KEY PRE-DISTRIBUTION APPROACH USING BLOCK LU-DECOMPOSITION IN WIRELESS SENSOR NETWORK

By

Areej Rasmi AL-Rabadi

Supervisor

Prof. Dr. Najib A. Kofahi

COMPUTER SCIENCES

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF REQUIRMENT FOR THE DEGREE OF THE MASTER OF COMPUTER SCIENCE AT YARMOUK UNIVERSITY, IRBID, JORDAN

November, 2018

The undersigned have examined the thesis entitled **“KEY PRE-DISTRIBUTION APPROACH USING BLOCK LU-DECOMPOSITION IN WIRELESS SENSOR NETWORK”**

By

Areej Rasmi AL-Rabadi

B.SC. Computer Science/ Yarmouk University, 2015

A Candidate For The Degree of **Master of Computer Science** and Hereby Certify That is Worthy of Acceptance.

Approved By:

Najib A. Kofahi ....Najib A. Kofahi........ **Supervisor**
Professor of Computer Science, Yarmouk University

Malek M. Barhoush ....**Committee Member**
Assistant Professor of Computer Science, Yarmouk University

Ayoub M. Alsarhan ....**Committee Member**
Associate Professor of Computer Science, Hashemite University

November, 2018

Acknowledgment

First of all, I would thank god for being able to conduct and complete this step in my study with success. Then, I would like to express my appreciation and my special thanks to my supervisors for their many helpful comments and suggestions and for their close supervision to complete this thesis, Prof. Dr. Najib A. Kofahi, whose valuable suggestions and guidance were very helpful in the various phases for the completion of my thesis and make it a full proof success. I wish also to express my gratitude to Dr. Ahmad M. Manasrah, the person who gave me the opportunity to do this research in the field of the key agreement problem in the wireless sensor network and to take care of me during the entire thesis.

Last but not the least, I wish to express a sense of gratitude and love to my friends and my beloved family for their support, elevating inspiration, and encouraging guidance to complete this step in my study.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Notation	Description
WSN	Wireless Sensor Network
LU-Decomposition	Lower Upper Decomposition
BLU-Decomposition	Block LU-Decomposition
L	Lower Triangular Matrix
U	Upper Triangular Matrix
BL	Block Lower Triangular Matrix
BU	Block Upper Triangular Matrix
DH	Diffie Hellman
MAC	Message Authentication Code
F_q	Finite Field Contains (q) Elements
t	The Degree of The Polynomials

الملخص

الربضي، أريج رسمي. طريقة ما قبل التوزيع الرئيسية باستخدام خوارزمية تحليل المصفوفة المثلثية السفلى و العليا على شكل مجموعات في شبكة الاستشعار اللاسلكية. ماجستير في علوم الحاسوب، رساله، قسم علوم الحاسوب، جامعه اليرموك، ٢٠٠٨. (المشرف: أ.د. نجيب الكوفحي).

تعتمد العديد من التطبيقات الحديثة والمهمة في الوقت الحاضر على شبكة الاستشعار اللاسلكي بسبب مزاياها المتعددة. بيئات العمل لهذه الشبكات عادة ما تكون تابعة للمناطق المعادية لذلك يتم توزيع أجهزة الاستشعار بشكل عشوائي على هذه المناطق و نظرًا للتوزيع العشوائي لعقد جهاز الاستشعار، فأنه لا يمكن لأي من هذه الأجهزة التعرف على أجهزة الاستشعار الأخرى والتي قد تقع في نطاق اتصالها. لذلك، تحتاج أجهزة الاستشعار هذه إلى مخطط للاتفاق بين أجهزة الأستشعار على اليه لإنشاء مفاتيح مشتركة يتم من خلالها إنشاء اتصال آمن مع بعضها البعض من أجل حماية المعلومات المرسلة فيما بينها والتي يتم نقلها أخيراً إلى وحدة التحكم الرئيسية. يقترح هذا البحث مخططًا جديدًا للاتفاق على اليه لإنشاء مفاتيح بين اجهزة الاستشعار في شبكة الاستشعار اللاسلكية معتمداَ على مخطط متعدد الحدود و خوارزمية تحليل المصفوفة المثلثية السفلى و العليا على شكل مجموعات. لقد تم تقييم أداء الخوارزمية المقترحة في هذه الرسالة استنادًا إلى درجة الاتصال بين اجهزة الاستشعار ومدى تأثير اجهزة الاستشعار التي تم امساكها وكشفها من قبل الخصم على أجهزة الاستشعار الاخرى وسعة الذاكرة التي تحتاجها ومقدار الطاقة التي تستهلكها وعدد الرسائل التي يحتاج الخصم ان يتتبعها لكسر الخوارزمية المقترحة وكم من اجهزة الاستشعار يحتاج الخصم ان يضع بين اجهزة الاستشعار القانونية لكسر الشبكة. لقد تم الاعتماد على نظام المحاكاة لتقييم أداء الخوارزمية المقترحة ومقارنتها مع الأعمال الأخرى في هذا المجال حيث تظهر نتائج المحاكاة أن الخوارزمية المقترحة تحقق أداءً أفضل مقارنة بالأعمال الأخرى الموجودة حالياً.

الكلمات المفتاحية: شبكة الاستشعار اللاسلكية ,أجهزة الاستشعار ,مشكلة الأتفاق على مفتاح مشترك ,نهج ما قبل التوزيع الرئيسي ,نظام إدارة المفاتيح ,مخطط متعدد الحدود ,مخطط القائمة على المصفوفة ,خوارزمية تحليل المصفوفة المثلثية السفلى و العليا ,خوارزمية تحليل المصفوفة المثلثية السفلى و العليا على شكل مجموعات.

Abstract

AL-Rabadi, Areej Rasmi. Key Pre-Distribution Approach Using Block LU-Decomposition in Wireless Sensor Network. Master of Computer Science, Thesis, Department of Computer Sciences, Yarmouk University, 2018. (Supervisor: Prof. Dr. Najib A. Kofahi).

