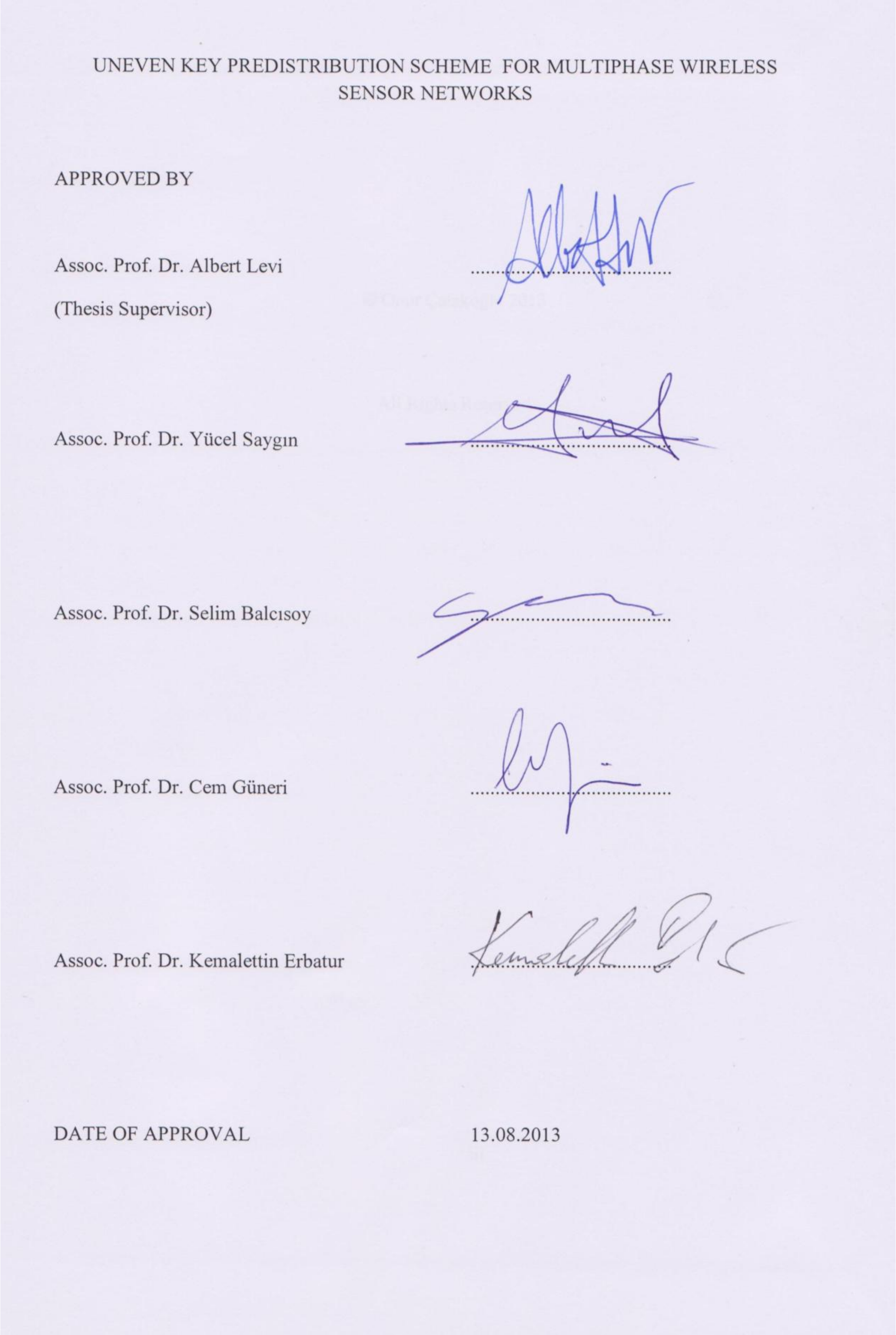
UNEVEN KEY PREDISTRIBUTION SCHEME FOR MULTIPHASE WIRELESS SENSOR NETWORKS

by

ONUR ÇATAKOĞLU

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



UNEVEN KEY PREDISTRIBUTION SCHEME FOR MULTIPHASE 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 ....

Assoc. Prof. Dr. Kemalettin Erbatur ....

DATE OF APPROVAL 13.08.2013

UNEVEN KEY PREDISTRIBUTION SCHEME FOR MULTIPHASE WIRELESS SENSOR NETWORKS

Onur Çatakoğlu
Computer Science and Engineering, MS Thesis, 2013
Thesis Supervisor: Assoc. Prof. Albert Levi

Keywords: Wireless Sensor Network, Key Predistribution, Multiphase Sensor Network, Resiliency, Security

Abstract

In multiphase Wireless Sensor Networks (WSNs), sensor nodes are redeployed periodically to replace nodes with depleted batteries. In order to keep the network resilient against node capture attacks across different deployment epochs, called *generations*, it is necessary to refresh the key pools from which cryptographic keys are distributed. In this thesis, we propose Uneven Key Predistribution (UKP) scheme that uses multiple different key pools at each generation. Keys are drawn unevenly from these key pools and loaded to sensor nodes prior to deployment. Nodes are loaded with keys not only from their current generation, but also from future generations. We conduct simulation based performance evaluation in mobile environments using three different mobility models. One of them, Circular Move Mobility model, is first proposed in this thesis. Our UKP scheme provides self healing that improves the resiliency of the network up to 50% under heavy attack as compared to an existing scheme in the literature. Moreover, our scheme provides almost perfect local and global connectivity.

ÇOK FAZLI TELSİZ DUYARGA DÜĞÜMÜ AĞLARI İÇİN EŞİTSİZ ÖN YÜKLEMELİ ANAHTAR DAĞITIM ŞEMASI

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

Anahtar Kelimeler: Telsiz Duyarga Ağları, Anahtar Ön Dağıtımı, Çoklu Fazlı Duyarga Ağları, Dayanıklılık, Güvenlik

Özet

Çok fazlı Telsiz Duyarga Ağlarında, duyarga düğümleri bataryaları tükenmiş düğümlerin yerine geçmek üzere periyodik olarak tekrar konuşlandırılır. Ağı, nesil adı verilen farklı konuşlandırma zaman aralıklarında düğüm ele geçirme saldırılarına karşı daha güçlü hale getirmek için kriptografik anahtarların dağıtımının yapıldığı anahtar havuzunu tazelemek gerekmektedir. Bu tezde, her nesil için farklı anahtar havuzları kullanan Eşitsiz Ön Yüklemeli Anahtar Dağıtım şeması anlatılmaktadır. Bu anahtar havuzlarından alınan farklı sayıda anahtarlar duyarga düğümlere konuşlandırılmanın öncesinde yüklenir. Düğümlerde yüklü olan anahtarlar, düğümün sadece kendi nesline değil, aynı zamanda gelecek nesillere ait anahtarlardan da oluşmaktadır. Simulasyonlarımızda, performans değerlendirmesini mobil ortamlar için üç tane mobilite modeli kullandık. Bunlardan bir tanesi olan Çembersel Hareket Mobilite modeli ilk olarak bu tezde sunulmaktadır. Eşitsiz Ön Yüklemeli Anahtar Dağıtım şeması literatürde bulunan bir şemaya göre daha ağın dayanıklılığını ağır saldırı altında %50'ye kadar arttıran bir öz iyileşme sağlamaktadır. Bunların yanı sıra, şemamızda yerel ve genel bağlantı oranı yaklaşık 100% olmaktadır.

To my family

Acknowledgements

Foremost, I would like to express my deepest gratitude to my advisor, Prof. Albert Levi for his patience, motivation and support throughout my graduate school career. He is one of the most big hearted and sincerest person I have ever known. Without his guidance and persistent help this thesis would not have been possible. One could not wish for a better or friendlier advisor.

I would like to thank my committee members: Prof. Yücel Saygın, Cem Güneri, Selim Balcısoy, Kemalettin Erbatur, for their support, comments and helpful suggestions.

I owe a very important debt to my research group, Merve Şahin and Salim Sarımurat. for working together before deadlines, and for all the fun and sharing we had in last two years. Their friendship and assistance has meant a lot to me that I could ever express. I could never complete my study without their help.

Special thanks also to Metallica, Iron Maiden, Jimi Hendrix, Queen, Aerosmith and Guns N' Roses for giving me the morale when I needed the most. These guys rock my world since my childhood and boost me for the hardest tasks.

I would also like to express my gratitude to TÜBİTAK (Scientific and Technological Research Council of Turkey) for providing scholarship for my graduate education and support this research project under grant 110E180. I also thank Sabancı University for offering me the tuition waiver scholarship. I am indebted to these foundations for supporting my education.

Last but not least, I would like to express the deepest appreciation to my parents Necati Çatakoğlu and Mahinur Çatakoğlu, and my sister Pınar Çatakoğlu. They always loved me and supported me unconditionally. I owe them everything from the start to the end.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are used to carry wide range of data for various kinds of applications such as military, security, smart homes, tele-health, environmental observation and industry automation. Information that is transferred via those networks may contain not only temperature readings for habitat monitoring but also classified military data for battlefield surveillance, which should not be seen by an unauthorized person. Therefore, security is important for these applications. WSNs have very limited resources in terms of memory and computational power. Hence, symmetric key cryptography is mostly used for existing key management schemes. However, predistribution of the symmetric keys effectively and efficiently in terms of resource usage have always been a challenge in WSNs.

An attacker can learn keys that are inside of any node by capturing the node and use these keys to compromise links between other sensor nodes. In Random Key Predistribution (RKP) scheme, Eschenauer and Gligor [1] attempt to solve this issue by distributing keys, which are drawn randomly from a collection of keys, called *key pool*, to sensor nodes. Since the same key pool is used for every node, an adversary who captures sensor nodes persistently, called *constant attacker*, would eventually learn the entire key pool of the corresponding sensor network.

Since WSNs are battery-powered systems, new nodes have to be redeployed periodically. In multiphase WSNs, sensor nodes with depleted batteries are replaced in time with periodical redeployment of the nodes. Castelluccia and Spognardi proposed RoK (A Robust Key Predistribution Protocol for Multiphase WSNs) scheme [5] for multiphase WSNs. In RoK, nodes' battery lives are divided into phases and they automatically *self heal* the network against node capture attacks by updating their keys at the end of each phase. Since the adversary has not captured newly updated keys yet, communications established by these keys cannot be compromised. Therefore, while RKP scheme cannot achieve such *self healing* mechanism, RoK scheme can establish a resilient network by utilizing the redeployment feature of WSNs. Yet, there is still room for enhancing the resiliency of the network in multiphase WSNs.

