. As the keyring size increases, the probability of forming a secure link between two nodes also increases. However, the network becomes less resilient to node capture attacks because more keys are compromised each time a node is captured.

To strengthen the security of basic scheme, many different approaches are proposed in literature. In q-composite random key predistribution scheme [24], two nodes are required to share at least q common keys to form a secure link. Moreover, the communication key is generated as the hash of all shared keys between these two nodes. q-composite scheme increases network resiliency at the cost of some computation overhead. However, to achieve the same level of connectivity with basic scheme, it requires an increase in the keyring size of nodes. When the keyring size is increased, more keys are compromised after a node capture. Thus, q-composite scheme can be disadvantageous in large-scale attacks. Another modification to basic scheme, called Hashed Random Key Predistribution, is proposed in [25]. In this study, the keys in each node are hashed different number of times. Nodes keep their hashed keys together with the hashing amount. In the shared key discovery phase, two nodes equalize their hashing amounts and obtain a common key to use in symmetric encryption. This scheme improves the resilience of network, however causes some communication, memory and computation overhead. The session key scheme proposed in [26] aims to provide session keys to neighboring node pairs after shared key discovery phase. This scheme improves security, however a session key can also be compromised if the initial keys used to generate the session key are compromised. Finally, the Key Redistribution scheme proposed in [27] replaces the original path key establishment phase of basic scheme with the key redistribution phase. In key redistribution phase, nodes analyze the key IDs received from their neighbors. If node *A* wants to communicate with node *B*, it finds an intermediary node *C* and asks node *C* to send a chosen key to node *B*. In this way, node B obtains a key to communicate with node A. Moreover, after a few iterations of key redistribution phase, node A has common keys with all its neighbors, so it deletes the unused keys in its keyring. This scheme increases the connectivity and resiliency of network, however it causes high communication overhead due to the operations performed at each iteration of key redistribution phase.

2.3.4. Deployment knowledge based schemes

To achieve better connectivity and resiliency than the random key predistribution schemes, some of the studies use other information such as the deployment location of sensor nodes. The scheme proposed by Du et al. [6] (will be referred as Du's scheme) utilizes the fact that sensor nodes will be deployed as groups. This deployment knowledge can be used to give common keys only to the neighboring groups, thus increasing connectivity. In this scheme, sensor nodes are divided into groups, and a key pool is constructed for each group. The key pools of horizontally, vertically or diagonally neighboring groups have certain amounts of overlapping keys. Figure 2.2 demonstrates the zone based key pools and the amount of shared keys between these key pools. If the size of the global key pool is $|S|$, two horizontally or vertically neighboring key pools share $a|S|$ keys where a is the overlapping factor and $0 \leq a \leq 0.25$. Two diagonally neighboring groups share $b|S|$ keys where b is another overlapping factor, $0 \leq b \leq 0.25$ and $4a + 4b \ = 1$. However, two non-neighboring key pools do not share any keys.

In Du's scheme, groups of nodes are deployed to the area using grid pattern. The center point of each grid cell becomes the deployment point for nodes as it can be seen in Figure 2.3.a. Deployment follows a two dimensional Gaussian distribution within each grid cell. Figure 2.3.b demonstrates that, in this deployment model, node density is higher in the middle area of deployment region, compared to the areas that are close to the border.

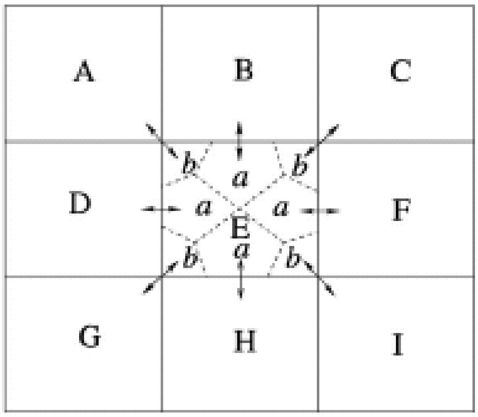
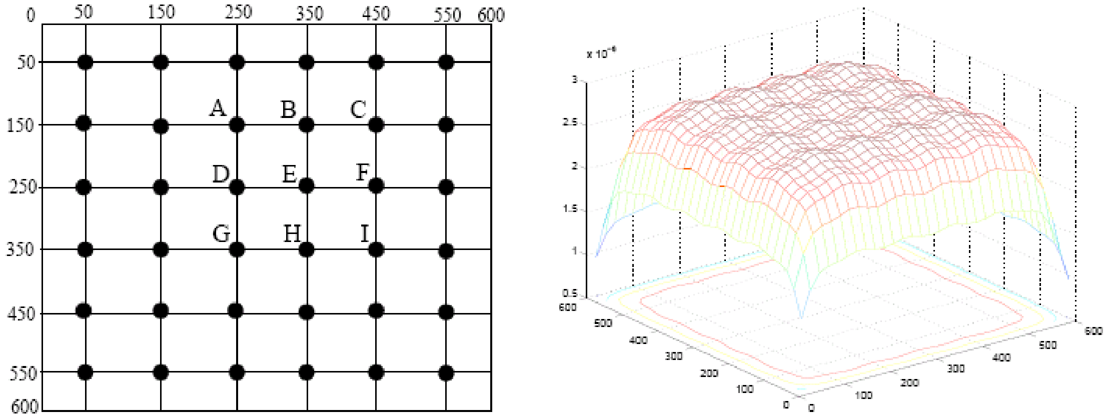


Figure 2.2 Shared keys between neighboring key pools in Du’s Scheme [6]



(a) Deployment points (each dot represents a deployment point).

(b) Deployment distribution on the entire region using the deployment strategy modeled by (a).

Figure 2.3 Deployment model of Du’s scheme [6]

After the key predistribution and deployment phases, Du's scheme follows the shared key discovery and path key establishment steps of basic scheme. To achieve the same connectivity level with basic scheme, Du's scheme requires less number of keys in keyrings of nodes. Hence, Du's scheme decreases the memory overhead and increases the resilience of network.

The Group Based Key Establishment scheme proposed in [28] assigns pairwise keys to the nodes in each group. Moreover, each sensor node is loaded with pairwise keys to communicate with nodes from other groups. If two nodes are in different groups and they do not have a pairwise key, they use intermediary nodes to establish a pairwise key. This scheme is applicable only if the size of the groups is small. Other location based schemes can be found in [29, 30, 31]. The problem with location based schemes is that when they are applied to MWSNs, usage of deployment knowledge becomes a disadvantage in time. In [32], it is showed that the location based schemes do not have any superiority over random key predistribution schemes regarding the MWSNs. Moreover, for certain mobility models, location based schemes may perform far worse than the probabilistic schemes. Detailed analysis will be given in Section 4.

2.3.5. Matrix-based schemes

These schemes are based on Blom's matrix based pairwise key distribution scheme proposed in [33]. In this scheme, a symmetric matrix of size *n* x *n* stores all the pairwise keys for a group of *n* nodes. Each element k_{ij} in the matrix is used to secure the link between node *i* and node *j*. This matrix of keys is calculated using a private matrix and a public matrix of size $(\lambda+1)$ x *n*. Each node stores a row from the private matrix and a column from the public matrix. When two nodes want to communicate, they exchange their columns and the key is computed as the product of their private row and the column of other's. This scheme provides perfect connectivity. Moreover, it has λ-secure property, which means that, if less than λ nodes are captured, none of the links are compromised. However, if λ nodes are captured, the whole network becomes compromised.

2.3.6. Polynomial-based schemes

The Polynomial based Key Predistribution scheme proposed in [34] uses randomly generated λ-degree polynomials. In key predistribution phase, each sensor is loaded with a partially evaluated polynomial share corresponding to its index. When node *i* and node *j* want to communicate, they evaluate the polynomial at point (*i*, *j*) and generate a key. This scheme also has perfect connectivity and λ-secure property.

2.3.7. Combinatorial designs

These schemes use deterministic approaches based on combinatorial design for key distribution. In some of these schemes, connectivity of network depends on the density of nodes, whereas some other studies provide full connectivity even in sparse networks. The schemes proposed by Çamtepe and Yener [35], uses several block design techniques to generate key chains and key pools. Their work provides better connectivity than probabilistic schemes, with smaller key size.

2.3.8. Schemes focusing on mobility

Although there is limited work in literature for the key distribution problem in MWSNs, some schemes designed for static networks can be applied to mobile networks to some extent such as [36] and [37]. The approach proposed in [38] use assisting nodes to distribute keys to sensor nodes. However, this scheme requires too many assisting nodes to achieve a high level of connectivity. Another study [39] proposes a key establishment scheme for MWSNs using the post deployment knowledge of sensor nodes. In this study, each key unit is mapped to a location before deployment. After deployment, sensor nodes determine their post-deployment locations and each node computes the distance between its post-deployment location and the locations associated with the keys. Next, keys are prioritized according to their distance: smaller distance keys have higher priority. Two nodes can communicate by using their common high priority keys. This study assumes that the sensor node locations are known through a location finding system such as GPS. Moreover, it requires a high amount of additional memory to achieve a reasonable connectivity level. The scheme proposed in [40] uses mobile base stations operating as key distribution centers. In this scheme, nodes are not preloaded with keys. After the deployment, base station moves among the nodes, generates and distributes pairwise keys to sensor nodes. This scheme is perfectly resilient to node capture attacks. Because each node pair uses a different key, node capture does not reveal any of the keys used in the rest of the network.

2.4. Need for Mobile WSNs

There are many studies in literature, that shows the importance of mobility in WSNs. Firstly, it is shown that using mobile entities in a sensor network improves the coverage area [41, 42]. Because the initial deployment of WSNs is usually done by scattering the sensor nodes from a plane or vehicle, complete coverage of whole area may not be guaranteed after initial deployment. Moreover, optimal initial deployment of nodes may not be known in many cases. Hence, mobility of nodes can be used to rearrange the network after deployment [43]. Mobility also becomes useful when some nodes in the network die due to their limited battery power or environmental conditions. Replacing or recharging these nodes may be difficult in many cases [44]. Mobile nodes can cover the holes in network, which are caused by the dead nodes.

Secondly, WSNs need to support various different missions. In many of these missions, such as battlefield surveillance, object tracking etc., sensor nodes are required to be mobile. In these conditions, mobile sensor nodes provide enhanced target tracking and better efficiency [45].

2.5. Mobility Models

There are many different mobility models proposed for MWSNs. These models can be categorized as the entity based models and group based models. In the entity based models sensor nodes move individually, whereas in the group based models each sensor node belongs to a group and move together with that group. The survey by Camp, Boleng and Davies [46] is one of the most important studies on mobility models in literature. This study concludes that performance of an ad hoc network can vary significantly with different mobility models. Also, during the performance evaluations, chosen mobility model should closely match the expected real-world scenario. These conclusions are also valid for MWSNs, as shown in [44, 45]. Considering these conclusions and following the recommendations in [46], we chose the Random Walk Mobility Model for entity based mobility and the Reference Point Group Mobility Model (RPGM) for group based mobility in our simulations. The implementations of these mobility models are downloaded from *http://toilers.mines.edu* and they are modified according to the requirements of our study.

2.5.1. Random Walk Mobility Model

In this model, nodes randomly choose a direction and speed from predefined ranges, [*speedmin*; *speedmax*] and [0;2π] respectively [46]. They move in that direction for a constant travel time or a constant distance, and then choose a different direction and speed. In our implementation, each node moves for one minute before they choose a different direction and speed.

2.5.2. Reference Point Group Mobility Model

RPGM model is said to be a generic method for group mobility, because various other group mobility models, such as the Nomadic Community and Pursue Mobility models, can be implemented by changing the input parameters of this model [46].

In RPGM model, a node is chosen as a logical center within each group. This node is also called the group reference point. Group center chooses a random destination point and starts moving to that destination with a randomly chosen speed. Other nodes in the group have individual reference points, updated according to the movement of group reference point. The nodes start to move to a randomly chosen point, which is in a predefined radius of their reference point. In our implementation, reference point of a node can be at most 70 meters away from the group reference point. Moreover, the random point chosen by the node can be at most one meter away from its own reference point. After the group center (group reference point) reaches its destination, it selects a new destination and all other nodes in the group move accordingly.

3. OUR SCHEME: DYNAMIC KEYRING UPDATE MECHANISM

In this section, we present our Dynamic Keyring Update (DKRU) mechanism for mobile wireless sensor networks. Our mechanism can be used together with different key predistribution schemes and it can be considered as an extension to the shared key discovery phase. The main purpose of our mechanism is to enable a sensor node to periodically update its keyring according to its neighbors. After each time the shared key discovery phase is performed, a node determines on a set of keys which are frequent among its neighbors, and requests the transmission of these keys from a base station. As a result, during the next shared key discovery phase, the probability of sharing common keys with neighbors increases for each sensor node.

The application process of DKRU mechanism can be examined in five steps for better explanation. A general overview of DKRU mechanism is given in Figure 3.1. These steps will also be explained in reference to the pseudo code in Figure 3.2. The list of symbols we use in our mechanism is provided in Table 3.1.

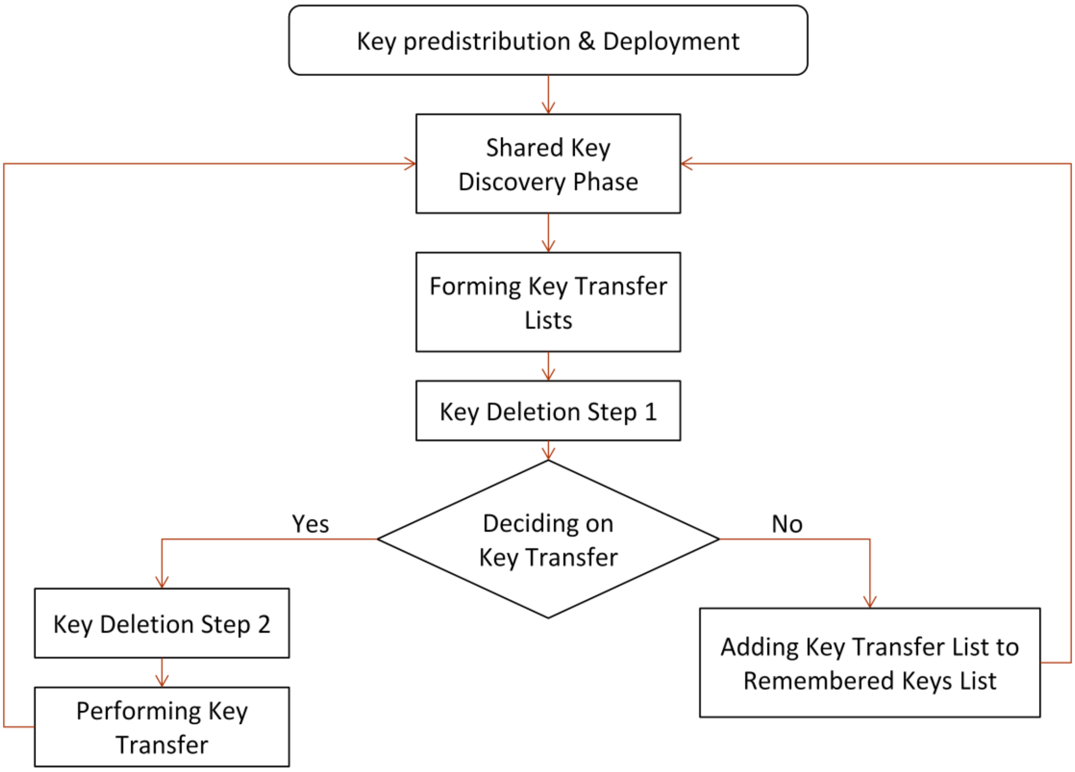

Key predistribution & Deployment
Shared Key Discovery Phase
Forming Key Transfer Lists
Key Deletion Step 1
Deciding on Key Transfer
Yes
No
Key Deletion Step 2
Performing Key Transfer
Adding Key Transfer List to Remembered Keys List