At present, many types of applications rely on the wireless sensor networks because of its availability and its other advantages. The working environments of the wireless sensor networks are usually hostile and it is difficult to access so that the sensor nodes are distributed randomly over these environments. Due to the random distribution of the sensor nodes, none of these devices can detect which of the other sensor nodes may fall within its range. Therefore, these sensor nodes need a key agreement scheme that allows them to establish a secure communication with one another in order to protect the information sent between them which is finally conveyed to the base station. This thesis proposes a new and efficient key management scheme for wireless sensor network based on the Polynomial Pool-based key pre-distribution scheme and the Block LU-Decomposition algorithm. Due to the resource limitations of the sensor nodes the performance of the proposed algorithm is evaluated based on network connectivity, resilience against node capture, memory overhead, energy consumption, and the number of Sybil nodes and the messages needed by the adversary to break the entire network and the proposed approach. The proposed approach was evaluated using the simulation techniques, and the simulation results were compared with related works. The simulation results show that the proposed scheme performs better compared to the existing schemes.

Keywords: Wireless Sensor Network, Sensor Node, Key Agreement Problem, Key Pre-Distribution Approach, Key Management Scheme, Polynomial-Based Scheme, Matrix-Based Key Pre-Distribution Scheme, LU-Decomposition Method, Block LU-Decomposition Algorithm.

Chapter One

Introduction

1.1 General Overview

The wireless sensor network (WSN) has been considered as one of the most important technologies for the twenty-first century (Zheng and Jamalipour, 2009). It consists of thousands of small and tiny devices called sensor nodes that are responsible for collecting readings from the targeted area (Singh, Singh and Singh, 2010; Tiwari *et al.*, 2015). The collected data are sent to a base station which is more powerful than the sensor nodes in terms of processing capabilities and memory size. These sensor nodes are of small size, limited memory, and limited computational capabilities (Tiwari *et al.*, 2015). This type of networks is used with a wide range of applications such as, seismic monitor, military operations, medical operations and to monitor the environmental conditions like the temperature, vibrations and the pressure (Zhao and Ye, 2014).

There are two main types of the environments in which sensor nodes can be deployed (Divya *et al.*, 2014); the controlled environment, such as home and office, where they can be deployed manually, and the uncontrolled environment such as the hostile and toxic areas in which the sensor nodes are randomly distributed in the targeted area by dropping them from the air or by soldiers. The random distribution of the sensor nodes to the target area produces an unknown network topology. It is, therefore, not easy to visit the sensors in this environment to perform some regular maintenance which, therefore, gives adversaries some advantages in these environments to physically damage the sensors or they can replace some by their own sensors for their own benefits.

The WSN is weaker against external attacks than the wired networks due to the limited power and memory of the sensor nodes and the nature of the communications (Khandke *et al.*, 2013; Anita, Geetha and Kannan, 2015). The WSN communication following a broadcast style to transmit the information, hence, the data collected and transmitted in this network can be intercepted easily by the eavesdroppers in the uncontrolled environments. Therefore, certain applications communications, such as military operations, must be protected from any external/internal attack. The perfect protection mechanism is through encrypting the communication channels between communicating ends. Therefore, various efforts have been made by different researchers to find an efficient mechanism to generate the secret keys for the sensor nodes to protect the data transmitted between them taking into consideration the nature of the communication between the sensor nodes, the size and the density of the WSN, and the network topology (Anita, Geetha and Kannan, 2015).

On the other hand, the random distribution of the sensor nodes and the wireless nature of WSN communications give the adversaries an opportunity, to distribute their own sensor nodes among the legitimate nodes in an uncontrolled environment or replacing some nodes with forged identities (i.e. malicious nodes) to be able to communicate with the neighboring legitimate nodes (Chan, Perrig and Song, 2003; Zia and Zomaya, 2011). These malicious nodes are called Sybil nodes and this type of attacks is called Sybil attack (Tandel, 2016). Sybil attack takes benefits from the fact that each node in the WSN can communicate only with the neighboring nodes, and not all sensor nodes can communicate with the base station directly (Ilakkiya, Jayakumar and Shobana, 2013). Therefore, the sensor nodes need to broadcast their messages to each other until the messages reach the base station. The Sybil node can take advantage of the retransmission of the same message between the sensor nodes until it reaches the base station and acts as a legitimate node to receive these messages. So, the Sybil nodes can alter or drop the messages before they forward it to the neighboring nodes, which may lead to degrading the performance of the network, hence violating the privacy, integrity and the secrecy of the transferred data.

1.2 Background Information

The WSN is used in a wide range of applications as the military operations which require a high level of security (Mansour, Chalhoub and Lafourcade, 2015). So, the sensor nodes need to transmit the data securely through a secure link. As presented in (Hsu *et al.*, 2014), the security of the WSN is achieved through two important outlines: (1) The key management schemes which are responsible for creating, calculating and distributing the secret keys to the sensor nodes to encrypt the transmitted data and protect it from the adversaries, (2) The key establishment protocols to provide the sensor nodes with secure routes within the network to exchange their information and establish their secret keys.

Two types of key establishment protocols were used in WSN (Hsu *et al.*, 2014). The first type is the key transfer protocol, where a trusted server is responsible for generating the secret keys and transfer these keys to the sensor nodes secretly. The second type is the key agreement protocol in which the sensor devices are involved in the key derivation process like the Diffie-Hellman (DH) protocol which is one of the most adopted key agreement protocols (Joshi, Verma and Verma, 2015). The key agreement protocol gives the sensor nodes the ability to derive their own shared keys to encrypt their secret messages without the need for a trusted server (Hsu *et al.*, 2014). However, how a pair of sensor nodes can agree on a shared key and what scheme can be used to derive this key is called the key agreement problem (Du *et al.*, 2005; Mansour, Chalhoub and Lafourcade, 2015). The security issue in WSN involves verifying the following list of requirements:

1) The data integrity which provides the receiver of the message with the ability to check if the message information has been modified or altered by the adversaries.
2) The sender authentication which provides the receiver of the message with the ability to verify the authenticity of the message source and its reliability.
3) The confidentiality of the communication which provides the sensor nodes with the ability to protect and hide the message information from disclosure to the unauthorized sensors (Hsu *et al.*, 2014; Anita, Geetha and Kannan, 2015).