Most of the recent studies do not consider mobile environment. In other words, they assume that sensor nodes are static. However, this is not always correct, because there are many types of applications in commercial, environmental and military studies such as housekeeping robots, service industry, wildlife tracking, patient tracking, autonomous deployment, shooter detection [3] which require a mobile network.

1.1. Contribution of the Thesis

In this thesis, we propose a novel method for key predistribution in multiphase and mobile wireless sensor networks, called Uneven Key Predistribution (UKP). The main idea behind our method is to employ distinct key pools and assign keys to the nodes from these key pools by utilizing the temporal likelihood information of the ages of the nodes. In this way, the number of keys taken from different generations become uneven. At each deployment, newly deployed nodes take their keys not only from the existing key pools, but also from a new distinct key pool. As in RoK [5], in UKP scheme, the future generation keys that a node can know is limited to its maximum life. Keys in the network are renewed at each redeployment phase and, correspondingly, the adversary can never compromise entire key pool. This feature provides *self healing* to the network. Differently from the RoK scheme, UKP uses multiple distinct key pools to refresh keys instead of using forward and backward hash operations for the sake of resiliency. In our scheme, hash operation is used only for creating a session key between two nodes from common keys not in key pool generations. Thus, cryptographic overhead is minimal.

In performance evaluation, we consider three models; *Random Walk Mobility model*, *Reference Point Group Mobility model* and *Circular Move Mobility model*. Among these, Circular Move Mobility is a novel model that we propose in this thesis. For the performance evaluation, we compare our UKP scheme with RoK [5]. Since RoK is proposed for static WSNs, we adopted it to work with mobility models. Our results show that we have better resiliency than RoK scheme without decreasing the local connectivity of network and without adding any additional memory overhead.

1.2. Organization of the Thesis

The rest of the thesis is organized as follows. Section 2 gives the background information on WSN security. Section 3 provides a details our scheme and explains it in more detail. Section 4 presents the mobility models and Section 5 gives performance evaluation of UKP. Finally, Section 6 concludes the thesis.

2. BACKGROUND INFORMATION

A *sensor node* can sense various type of phenomenon including the occurrence of events such as temperature drop or pressure. [23]. A *wireless sensor node* can communicate airborne to transfer the collected data to another sensor node. Wireless Sensor Networks (WSNs) consist of large collection of sensor nodes which senses and delivers information via a short-range wireless communication. They can sense and process wide range of data including humidity, temperature, vehicular movement, lightning condition, pressure, etc. for various kinds of applications such as military, security, smart homes, tele-health, environmental observation and industrial automation [16]. Sensor nodes do not only communicate with each other, but also communicate with *base stations*. The duty of the base station is to manage the network and collect the data that is gathered from environment. Base station can perform costly operations for the network and store considerable amount of data.

Sensor nodes that are deployed to a certain area transmit information by communicating each other in multihop manner. The information they carry finally reaches to a sink node (a.k.a. a base station). When an event is detected by a sensor node, it creates a corresponding data packet. Then, this packet is transmitted to the sink node(s) possibly via intermediate nodes. Illustration of this process is given in Figure 2.1. The sensed information can be aggregated and processed along the way by the nodes. Also, they can store the data which is sensed from the environment or received from another node. Additionally, sensor nodes can have supporting technologies, such as Global Positioning System (GPS), to determine their current location or their final destination [23].

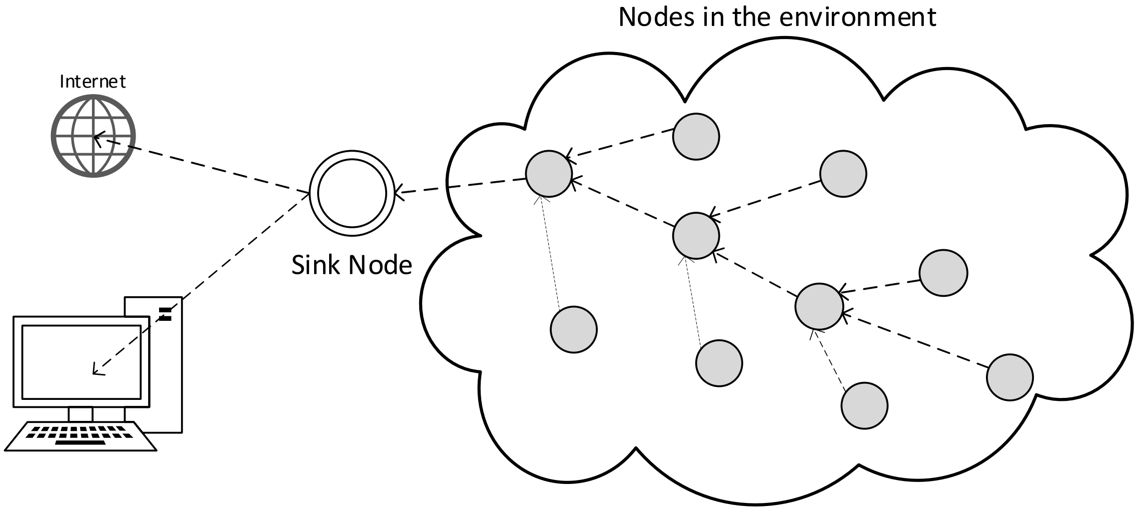


Fig. 2.1 Communication between nodes and sink node

The disadvantage of WSNs is their limited resources. These limitations do not allow them to have advanced technologies. Their computational power is insufficient to carry out costly tasks since embedded processors in nodes are not powerful as in wired networks. Further, they are not designed to store excessive data because of their inadequate memory capacity. Their memory usually consist of flash memory and RAM in order to store application code, sensed data and intermediate computations. Also, their communication range is limited and it is mostly dependent on environmental factors. These limitations are partly due to the limited energy and physical size of the sensor nodes [19]. WSNs are battery operated systems. Battery of a sensor node usually cannot be replaced and node becomes unusable after the depletion. Nodes will eventually stop functioning and become unable to send and receive messages. This leads to lack of connectivity in the network. Thus, nodes are redeployed periodically to replace nodes with depleted batteries. This kind of sensor networks are called *multiphase* WSNs.

Sensor nodes may be static or mobile depending on the application. Most of the recent studies do not consider mobile environments and assume that sensor nodes are static. However, it is not always correct, because there are many types of applications in commercial, environmental and military studies such as housekeeping robots, service industry, wildlife tracking, patient tracking, autonomous deployment, shooter detection [3]. All of these applications require mobility of nodes. Also, mobile nodes can improve the performance of the network in terms of energy efficiency, throughput and connectivity with small impact to data routing and end-to-end latency [20].

2.1. Security of Wireless Sensor Networks

Information that is transferred via WSNs may contain not only temperature readings for habitat monitoring, but also classified military data for battlefield surveillance which should not be observed by an unauthorized person. Since WSNs are deployed to an open and unattended field, they are vulnerable to many types of attacks. It is harder to detect an intrusion, capturing or corruption in the network compared to wired networks as the communication medium is air or underwater in some cases. Therefore, security is very important in WSNs.

Security requirements of WSNs are listed as follows [15] [19].

- *Confidentiality:* This is a security service that provides secrecy for transmitted data between two nodes. Nodes encrypt critical information before the transmission. Receiving node decrypts this information after the data is fully transmitted. An attacker should not be able to decrypt this information even if he monitors the entire communication. Confidentiality is a requirement against these attacks.

- *Authenticity:* This service is used to prevent unauthorized access. Nodes check the identity of each other to decide if the message comes from a real sender or not. An attacker can spoof or imitate identities of nodes in the network in order to obtain sensitive data or corrupt the network by spreading false information. Authenticity is used to prevent this kind of attacks.

- *Integrity:* Attacker can modify data by changing some of the bits of the message or changing it completely. Integrity feature ensures that message transmitted between two nodes is not modified by an malicious person.

In order to establish a secure communication, there is need for encrypted links and/or authenticated nodes. Several existing security protocols that are used in wireless networks may not be suitable for WSNs. Public key (asymmetric key) cryptographic algorithms such as RSA [17] and Diffie-Hellman [18] are inapplicable considering nodes' inadequate computational power for costly encryption and decryption operations. In order to fulfill the requirement for secure communication, symmetric key cryptography is the optimal solution to cover the limitations of WSNs. AES and DES [21] are some of the well known and standardized algorithms for symmetric key cryptography. There are also other lightweight symmetric key algorithms proposed for WSNs [22].

In symmetric key cryptography, encryption and decryption operations are performed using a single key. Hence, sender and receiver parties should be supplied with the same key in order to form a proper secure communication. This shared key between communication parties should be secret, because if an adversary learns this key, he also gains the encryption/decryption capabilities in that network. As a result, all the communication links which use this shared key become compromised.

Although using symmetric key cryptography meets most of the constraints of WSNs, it brings up another problem which is *key distribution*. Sensor nodes should be able to transfer data to intermediate nodes which will deliver the information when a sink node is not available. Owing the fact that classified data should not be monitored along the way, these paths are required to be secure. If two nodes share a secret key, they can establish a secure communication. However, distribution of the symmetric keys have always been a challenge in WSNs due to the limitations of sensor nodes. An obvious solution to key distribution problem is to predistribute the keys to sensor nodes before deployment. One method is to load a single key to all sensor nodes and using this key in all communication links. However, if all nodes are loaded with the same key, the security of network relies on this single key. In this case, when a sensor node is captured by an attacker, the communication key is revealed and attacker can compromise the entire network. Another method is to generate pairwise keys for all node pairs. Then, each node is loaded with its pairwise keys for all other nodes in network. In this case, even if an attacker captures a sensor node and learns all its keys, he cannot use these keys to compromise the communication links between other nodes. Yet, this method leads to very high memory consumption because a sensor node needs to keep all its pairwise keys in memory. Considering the huge number of sensor nodes in a WSN, this method seems to be infeasible in practice. The key distribution problem in WSNs is widely studied in the literature. More detailed explanation about this topic will be given in Section 2.2.