Figure 3.1 Overview of DKRU mechanism

Table 3.1 List of symbols used in our DKRU mechanism

n_i	Sensor node *i*
K_{i-BS}	Pairwise *key shared between node i and base station (BS)*
C_i	List of the Most Frequent Keys belonging to node *i*
T_i	Key Transfer List belonging to node *i*
R_i	List of Remembered Keys belonging to node *i*
m	Size of the keyring
q	Minimum number of common keys required for two neighboring nodes to establish a secure communication (a parameter for q-composite scheme)
tc	Number of frequent key IDs added to Key Transfer List from 1-hop neighbors
p	Probability for adding a frequent key ID to Key Transfer List
t_{max}	Maximum number of keys that a sensor node can transfer from the base station at one time (Maximum Transfer Count)
nc	Node connectivity threshold for key transfer decision
rc	Maximum size for List of Remembered Keys
uc	Usage count threshold for deletion of keys

```
1- Nodes and base stations are predistributed with keys. Then they are deployed to the
deployment area.
2- During the movement of nodes, the following steps are executed periodically:
    3- Shared key discovery phase is performed.
    4- Sensor node pairs, who share at least $q$ common keys, establish a secure
    communication using all their shared keys.
    5- For each node $n_i$;
        6- The Most Frequent Keys list ($C_i$) is formed and sorted in decreasing order.
        7- Starting with the first key in $C_i$, $tc$ number of keys are added to Key Transfer
        List ($T_i$), each with a probability of $p$.
        8- $T_i$ list is sent to neighboring nodes and their lists are received.
        9- According to the $T$ lists coming from neighbors and the Remembered Keys
        list ($R_i$), $T_i$ list is updated.
        10- If the size of $T_i$ list is greater than $t_{max}$, some of the keys in $T_i$ list are
        deleted randomly, until the size of the list becomes equal to $t_{max}$.
        11- Keys that exceed the usage count ($uc$) are deleted from keyring.
        12- Node connectivity is calculated.
            12.a- If node connectivity is below the $nc$ threshold, the keys in $T_i$ list
            will be transferred;
                12.a.i- If there is not enough space in keyring for the transfer of
                new keys, some of the current keys are deleted, starting with
                the earliest used ones.
                12.a.ii- The keys in $T_i$ list are transferred from the Base Station.
            12.b- If node connectivity is above the $nc$ threshold, the keys in $T_i$ list
            are added to the $R_i$ list. If the size of $R_i$ list becomes greater than $rc$,
            the oldest keys in $R_i$ list are deleted, until enough space is opened for
            the latest remembered keys.
        13- $C_i$ and $T_i$ lists are cleared.
```

Figure 3.2 Pseudo code for DKRU mechanism

3.1. Key Predistribution and Deployment

In this phase, sensor nodes are initialized and keys are distributed to each node before deployment. For key distribution, any chosen key predistribution model can be used. In addition, base stations share preloaded pairwise keys with each sensor node and they store all the keys of the global key pool in their memory. The pairwise key between node i and a base station is denoted as K_{i-BS}.

After the key predistribution phase, nodes and base stations are deployed. This part covers the steps 1 to 4 in Figure 3.2.

3.2. Forming the Key Transfer List

After deployment, sensor nodes try to communicate by performing the shared key discovery phase periodically. In shared key discovery phase, sensor nodes broadcast the key IDs in their keyrings to see if they share any common keys with their neighbors. Consequently, a node learns the IDs of all keys that exist in its neighbors' keyrings. Using this information, a node can easily calculate the frequency of each key that is found in its neighbors' keyrings, but not found in its own keyring. The IDs of these keys constitute the List of the Most Frequent Keys (C_i) for this node. Then, these frequencies are sorted in decreasing order. Starting with the most frequent key, a node selects tc number of keys for its Key Transfer List (T_i). Each key is selected with a probability of p. In this initial state, T_i list consists of the frequent keys that are found in n_i's 1-hop neighbors. This part corresponds to steps 6 and 7 in Figure 3.2.

After nodes establish their initial Key Transfer Lists, they broadcast these lists to their neighbors. In this way, nodes can learn the frequent keys found in their 2-hop neighbors. Nodes have a high probability of meeting with their 2-hop neighbors in the future steps, so this broadcast operation can be considered as an investment for the future. The IDs of unique frequent keys coming from the 2-hop neighbors are added to T_i list.

At this point, number of key IDs in T_i list may be more than the allowed Maximum Transfer Count (t_{max}). In this case, some of these key IDs are deleted randomly, until the Maximum Transfer Count is reached. The reason for adding randomness to the process of forming Key Transfer List is to prevent the transfer of same set of keys repeatedly. If the transfer lists become repetitive, many of the links are secured by the same set of keys, which will deteriorate the resiliency of the network. Another precaution against repetitive transfer lists is to have a List of Remembered Keys (R_i) in each node. While forming the T_i list, the keys in R_i list are also checked and these keys are certainly excluded from T_i list. The detailed usage of R_i list will be explained in the next subsection. The steps 8, 9 and 10 in Figure 3.2 corresponds the process of finalizing the Key Transfer List for each node.

3.3. Deciding on Key Transfer

After a node forms its Key Transfer List, it decides whether it needs to transfer these keys or not, according to its *node connectivity*. *Node connectivity* is the ratio of number of neighbors with which a node shares common keys over the number of all neighbors. This ratio can easily be calculated at the end of the shared key discovery phase. If the connectivity of a node is less than a threshold value (nc), this node requests the transfer of new keys from the base station. However, if connectivity of a node is greater than the nc threshold, it does not transfer any keys. Instead, the key IDs in its Key Transfer List (T_i) are added to the List of Remembered Keys (R_i). Node remembers these keys because when forming the Key Transfer List next time, these keys will be excluded even if they are among the most frequent keys. The purpose of List of Remembered Keys is again to prevent the transmission of same set of keys repeatedly. If size of the R_i list has already reached its maximum value (rc), then enough number of keys are deleted from R_i list, starting with the oldest ones. In this way, the latest remembered keys are prioritized. This part covers the step 12 in Figure 3.2, excluding 12.a.i and 12.a.ii, which will be explained in following subsections.

3.4. Key Deletion Process

Another property of our DKRU mechanism is that the size of the keyring of a node never exceeds the predefined keyring size m. Before a node transfers new keys, it deletes the required number of existing keys. Key deletion process has two steps. For the first step, each node stores *key usage count* values for all of its keys. *Key usage count* is calculated as the number of times a key is used in securing links. Keys are deleted if their usage count exceeds a predefined threshold (uc). This step is executed regardless of the key transfer decision, because keys whose usage counts exceed the threshold are not allowed to be used in any links again. After this step, if the node is going to transfer new keys and if it does not have enough space in its keyring, then it deletes some of its existing keys starting with the earliest used ones, until enough space is created for new keys. Key transfer operation is performed after the key deletion process. Hence, keyring size can never exceed m. The steps 11 and 12.a.i in Figure 3.2 corresponds to this key deletion process.

3.5. Performing Key Transfer

When a sensor node wants to request the keys in its T_i list from the base station, the node encrypts the requested key IDs with key K_{i-BS} and sends this message to the base station. Base station sends these keys to sensor node again by encrypting them with key K_{i-BS}. The number of keys that a sensor node can request from the base station at the end of each shared key discovery phase cannot exceed the Maximum Transfer Count (t_{max}). This part corresponds to the step 12.a.ii in Figure 3.2.

After the key transfer operation is performed, or the keys in T_i list are added to the R_i list; node prepares itself for the next shared key discovery phase by clearing the C_i and T_i lists.

The main assumptions of this mechanism are as follows. Base stations are tamper-proof and they cannot be captured by an attacker. In addition, we assumed that each node can directly communicate with a base station in its communication range. These assumptions require a powerful base station with high memory capacity and large communication range. The number of base stations needed depends on the wireless communication range of the base stations and the area of the deployment zone.

4. PERFORMANCE EVALUATIONS

We evaluated the performance of DKRU mechanism by applying it to two different key predistribution schemes, which are the basic scheme [3] and Du's scheme [6]. In this section, we first define the threat model and metrics we used for performance evaluations. Then, we explain how we incorporate the key predistribution schemes with mobility models. Finally, we present the detailed evaluation results of our mechanism when it is used together with basic scheme and Du's scheme respectively.

4.1. Threat Model

Security of a symmetric key cryptosystem rely on the secrecy of the key it uses [7]. If an attacker learns the key and intercepts the encrypted messages, he/she can decrypt these messages easily and confidentiality of system is destroyed. An important problem for WSNs is that, because sensor nodes are vulnerable to physical capture attacks, an attacker can easily retrieve all the keying material stored in a sensor node. Then, attacker can use these keys to decrypt the eavesdropped communications.

4.2. Performance Metrics

To evaluate the performance of our mechanism, we use four different performance metrics, which are global connectivity, local connectivity, resiliency and communication overhead. These metrics are explained in following subsections in detail.

4.2.1. Global Connectivity

Wireless Sensor Networks can also be viewed as key-sharing graphs where nodes are the vertices and secure links are the edges. Global connectivity is defined as the ratio of the size of the largest isolated component in this graph to the size of the whole network [6]. Nodes that are not connected to largest isolated component are considered as disconnected from the secure network. Hence, it is important to have high global connectivity in a network.

4.2.2. Local Connectivity

We define local connectivity as the probability of two neighboring nodes being able to find at least 2 common keys to establish a secure communication link between them. Path key establishment phase is not taken into consideration in the computation of local connectivity, due to its high communication overhead. Hence, it is important for a network to achieve good local connectivity using shared key discovery phase alone. Moreover, path key establishment phase should be avoided in MWSNs because it involves much more communication and computational overheads compared to static WSNs [39].

Connectivity of a random key predistribution scheme is directly proportional to the keyring size of nodes. As the keyring size increases, each node gets more keys from the global key pool and probability that two nodes share common keys also increases. However, there is always a limitation on the keyring size, because sensor nodes have restricted memory capacity. Moreover, increasing keyring size too much deteriorates the resiliency of network. This issue will be explained in the next subsection.

4.2.3. Resiliency

One of the most important security threats for WSNs is the physical capture of sensor nodes by an attacker. Because the sensor nodes are not tamper proof, the attacker can access the keyrings of sensor nodes and decrypt their communication. Moreover, these compromised keys may be used in communication links of non-captured nodes, too. In this case, the attacker can also decrypt the communications among non-captured nodes. Resiliency of a network is inversely proportional to the amount of compromised links between non-captured nodes. In resiliency analysis, it is assumed that when an attacker captures a node, it retrieves all the keys in the node's keyring. Also, attacker has the ability to eavesdrop all message exchanges in the network. However, our attack model does not involve an active attacker who manipulates captured nodes to do further actions. In our simulations, attacker captures one node in each minute. Then we compute the ratio of additionally compromised links due to these node captures.

Resiliency of a random key predistribution scheme is inversely proportional to the keyring size of nodes. As the keyring size increases, more keys become compromised when a node is captured by an attacker. Consequently, these compromised keys give attacker the opportunity to decrypt more communication links among non-captured nodes. On the other hand, decreasing keyring size to provide higher network resiliency is not always a good solution, because it reduces the local and global connectivity of network. Hence, random key predistribution schemes always have a trade-off between connectivity and resiliency.

4.2.4. Communication Overhead

Communication overhead is defined as the average number of bytes sent and received by a node at each shared key discovery phase. Without the Dynamic Keyring Update mechanism, a node sends/receives all of the key IDs to/from its neighbors for the shared key discovery phase. However, using Dynamic Keyring Update mechanism results in additional communications. Firstly, nodes send the key IDs in their initial Key Transfer Lists to their neighbors and receive the key IDs from their neighbors. tc parameter affects the communication overhead of this step. Secondly, if a node is going to perform key transfer, it sends the requested key IDs to the base station and receives the encrypted keys. t_{max} parameter is important here because it determines how many keys will be requested from the base station. In our computations, we considered 4-byte key IDs and 32-byte keys.

4.3. Incorporation of Mobility Models into Key Pre-Distribution Schemes

To understand the behavior of key predistribution schemes in Mobile WSNs, we should first understand the behavior of different mobility models with different initial deployment types. To be able to see the effects of mobility models, we visualized the network for a limited number of nodes. Figures 4.1, 4.2, 4.3 and 4.4 demonstrate the node locations at two specific times: beginning of the simulation ($t = 0$) and end of the simulation ($t = 200$). These figures indicate that the initial deployment model and the chosen mobility model have significant impact on the operation of sensor network. Please note that, because the number of nodes used in performance evaluation simulations are very high, it is difficult to visualize the network with that many number of nodes. Thus, we used 500 nodes for basic scheme and 900 nodes for Du's scheme in this visualization process.

4.3.1. Basic Scheme

For the basic scheme, there is no specific initial deployment model because key predistribution is done randomly. Thus, we can deploy the nodes according to the related mobility model.

For the Random Walk mobility model, node deployment follows a uniform random distribution in the deployment area. Figure 4.1 shows that there is no specific pattern in node deployment. The distribution of nodes looks similar both at the beginning of the simulation and at the end of the simulation.

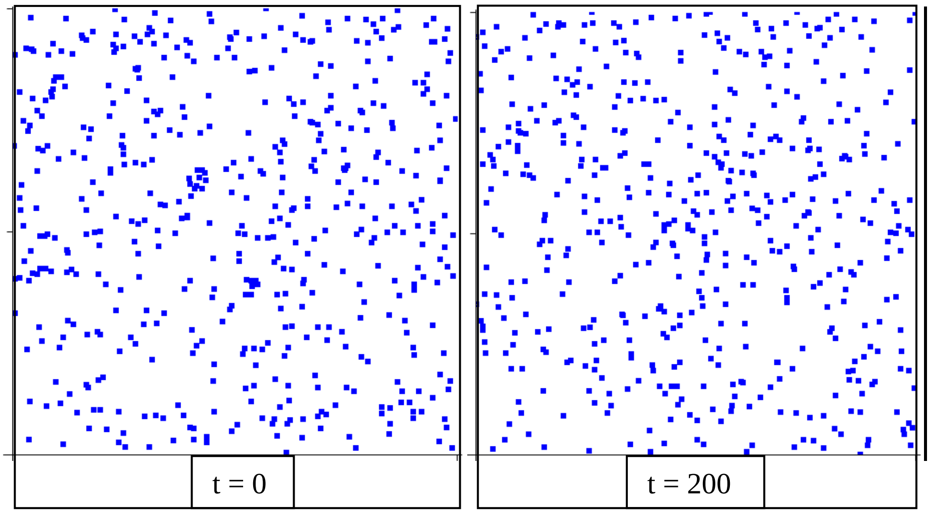


Figure 4.1 Visualization of Random Walk Mobility Model with Basic Scheme

For the initial deployment of RPGM model, firstly the group reference point is deployed to a random point in deployment area. Then, other nodes in the group are placed within in a predefined range of the group center, again randomly.

Figure 4.2 shows the initial and final locations of 5 different node groups. As we can see, these node groups are in very different locations at the end of the simulation. Moreover, some of these groups seem to be merged because they move close to each other during the simulation.