The previous requirements can be ensured using an efficient and strong key management scheme (Ilakkiya, Jayakumar and Shobana, 2013). The key management schemes that have been proposed to provide the security requirements and to solve the key agreement problem are classified into three categories (Dai and Xu, 2010):

1) **Trusted Third Party:** Every node has a secret key with the base station only and the sensor nodes cannot communicate with each other. However, such a protocol increases the communication overhead because it depends on using two types of keys; a public and a private key (Asymmetric Cryptographic scheme) (Rani and Kumar, 2012; Alshanty and Erşan, 2016).

2) **Public Key:** It has two types of keys (a private and a public key) between each pair of sensor nodes. However, this protocol increases the communication overhead and requires large memory. Therefore this scheme is not suitable for the WSN.

3) **Key Pre-Distribution:** This protocol operates in three steps (Giri and Mahadevan, 2013):

    1. Keying Information Pre-Distribution: In this step, each sensor node receives an important information from the base station before their deployment to generate the keys.

    2. A Secret-Key Establishment: Each node can use the keying information that is received from the base station to establish a secure key with the neighboring nodes based on the key management scheme that is used.

    3. Path-Key Establishment: If each pair of nodes could not establish a secret key directly, then they can communicate through other nodes (i.e. using the other nodes as intermediates). But every node in this path must have a secret key with the neighboring nodes.

The traditional key pre-distribution schemes such as the Single Master Key Pre-Distribution scheme and the Pairwise Key Pre-Distribution scheme are not suitable for the

WSN (Banaie *et al.*, 2015). In the Single Master Key Pre-Distribution scheme, all the sensor nodes use the same key (i.e. master key) and this key is shared by all the sensor nodes within the same network. Each sensor node wants to communicate with the adjacent nodes must use this shared key. Despite that, this scheme requires less storage space because the sensor nodes need to store only one shared key, but it is not suitable for the WSN security because if any of these sensor nodes were compromised (i.e. trusted nodes have been taken over by the attacker), then the master key will be known by the adversary and this will lead to compromise the entire network. In The Pairwise Key Pre-Distribution scheme, the base station distributes the shared keys to the sensor nodes before their deployment (Banaie *et al.*, 2015). The sensor nodes can use these keys to communicate with the adjacent sensor nodes. Example of the Pairwise Key Pre-distribution scheme in (Zhao and Ye, 2014). In this scheme, the sensor nodes and the keys are divided into groups and each group receives two sets of keys; the key chain and the cross-group key chain. If both nodes are in the same group they can communicate using their key chains, otherwise, they can use the cross-group key. Despite the fact that it increases the resilience (i.e. if the adversary discloses some secret keys by compromising some nodes, the other sensor nodes in the network can keep its secret keys secure from the adversary) of the network and provide mutual authentication between the sensor nodes, but it is also not suitable for the WSN because each sensor node needs to store all the pairwise keys in its private memory, which may exceed the storage size of its memory space especially if the network size is large (Ilakkiya, Jayakumar and Shobana, 2013).

The disadvantages of the previous solutions are more than its advantages, and they are not suitable for the WSN. This motivated the researchers to explore different solutions to protect the WSN and the sensor nodes from the adversary attacks based on the key pre-distribution protocol. In (Eschenauer and Gligor, 2002; Rani and Kumar, 2012), the researchers proposed a random key Pre-distribution scheme, where the keys are distributed randomly to the sensor nodes from a private key pool before their deployment and any pair of nodes can communicate if they can find a common key, otherwise they can communicate using another path through other sensor nodes (i.e. Multi-hop communication). Another random key pre-distribution scheme, the Pairwise Key Pre-distribution approach that is presented in (Mu and Li, 2014). This approach works as a random key pre-distribution approach but before distributing the keys to the sensor nodes, the base station performs (ι) times has function operation on these keys to generate new keys. The value of the (ι) variable varies from one sensor node to another. However, in the random key Pre-distribution scheme, it is not always possible for a sensor node to find a common key with the other sensor nodes due to the random distribution of the keys to the sensor nodes. Moreover, this approach may require more memory space as well as the limited view on the routing paths (i.e. it is not always possible for a sensor node to find a common key with the other neighbor sensor nodes), where this may require an increase in the number of keys that are distributed to the sensor nodes to increase its possibility to find a common key.

As a result of the limited view on the routing paths between any two communicating sensor nodes, the deterministic key pre-distribution scheme has appeared. This scheme distributes the keys to the sensors based on the combinatorial design, where it decides how many keys and which keys will be distributed to the sensors before their deployment. In (Çamtepe and Yener, 2004), the researchers used two types of combinatorial designs; the symmetric designs and the generalized quadrangles design with the aim to generate a set of keys for the sensor nodes in the WSN. Another deterministic approach is proposed in (Anzani, Javadi and Moeni, 2018). This approach based on the multivariate polynomials, where the authors applied the combinatorial design on the multivariate polynomial to derive the shared keys.

1.3 Matrix-Based Key Pre-Distribution Scheme

Most of the key pre-distribution approaches that are based on Matrix of keys (i.e. keys are presented within a matrix) to distribute the keys to the sensor nodes are based on the LU-Decomposition procedure. The LU-Decomposition is a procedure that is used to find two special matrices; (L) and (U) that represent the original matrix as represented in Figure 1.1, where (L) is the lower triangular matrix and (U) is the upper triangular matrix. The (L) matrix has all elements above the diagonal to be zeros, where in the (U) matrix all elements below the diagonal are zeros. (Bandara and Ranasinghe, 2005).