2.1. Hash Functions

Hash functions are used to generate fixed-length fingerprints of arbitrarily large data. Output of the hash function is denoted as $H(M)$, where M is the message of variable length and $H(.)$ is the hash function. The calculated $H(M)$ has fixed length for any message M. In Figure 2.2, this process is illustrated. A hash function is a one-way function. In other words, for a given $h = H(M)$, it should be practically infeasible to compute message M.

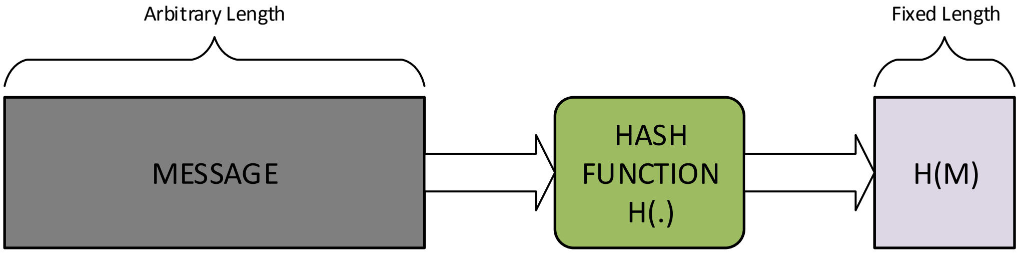


Fig. 2.2 Illustration of hashing process

One important requirement for secure (cryptographic) hash functions is collision-resistance. Since hash functions map various length of data to a fixed size data, there is a possibility of two input values give the same output value. This situation is called *collision*. In other words, collision is having the same hash value, $H(x) = H(x')$ for two distinct pieces of data, x and x'. This is an unwanted situation, because an adversary can intentionally search for collisions. Hence, a *collision resistant* hash function, where it is infeasible to find any pair of inputs sharing the same hash value, is desired in WSNs. Reader can refer [24] for detailed information.

2.2. Related Works

Because public key cryptography is a very costly option for WSNs in terms of computational power and memory consumption, most of the studies use symmetric key cryptography to secure WSNs. In symmetric key cryptography, two nodes must have the same secret key, in order to establish a secure communication. Distribution of these secret keys to a node is difficult after node deployment because environment may be monitored by an attacker. Thus, keys are needed to be preloaded to nodes properly before deployment.

Random Key Predistribution (RKP) is the most popular scheme that is proposed by Eschenauer and Gligor [1]. It is a basis to many existing schemes in wireless sensor networks. RKP has three phases: key setup, shared key discovery and path key establishment.

Key setup: In this phase, each node receives a group of keys, called key chains, from a large pool, *P*. Each key in the key pool has a unique ID for key discovery. Key chains are loaded into nodes prior to deployment.

Shared key discovery: In the field, sensor nodes try to securely communicate with their neighbors if they share a secret common key. If two nodes have the key(s) with the same ID(s), then they can securely communicate with each other by encrypting the data with same keys. Established link is said to be a direct secure link.

Path key establishment: When two neighboring nodes do not have a secret key in common, they can look for a common intermediate node with where both share a secret key. With the help of common secure neighbor, they can establish a secure link. This phase called path key establishment.

Figure 2.3 gives the illustration for key pool and key chain for RKP scheme. If length of the key chain, *m*, and size of the key pool, *P*, are chosen properly, a resilient network can be achieved whilst maintaining a fair network connectivity. However, a constant attacker eventually learns all the keys in the key pool in this scheme and he can compromise the entire network [5].

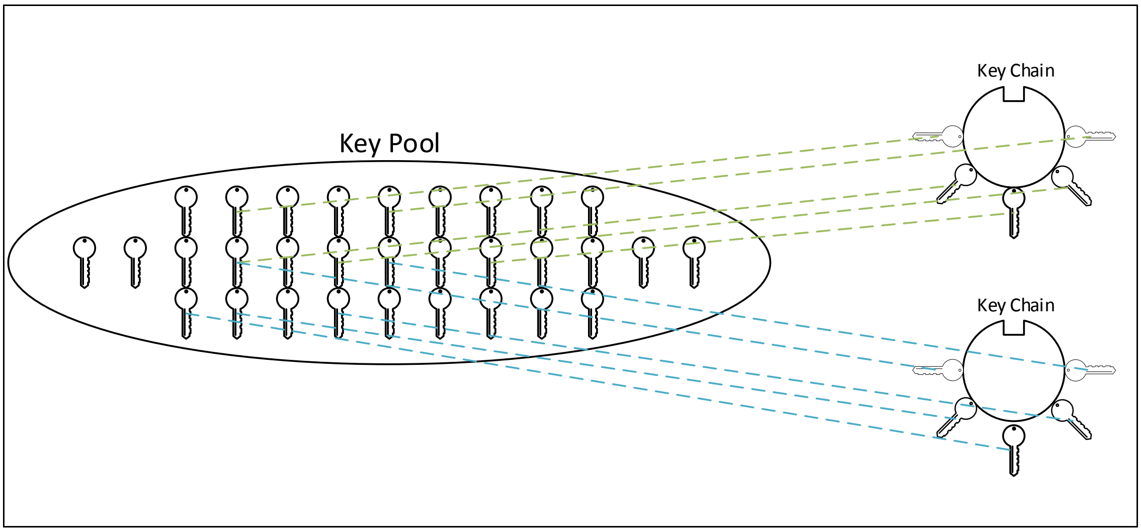


Fig. 2.3 Key Pool and Key Chains for RKP

Several other probabilistic schemes are proposed in [2, 11, 12, 13] after RKP scheme. Among them, Chan et al. [2] improved RKP scheme by using a threshold value, $q > 1$, for the number of common keys that are needed for establishing a secure connection. If two nodes meet this requirement, they hash their all common keys to create a *session key.* Session key is used to encrypt the communication between these nodes. Yet it requires more keys to be stored before the deployment or fewer keys in the key pool to achieve a good connectivity ratio. Increase in number of stored keys results in an additional memory overhead.

Since sensor nodes are battery operated systems, they have to be redeployed periodically for the sake of connectivity of the network. These new nodes are assumed to be deployed at regular epochs which are called *generations*. Also, lifetime of a node is assumed to have an upper bound and it is determined by *generation window*, G_w. A newly deployed sensor node's battery at generation j will deplete before generation $j + G_w$. In the RoK scheme [5], key pools evolve for each new generation and sensors update their key rings by hashing their keys. In other words, keys have lifetimes and they are refreshed when a new generation is deployed. While they are limiting the lifetime of predistributed keys, they achieved to maintain high connectivity. This mechanism is achieved by using forward and backward hash chains. In every redeployment phase, previously deployed nodes hash their keys and new nodes with fresh keys are replaced with nodes whose batteries are empty. Each sensor node takes its keys from both forward and backward key pools, FKP and BKP, that are associated to its generation. Each key pool has $P/2$ random keys.

Notation used for RoK is explained in Table 2.1. This notation also will be used for explanation of our proposed scheme Uneven Key Predistribution (UKP) scheme in Section 3.

Table 2-1 Symbols used for RoK and UKP

Symbol	Explanation
A	Sensor A
n	Last generation of the network
FKP^j	Forward key pool at gen.
BKP^j	Backward key pool at gen.
P^j	Key pool of gen. j
m^j	Number of keys that are taken from key pool j
p_A^j	Number of keys that are taken from key pool j for node A
P	Key pool size
FKR_A^j	Forward key ring of A at gen.
BKR_A^j	Backward key ring of A at gen.
KR_A^j	Key ring of A that deployed at gen. j
G_w	Generation window
fk_t^j	t-th forward key at gen. j
bk_t^j	t-th backward key at gen. j
$k_{t_u}^j$	t_u -th key of P^j
k_{AB}	Common secret key between sensor A and B
$H(.)$	Secure hash function
m	Key ring size

In the RoK scheme [5], key pools evolve for each new generation and sensors update their key rings by hashing their keys. In other words, keys have lifetimes and they are refreshed when a new generation is deployed. While they are limiting the lifetime of predistributed keys, they achieved to maintain high connectivity. This mechanism is achieved by using forward and backward hash chains. In every redeployment phase, previously deployed nodes hash their keys and new nodes with fresh keys are replaced with nodes whose batteries are empty. Each sensor node takes its keys from both forward and backward key pools, FKP and BKP, that are associated to its generation. Each key pool has $P/2$ random keys.

Forward key pool at generation j defined as $FKP^{j} = \{fk_1^j, fk_2^j, \dots fk_{P/2}^j\}$ where $fk_t^{j+1} = H(fk_t^j)$. Similiarly backward key pool at generation j will be $BKP^j = \{bk_1^j, bk_2^j, \dots bk_{P/2}^j\}$ where $bk_t^j = H(bk_t^{j+1})$. While a node at generation j takes its forward keys from FKP^j, it takes its backward keys from BKP^{j+G_w-1}. Therefore, key rings of the node will be formally represented as;

$FKR_A^j = \{fk_u^j | u = h(id_A \| i \| j), i = 1,2,\dots,{}^{m}/_{2}\}$ and

$BKR_A^j = \{bk_u^{j+G_w-1} | u = h(id_A \| i \| j), i = 1,2,\dots,{}^{m}/_{2}\}$

for forward and backward key ring respectively.

A sensor B deployed at generation i in the range of $[j - G_w, j + G_w]$ communicates with sensor A while their common keys' indices are $t_1, t_2, \dots, t_z$ respectively as follows.

while $i \leq j$,

$$k_{AB} = H(fk_{t_1}^j \| bk_{t_1}^{i+G_w-1} \| fk_{t_2}^j \| bk_{t_2}^{i+G_w-1} \| \dots fk_{t_z}^j \| bk_{t_z}^{i+G_w-1})$$

If two neighboring nodes have multiple shared common keys, all of them are used for the session key, k_{AB}. An adversary cannot compute keys from past generations by using forward keys, and cannot compute keys from future generations by using backward keys. Since all of the common keys including both forward and backward keys are used for establishing a secure channel, this mechanism provides forward and backward secrecy.

There are some other works inspired by RoK that focus on multiphase networks. RPoK [6] is a polynomial-based RKP scheme proposed by Ito et al. for multiphase WSNs. Using private subkey that is indirectly stored into every each node, they are able to establish a resilient network. Yi et al. [4] proposed a hash chain based scheme (HM scheme) for multiphase WSNs by using different key matrixes for every phase which is the separated work time of the nodes.