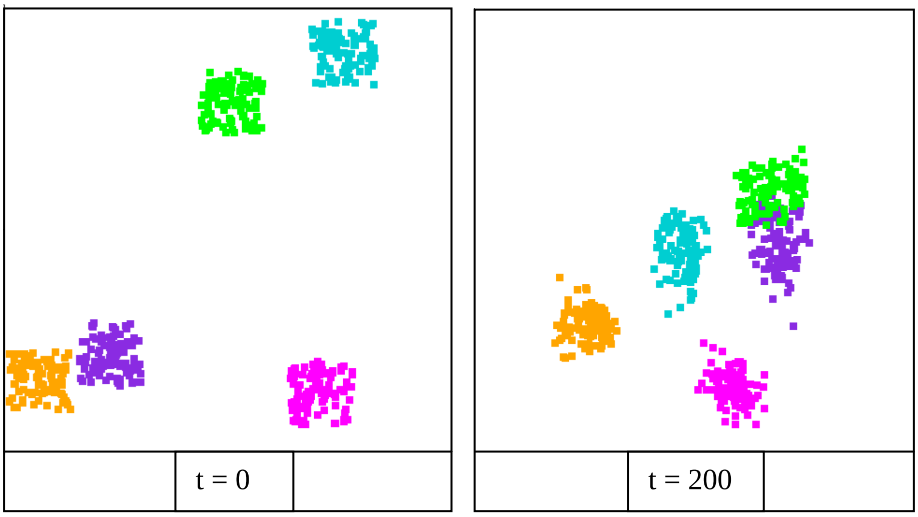


Figure 4.2 Visualization of RPGM Model with Basic Scheme

4.3.2. Du's Scheme

Because Du's Scheme uses deployment knowledge in key predistribution, it follows a specific initial deployment model. Thus, we used the grid pattern in initial deployment, regardless of the mobility model. Our simulation area consists of 100 grid cells, each with a size of 100x100 meters. At each grid cell, a node group consisting of 100 nodes is deployed following a two dimensional Gaussian distribution. The center of each grid cell becomes the deployment point. The standard deviation parameter for Gaussian distribution is set to 50 meters, which means most of the sensor nodes of a group will be within 50 meters range of the center of the grid cell. For visualization purposes, we deployed 9 groups to the area and analyzed the mobility models in Figures 4.3 and 4.4.

Figure 4.3 shows the behavior of Random Walk mobility model. Because the sensor nodes move individually and choose new direction and speed at each minute, they tend to stay in the same neighborhood. As time progresses, nodes start to spread over the simulation area.

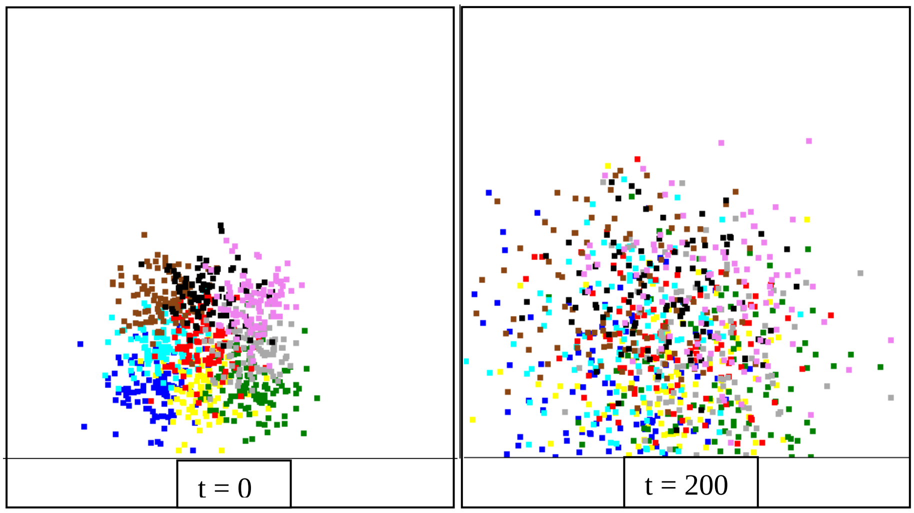


Figure 4.3 Visualization of Random Walk Mobility Model with Du's Scheme

RPGM model, on the other hand, behaves differently than the Random Walk model. Because all nodes in a group move according to the group reference point, groups get separated quickly and start to move in random directions. As we can see in Figure 4.4, node groups may end up in different places at the end of the simulation. Moreover, some node groups may merge into each other during the simulation.

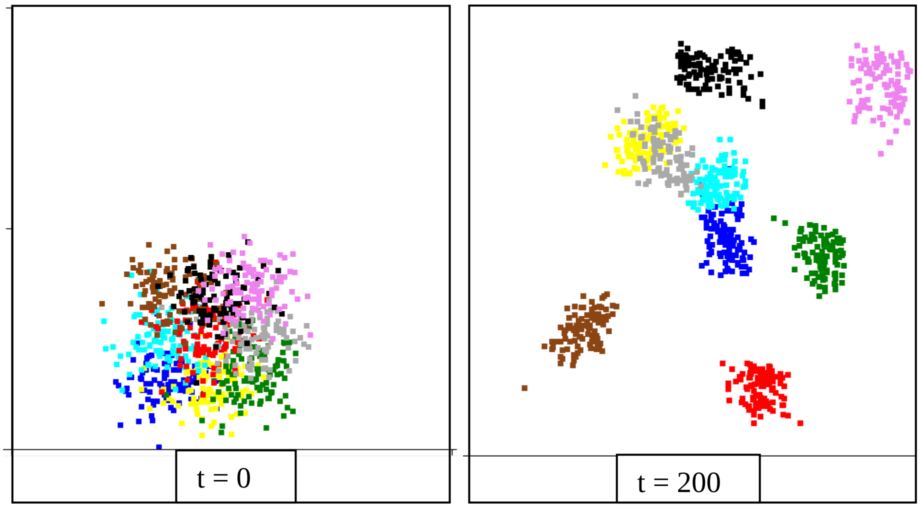


Figure 4.4 Visualization of RPGM Model with Du's Scheme

4.4. Using Basic Scheme As Key Pre-Distribution Basis

In this part, we used basic scheme [3] together with q-composite scheme [24] as the key predistribution basis for sensor nodes. In key predistribution phase, a certain number of keys (m) are randomly chosen from a global key pool for each sensor node. Then, these keys are loaded to nodes, forming their keyrings. In addition, base stations share a pairwise key with each sensor node and they are loaded with all the keys of the global key pool. q value for the q-composite scheme is set to 2, which means at least two common keys are required for secure communication of two nodes.

After all the keys are distributed, sensor nodes are deployed to the field according to the related mobility model, as described earlier.

The performance of our mechanism is evaluated via simulations, using C# for code development. A comparative analysis of basic scheme with and without our Dynamic Keyring Update mechanism is given in following subsections. For a fair comparison, we measured performance of basic scheme with two different m values. Both random walk and RPGM mobility models are evaluated separately in each subsection. The common parameters and system configuration are as follows.

- The number of sensor nodes in the network is 10,000.
- Deployment area is 1,000 x 1,000 square meters.

- Size of the global key pool is 100,000.
- Wireless communication range of sensor nodes is 40 m.
- For mobility models, minimum and maximum speed of nodes are 5 and 15 meters/minute respectively.

Additional parameters are given in Table 4.1.

Table 4.1 List of other parameters used in simulations

	Basic scheme	**DKRU with RPGM**	**DKRU with Random walk**
m	300 and 475	300	300
tc	-	3	3
p	-	0.6	0.6
t_{max}	-	10	10
nc	-	0.9	0.9
rc	-	80	80
uc	-	50	40

4.4.1. Local Connectivity Analysis

In basic scheme, keys are distributed randomly to sensor nodes. Therefore, a node shares common keys with any other node with equal probability, regardless of their coordinates in deployment area. For this reason, mobility of nodes do not significantly affect the local connectivity performance of basic scheme. As it can be seen from Fig. 6 and Fig. 7, local connectivity value of basic scheme is approximately 0.22 when m=300 for both mobility models.

Adding DKRU mechanism to basic scheme increases local connectivity approximately to 0.66, while the keyring size does not change. In other words, DKRU mechanism provides three-times better local connectivity with the same number of keys in keyrings of nodes. This increase is valid for both mobility models, as shown in Figures 4.5 and 4.6.

Another important issue about DKRU mechanism is the time it requires to reach a steady-state local connectivity. Because keyring size is 300, its local connectivity ratio starts with 0.22, which is same as the basic scheme with 300 keys. Then, as the key transfer operation is performed, local connectivity starts to increase. Once node connectivity of many of the nodes exceeds the 0.9 threshold, they stop transferring keys for a while. When node connectivity drops again, nodes start to perform key transfer. This process causes the fluctuations that are observed at the beginning of the simulation results.

To be able to make a fair comparison on the resiliency of basic scheme and DKRU mechanism, we used same local connectivity in both schemes. To do this, we increased the keyring size of basic scheme to 475 keys. In this case, basic scheme also achieves a local connectivity around 0.66 for both mobility models, as seen in Figures 4.5 and 4.6. In other words, basic scheme requires an increase of 175 keys in keyring size to achieve the same local connectivity with DKRU mechanism.

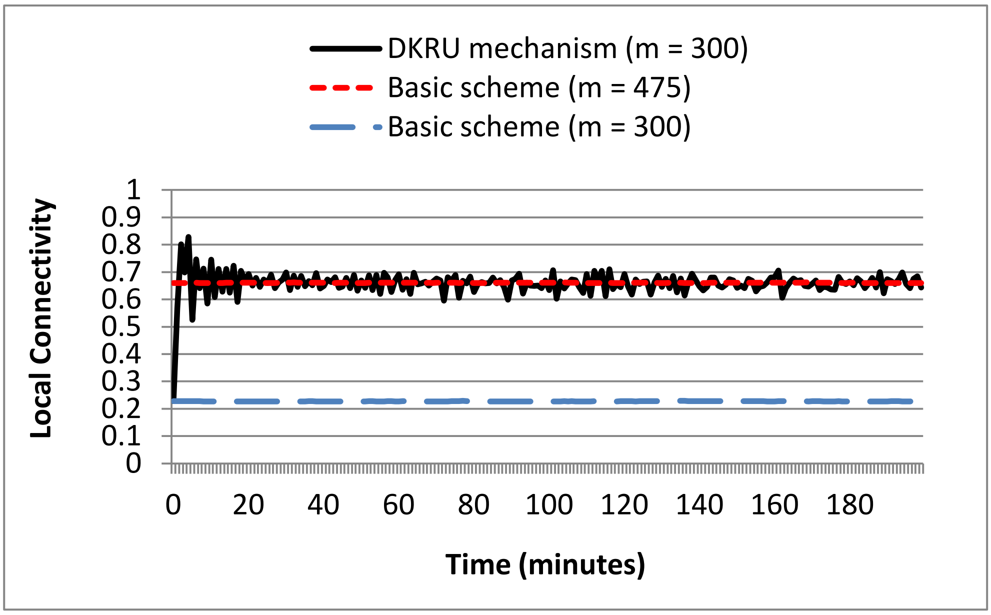


Figure 4.5 Local connectivity versus time for RPGM model

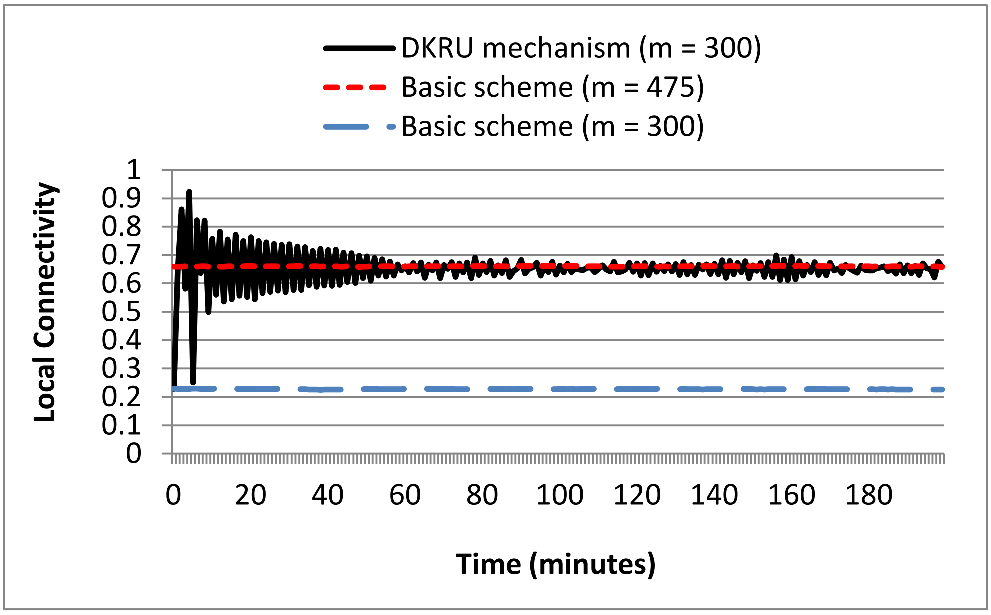


Figure 4.6 Local connectivity versus time for random walk mobility model

4.4.2. Global Connectivity Analysis

When we analyze the global connectivity of basic scheme for RPGM and Random Walk mobility models (as shown in Figures 4.7 and 4.8), we can see that for both values of m, network have almost perfect global connectivity. Using DKRU mechanism also gives similar connectivity values.

In Figure 4.7, however, there are some minor decreases in global connectivity for certain time periods. These decreases are due to the underlying RPGM model. In RPGM model, node groups move independently and sometimes, some of these groups may move away from others and get out of the communication range of the largest isolated component of network. In this case, even if the nodes in these groups share common keys with other nodes, they cannot form any kind of communication links and global connectivity decreases. When these isolated groups get close to the rest of the network again, global connectivity also recovers to its original value. In random walk mobility model each node moves individually and randomly, so this mobility model does not result in isolated components in network.

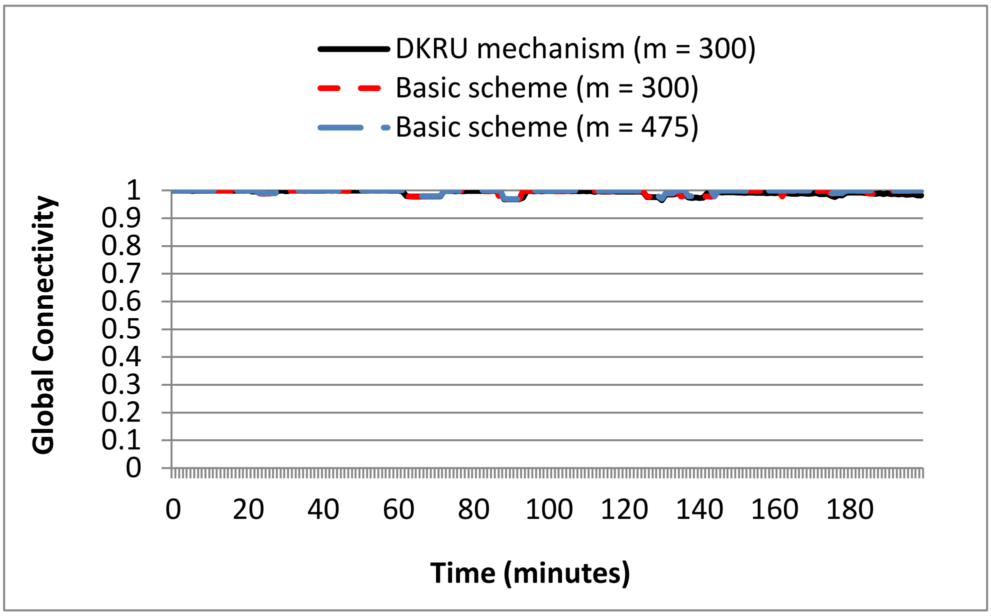


Figure 4.7 Global connectivity versus time for RPGM model

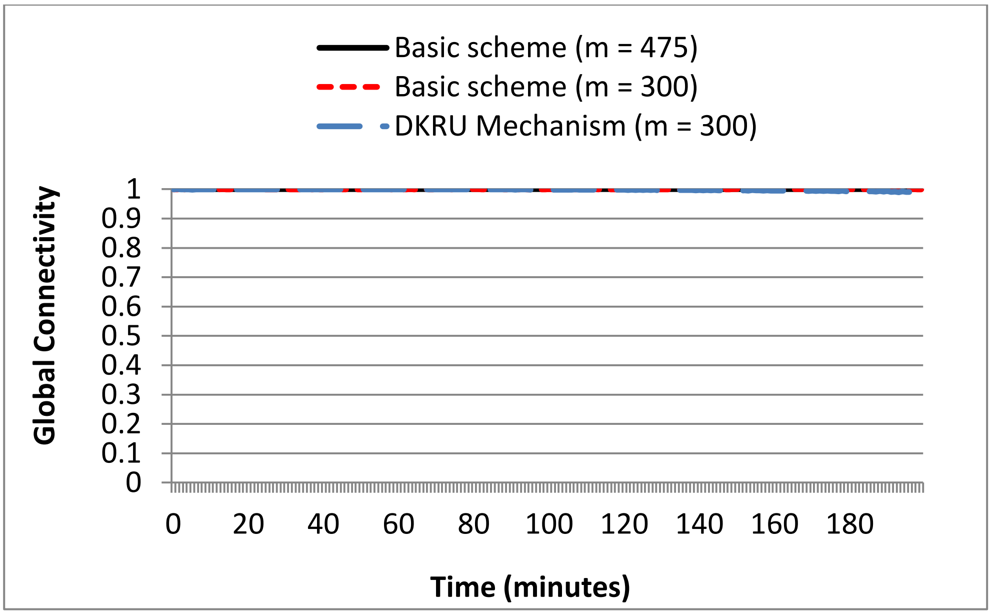


Figure 4.8 Global connectivity versus time for random walk mobility model

4.4.3. Resiliency Analysis

We make a comparative analysis of basic scheme with and without DKRU mechanism in terms of network resiliency. We make the analysis from two perspectives; fixed keyring size and fixed local connectivity.

a) **Fixed keyring size (m=300)**

When the keyring size is fixed for basic scheme and DKRU mechanism, the latter achieves a much higher local connectivity compared to the former. Moreover, DKRU mechanism does not worsen the resiliency of network. Actually, it even brings about a slight improvement to resiliency. As Figures 4.9 and 4.10 demonstrate, additionally compromised links ratio for DKRU mechanism is around 0.13 at the end of the simulation, whereas this ratio is around 0.17 for basic scheme when same number of keys are used. Both mobility models show similar results.