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \mathrm{l}_{21} & 1 & 0 & 0 \\ \mathrm{l}_{31} & \mathrm{l}_{32} & 1 & 0 \\ \mathrm{l}_{41} & \mathrm{l}_{42} & \mathrm{l}_{43} & \mathrm{l}_{44} \end{bmatrix} * \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Figure 1.1: LU-Decomposition

The matrix has an LU-Decomposition if and only if its determinant does not equal (0). The determinant of a matrix is a value that can be obtained from the elements of the matrix, it is denoted as $det(A)$. Equation (1) illustrates the determinant of a (3×3) matrix. The matrix that has LU-Decomposition is called an invertible matrix (Polok and Smrz, 2017).

$$\det(A) = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = a(ei - hf) - d(bi - hc) + g(bf - ec) \qquad (1)$$

A special case of the matrix is the symmetric matrix that is equal to its transpose (Huckle, Waldherr and Schulte-Herbrüggen, 2013). The transpose is an operator which flips a matrix over its diagonal, where the rows and the columns are interchanged, it is denoted as (A^T). Figure 1.2 illustrates how the transpose operator works.

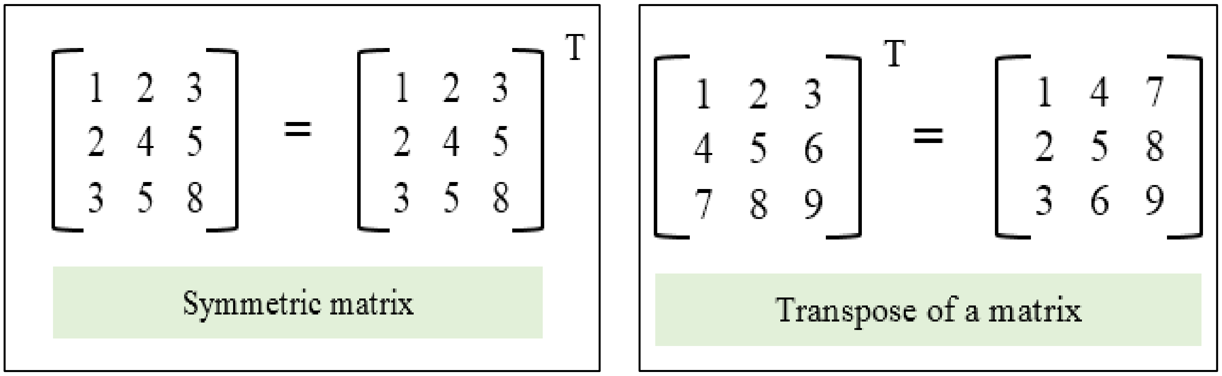

$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 8 \end{bmatrix}^T$
Symmetric matrix
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$
Transpose of a matrix

Figure 1.2: The Transpose Operator

In this thesis, the LU-Decomposition will be used, in particular, the Block LU-Decomposition that is presented in (Demmel, Higham and Schreiber, 1995). The Block LU-Decomposition is a modified LU-Decomposition which acts on blocks of the matrix as represented in (Chen *et al.*, 2009). The main purpose behind creating the Block LU-

Decomposition procedure is to speed up the time required to solve the matrix. In this work, the Block LU-Decomposition procedure will be used to decompose the block matrix into Block Lower Triangular (BL) and Block Upper Triangular (BU) matrices as shown in Figure 1.3.

$$\begin{array}{|c|c|c|}\hline A_{11} & A_{12} & A_{13} \\ \hline A_{21} & A_{22} & A_{23} \\ \hline A_{31} & A_{32} & A_{33} \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline L_{11} & L_{12} & L_{13} \\ \hline L_{21} & L_{22} & L_{23} \\ \hline L_{31} & L_{32} & L_{33} \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline U_{11} & U_{12} & U_{13} \\ \hline U_{21} & U_{22} & U_{23} \\ \hline U_{31} & U_{32} & U_{33} \\ \hline \end{array}$$

Figure 1.3: The BLU-Decomposition

The BLU-Decomposition is a procedure deals with the matrix as blocks, where each block contains a set of elements. After applying the BLU-Decomposition to the matrix blocks, two matrices will be resulted (BL and BU). Each matrix contains a set of blocks, each block contains a set of elements. This is the main advantage of the proposed approach, where more than one space will be used to generate the shared keys for the sensor nodes.

1.4 Polynomial-Based Key Generation

The polynomial is an expression contains variables, coefficients, and operations (Hammack, 2013). There are several types of polynomials like the univariate polynomial (contains only one variable), bivariate polynomial (contains two variables), and the trinomial polynomial (contains three variable) (Hoffmann, 2011). The polynomial function can be expressed using the summation notation, such as $p(x) = \sum_{k=0}^{n} a_k\, x^k$. If the variables of the a polynomial can be interchanged, then this polynomial is a symmetric polynomial (Sziklai, 2013). The same polynomial can be obtained by any of its variables or its variables can be switched, such as $p(x,y) = x^2 + y^2$ is a symmetric polynomial, where any switching in its variables will give the same polynomial $p(x,y) = p(y,x)$ (i.e. $x^2 + y^2 = y^2 + x^2$).

The symmetric polynomial can be written in terms of a set of elementary symmetric polynomials$(e_1, e_2, \ldots, e_n)$ (Daoub, 2012). For example the elementary symmetric polynomials of the polynomial $p(x,y,z) = x^3yz + xy^3\, z + xyz^3$ are $\mathrm{e}_1(x,y,z) = xyz,\ e_2(x,y,z) = (x+y+z)^2, e_3(x,y,z) = -2(xy + xz + yz),$ so the polynomial *p(x, y, z)* can be written as $\mathrm{e}_1 * ({e_2}^2 - 2 * e_3)$. The symmetric polynomials can be expressed as $p(x\,,y) = \sum_{i,j=0}^{n} a_{ij}\, x^i y^j$ using the summation notation, where (a_{ij}) represents the coefficients of the elementary symmetric polynomials. The coefficients of the symmetric polynomials are generated from a finite field $\left(F_q\right)$ that contains a set of elements (e.g., $2^{20}, 2^{64}$, less or more according to the number of the variables of the polynomials). The number of the elements in the finite filed expresses the order of the field, such as a field of order (q) contains (q) elements $\{0,1, \ldots \ldots, q-1\}$ (Benvenuto, 2012).