Moreover, there are some studies focusing on the mobility of the WSN. Tas et al. [8] proposed Mobile Assisted Key Distribution in Wireless Sensor Network. In this paper, keys are distributed by a mobile element that handles the overload of key distribution. In their proposed schemes, mobile robot broadcasts key material to sensor nodes and all nodes in the radius of the robot can receive the keys. They discuss and evaluate feasibility of their schemes by simulation. Another scheme, proposed by Das [9], utilizes post deployment knowledge in mobile sensor networks. In this work, nodes are assumed to know their current coordinates as they move. They assign location information to each keying materials and keys are given priority when the distance between node's current location and post deployment location that is stored in the key is smaller. Nodes in that location can establish a secure communication link by using these prioritized key. Karaca et al. [10] used mobile base stations to distribute keys to sensor nodes. Nodes have only keys to communicate with this base station at first. When two nodes want to establish a secure communication, they ask for a session key from the base station.

3. UNEVEN KEY PREDISTRIBUTION SCHEME

In this section we present our proposed Uneven Key Predistribution (UKP) scheme for mobile multiphase wireless sensors.

Communication between generations is a must; because every non-communicating sensor node pair will lead to decrease in connectivity. In RoK [5], keys are updated and refreshed at the end of each phase. Therefore, two nodes which are from different generations can establish a secure channel with this update mechanism.

UKP follows a different mechanism for that purpose. It is based on average age of nodes in the network i.e. keys are predistributed to a node according to its life time. Every sensor node from generation j can communicate with another sensor node from different generation in the range of $(j - G_w, j + G_w)$ as in the RoK. However in UKP, instead of taking m number of keys from a key pool, a node takes its keys from G_w number of key pools. In other words, our scheme predistributes the keys not just from the key pool of the current generation, but also from key pools of future generations.

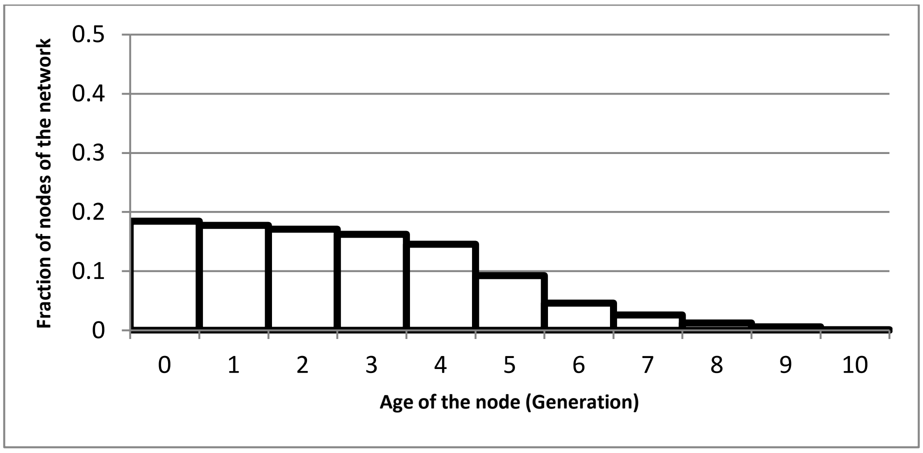


Fig. 3.1 Distribution of average age of the nodes.

The main idea of UKP is to distribute keys considering nodes' average age distribution. Our sensor node life time modeling, which is also used in RoK [5], is a probabilistic one based on Gaussian (normal) distribution with mean $\frac{G_w}{2}$ and standard deviation $\frac{G_w}{6}$ where G_w is set to 10. Figure 3.1 shows average age distribution of the sensor nodes. As can be seen from this figure, most of the nodes in the network are newly deployed or young. Starting with age 5, the number of old nodes decreases significantly. Since the number of younger nodes is significantly larger than the old ones, we distribute more keys from the key pools closer to a node's generation than the older ones with the aim of increasing connectivity. These details will be elaborated in the next subsection.

3.1. Pools and Key Assignments

In UKP, there are n distinct pools that are not associated with each other. A sensor node takes its keys from G_w number of consecutive key pools in terms of generations. In order to decide how many keys to pick from a particular key pool, we use the distribution shown in Figure 3.1. In that case, a sensor has the most keys from its own generation key pool and takes fewer keys from key pools that belong to further generations.

More formally, a node at generation j takes its keys from $P^j, P^{j+1} \ldots P^{j+G_w-1}$. The number of the keys taken from these generations are denoted as $m^j, m^{j+1} \ldots m^{j+G_w-1}$ where $m^j > m^{j+1} > \cdots > m^{j+G_w-1}$. Actual m values that we use in our UKP scheme are given in Table 3.1. In our UKP scheme, a node deployed at generation j, may have common keys with the nodes deployed in the generation range $[j - G_w + 1, j + G_w - 1]$. Thus, if a j^{th} generation node is captured, only the nodes deployed at generation in the abovementioned range are affected. This provides forward and backward secrecy.

Table 3-1 Actual m percentages of key size for UKP where G_w set to 10

m^j	m^{j+1}	m^{j+2}	m^{j+3}	m^{j+4}	m^{j+5}	m^{j+6}	m^{j+7}	m^{j+8}	m^{j+9}
19,1%	18,8%	18%	16%	13%	8%	%4	2%	~1%	0,01%

The key ring of node A, which is denoted as KR_A^j is composed of all the key sets coming from different generations of key pools as stated above. Similarly, if node B is at generation $j + G_w$, key ring of node B which is denoted as $KR_B^{j+G_w}$ will take its keys from key pools P^{j+G_w}, P^{j+G_w+1} ... P^{j+2G_w-1}. Node B does not have any keys from P^j, P^{j+1} ... P^{j+G_w-1} key pools. In other words, if there is at least G_w number of generations difference between two nodes, these two nodes do not share any common keys. In this way, we provide self healing, because compromised keys become outdated in time.

We represent key ring of a node A at generation j as follows.

$$KR_A^j = \begin{cases} k_{t_u}^j \mid u = 1,2, \dots m^j \\ k_{t_u}^{j+1} \mid u = 1,2, \dots m^{j+1} \\ \dots \\ k_{t_u}^{j+G_w-1} \mid u = 1,2, \dots m^{j+G_w-1} \end{cases}$$

where, $\mathrm{k}_{\mathrm{t_u}}^{\mathrm{i}}, i = j \dots j + G_w - 1$, are the keys selected from corresponding P^i using uniform random distribution with replacement.

The size of the key ring produced in this way, m, is calculated as follows.

$$m = m^j + m^{j+1} \dots + m^{j+G_w-1}$$

The purpose of having an uneven key distribution, i.e., using more keys from closer key pools in terms of generation is to achieve higher local connectivity in network. Moreover, this will strengthen the *self healing* property, since a compromised key has less chance of existence in further generations. In other words, most of the keys will be outdated sooner than the remaining ones and resiliency will be enhanced by the arrival of the new nodes with fresh keys. Each key in the key pools has a unique ID for key discovery similar to RKP [1] scheme. Sensor nodes broadcasts IDs of its keys and when two neighboring nodes share at least one common key, session key establishment phase starts.

3.2. Session Key Establishment

Any two nodes, say node A and node B, can establish a session key only if they share at least one common key in their key rings. The session key is computed as the hash of all common keys that nodes A and B share. This key is denoted as k_{AB}. The common keys used in session key establishment are chosen irrespective of the generations of keys. In other words, even if the two nodes come from different generations, they use all of the keys in their key rings to find common keys for session key establishment. Let us say that node A comes from generation j, node B comes from generation i and the condition $i \leq j$ holds for node generations. Then, if the common keys A and B share are denoted as

$$CK_{AB}^{x} = \{\forall\, k_v^x \,\middle|\, k_v^x \in KR_A^j\,, k_v^x \in KR_B^i \; and \; k_v^x \in P^x \; for \; i \leq j \leq x\}$$

then, the session key is computed as follows.

$$k_{AB} = H\{CK_{AB}^{x} \,\middle|\, x = j \ldots i + G_w - 1\}$$

As an example, if node A comes from generation j, node B comes from generation $j-2$ as shown in Fig. 2, and the set of common keys they share are $k_{t_1}^{j}$, $k_{t_2}^{j}$, $k_{t_1}^{j+1}$, $k_{t_2}^{j+1}$, $k_{t_9}^{j+4}$, $k_{t_3}^{j+5}$ and $k_{t_{11}}^{j+7}$, the session key is computed as follows.

$$k_{AB} = H(k_{t_1}^{j} \| k_{t_2}^{j} \| k_{t_1}^{j+1} \| k_{t_2}^{j+1} \| k_{t_9}^{j+4} \| k_{t_3}^{j+5} \| k_{t_{11}}^{j+7})$$

Again, the common keys coming from different key pools $(P^j, P^{j+1}, P^{j+4}, P^{j+5}, P^{j+7})$ and consequently different generations are used together to form the session key.

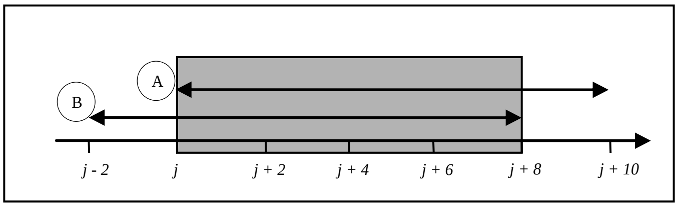


Fig. 3.2 Generation windows and overlapping generations of nodes A and B.

4. MOBILITY MODELS

In order to simulate node mobility, we used three models: (a) Random Walk Mobility model, (b) Reference Point Group Mobility model and Circular Move Mobility model (c). Among these, Circular Move Mobility model is a new one that we propose together with our research group. These mobility models are explained below.

4.1. Random Walk Mobility Model

In this model, a sensor node chooses a direction and speed randomly using uniform distribution. Then it moves in that direction for a fixed amount of time, which is taken as one minute in our simulations. When it finishes its movement, this process repeats itself with new direction and speed. A node which reaches the boundary of simulation area is reflected back with the same angle. Past location and speed information are not stored, so no memory usage is needed. Therefore, this model is suitable for the sensor nodes. In Figure 4.1, movements of the 5 nodes are illustrated for Random Walk Mobility model. The reader can refer to [7] for more detailed information about Random Walk Mobility model.