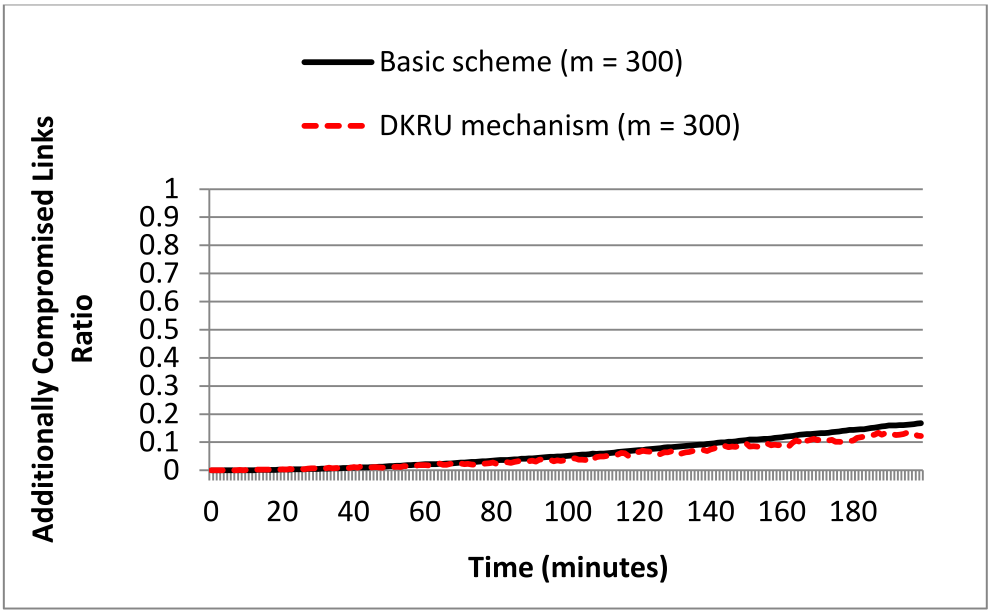


Figure 4.9 Additionally compromised links ratio versus time for RPGM model when keyring size is fixed

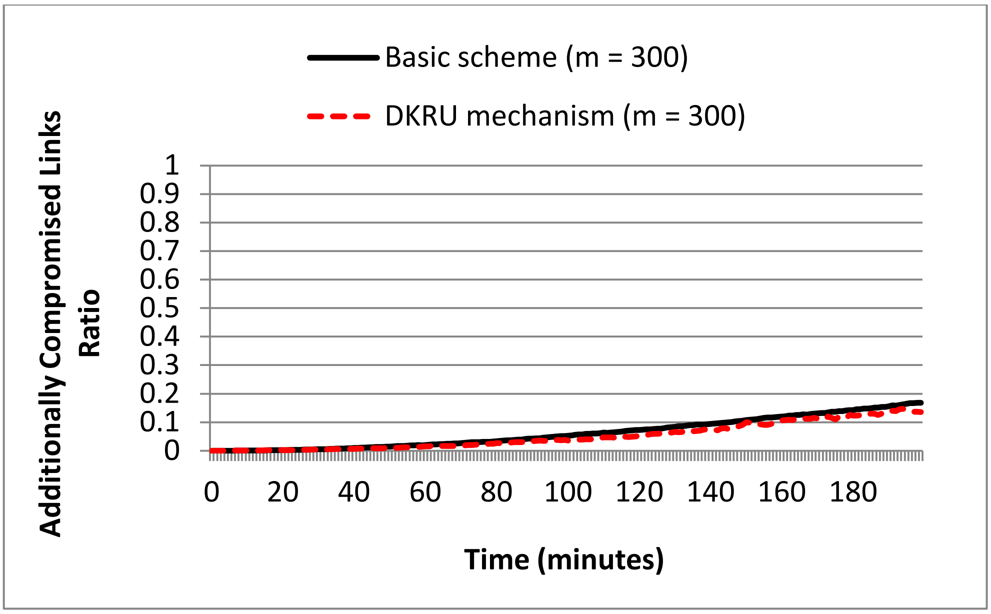


Figure 4.10 Additionally compromised links ratio versus time for random walk mobility model when keyring size is fixed

***b)* Fixed local connectivity**

When the local connectivity of network is fixed to 0.66, basic scheme requires the keyring size to be 475. In this case, additionally compromised links ratio of basic scheme increases approximately to 0.25 at the end of the simulations for both mobility models, as shown in Figures 4.11 and 4.12. For the same local connectivity level, additionally compromised links ratio of DKRU mechanism raises only to 0.13 for both mobility models. In other words, DKRU mechanism provides higher network resiliency when local connectivity of basic scheme and DKRU mechanism are at the same level.

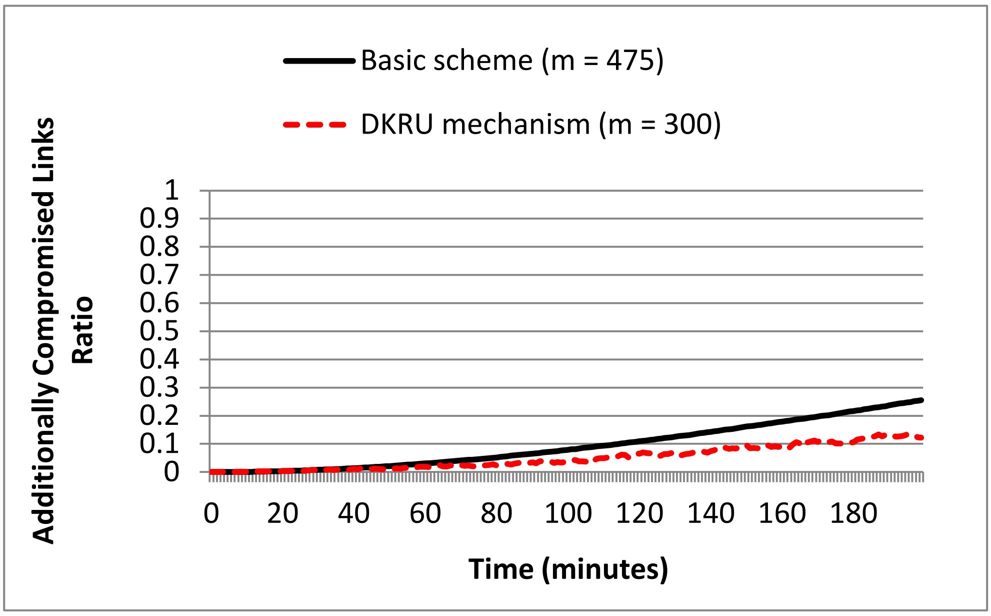


Figure 4.11 Additionally compromised links ratio versus time for RPGM model when local connectivity is fixed

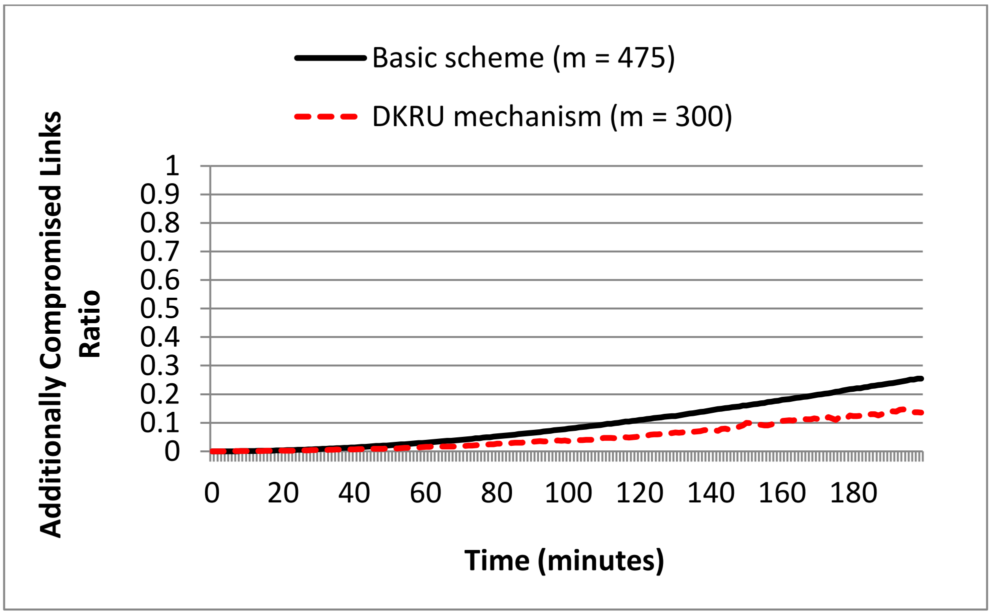


Figure 4.12 Additionally compromised links ratio versus time for random walk mobility model when local connectivity is fixed

4.4.4. Communication Overhead Analysis

As stated earlier, the shared key discovery phase of basic scheme introduces some communication overhead due to the exchange of key IDs with neighboring nodes. DKRU mechanism increases this overhead with the broadcast of initial key transfer lists and key transfer operations. tc and t_{max} parameters affect the additional overhead of DKRU mechanism. In our simulations, tc parameter is set to 3 and t_{max} parameter is set to 10. Once again, we considered the key IDs as 4 bytes and keys as 32 bytes.

Figures 4.13 and 4.14 shows the communication overhead results for RPGM and random walk mobility models. Communication overhead of RPGM model is generally higher than the random walk mobility model, because in RPGM model nodes in each node group move closely to each other. Consequently, a node has more neighboring nodes in its communication range. This increases the number of bytes sent and received during the shared key discovery phase.

As it can be seen in Figures 4.13 and 4.14, communication overhead of our mechanism is very close to the communication overhead of basic scheme when the keyring size m is set to 300 in both schemes. The reason is that, tc and t_{max} parameters do not require high values in our mechanism. As an example, tc parameter is set to 3, which means that a sensor node sends 3 additional key IDs to each of its neighbors and receives 3 additional key IDs from each of its neighbors. In terms of communication overhead, this process corresponds to increasing the node's keyring size by 3. Besides, t_{max} parameter is set to 10, which results in additional communication overhead of 320 bytes. Moreover, key transfer operation is performed only if node connectivity is below the nc threshold. Hence, additional communication overhead due to key transfer does not always occur.

When DKRU mechanism is compared to basic scheme with keyring size of 475 keys, it can be seen that communication overhead of DKRU mechanism is much less than the basic scheme. In other words, when local connectivity of DKRU mechanism and basic scheme are at the same level, DKRU mechanism is more advantageous in terms of communication cost. This result is an expected one because DKRU mechanism achieves this local connectivity level with only 300 keys, whereas basic scheme requires 475 keys for the same level. Hence, at each shared key discovery phase of basic scheme, sensor nodes send/receive additional 175 key IDs to/from each of their neighbors. This result also shows the importance of keyring size for the communication cost.

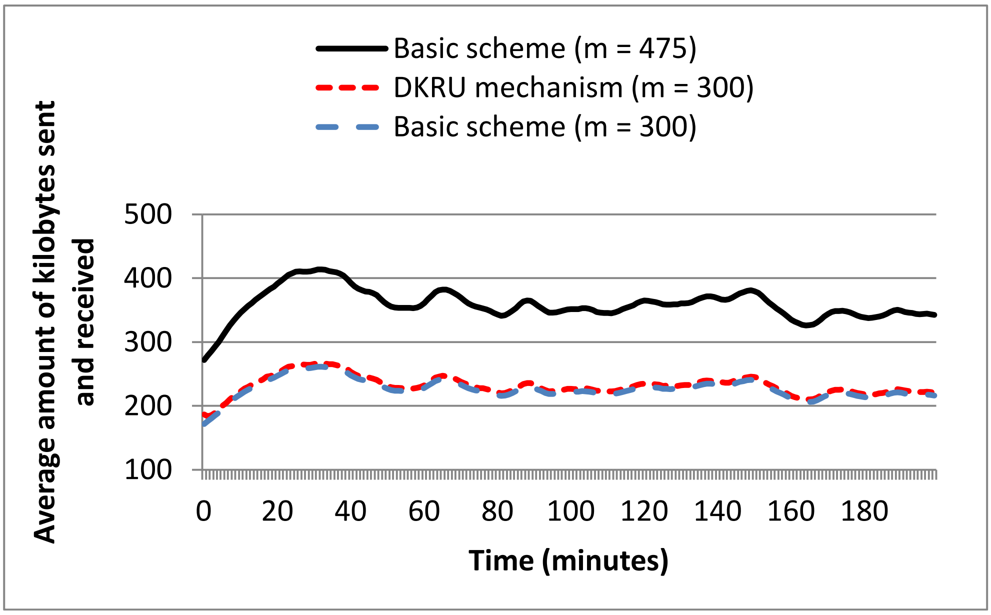


Figure 4.13 Communication overhead versus time for RPGM model

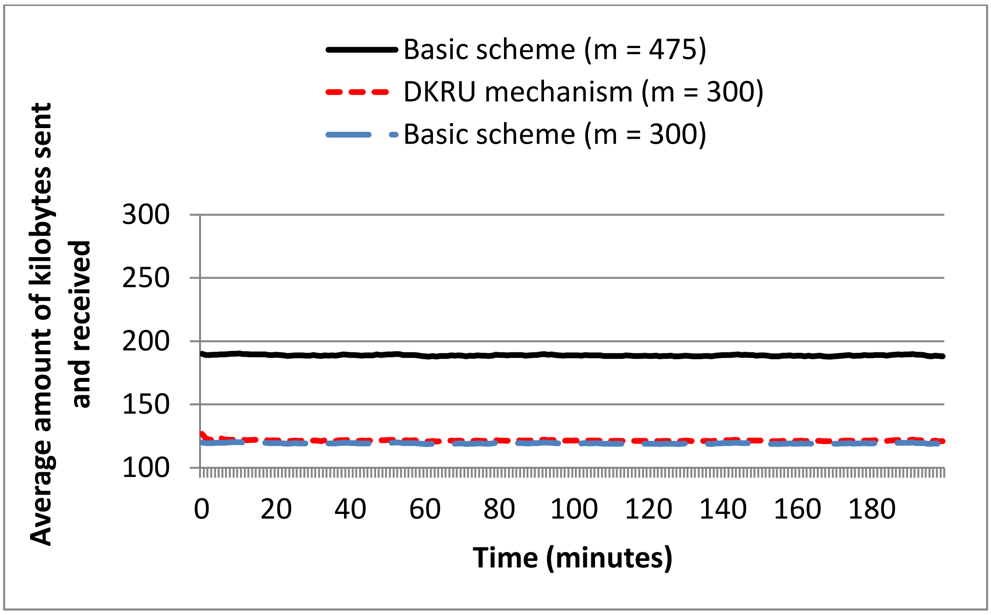


Figure 4.14 Communication overhead versus time for random walk mobility model

4.5. Using Du's Scheme As Key Pre-Distribution Basis

In this part, we used Du's scheme [6] together with q-composite scheme [24] as the key predistribution basis for sensor nodes. Sensor nodes are divided into equally-sized groups and a group key pool is prepared for each group. Keys in group key pools are selected from a global key pool, considering the neighboring relations of groups after deployment. Then, a certain number of keys (m) are distributed randomly to each sensor node, from the related group key pool. q value is set to 2, which means at least 2 common keys are required for secure communication of two nodes. Moreover, base stations are loaded with all the keys of the global key pool and they share pairwise keys with each sensor node.