1.5 Secure Communication Protocol

The DH protocol is a protocol that is used to securely exchange the secret keys between any two communicating parties to encrypt and decrypt a transmitted data without having any prior knowledge about each other (Joshi, Verma and Verma, 2015). The DH protocol does not provide the desired authenticity for the communicating parties, where it does not give any information about the identities of both parties (Kallam, 2015). Therefore, the protocol is used with three different authentication methods; the Digital Signature, Public-Key Encryption and Symmetric-Key encryption (Li, 2010). However, due to the limited power of the sensor nodes in the WSN, the Symmetric key is more suitable for the WSN compared to the other methods (Khuraijam and Radhika, 2013), where the digital signature method and the public key encryption employ asymmetric cryptographic. The asymmetric cryptographic uses two keys; a public key and a private key, which requires higher computational cost (Gundimeda, 2014). In addition to that, the public key encryption does not provide the required authentication, where the message can be sent by any sensor node that has access to the receiver's public key (Blumenthal, 2007). Due to the asymmetric key encryption limitations, the symmetric key encryption method where both of the sensor nodes have the same key for encryption and decryption will be chosen in this work to provide a mutual authentication between communicating parties. In addition to the symmetric key encryption method, the DH protocol will be chosen to provide the sensor nodes a secure route to exchange the appropriate information within the network and establish their shared keys as presented in Figure 1.4.

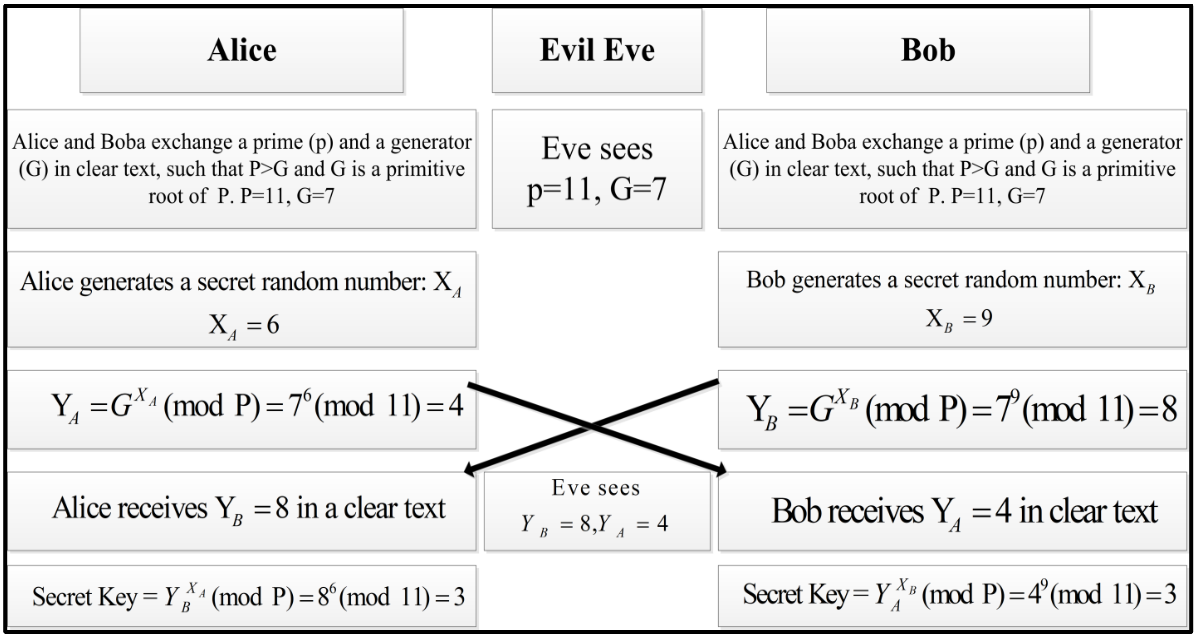

Alice
Evil Eve
Bob
Alice and Boba exchange a prime (p) and a generator (G) in clear text, such that P>G and G is a primitive root of P. P=11, G=7
Eve sees p=11, G=7
Alice and Boba exchange a prime (p) and a generator (G) in clear text, such that P>G and G is a primitive root of P. P=11, G=7
Alice generates a secret random number: X_A
$X_A = 6$
Bob generates a secret random number: X_B
$X_B = 9$
$Y_A = G^{X_A} \pmod{P} = 7^6 \pmod{11} = 4$
$Y_B = G^{X_B} \pmod{P} = 7^9 \pmod{11} = 8$
Alice receives $Y_B = 8$ in a clear text
Eve sees
$Y_B = 8, Y_A = 4$
Bob receives $Y_A = 4$ in clear text
Secret Key $= Y_B^{X_A} \pmod{P} = 8^6 \pmod{11} = 3$
Secret Key $= Y_A^{X_B} \pmod{P} = 4^9 \pmod{11} = 3$

Figure 1.4: Diffie Hellman (DH) Key Exchange (Li, 2010)

Despite that the DH protocol provides a secure route for the sensor nodes to exchange their information, it does not provide the desired authenticity for the sensor nodes (Li, 2010). In order to provide the needed authenticity in WSN, the message authentication code (MAC) is normally employed to provide the authenticity and the integrity to the transmitted messages between the sensor nodes within the same network (Chowdhury and Dasbit, 2015). The MAC is a small piece of information which is generated by the sender of the message and verified by the receiver of the message as shown in Figure 1.5. This code can be built and calculated using any block cipher that is suitable for the WSN in terms of computation for encryption and decryption, speed, memory usage, and the security against any type of attack (Rehman *et al.*, 2012).