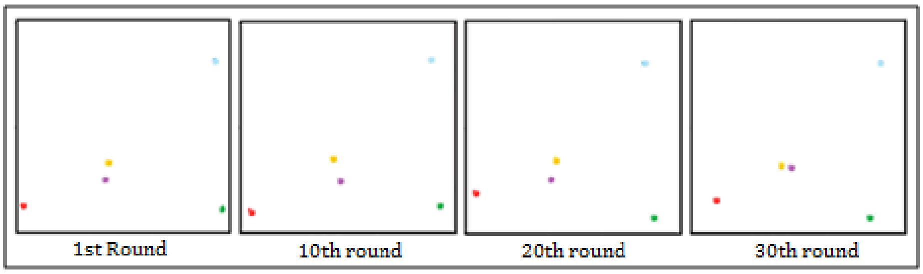


Fig. 4.1 Illustration of Random Walk Mobility model.

4.2. Reference Point Group Mobility Model

This model covers both groups' random movement and random movement of individual nodes inside a group. Each group moves based on a node that is chosen as central node. This feature is provided with reference points. Individual nodes pick a reference point randomly around the central node and this reference points are updated with the movement of central node. Individual nodes moves around the central node with minor randomness. In Figure 4.2, movements of the 3 groups of nodes are illustrated for Reference Point Group Mobility model. The reader can refer to [7] for more detailed information about Reference Point Group Mobility model.

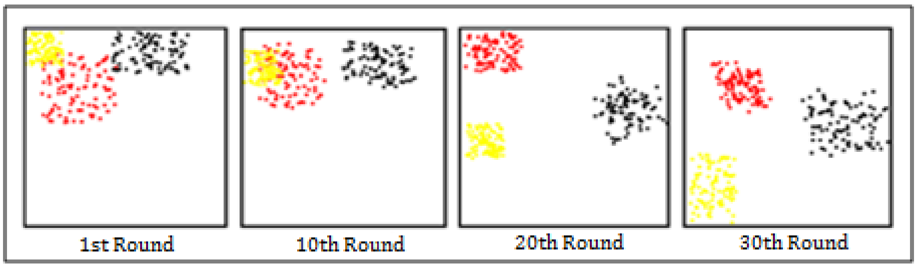


Fig. 4.2 Illustration of Reference Point Group Mobility model.

4.3. Circular Move Mobility Model

A novel mobility model called Circular Move Mobility is presented in this section. In this model, nodes are deployed around the perimeter of a circular area and move towards the center of this circle as time passes. In this way, the area is fully covered in time. Hence, we end up with a mobility model to cover a 2D area with a simple one-dimensional deployment strategy. This is, actually, one of our motivations in developing this model. We assume that a vehicle circles around the area and deploys nodes from eight different points on the perimeter. After deploying nodes on the perimeter, they start to scan the area as they move towards to the center. Nodes, whose batteries are depleted, can be collected easily since they cluster around the center of the deployment area. In this way, dead nodes can easily be removed out of the field. This is our second motivation behind this new model. There are two phases of this model: (i) deployment phase, and (ii) movement phase.

In deployment phase, nodes are deployed from eight different and equidistant points on circle that we call *bunch points* from now on as in the Figure 4.3. We assume that nodes deployed from these points will spread along and off the arc. Nodes are distributed through the arc of the circle, $\widehat{AOB}$, according to Gaussian distribution. Similarly, nodes are distributed off the arc through a line, l, which is congruent to the radius of the circle. The distance between this point and the center of the circle is decided according to the Gaussian distribution.

Majority of nodes are expected at around bunch point and fewer nodes are expected to be closer to the arc as distance from the bunch point increases. Hence, nodes cluster in certain area called *density ellipse.* In Figure 4.3, this area is illustrated. In that area, node population should be higher due to the Gaussian distribution model.

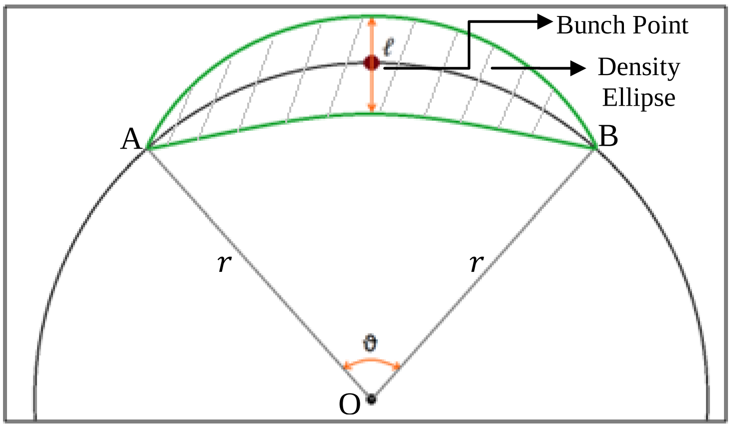


Fig. 4.3 Illustration of Density Ellipse.

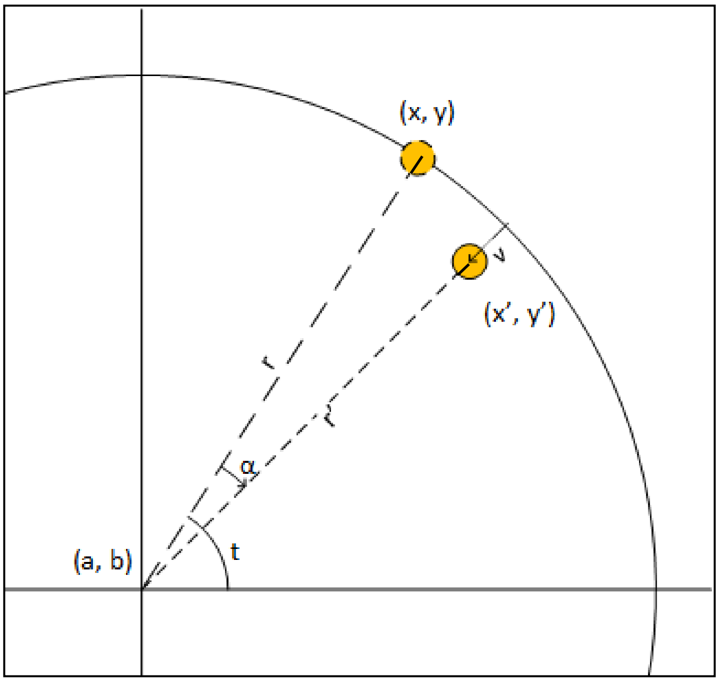


Fig. 4.4 Movement of a sensor node in one round.

In movement phase, nodes start moving towards to the center after they are deployed. They do not stop until they reach the center or their batteries are depleted. Nodes choose a speed vector which is directed at center and congruent to the radius. We refer to this speed vector as *linear speed* from now on. Linear speed is decided using uniform random distribution between a maximum and a minimum value at each round. While moving in the direction of center, nodes have an angular displacement at each round. The rate of change of this displacement is defined as *angular speed*. If a node's maximum angular speed is given as Ă, an angular speed, $\overline{\alpha}$, is decided using uniform random distribution within the range of $[-$Ă, Ă$]$. Hence, node can move clockwise or counter-clockwise according to the $\overline{\alpha}$ speed vector.

As an example consider a node with initial point (x, y) is

$$x = a + r\cos t,$$

$$y = b + r\sin t$$

where (a, b) is the center coordinates, r is the radius of the circle and t is initial the angle as depicted in the Figure 4.4. If the linear speed of the node is $\bar{v}$ per round, then new distance to the center in terms of r and $\bar{v}$ becomes $r' = r - |\bar{v}|$ for the next round where $|\bar{v}|$ is the magnitude of the vector $\bar{v}$. If the angular speed is $\overline{\alpha}$ per round which is chosen randomly between $[-$Ă, Ă$]$, then new coordinates of the node is calculated as

$$x' = a + r'\cos(t - |\overline{\alpha}|),$$

$$y' = b + r'\sin(t - |\overline{\alpha}|)$$

where $|\overline{\alpha}|$ is the magnitude of the vector $\overline{\alpha}$.

As a result, node moved from its initial point, (x, y), to a new point, (x', y') with two random variables, v and α. This process is repeated by starting (x', y') to a new coordinate using two random variables generated for the next round. Movement of the nodes continue until they get close to the center or their batteries are depleted. Note that, nodes stop at a certain distance from the center before reaching it as if there is a boundary. The distance is determined by a threshold value which is 1% of the radius.

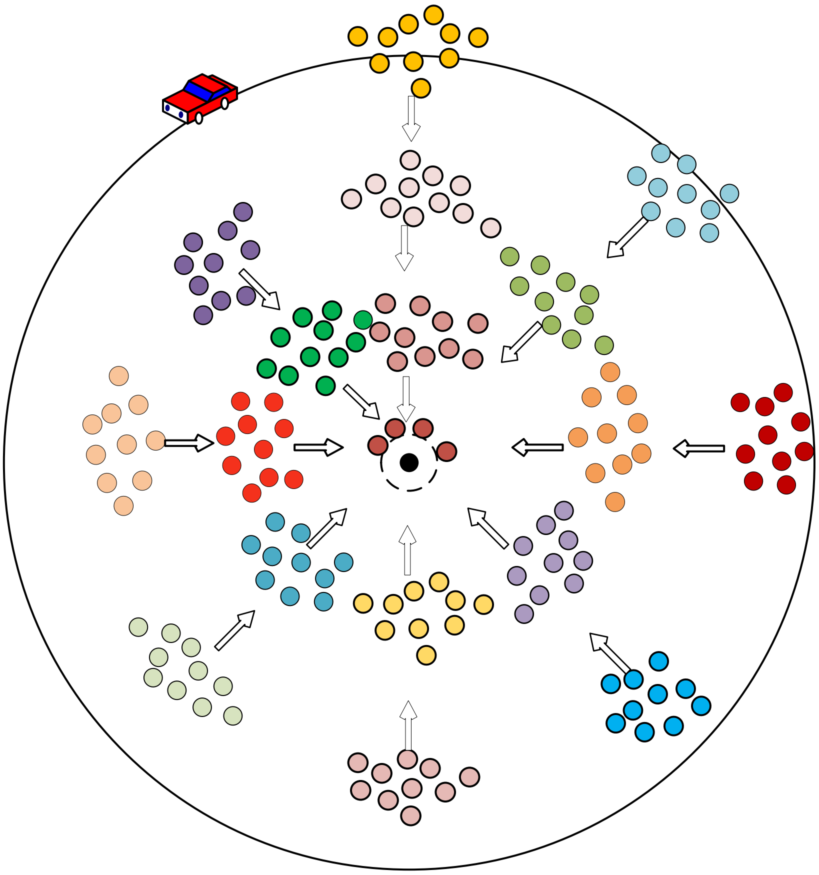


Fig. 4.5 Movement of sensor nodes from the perimeter to the circle as groups.