After the key predistribution phase, nodes and base stations are deployed to simulation environment, as described earlier in Section 4.3.

The performance of our Dynamic Keyring Update mechanism, when it is applied to Du's scheme is given in following subsections. We evaluate both the random walk and RPGM mobility models and compare the performance of DKRU mechanism and Du's scheme. The common parameters and system configuration used in simulations are as follows:

- The number of sensor nodes in the network is 10,000.
- Deployment area is 1,000 x 1,000 square meters.
- Deployment area is divided into a grid of 10 x 10 cells and each cell has a group of 100 nodes in initial deployment.
- Area of each grid cell is 100x100 square meters.
- Size of the global key pool is 100,000.
- Size of the key pool for each group of nodes is 1789.
- Two horizontally and vertically neighboring key pools share exactly 0.2x1789 keys.
- Two diagonally neighboring key pools share exactly 0.05x1789 keys.
- Two non-neighboring key pools share no keys.
- Wireless communication range of sensor nodes is 40 meters.
- Nodes are deployed to the grid cells using two dimensional Gaussian distribution.
- For mobility models, minimum and maximum speed of nodes are 5 and 15 meters/minute respectively.

Additional parameters are given in Table 4.2.

Table 4.2 List of other parameters used in simulations

	Du's scheme	**DKRU with RPGM**	**DKRU with Random walk**
m	300	300	300
tc	-	3	3
p	-	0.6	0.6
t_{max}	-	10	10
nc	-	0.9	0.9
rc	-	80	80
uc	-	200	150

4.5.1. Global Connectivity Analysis

When global connectivity of Du's scheme is examined for Reference Point Group Mobility model (Figure 4.15), it can be seen that even the network is fully connected at the beginning, in a short amount of time, only 10% of the network remains connected. This major decline results from the fact that two non-neighboring key pools do not share any keys in Du's scheme. When the initially non-neighboring groups become neighbors due to mobility, they cannot communicate and each group forms its own isolated component, which constitutes only 10% of the network. Our mechanism fixes this issue because nodes update their keyrings according to their new neighbors. In Figure 4.15, it can be seen that our mechanism provides almost perfect network connectivity for RPGM model.

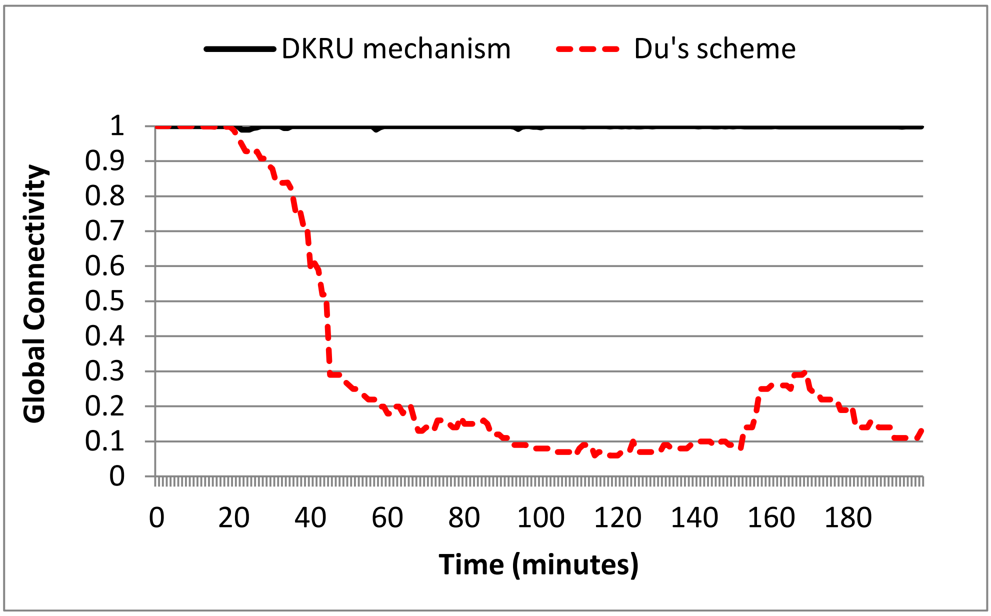


Figure 4.15 Global connectivity versus time for RPGM model

In the random walk mobility model, global connectivity does not decrease significantly for Du's scheme. The reason is that because each node selects a new direction and speed periodically and randomly, they mostly stay in the same neighborhood. Their neighborhood consists of the vertically, horizontally and diagonally neighboring grid cells. Because the key pools of these cells have some overlapping, nodes can continue to establish secure links between them. As shown in Figure 4.16, our mechanism also provides almost perfect global connectivity for this mobility model.

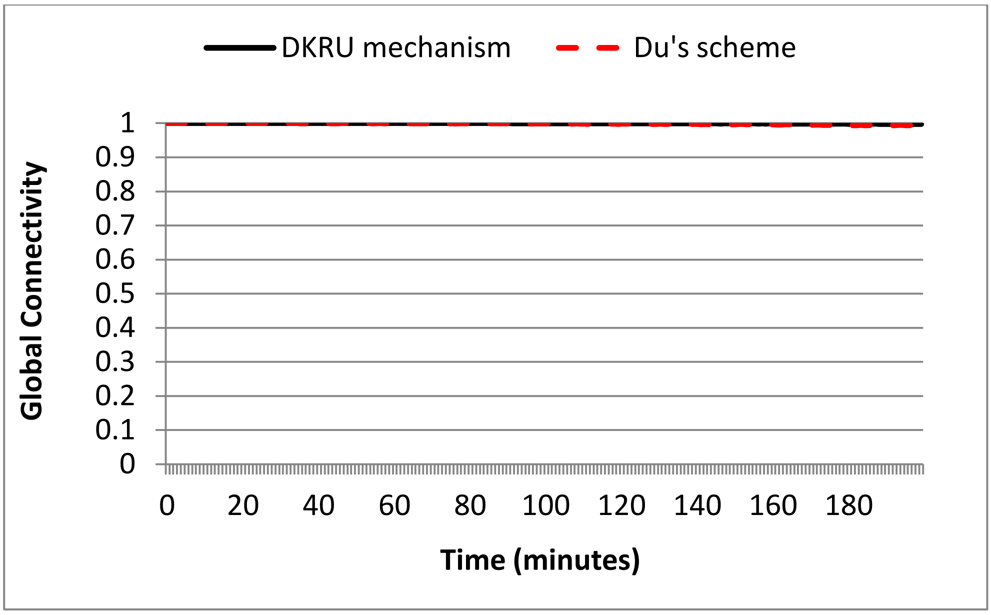


Figure 4.16 Global connectivity versus time for random walk mobility model

4.5.2. Local Connectivity Analysis

Despite the fact that location based key distribution schemes provide better local connectivity than the probabilistic schemes for static WSNs, same situation is not valid for MWSNs. In Figures 4.17 and 4.18, it can be seen that the local connectivity of Du's scheme decreases from 90% to 30% over time for both mobility models. This decrease is sharper for RPGM model because neighboring relationships break off faster in this model. When the DKRU mechanism is added to Du's scheme, local connectivity can be improved to 60% in steady state, without requiring any increase in keyring size.

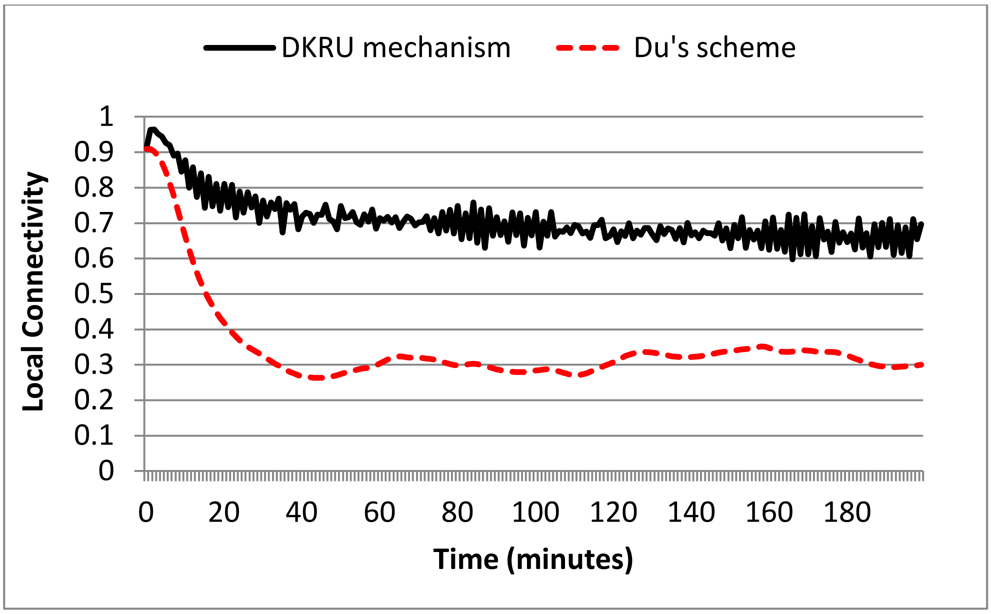


Figure 4.17 Local connectivity versus time for RPGM model

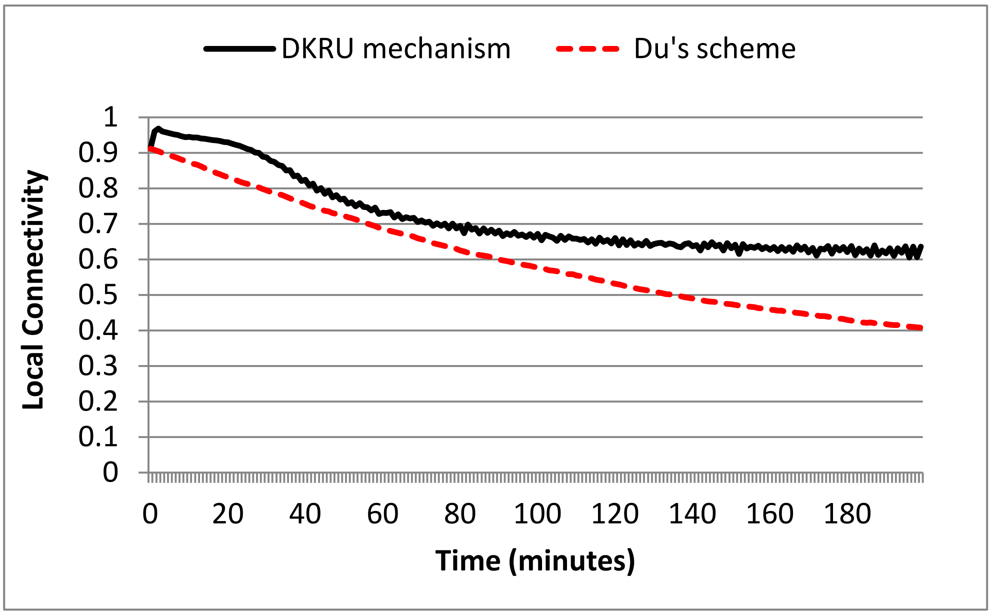


Figure 4.18 Local connectivity versus time for random walk mobility model

4.5.3. Resiliency Analysis

For the RPGM model, Du's scheme has very low global and local connectivity. Due to this low connectivity of network, it is hard to make judgments about the resiliency of network. As shown in Figure 4.19, additionally compromised links ratio for Du's scheme is close to zero after 200 minutes of simulation. However, this does not indicate that the network is resilient. Actually, this indicates that there are not enough links in the network to be compromised. On the other hand, our mechanism provides high local and global connectivity for this mobility model. Despite this high connectivity, additionally compromised links ratio reaches only to 0.1 in our mechanism. This means, about 90% of the communication links between non-captured nodes are still secure.

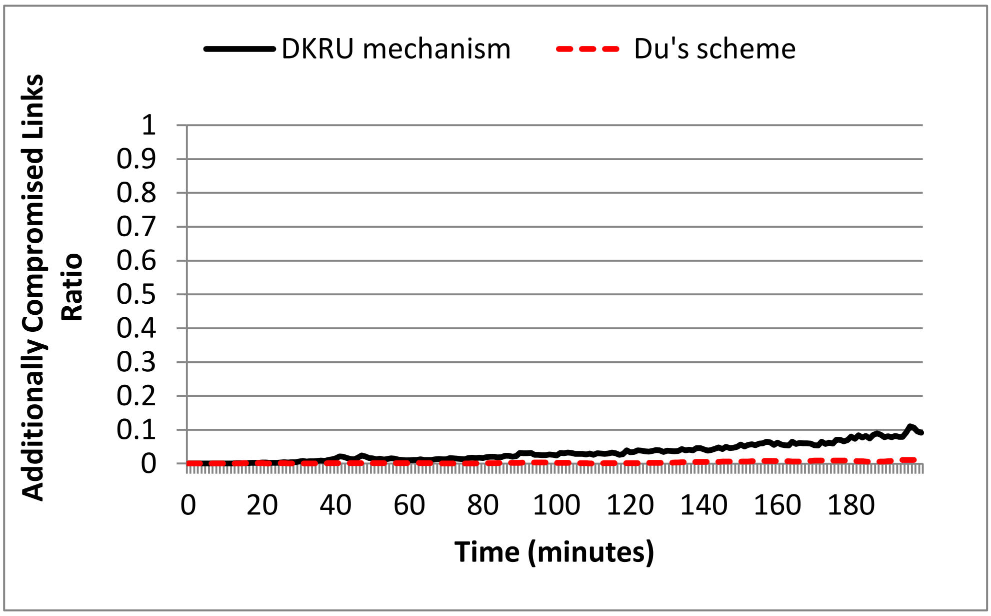


Figure 4.19 Additionally compromised links ratio versus time for RPGM model

For the random walk mobility model, Figure 4.20 shows that the additionally compromised links ratio of our mechanism is almost equal to the Du's scheme, except for a slight increase towards the end of the simulation. These results demonstrate that application of our mechanism does not significantly deteriorate the resiliency of network.