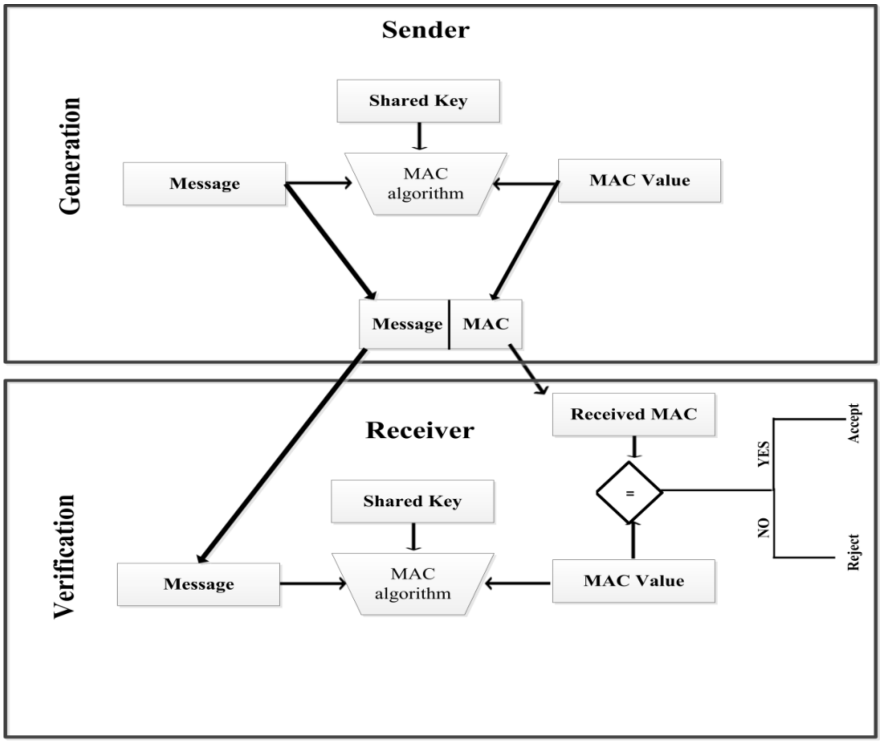


Figure 1.5: The Message Authentication Code (MAC)

1.6 Problem Statement

This section presents several aspects of the research; its main purpose, its motivation, the research questions, and the importance of this research. In addition, this section represents a set of aspects that will not be covered in this research which will be the limitations of the scope of this thesis.

1.6.1 Research Purpose

Since the data transmitted between the sensor nodes in the WSN are exposed to external attacks, controlling these sensor nodes are difficult and might be impossible. Therefore, the proposed approach aims to help the sensor nodes to transmit the data between them securely and guarantee the integrity and the privacy of these data. The proposed approach should also grant these sensor nodes to have the ability to establish a secure communication with the neighboring sensor nodes with enough key pairs and authenticate this communication.

1.6.2 Research Motivation

The wireless sensor network consists of a set of sensor nodes which are distributed randomly to a specific area to collect readings from the surroundings and transfer these readings between them until it reaches the base station. However, certain data has to be secret such as the readings from military deployments. Since the nature of the communication between sensor nodes in WSN is broadcast transmissions, the same key is used by a group of sensor nodes within the same range. Therefore, these communications are vulnerable to external and internal attacks and should be kept protected especially if the attacker manages to compromise a few sensor nodes or manage to intercept the communication channel. Most of the recent approaches that are created to solve this problem based either on the Polynomial-Based or the Matrix-Based scheme. Other approaches tried to merge between them in order to increase the resilience of the network and to increase the level of the connectivity between the sensor nodes. The problem with these approaches is that every sensor node receives a number of entries equal to the number of the sensor nodes in the network. This problem leads to several issues, such as it limits the number of the sensor nodes in the network because each node needs to store a large number of entries in its limited memory and consumes high energy while transmitting these entries between each other in order to create their shared keys.

Therefore, the main goal of the proposed approach is to increase the resilience of the WSN and at the same time keep the connectivity high between the sensor nodes using the BLU-Decomposition method without affecting on the limitations of the sensor nodes such as their limited memory and their limited energy.

1.6.3 Research Questions

The main condition and challenge that is faced by the proposed approach is the security characteristic, such that the proposed approach must give the desired level of security for the wireless sensor network to protect its sensor nodes from the Sybil nodes. The security level and the efficiency of the proposed approach will be expressed by answering the following questions:

1. Is the Block LU- Decomposition approach secure?
2. How many Sybil nodes does the adversary need in order to break the entire network?
3. How many messages does the adversary need to intercept in order to break the used scheme?

1.6.4 Research Significance

Most of the known key pre-distributing approaches are based on pre-distributing the keying information to the sensor nodes before they are deployed into the field. Thus, the base station does not need to transfer this information through the network. In this regards, the main characteristics of the proposed approach that will distinguish it from other existing approaches are:

1) The proposed approach employs the polynomial-based key pre-distribution to ensure a better resilience to the WSN due to the t-security property of the polynomials, where (t) is the number of sensor nodes the adversary needs to compromise (disclose) to break the entire network. The resilience means that, if the compromised a set of sensor nodes, the proposed approach should ensure that the other sensor nodes are not disclosed. (Dai and Xu, 2010; Banaie *et al.*, 2015).
2) The proposed approach is based on the Block LU-Decomposition procedure to ensure that any neighboring pair of sensors can find a common key. In the proposed approach, the matrix is decomposed into two matrices; a block lower triangular matrix (BL) and a block upper triangular matrix (BU), each node receives one row from the (BL) matrix and one column from the (BU) matrix. Any pair of neighboring nodes want to communicate they can exchange their rows, then each node multiplies its column with the other node's row to obtain a shared key. Since the matrix is symmetric, then the multiplication of this matrix will ensure that the two communicating parties will get the same shared key (Dai and Xu, 2010).
3) The proposed approach employs the block matrix, where public matrix will be divided into four blocks of the same sizes to make it harder for the adversary to guess the right block that is used for establishing the key pair between the

sensor nodes. So there will be more options for the adversary which will increase the resilience of the sensor nodes.