Figure 4.5 shows a simplified version of deployment and movement phases. As it is stated before, we assume a vehicle circles around to deploys nodes. After deployment of one group is finished, next deployment point is chosen as the closest bunch point on the way. Since nodes starts moving right after they are deployed, movement of the nodes looks like a spiral as shown in the Figure 4.5.

Algorithms that are used for deployment phase and movement phase is explained in pseudocode in Figure 4.6 and Figure 4.7 respectively.

```
/*  Terminology
    sda=standard deviation of angular displacement
    sdl=standard deviation of linear displacement
    a=x coordinate of the center point
    b=y coordinate of the center point
    GaussianDistribution(mean, std.dev.)= Generates a random
        number following Gaussian distribution for given
        mean and standard deviation
*/
SET angle to 0
FOR each generation
  FOR each deployment point on circle (total 8)
      FOR each node that is deployed
          SET r' to GaussianDistribution with (Radius, sdl)
          SET alpha to GaussianDistribution with (angle, sda)
          $x = a + r' \cos(alpha)$
          $y = b + r' \sin(alpha)$
          STORE initial values in NodeList as a Node
      END FOR
      START moveNodes with NodeList
      EMPTY NodeList
      INCREMENT angle by $2\pi/8$
  ENDFOR
ENDFOR
```

Fig. 4.6 Deployment phase algorithm for circular move mobility model

```
/*  Terminology
    a=x coordinate of the center point
    b=y coordinate of the center point
*/
SET currentRound to 0
WHILE currentRound < totalRounds
    FOR each Node in NodesList
        IF r' is bigger than the threshold value THEN
            COMPUTE random value RandSpeed for minSpeed and maxSpeed
            COMPUTE random value RandAngle for -angSpeed and angSpeed
            SET r' to (r'-RandSpeed)
            SET alpha to (alpha-RandAngle)
            $x = a + r' \cos(alpha)$
            $y = b + r' \sin(alpha)$
        ENDIF
    ENDFOR
    INCREMENT currentRound
ENDWHILE
```

Fig. 4.7 Movement phase algorithm for circular move mobility model

5. PERFORMANCE EVALUATION

We evaluated performance of our scheme with various simulations. In this section, we first explain threat model and then performance metrics. Finally, we give simulation results together with configuration and parameters.

5.1. Threat Model

We assume that attacker can learn keys of a node by capturing it. In RoK[5] scheme, attacker can compute forward and backward keys separately. In other words, if a forward key captured in generation j, it is possible to compute key with same index for generations after j and it is also possible for backward keys for generations before $j + G_w$. Because key pools in our UKP scheme are distinct, there is no such association between keys of different generations. However, the attacker learns all the keys in a captured node including keys that belong to further generations. In our model, we considered attacker as an eager attacker, which means nodes will be captured at constant rate at each round starting with 5th generation and attack does not stop until the end of the simulation.

5.2. Performance Metrics

In WSNs, sensor nodes are assumed to carry sensitive information. There should be a secure paths among the nodes. Thus secure connectivity is an important metric. Whilst delivering data, nodes should be communicating securely against an eavesdropping attacker. Hence, resiliency of the network is another significant metric. In this section, we explain these metrics.

5.2.1. Local and Global Connectivity

Local connectivity is an important metric that shows the performance of key distribution mechanism. It is defined as the probability of sharing a common key between two neighboring sensor nodes. If this value close to one, then most of the nodes in the network can communicate securely with almost all neighbors that in the range of communication.

High local connectivity shows that a node can establish a secure communication with most of its neighbors. However, high local connectivity does not guarantee high global connectivity. Global connectivity is used to check if there are any nodes that are not reachable from the rest of the network. It is calculated as the ratio of the number of nodes in the largest isolated component to the number of nodes in the whole network.

5.2.2. Resiliency

When an attacker captures a node, he learns all the keys stored in the node. Hence, any established connection between the corrupted node and its neighbors are automatically compromised. Moreover, attacker can compromise some other additional links, if he knows the keys that are used to establish those links. As more nodes are captured, the attacker learns more keys and he can use them to monitor additional channels. This presents a threat to the resiliency of network. Thus, ratio of the of compromised links is an important metric to evaluate security performance of the WSNs.

In order to evaluate resiliency of the network, we measure the ratio of additionally compromised links after a node capture. This ratio is computed as the number of additionally corrupted channels divided by the number of all establishes links, which are currently active. The network has better resiliency when the ratio of compromised links is smaller.

5.3. Configurations

Simulation code developed with C# using MS Visual Studio 2010. Simulations are conducted on a computer with 64-bit Windows 7 running on Intel Core i7-2600 CPU, 8.00 GB RAM.

For the sake of a fair comparison, we used similar setup as in the RoK [5] scheme for our simulation. We set the number of keys in each pool, P, to 10.000 and key ring size, m, as 500 for each node. Note that key chain size is $m/2$ for each of forward and backward key rings in the RoK scheme, in order to compare results under same memory consumption. Generation window, G_w, is taken as 10 and we assume that a node's lifetime is determined according to a Gaussian distribution with mean and G_w / 2 with standard deviation G_w / 6.

5.3.1. For Random Walk and Reference Point Group Mobility Models

The number of nodes in the network is taken as 1.000. Deployment area is 500×500 meters and sensor node's wireless communication range is 40 meters. Nodes are distributed in that area with uniform random distribution. Speed of a node is decided randomly between 5 to 15 meters per minute. Note that, we assume both schemes use same mobility patterns for the sake of fairness. *N*odes, whose lifetimes *are* expired, *are* replaced with new ones.

5.3.2. For Circular Move Mobility Model

The number of nodes in the network starts with 200 in one generation and increase in time. It stabilizes at average 1.000. At each round 25 nodes are deployed from a bunch point and then a new bunch point is picked as mentioned in Circular Move Mobility model. When a full circle completed (from initial point to that point again), a new generation is started to be deployed. Deployment area is a circle with radius 500 meters and sensor node's wireless communication range is 40 meters. Nodes are distributed on eight bunch points with 500 meters of mean and 20 meters of standard deviation of linear displacement off the arc. For angular displacement, mean values vary due to different bunch points on circle. These values are $0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4$ respectively for each bunch point and standard deviation is 15 degrees at each bunch point. Linear speed of a node is decided using uniform random distribution between 5 to 10 meters per minute and similarly, angular movement speed decided using uniform random distribution between -5 to 5 degrees. We assume both schemes use same mobility patterns.

5.4. Simulation Results

We compute local connectivity, global connectivity and resiliency performance of our UKP scheme and compare with RoK scheme [5]. Note that one generation is 10 rounds for random walk and Reference Point Group Mobility (RPGM) model and 8 rounds for Circular Move Mobility model. We run simulations for 300 rounds for random walk and RPGM model, and 150 rounds for Circular Move Mobility model. Moreover, each result is obtained by taking the average of 25 runs for the sake of smoothness of the results.

Local connectivity performance for both RoK and our UKP schemes under random walk and RPGM models is shown in Figure 5.1. In Figure 5.2, local connectivity performance is shown for Circular Move Mobility model. As shown in both figures, local connectivity is around the level of 1.0 in all cases. This level is almost perfect.

In Figure 5.3, global connectivity results are given for two mobility models, Random Walk and Reference Point Group Mobility model. The global connectivity performance is almost perfect level in all cases. Global connectivity ratio results for circular move model is given in Figure 5.4. Differently from other models, global connectivity decreases significantly at the beginning and then stabilizes at a nearly perfect level for both RoK and UKP in Circular Move Mobility model. Such a poor global connectivity at the beginning of the deployment is not unexpected in Circular Move Mobility model, because nodes are deployed as separate groups at the perimeter.

As nodes move in time, they start to spread to the environment and isolated groups get close to each other. Hence, global connectivity increases during the movement.

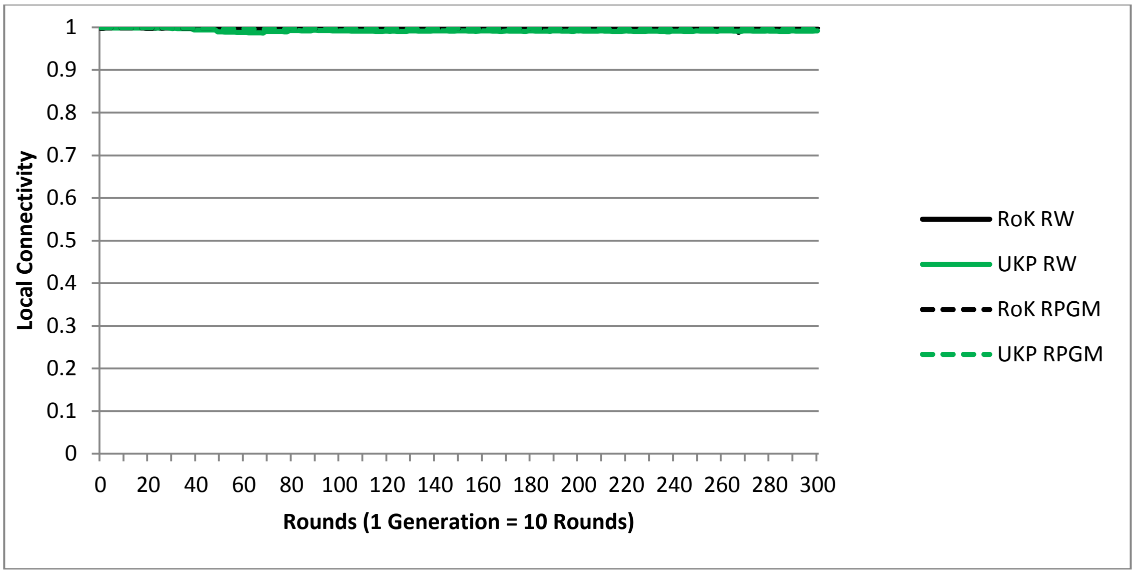


Fig. 5.1 Local connectivity of RoK and UKP for Random Walk (RW) and Reference Point Group Mobility (RPGM) models