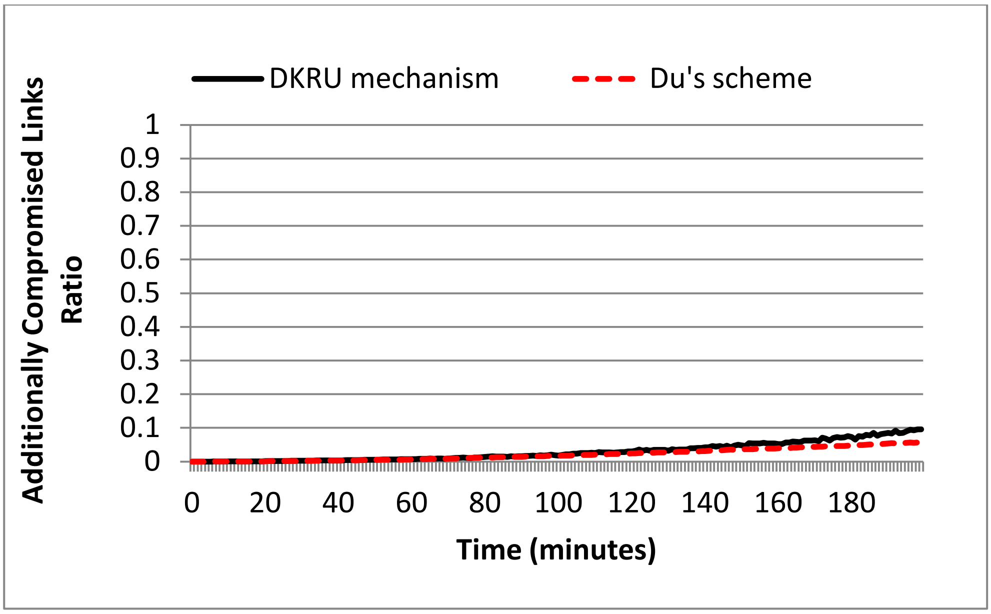


Figure 4.20 Additionally compromised links ratio versus time for random walk mobility model

4.5.4. Communication Overhead Analysis

As it can be seen in Figures 4.21 and 4.22, communication overhead of our mechanism is very close to the communication overhead of Du's scheme for both mobility models. Our mechanism does not introduce a significant increase in communication overhead, because values given to the tc and t_{max} parameters bring about a limited amount of communication cost.

Figures 4.21 and 4.22 also show that the communication overhead is higher in the RPGM model compared to the random walk mobility model. The reason is that, when nodes move together as a group, they have more neighbors in their communication range. Consequently, in the shared key discovery phase, they send and receive higher number of key IDs. Moreover, when two groups get close to each other during their movement, nodes can have even more neighbors in their communication range.

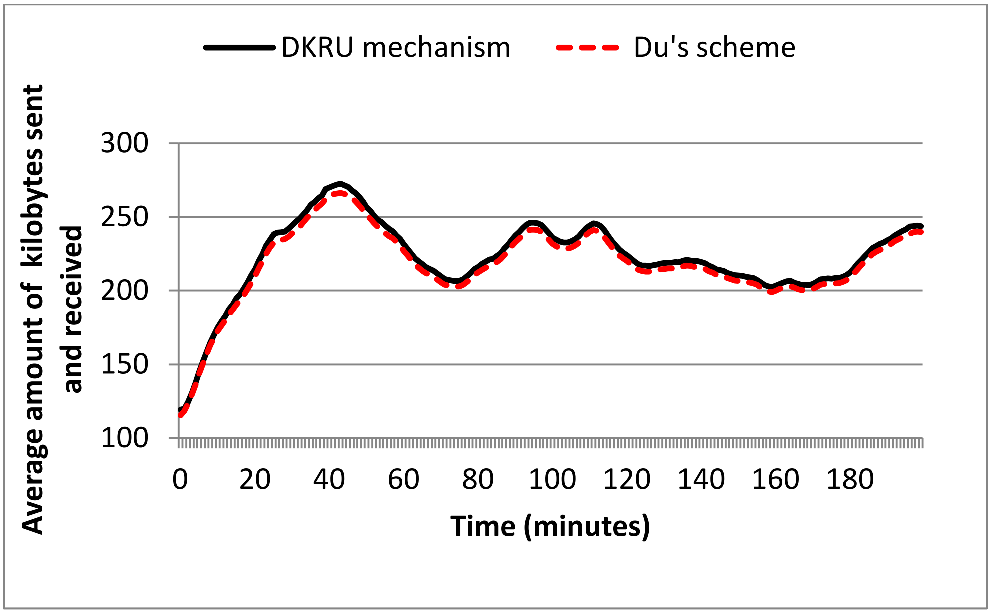


Figure 4.21 Communication overhead versus time for RPGM model

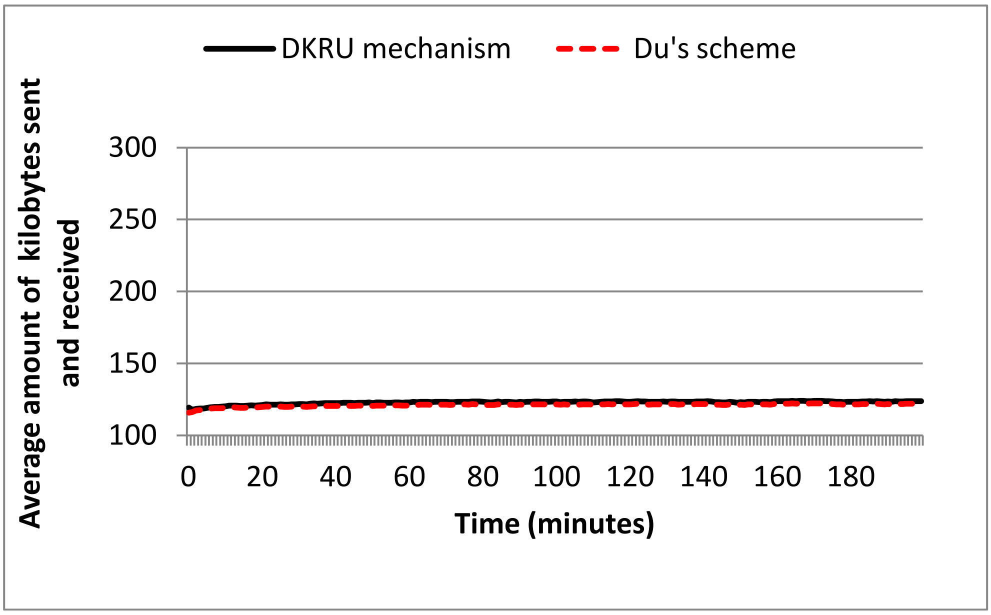


Figure 4.22 Communication overhead versus time for random walk mobility model

4.6. Choosing Simulation Parameters

Our DKRU mechanism has a number of parameters that affect its performance and need to be chosen carefully. These parameters are probability of adding a frequent key ID to Key Transfer List (p), maximum transfer count (t_{max}), maximum key usage count (uc) and node connectivity threshold (nc). These parameters can be selected according to the key predistribution basis and requirements of the network. To be able to find out to the optimal parameters for each key predistibution basis and mobility model, we perform unit analyses on the parameters. In this section, we explain how the parameters of our scheme affect its performance and how we chose the optimal parameters.

- Probability of adding a frequent key ID to Key Transfer List (p): After a node forms its Most Frequent Keys list (C_i), it selects tc number of keys from this list to form its initial Key Transfer List (T_i). As the p parameter increases, a node has more chance to transfer the keys with higher frequency. This decreases the randomness in the keyrings, because frequent keys become more frequent, and after a while, all nodes start to use the same set of keys. However, connectivity of network should not depend on a set of keys, because this decreases network resiliency. Hence, p parameter should not be set to a very high value. Also, it should not be less than 0.5 because otherwise the keys with lower frequencies have more chance to join the T_i list of a node. To prioritize the frequent keys in transferring operation without losing the randomness of keyrings, we set the p parameter to 0.6 in our simulations.

- Maximum transfer count (t_{max}): This parameter limits the number of keys that can be transferred to a node after each shared key discovery phase. In other words, it determines the maximum size of T_i list. Increasing t_{max} parameter improves connectivity because a node gets more keys which are frequent among its neighbors. However, increasing t_{max} also causes higher communication overhead because transferring more keys is a more costly operation. Our empirical studies show that setting t_{max} parameter to 10 is enough to increase connectivity without causing high communication overhead.

- Maximum key usage count (uc): This parameter is one of the most important parameters that affect the connectivity of network. When uc parameter is low, each key is used in only a few links and then deleted. This decreases network connectivity and causes fluctuations in connectivity level because keys used in communication links are deleted quickly. If the node density is high in a network, uc parameter should also be raised to increase the lifespan of keys. Therefore in our simulations, uc parameter has higher values in Reference Point Group mobility model, compared to Random Walk mobility model. The values for uc parameter in different network settings are chosen empirically to minimize the fluctuations in connectivity level of network.

- Node connectivity threshold (nc): Key transfer decision of nodes depends on this parameter. Decreasing this threshold also decreases the connectivity of the network because nodes do not transfer new keys frequently. However, when nc threshold is set to 1, it means that a node always requests the transfer of new keys, even if it can communicate with all of its neighbors. This may cause unnecessary communication overhead because if a node is already connected, it does not need new keys. Moreover, transferring new keys unnecessarily decreases the randomness of keyrings. Thus, we set the nc threshold to 0.9 in our simulations.

- Remembered keys count (rc): If a node decides not to transfer the keys in its Key Transfer List (T_i), it adds these keys to Remembered Keys list (R_i). Keys in R_i list are excluded from the T_i list when it is being prepared. Hence, as long as a key stays in R_i list, it cannot be transferred. rc parameter determines the maximum size of R_i list. As an example, if rc is set to 80 and t_{max} is set to 10, a key in R_i list needs 8 rounds of negative key transfer decision to leave the R_i list. Decreasing rc parameter deteriorates the resiliency of network, because same set of keys starts to circulate in the Key Transfer Lists of nodes.

4.7. Detailed Analysis of q Value

In our performance evaluations, we used the basic scheme [3] and Du's scheme [6] as different key predistribution bases. As in [24], we require q common keys for two nodes to establish a secure link between them. We set the q value to 2, which means that a node pair needs at least 2 common keys to form a secure direct link. This approach increases the resiliency of network, however it decreases connectivity because more common keys are required to form direct secure communication links.

Another disadvantage of this approach is that, as the q value increases, network becomes more vulnerable to large-scale attacks. When a small amount of nodes are captured, the adversary can decrypt little number of links with the compromised keys. However, when large number of nodes are compromised, larger fractions of network are revealed to adversary [24].

To find the optimal q value for our mechanism, we evaluated its performance for different q values. In this evaluation, we fixed the local connectivity of network to approximately 65% and we compared the resiliency values when q is equal to 1, 2, 3 and 4. Our mechanism provides almost perfect global connectivity in all cases, so we do not consider global connectivity metric in this comparison. To fix the local connectivity, we sometimes used different parameters for different q values. Our results for two different key predistribution bases (DKRU with basic scheme and DKRU with Du's scheme) and two different mobility models are as follows.

1. DKRU with basic scheme

Tables 4.3 and 4.4 give the parameters used to fix the local connectivity of network to 65% for different q values. As it can be seen in Table 4.3, node connectivity (nc) threshold is lower when q is equal to 1 in Random Walk Mobility model, compared to other q values. The reason is that, local connectivity of network is already close to 60% when q is equal to 1, and neighborhood of nodes does not significantly change in Random Walk mobility model. Hence, nodes do not need to transfer new keys frequently to achieve 65% local connectivity.

Table 4.3 Parameters for different q values when DKRU mechanism is used with basic scheme and Random Walk mobility model

	DKRU with basic scheme			
	q = 1	q = 2	q = 3	q = 4
m	300	300	300	300
tc	3	3	3	3
p	0.6	0.6	0.6	0.6
t_{max}	10	10	10	10
nc	0.55	0.9	0.9	0.9
rc	80	80	80	80
uc	40	40	40	40

Table 4.4 Parameters for different q values when DKRU mechanism is used with basic scheme and RPGM model

	DKRU with basic scheme			
	q = 1	q = 2	q = 3	q = 4
m	300	300	300	300
tc	3	3	3	3
p	0.6	0.6	0.6	0.6
t_{max}	10	10	10	10
nc	0.9	0.9	0.9	0.9
rc	80	80	80	80
uc	50	50	50	50

Figures 4.23 and 4.24 show the additionally compromised links ratios for different mobility models. These figures demonstrate that the best resiliency value is achieved when q is equal to 2. For larger values of q, DKRU with basic scheme suffers from the abovementioned disadvantage of requiring at least q keys for secure communication. As a result, network resiliency deteriorates. Considering these results, setting q value to 2 seems to be an optimal decision for our mechanism.

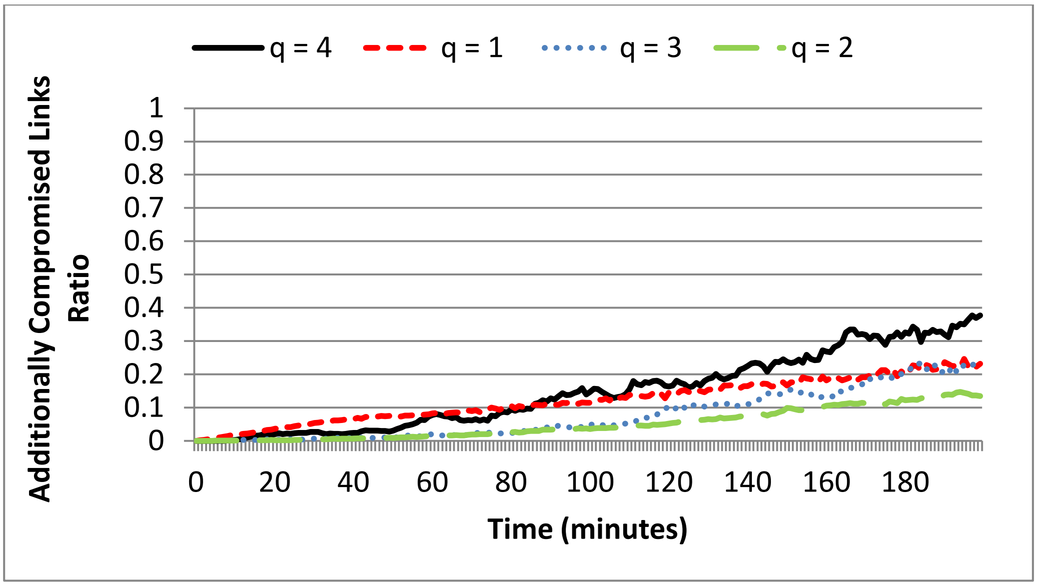


Figure 4.23 Additionally compromised links ratio in DKRU with basic scheme for different q values and random walk mobility model

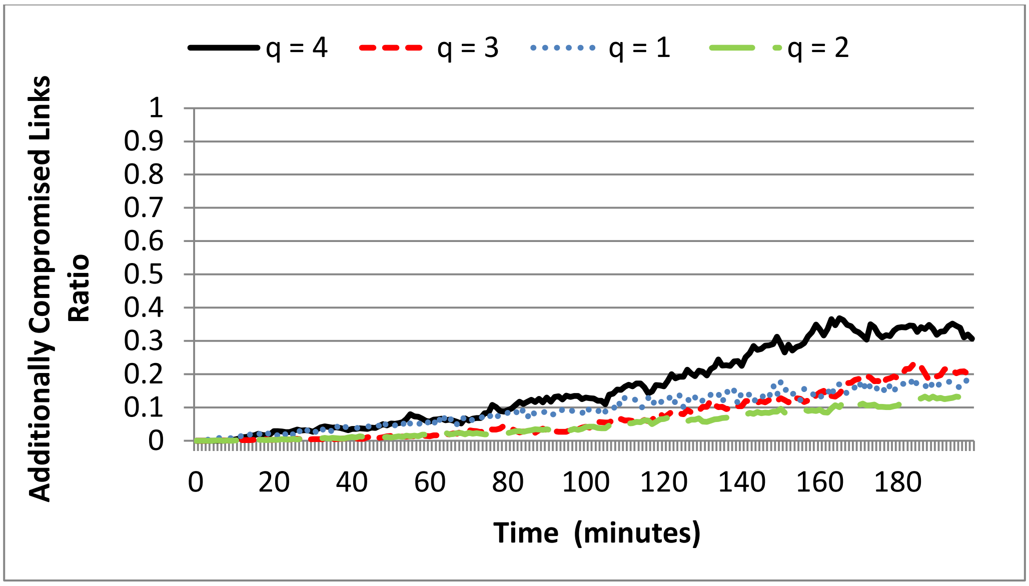


Figure 4.24 Additionally compromised links ratio in DKRU with basic scheme for different q values and RPGM model

2. DKRU with Du's scheme

Tables 4.5 and 4.6 give the parameters for different q values when DKRU mechanism is used with Du's scheme. As it can be seen in Table 4.5, for Random Walk mobility model the usage count (uc) parameter varies with different q values. The reason for these changes in uc value is to minimize the fluctuations in local connectivity level of network.