1.6.5 Research Limitation

The proposed algorithm will focus only on the sensor nodes; how they can establish a secure communication link between each other after the deployment taking into consideration its limited resources, such as its computational capabilities and its memory capacity. The base station (setup server) and its limitations (i.e. like its computational capabilities or its memory capacity), will not be considered in this thesis. In addition to that, the encryption and the authentication algorithms will not be mentioned in details in this thesis because the proposed approach focuses only on how the sensor nodes can generate secret shared keys between each other.

Chapter Two

Literature Review

Due to the random distribution of the sensor nodes in the WSN especially in the uncontrolled environment, it is very difficult to distribute the pairwise keys to these nodes before the deployment process. So, a key management scheme is needed to solve the key agreement problem by establishing a secure transmission medium with the neighboring nodes after the deployment. Several key management schemes have been used in the literature to solve this problem by taking into account the nature of the wireless communication in the WSN and the limited resources of the sensor nodes.

The recent schemes that have been used to solve the key agreement problem are classified into several types, some of these types are the Probabilistic techniques (Random techniques), the Deterministic techniques, the Polynomial–Based techniques and Matrix-Based techniques.

2.1 Probabilistic Key Pre-Distribution Scheme

The main reason for the probabilistic key pre-distribution scheme is to solve the resilience problem in the Single Master Key pre-distribution scheme and the storage problem in the pairwise key pre-distribution scheme (Yum and Lee, 2012). It is also called random scheme because the keys are randomly distributed to the sensor nodes. The first random approach refers to Eschenauer and Gligor (Eschenauer and Gligor, 2002) which is called the Basic scheme. In this scheme, the keys are randomly distributed to the sensor nodes from a common space called key pool, and each one of the sensor nodes receives a certain number of keys called key chain. Any pair of nodes wants to communicate they must have at least one common key.

The problem with the basic scheme is that the same key may be used by several pairs of nodes, so it has a low resilience. The enhanced random scheme that is proposed in (Du *et al.*, 2005) merged the Basic scheme with the Blom's scheme that is presented in (Blom, 1985) to improve the resilience of the WSN without using more memory. In this enhanced random scheme, Each sensor node receives (τ) key information selected randomly from (ω) key spaces and any neighboring nodes use the same space can communicate. Otherwise, they can communicate by finding another path which is called the key path through other nodes. The main result of this approach is the threshold value. This value represents the number of sensor nodes the adversary needs to compromise in order to be able to break the entire network.

The key management scheme that is presented in (Mu and Li, 2014) enhances the resilience of the previous random schemes by using different security level for each sensor node. In this approach, each sensor node generates an integer number (ι) represents the security level for it, after that it randomly chooses (m) keys from the key pool that is generated from the base station in the initialization phase and then performs (ι) times hash function operation on these keys to generate (m) new keys. In (Zhu *et al.*, 2016) the researchers enhances the security level of the network by increasing the number of pairwise keys between each pair of nodes. The key pool (i.e. the whole keys in the network) and the sensor nodes are divided into equal size groups. The sensor nodes are divided into groups according to their expected location. Each group of sensor nodes receives a set of keys from the corresponding key pool.

All previous schemes face a challenge to achieve high network connectivity and high resilience, where the same key may be used by more than one pair of nodes. In these schemes, the neighboring nodes might not be able to find a shared key because of the random distribution of the keys. In addition to that, these approaches require large memory to store more keys to increase the network connectivity and to reduce the ability to duplicate the shared keys that can be distributed to the sensor nodes by the base station. Table 2.1 shows a summary for the literature review of the probabilistic key pre-distribution approaches.

Table 2.1: Summary Table For Literature Review of The Probabilistic Key Pre-Distribution Schemes

Author Name & Year of Publication	The Technique Name	Main Reasons	Main Results
(Eschenauer and Gligor, 2002)	The Basic Scheme (E-G Scheme)	• Solve the memory overhead problem of the pairwise scheme. • Solve the resilience problem of the Single Master key scheme.	• Increased the resilience of the Single Master key pre-distribution. • Decreased the memory overhead of the Pairwise scheme.
(Du *et al.*, 2005)	Enhanced Random Scheme	• Increase the resilience of the WSN without using more memory.	• Improved the resilience of the random schemes. • Exhibited λ-secure property.
(Mu and Li, 2014)	Pairwise Key Pre-distribution Scheme	• Increase the resilience against node capture.	• offered a stronger resilience against node capture.
(Zhu *et al.*, 2016)	Improved Random Key Pre-Distribution Technique	• Improve the resilience and the network connectivity of the random key pre-distribution scheme.	• Increased the resilience of the random schemes. • Increased the network connectivity of the random schemes.

2.2 Polynomial-Based Key Pre-Distribution Scheme

The polynomial-based scheme was proposed to increase the resilience of the WSN. The basic polynomial-based key pre-distribution scheme was proposed in (Blundo *et al.*, 1998). This scheme goes through two steps; step (1): before the deployment process, the base station generates a unique identifier for each sensor node, then it generates a symmetric bivariate t −degree polynomial $f(x, y)$ with coefficients from F_q, step (2): the base station distributes the polynomials as polynomial shares to the sensor nodes. After the deployment, the sensor nodes start communicating to compute the shared keys.