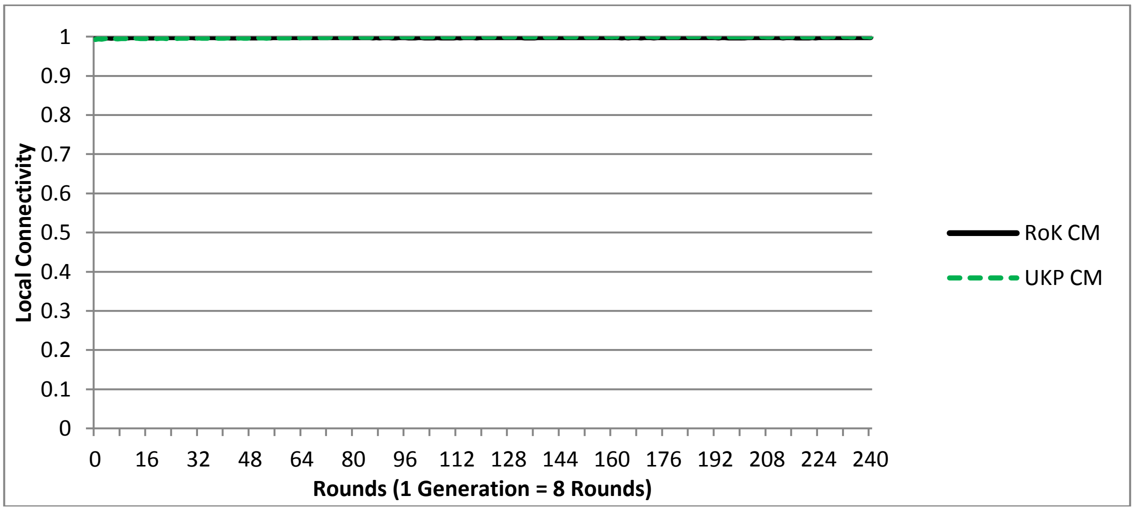


Fig. 5.2 Local connectivity of RoK and UKP for Circular Move (CM) Mobility model

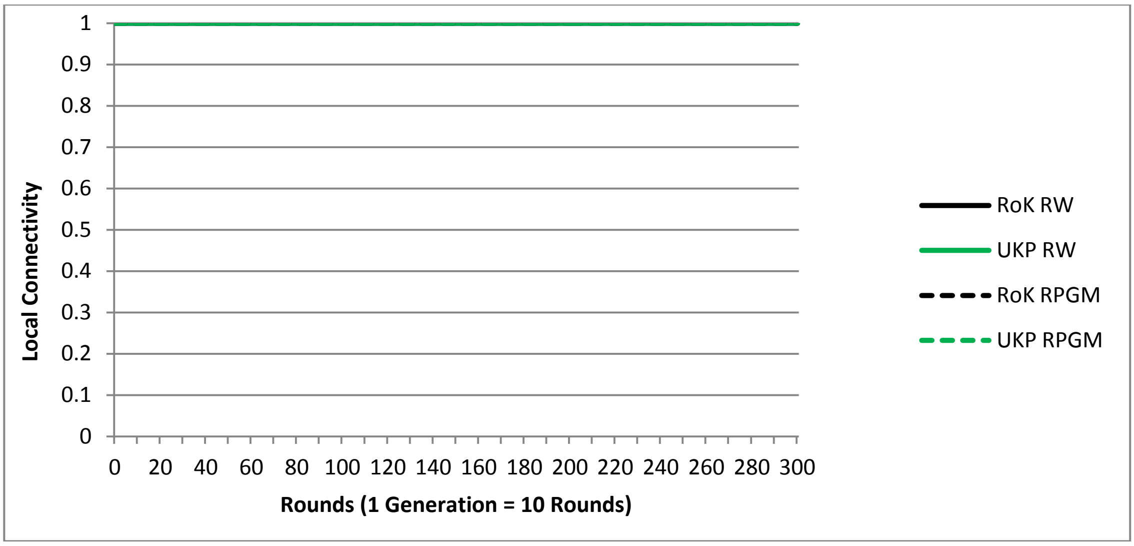


Fig. 5.3 Global connectivity of RoK and UKP for Random Walk (RW) and Reference Point Group Mobility (RPGM) models

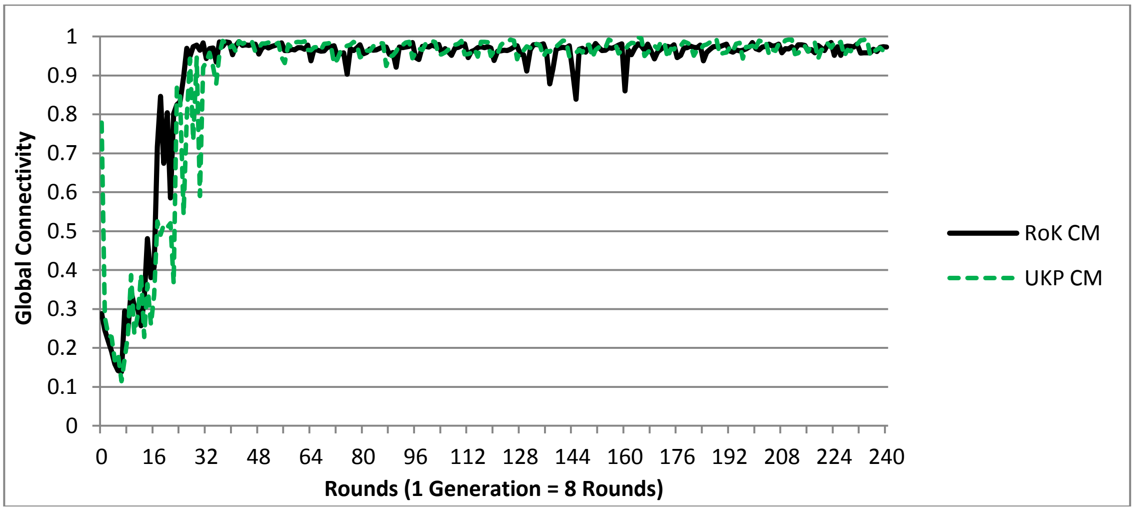


Fig. 5.4 Global connectivity of RoK and UKP Circular Move (CM) Mobility model

For the evaluation of the network resiliency, we consider an attacker who captures sensor nodes with rates 1, 3 and 5 nodes per round. In our simulations, attacker starts compromising nodes at generation 5 in order to allow some time for network stabilization.

Figure 5.5 gives the resiliency results for RoK and UKP for Random Walk Mobility model. It shows that our scheme outperforms RoK [5] scheme in terms of resiliency under heavy attack. Resiliency is improved in UKP, as compared to RoK by decreased the ratio of the compromised links from almost 50% to 38% where the capture rate is 5 nodes per round, and 30% to 22% with the capture rate 3 nodes per round. Only under light attack (capture rate = 1 node per round), RoK performs slightly better than UKP scheme. Figure 5.6 also gives us similar results with Reference Point Group Mobility model. Compared to RoK, we have better results with capture rates 3 nodes/round and 5 nodes/round. Resiliency of RoK is again slightly better than UKP when the capture rate is 1 node/round.

In Figure 5.7, ratio of the compromised links is presented for Circular Move Mobility model. For high capture rates, UKP performs much better than RoK. Our scheme lowers the ratio of compromised links from ~45% to 30% for capture rate 5 nodes/round, ~25% to 18% for capture rate 3 nodes/round as compared to RoK scheme. When the capture rate is 1 node/round, UKP has higher ratio of compromised links than RoK but the difference is minimal. As it can be seen from Figure 5.7, additionally compromised link ratio has less fluctuation over time for UKP scheme compared to RoK scheme. This shows that UKP scheme is more stable in terms of network resiliency.

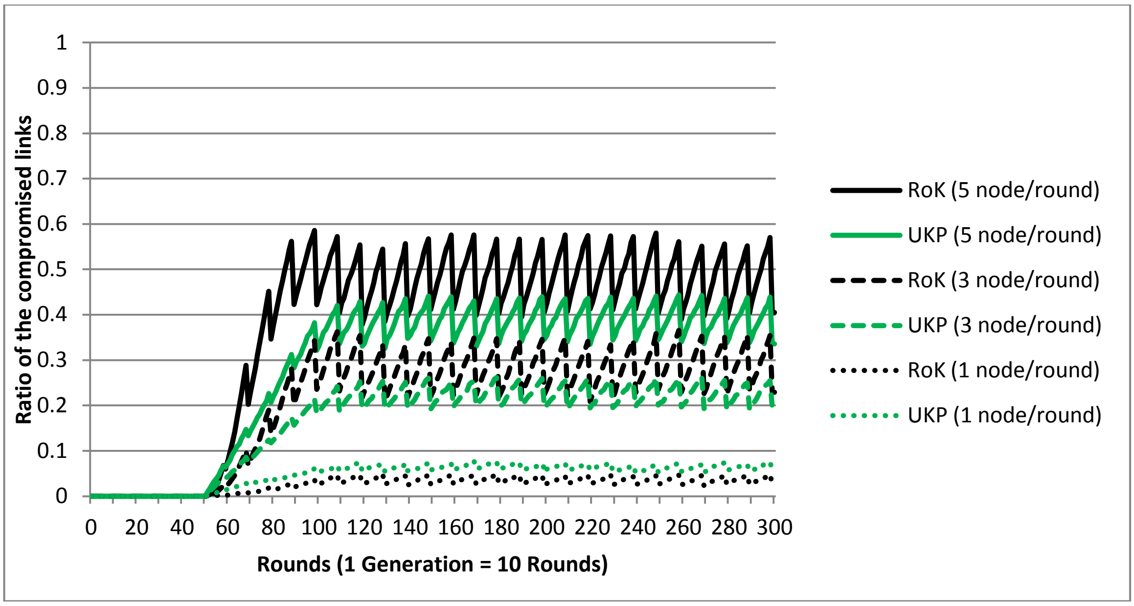


Fig. 5.5 Resiliency of RoK and UKP in case of an eager attacker with capture rates of 1, 3, and 5 nodes per round with Random Walk (RW) model