Table 4.5 Parameters for different q values when DKRU mechanism is used with Du's scheme and Random Walk mobility model

	DKRU with Du's scheme			
	q = 1	q = 2	q = 3	q = 4
m	300	300	300	300
tc	3	3	3	3
p	0.6	0.6	0.6	0.6
t_{max}	10	10	10	10
nc	0.9	0.9	0.9	0.9
rc	80	80	80	80
uc	100	150	80	80

Table 4.6 Parameters for different q values when DKRU mechanism is used with Du's scheme and RPGM model

	DKRU with Du's scheme			
	q = 1	q = 2	q = 3	q = 4
m	300	300	300	300
tc	3	3	3	3
p	0.6	0.6	0.6	0.6
t_{max}	10	10	10	10
nc	0.9	0.9	0.9	0.9
rc	80	80	80	80
uc	200	200	200	200

Figures 4.25 and 4.26 show the additionally compromised links ratios for Random Walk mobility and RPGM models. According to these schemes, resiliency performance of DKRU with Du's scheme does not change significantly when q is equal to 2, 3 and 4. The reason is that, neighboring nodes share more common keys in Du's scheme, compared to the basic scheme. Hence, the number of keys used in communication links may already be more than or equal to 2. However, when q is 1, the resiliency of network is worsened because two nodes can communicate even if they share only one key. When this shared key is compromised, their communication link can be easily decrypted by an attacker.

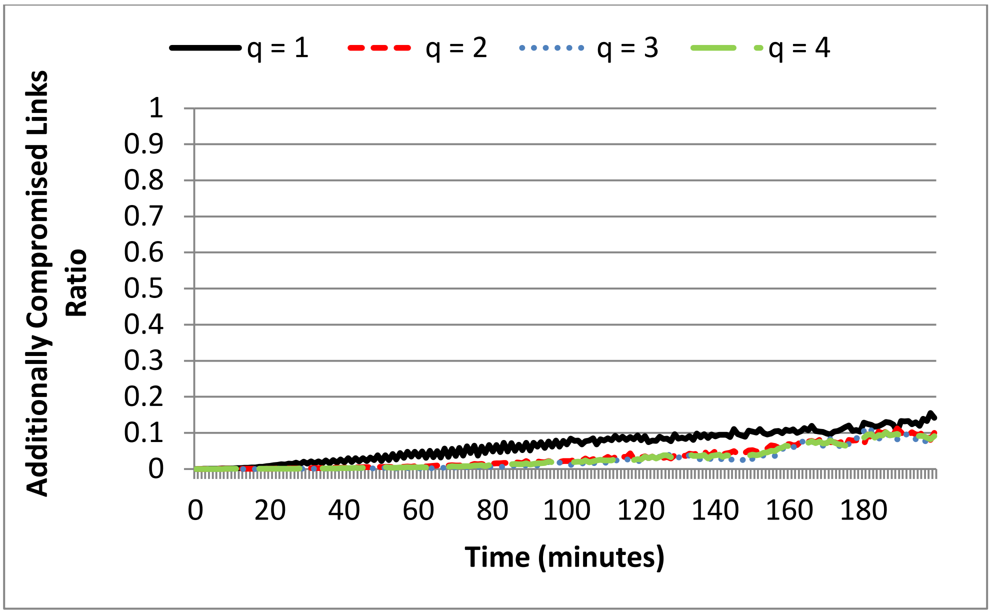


Figure 4.25 Additionally compromised links ratio in DKRU with Du's scheme for different q values and random walk mobility model

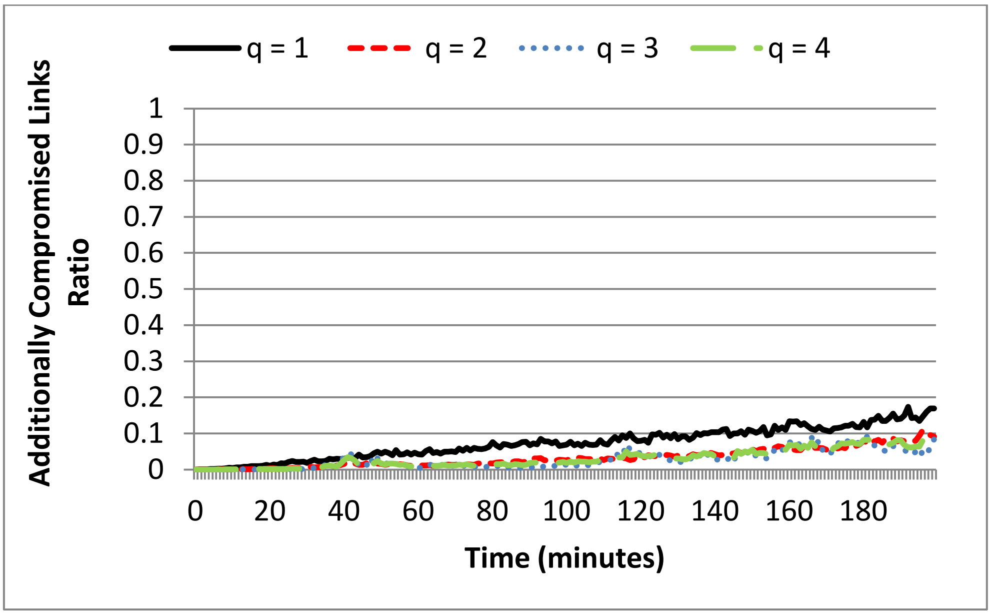


Figure 4.26 Additionally compromised links ratio in DKRU with Du's scheme for different q values and RPGM model

4.8. Scalability Analysis

The number of sensor nodes used in a WSN may vary significantly according to the application type. As the number of nodes in a network grows, the connectivity and resiliency performance of the network should not be worsened. *Scalability* of a security mechanism can be defined as the ability to support various network sizes [18]. To analyze the scalability property of our mechanism, we analyze its performance for different network sizes. In this analysis, we use Du's scheme as key predistibution basis and apply our DKRU mechanism when the deployment area is 800×800, 1000×1000 and 1200×1200 square meters, respectively. In each of these network sizes, we measure the global connectivity, local connectivity and resiliency performance of our mechanism. Results of our analysis show that the performance of our mechanism is not significantly affected by the network size. Thus, our mechanism is fairly scalable. Details of our analysis are given in the following subsections.

4.8.1. Parameters for different network sizes

Parameters we use in key predistribution and deployment phases for different network sizes are given in Table 4.7. The parameters of DKRU mechanism for different mobility models are the same as the parameters given in Table 4.2.

Table 4.7 Parameters for different network sizes

	Deployment area (m^2)		
	800 x 800	1000 x 1000	1200 x 1200
Number of sensor nodes	6,400	10,000	14,400
Number of grid cells	64	100	144
Number of nodes in each grid cell	100	100	100
Area of each grid cell (m^2)	100 x 100	100 x 100	100 x 100
Size of global key pool	100,000	100,000	100,000
Size of group key pools	2724	1789	1264
Horizontal and vertical key pool overlapping factor	0,2	0,2	0,2
Diagonal key pool overlapping factor	0.05	0.05	0.05

4.8.2. Global Connectivity Analysis

Figures 4.27 and 4.28 show that our mechanism provides almost perfect global connectivity for different networks sizes. For RPGM model, there are some minor decreases in global connectivity for certain time periods; however, these decreases are due to the underlying mobility model. Overall, we can say that our DKRU mechanism is scalable in terms of global connectivity.

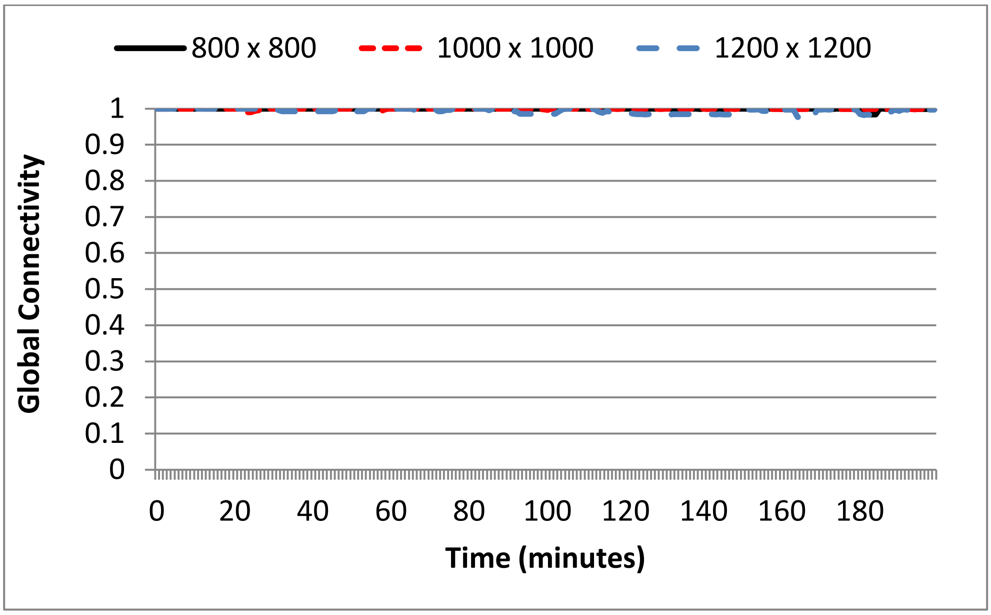


Figure 4.27 Global Connectivity in DKRU with Du's scheme for different network sizes and RPGM model

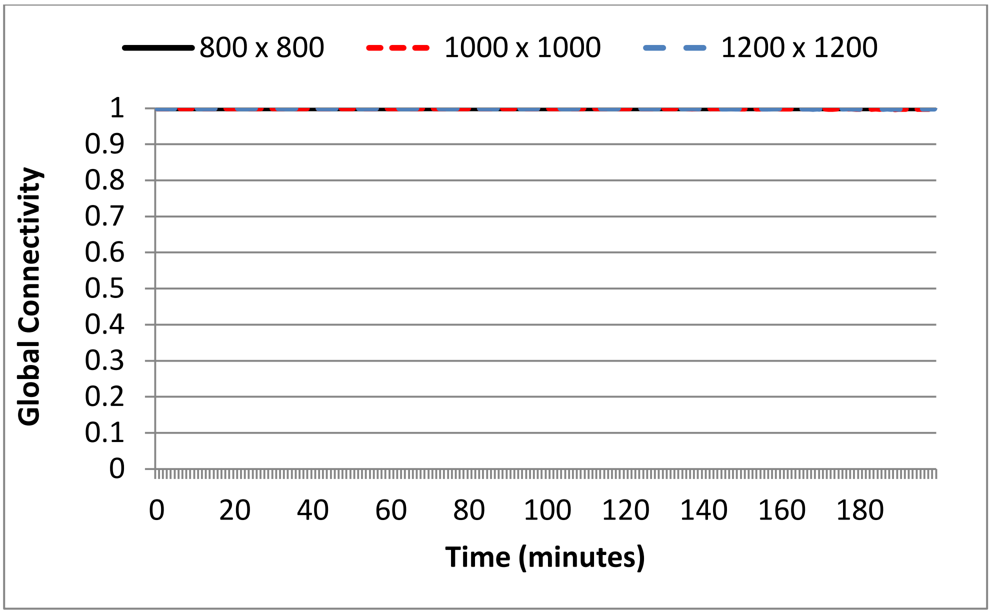


Figure 4.28 Global Connectivity in DKRU with Du's scheme for different network sizes and random walk mobility model

4.8.3. Local Connectivity Analysis

When we analyze local connectivity of our mechanism for different network sizes (Figure 4.29 and 4.30), we can see that local connectivity values converge to the same connectivity ratio regardless of the network size. In Figure 4.30, for 800 x 800 m^2 deployment area, local connectivity decreases more slowly compared to other network sizes. However, its convergence value is very close to the ones of 1000 x 1000 and 1200 x 1200 m^2 network sizes.

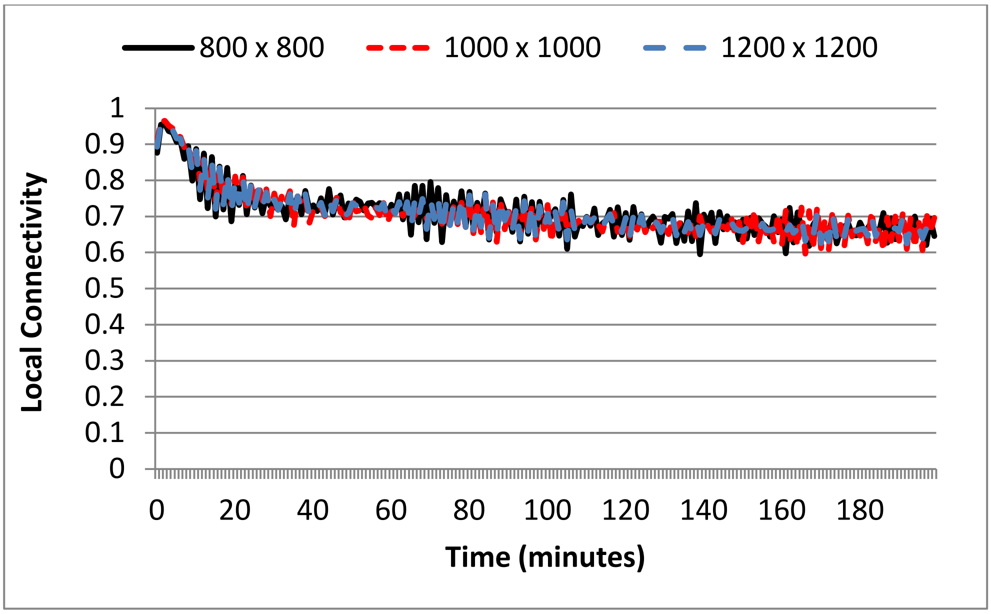


Figure 4.29 Local Connectivity in DKRU with Du's scheme for different network sizes and RPGM model

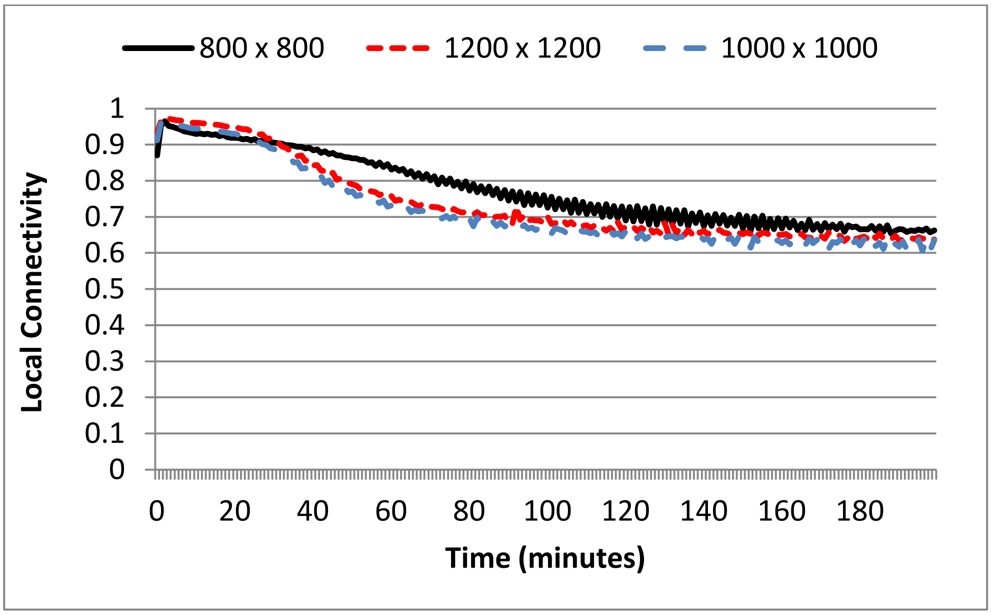


Figure 4.30 Local Connectivity in DKRU with Du's scheme for different network sizes and random walk mobility model

4.8.4. Resiliency Analysis

Figures 4.31 and 4.32 demonstrate that our mechanism provides almost the same level of resiliency for different network sizes. In other words, security of our mechanism is not weakened as the network gets larger. This flexibility against the increase in number of sensor nodes implies that our mechanism shows good scalability feature.

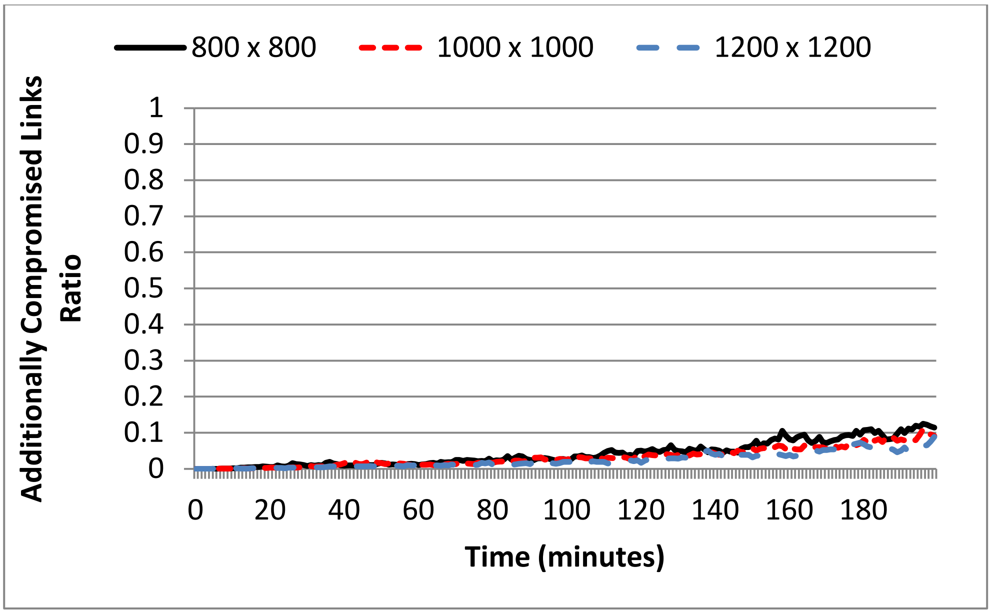


Figure 4.31 Additionally compromised links ratio in DKRU with Du's scheme for different network sizes and RPGM model

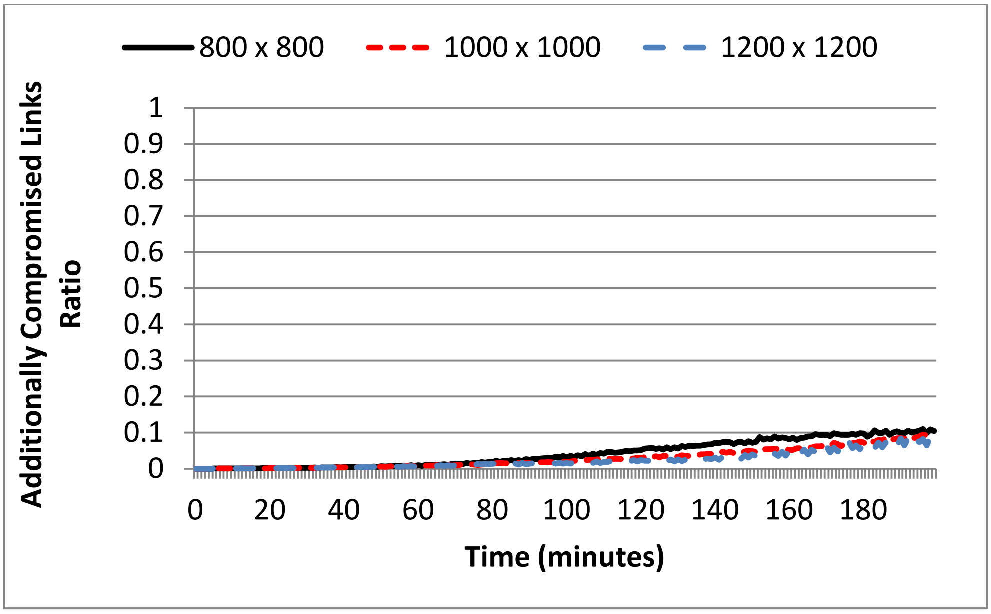


Figure 4.32 Additionally compromised links ratio in DKRU with Du's scheme for different network sizes and random walk mobility model

5. CONCLUSIONS

In this thesis, we proposed Dynamic Keyring Update (DKRU) mechanism for Mobile Wireless Sensor Networks (MWSNs). Our mechanism can be used together with different key predistribution schemes and it increases the local and global connectivity performance of these schemes. Using DKRU mechanism, a sensor node can update its keyring by observing the most frequent keys in its 1-hop and 2-hop neighbors' keyrings. Due to the mobile nature of network, neighbors of a node change continuously. Yet, DKRU mechanism helps sensor nodes to adapt to the network, regardless of their predeployment key distribution model.

We analyze performance of DKRU mechanism when it is used together with two different key predistribution schemes which are the basic scheme [3] and Du's scheme [6]. We also use two different mobility models for performance evaluation. Our results show that DKRU mechanism provides a significant increase to local and global connectivity of these key predistribution schemes in mobile case. We provide almost perfect global connectivity, which means that the global connectivity ratio is close to one in all cases. Moreover, local connectivity of DKRU-powered mechanisms is 40% higher than the local connectivity of original schemes. Another advantage of our mechanism is that, it does not increase connectivity at a high cost of resiliency and communication overhead. Additionally compromised links ratio of our mechanism is very close to original schemes and for some cases we provide better resiliency. Moreover, DKRU mechanism is scalable and brings about only a small amount of additional communication overhead.

6. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, 38(4) pp. 393–422, 2002.

[2] H. Chan, A. Perrig, D. Song. "Key distribution techniques for sensor networks," In Wireless sensor networks, C. S. Raghavendra, Krishna M. Sivalingam, and Taieb Znati (Eds.). Kluwer Academic Publishers, Norwell, MA, USA pp. 277-303, 2004.

[3] L. Eschenauer, V. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of the Ninth ACM Conference on Computer and Communications Security (CCS'02), ACM, New York, NY, USA, pp. 41–47, 2002.

[4] M. A. Simplício, Jr., M. Barreto, B. C. Margi, T. Carvalho, "A survey on key management mechanisms for distributed Wireless Sensor Networks." Comput. Netw. 54, 15 October 2010.

[5] S. A. Munir, R. Biao, J. Weiwei, W. Bin, X. Dongliang, M. Man, "Mobile wireless sensor network: Architecture and enabling technologies for ubiquitous computing." In Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on, vol.2, no., 21-23, pp.113-120, May 2007.

[6] W. Du, J. Deng, Y. Han, S. Chen, P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04), IEEE Computer Society, Los Alamitos, CA, USA, pp. 586–597, 2004.

[7] Y. Zhou, Y. Fang, Y. Zhang, "Securing wireless sensor networks: a survey," Communications Surveys & Tutorials, IEEE , vol.10, no.3, pp. 6-28, Third Quarter, 2008.

[8] Y. Wang, G. Attebury, B. Ramamurthy, "A survey of security issues in wireless sensor networks," Communications Surveys & Tutorials, IEEE , vol.8, no.2, pp.2,23, Second Quarter 2006.

[9] V. Rathod, M. Mehta, "Security in Wireless Sensor Network: A survey", Ganpat University Journal of Engineering & Technology, Vol.1 Issue 1, 2011.

[10] G. Gaubatz, J.-P. Kaps, E. Öztürk, B. Sunar, "State of art in ultra low-power public key cryptography for wireless sensor networks," In PerCom Workshops, pp. 146-150, 2005.

[11] F. Amin, A. H. Jahangir, H. Rasifard, "Analysis of public key cryptography for wireless sensor netwoks security." In PWASET ’08: Proceedings of World Academy of Science, Engineering and Technology, July 31, 2008.

[12] B. Arazi, I. Elhanany, O. Arazi, H. Qi "Revisiting public-key cryptography for wireless sensor networks." Computer, col. 38, no 11, pp. 103-105, 2005.

[13] A. R. Mishra, M. Singh, "Elliptic Curve Cryptography (ECC) for Security in wireless Sensor Network," International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 3, May 2012.

[14] W. Stallings, “Symmetric Ciphers,” in Cryptography and Network Security: Principles and Practice, 5th ed. Prentice Hall, 2010.

[15] X. Zhang, H.M. Heys, C. Li, "Energy efficiency of symmetric key cryptographic algorithms in wireless sensor networks," Communications (QBSC), 25th Biennial Symposium on , vol., no., pp.168,172, May 2010.

[16] W. K. Koo, H. Lee, Y. H. Kim and D. H. Lee, "Implementation and Analysis of New Lightweight Cryptographic Algorithm Suitable for Wireless Sensor Networks," in *Proc of 2008 Information Security and Assurance (ISA 2008),* pp.73-76, Korea, April 2008.

[17] C. Karlof, N. Sastry, and D. Wagner, “TinySec: a Link Layer Security Architecture for Wireless Sensor Networks,” in Proc of the 2nd international

Conference on Embedded Networked Sensor Systems (SenSys '04), pp. 162-175, New York, November 2004.

[18] S. A Çamtepe, B. Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: a Survey." Technical Report TR-05-07 Rensselaer Polytechnic Institute, Computer Science Department, March 2005.

[19] J.Zhang, V. Varadharajan, "Wireless sensor network key management survey and taxonomy". Journal of Network and Computer Applications, 2009.

[20] J. C. Lee, V. C. M. Leung, K. H. Wong, J. Cao, H. C. B Chan, "Key management issues in wireless sensor networks: current proposals and future developments." IEEE Wireless Communications, vol. 14, no. 5, pp. 76-84., 2007.

[21] Y. Xiao, V. K. Rayi, B.Sun, X.Du, F.Hu, M. Galloway, "A survey of key management schemes in wireless sensor networks." Comput. Commun. 30, 11-12 Sep. 2007.

[22] A. S. Reegan, E. Baburaj, "Key management schemes in Wireless Sensor Networks: A survey," Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on , vol., no., pp.813,820, 20-21 March 2013.

[23] B. Lai, S. Kim, I. Verbauwhede, "Scalable session key construction protocol for wireless sensor networks," in: IEEE Workshop on Large Scale Real-Time and Embedded Systems (LARTES), IEEE Computer Society, Washington, DC, USA, 2002.

[24] C. Haowen, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks," Security and Privacy, 2003 Symposium on , vol., no., pp. 197- 213, 11-14 May 2003.

[25] T. Shan, C. Liu, "Enhancing the key pre-distribution scheme on wireless sensor networks," in: IEEE Asia-Pacific Conference on Services Computing, IEEE Computer Society, Los Alamitos, CA, USA, pp. 1127–1131, 2008.

[26] S. Hussain, M. Rahman, L. Yang, "Key pre-distribution scheme using keyed-hash chain and multipath key reinforcement for wireless sensor networks," IEEE Computer Society, Los Alamitos, CA, USA, pp. 1–6, 2009.

[27] C.-F. Law, K.-S. Hung, Y.-K. Kwok, "A novel key redistribution scheme for wireless sensor networks," in: IEEE International Conference on Communications (ICC’07), IEEE Computer Society, Washington, DC, USA, pp. 3437–3442, 2007.

[28] L. Zhou, J. Ni, C. Ravishankar, "Efficient key establishment for groupbased wireless sensor deployments," in: Proceedings of the Fourth ACM workshop on Wireless security (WiSe’05), ACM, New York, NY, USA, pp. 1–10, 2005.

[29] D. Liu, P. Ning, "Location-based pairwise key establishments for static sensor networks," in Proceedings of the First ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN’03), ACM, New York, NY, USA, pp. 72–82, 2003.

[30] Z. Yu, Y. Guan, "A key management scheme using deployment knowledge for wireless sensor networks," IEEE Transactions on Parallel Distribution and Systems, vol. 19, no. 10, pp. 1411–1425, 2008.

[31] D. Huang, M. Mehta, D. Medhi, L. Harn, "Location-aware key management scheme for wireless sensor networks," 2nd ACM workshop on Security of Ad Hoc and Sensor Networks, 2004.

[32] K. Karaca, "A key distribution scheme tailored for mobile wireless sensor networks," Unpublished MS Thesis, Sabancı University, 2011.

[33] R. Blom, "An optimal class of symmetric key generation systems," in: Proceedings of the EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques, Springer, New York, NY, USA, pp. 335–338, 1985.

[34] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, "Perfectly secure key distribution for dynamic conferences," in: LNCS, vol. 740, Springer, New York, NY, USA, pp. 471–486, 1993.

[35] S. A. Çamtepe, B Yener, “Combinatorial design of key distribution mechanisms for wireless sensor networks,” IEEE/ACM Trans. Netw. 15, pp. 346-358, 2 April 2007.

[36] L. Zhou, J. Ni, C.V. Ravishankar, "Efficient key establishment for group-based wireless sensor deployments," in Proceedings of the 4th ACM Workshop on Wireless Security (Cologne, Germany, September 02 - 02, 2005). WiSe '05. ACM, New York, NY, pp. 1-10, 2005.

[37] A. Ünlü, A. Levi, "Two-tier, scalable and highly resilient key predistribution scheme for location-aware wireless sensor network deployments," Mob. Netw. Appl. 15, 4, pp. 517-529, August 2010.

[38] Q. Dong, D. Liu, "Using auxiliary sensors for pair-wise key establishment in WSN," in Proceedings of the 6th international IFIP-TC6 Conference on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation internet Atlanta, GA, USA, May 14 - 18, 2007.

[39] A. Kumar Das, "A Key Establishment Scheme for Mobile Wireless Sensor Networks Using Post-Deployment Knowledge," published in International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.4, July 2011.

[40] K. Karaca, A. Levi, "Resilient key establishment for mobile sensor networks," Distributed Computing in Sensor Systems and Workshops, International Conference on, pp. 1-6, 2011.

[41] G. Wang, G. Cao, T. Porta, W. Zhang, "Sensor relocation in mobile sensor networks." In: IEEE INFOCOM 2005.

[42] B. Liu, P. Brass, O. Dousse, P. Nain, D. Towsley, "Mobility improves coverage of sensor networks." In: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc), pp. 300–308, 2005.

[43] I. Amundson, X. D. Koutsoukos, "A Survey on Localization for Mobile Wireless Sensor Networks," Mobile Entity Localization and Tracking in GPS-less Environnments Lecture Notes in Computer Science Volume 5801, pp 235-254, 2009.

[44] J. Rezazadeh, M. Moradi, A. S. Ismail, "Mobile Wireless Sensor Networks Overview," IJCCN International Journal of Computer Communications and Networks, Volume 2, Issue 1, February 2012

[45] R.A. Pushpa, A. Vallimayil, V.R.S. Dhulipala, "Impact of mobility models on mobile sensor networks," Electronics Computer Technology (ICECT), 2011 3rd International Conference on , vol.4, no., pp.102,106, 8-10 April 2011.

[46] T. Camp, J. Boleng, V. Davies, "A survey of mobility models for ad hoc network research," Wireless Communications and Mobile Computing 2, pp. 483-502, 2002.