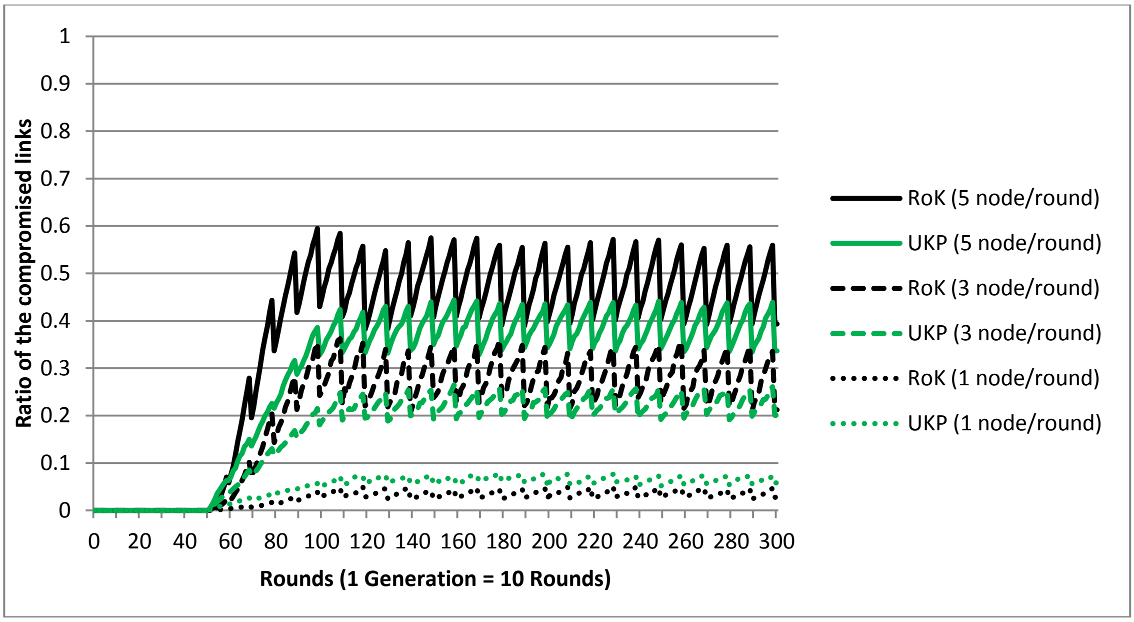


Fig. 5.6 Resiliency of RoK and UKP in case of an eager attacker with capture rates of 1, 3, and 5 nodes per round with Reference Point Group Mobility (RPGM) model.

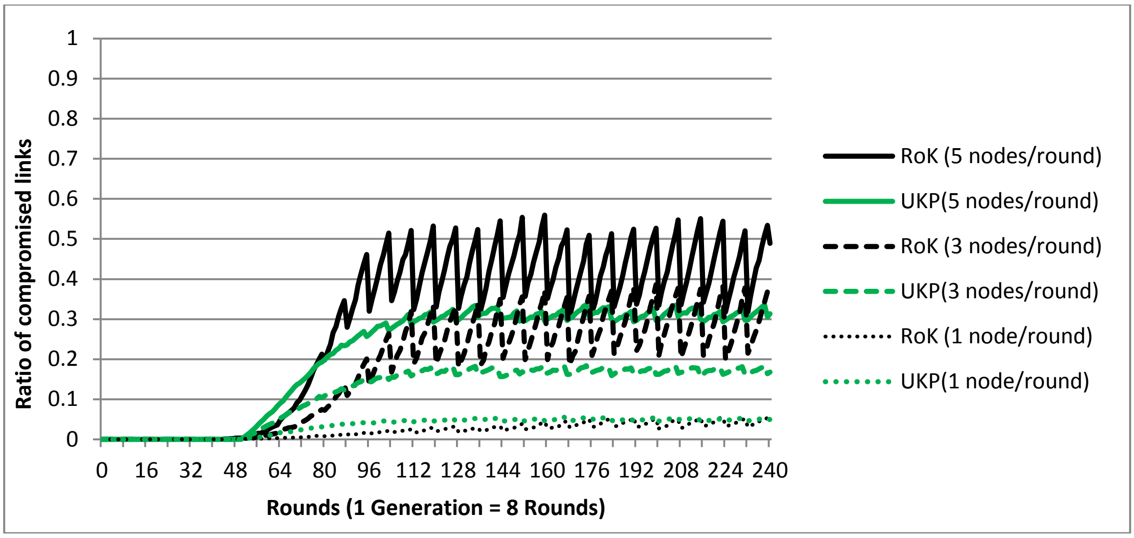


Fig. 5.7 Resiliency of RoK and UKP in case of an eager attacker with capture rates of 1, 3, and 5 nodes per round with Circular Move (CM) Mobility model.

6. CONCLUSIONS

In this thesis, we propose Uneven Key Predistribution (UKP) scheme for multiphase wireless sensor networks in mobile environment. Our scheme is based on using different and distinct key pools at each generation and usage of keys that are drawn unevenly from these pools based on measured age distribution of the nodes. At each generation, nodes choose a new set of keys from a new key pool. Therefore, keys in the network will be renewed partially at each redeployment phase; this provides *self healing* to the network.

We employed simulation techniques for the performance evaluation of UKP. We also implemented RoK scheme [5] since it serves as a basis to multiphase wireless sensor network security. We run simulations for both schemes with same parameters and same movement patterns for the sake of a fair comparison. In both schemes, nodes use same memory and same computational power in all models and their battery lives are randomized by a Gaussian distribution with same mean and standard deviation.

We run our simulations under different mobility models. Apart from the two existing mobility models, we proposed a new one, Circular Move Mobility model, for monitoring a circular area by deploying nodes only around the perimeter.

We improve the resiliency against heavy node capture attacks, whilst maintaining almost perfect local and global connectivity for Random Walk and Reference Point Group Mobility models. For Circular Move Mobility model, UKP still outperforms in terms of resiliency under heavy node capture attacks with almost perfect local connectivity. Global connectivity for both RoK and UKP under Circular Move Mobility model is very low at the beginning, but it stabilizes around 0.95 at steady state. Also in Circular Move Mobility model, UKP scheme's resiliency performance is much more stable than RoK.

7. REFERENCES

[1] Eschenauer, L. and Gligor, V. D.: A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 41--47. Washington, DC, USA (2002)

[2] Chan, H., Perrig, A., Song, D.: Random Key Predistribution Schemes for Sensor Networks. In: IEEE SP'03, pp. 197--213 (2003)

[3] Amundson, I., Koutsoukos, X.D.: A Survey on Localization for Mobile Wireless Sensor Networks. In: Proceedings of the 2Nd International Conference on Mobile Entity Localization and Tracking in GPS-less Environments, pp. 235--254, Springer (2009)

[4] Yi, S., Youngfeng, C., Liangrui, T.: A multiphase key predistribution scheme based on hash chain. In: Fuzzy Systems and Knowledge Discovery, pp. 2061--2064, (2012)

[5] Castelluccia, C. and Spognardi, A.: RoK: A robust key predistribution protocol for multiphase wireless sensor networks. In: SecureComm2007, Third International Conference on Security and Privacy in Communication Networks. Baltimore, MD, USA (2007)

[6] Ito, H., Miyaji, A., Omote, K.: RPoK: A Strongly Resilient Polynomial-Based Random Key PreDistribution Scheme for Multiphase Wireless Sensor Networks. In: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, pp.1--5. (2010)

[7] Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. In: Wireless Communication & Mobile Computing (WCMC):

Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483--502. (2002)

[8] Tas, B., Tosun, A.S.: Mobile Assisted Key Distribution in Wireless Sensor Networks. In: Communications (ICC), 2011 IEEE International Conference, pp. 1--6. (2011).

[9] Kumar Das, A.: A Key Establishment Scheme for Mobile Wireless Sensor Networks Using Post-Deployment Knowledge. In: International Journal of Computer Networks & Communications (IJCNC), vol. 3, no. 4, (2011)

[10] Karaca, K., Levi A.: Resilient key establishment for mobile sensor networks. In: Distributed Computing in Sensor Systems and Workshops (DCOSS), pp. 1--6. (2011)

[11] Shan, T., Liu, C.: Enhancing the key predistribution scheme on wireless sensor networks. In: IEEE Asia-Pacific Conference on Services Computing, IEEE Computer Society, pp. 1127--1131, Los Alamitos, CA, USA (2008).

[12] Hussain, S., Rahman, M., Yang, L.: Key predistribution scheme using keyed-hash chain and multipath key reinforcement for wireless sensor networks. In: IEEE Computer Society, pp. 1--6, Los Alamitos, CA, USA, (2009).

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

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

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

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

[17] Rivest R.L., Shamir A, Adleman L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In: Communications of the ACM, vol. 21, pp. 120--126. (1978)

[18] Diffie, W.; Hellman, M.E.,: New directions in cryptography. In: Information Theory, IEEE Transactions on , vol.22, no.6, pp.644--654, Nov (1976) doi: 10.1109/TIT.1976.1055638

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

[20] Basagni, S., Carosi, A. and Petrioli, C.: Mobility in Wireless Sensor Networks. In: Algorithms and Protocols for Wireless Sensor Networks (ed A. Boukerche), John Wiley & Sons, Inc., Hoboken, NJ, USA. (2008) doi: 10.1002/9780470396360.ch10

[21] Stallings, W.: Symmetric Ciphers. In: Cryptography and Network Security: Principles and Practice, 5th ed. Prentice Hall, (2010).

[22] Zhang, X., Heys, H.M., Li, C.: Energy efficiency of symmetric key cryptographic algorithms in wireless sensor networks. In: Communications (QBSC), 25th Biennial Symposium on , pp.168--172, May (2010).

[23] Dargie, W. and Poellabauer, C., "Fundamentals of wireless sensor networks: theory and practice", John Wiley and Sons, pp. 3--14 (2010). ISBN 978-0-470-99765-9,

[24] Schneier, B.: Applied Cryptography (2nd Ed.): Protocols, Algorithms, and Source Code. in: *C*. John Wiley & Sons, Inc., New York, NY, USA (1995).