In order to increase the resilience of the WSN against node capture, Banaie et al (Banaie *et al.*, 2015) have added the matrices to the polynomial–based key pre-distribution scheme. In this approach, the base station generates a set of multivariate polynomials from a polynomial pool to construct symmetric matrices. Then, each sensor node receives one row from a set of these matrices based on its identities and the identity of the matrices that are pre-loaded to it. The researchers in this approach claim that they increase the resilience of the WSN without increasing the communication overhead or the memory overhead. In (Zhang, Li and Li, 2018), the researchers have merged polynomial–based key pre-distribution scheme with the probabilities key pre-distribution approaches. In this approach, each sensor node receives two parts of information; the first part of the preloaded information is polynomial shares (i.e. $f(ID(i), x)$ for $i=1,2,\ldots,N$), and the second part is key generated by the preloaded polynomial shares.

In (Mahmood, Ning and Ghafoor, 2017), the researchers focused on how to reduce the number of the messages transmitted between the sensor nodes and at the same time increase the security level of the sensor nodes. The researchers focused on the hierarchal distribution of the sensor nodes (i.e. the sensor nodes are divided into groups and each group has a cluster head). Each group receives two symmetric polynomials; one is used between the sensor nodes within the same group (inter-sensor nodes polynomial), while the other one is used between the groups in the same network (intergroup polynomial). To increase the security of the transmitted messages each node receives a set of symmetric keys before the deployment to exchange its identities securely after the deployment. Table

2.2 shows a summary for the literature review of the polynomial-based key pre-distribution schemes.

Table 2.2: Summary Table For Literature Review of The Polynomial-Based Key Pre-Distribution Schemes

Author Name & Year of Publication	The Technique Name	Main Reasons	Main Results
(Blundo *et al.*, 1998)	Basic Polynomial-Based Key Pre-Distribution Protocol	• Allows a group of users of a given size (a dynamic conference) to be able to compute a common secure key.	• Modeled, analyzed, and designed dynamic optimal conference key distribution scheme.
(Banaie *et al.*, 2015)	Polynomial-Based Pairwise Key Management Scheme	• Increase the resilience of the WSN. • Enhance the storage usage of the sensor nodes.	• Decreased the memory overhead. • Increased the resilience of the WSN.
(Mahmood, Ning and Ghafoor, 2017)	A Polynomial Subset-Based Multi-Party System	• Decrease the computational cost of the Polynomial-Based schemes.	• Increased the lifetime of the network. • Decreased the communication cost and memory overhead.
(Zhang, Li and Li, 2018)	Key Establishment Scheme Based on Polynomial and Random Key Pre-Distribution Scheme	• Increase the resilience of the WSN takes into account the limited resources of the sensor nodes.	• Increased the resilience against node capture attacks.

2.3 Deterministic Key Pre-Distribution Scheme

The deterministic key pre-distribution scheme was proposed to solve the network connectivity problem of the random approaches. Çamtepe and Yener in (Çamtepe and Yener, 2004) proposed the first deterministic approach to increase network connectivity without increasing the size of the key pool and the key chain. They used two classes of the combinatorial design; a symmetrical balanced design, in particular, the finite projective plane of order (q), where (q) is a prime power to generate symmetric design. The second class is Generalized Quadrangles (GO) design to represent the WSN.

Sanchez and Baldus in (Sanchez and Baldus, 2005) proposed another approach to enhance the scalability and the resilience of the previous approach. In this approach, the researchers used the combinatorial theory to distribute a number of bivariate polynomials to the sensor nodes. Any node wants to communicate with the neighboring nodes must evaluate its shared polynomials using a specific index according to its block and the identity of the other node. The researchers in (Khandke *et al.*, 2013) used asymmetric matrices to increase the resilience. The base station in this approach generates a positive integer and a base of form $\{1, t^1, t^2, \ldots.., t^{n-1}\}$. Then, the base station generates a set of keys from the key pool to build a secret information and distribute both of the information and the base (t) to the sensor nodes. Each one of these sensor nodes has two keys. So each pair of nodes will have two separate links to communicate. In this case, if the adversary compromises one of these links, there will be another secure link.

Another deterministic approache that is proposed in (Anzani, Javadi and Moeni, 2018). In this approach, the researchers based on multivariate polynomials and the combinatorial design. The multivariate polynomials are distributed to the sensor nodes before the deployment based on both of its identities and the combinatorial design. The problem with this approach is that the resilience is constant and cannot be increased, where it requires an increase in the number of the independent key path to establish an indirect key to increase the security level that may increase the communication overhead. Table 2.3

shows a summary for the literature review of the deterministic-based key pre-distribution schemes.

Table 2.3: Summary Table For Literature Review of The Deterministic Key Pre-Distribution Schemes

Author Name & Year of Publication	**The Technique Name**	**Main Reasons**	**Main Results**
(Çamtepe and Yener, 2004)	(C-Y scheme)	• Solve the connectivity problem of the probabilistic approaches.	• Increased the network connectivity without increasing the size of the key pool and the key chain.
(Sanchez and Baldus, 2005)	A Deterministic Pairwise Key Pre-Distribution Scheme	• Solve the existence and the authentication problems of Ç-Y scheme • Solve the scalability problem of the C-Y scheme and the Blundo's protocol.	• Increased the scalability of the C-Y Scheme. • Solved the existence problem of the C-Y Scheme. • Solved the mutual authentication of the Deterministic techniques.
(Khandke *et al.*, 2013)	A Novel Deterministic Key Pre Distribution Scheme	• Improve the performance and the resilience of the sensor nodes.	• Improved the resilience of the sensor nodes against the sensor nodes captured. • Increased the scalability and the flexibility of the WSN.
(Anzani, Javadi and Moeni, 2018)	Deterministic Key Pre-Distribution Method Based on Hypercube Multivariate Scheme.	• Improve the resilience of the WSN against sensor nodes captured.	• Improved the resilience of the sensor nodes.

2.4 Matrix-Based key Pre-Distribution Scheme

The matrix–based scheme always ensure that any pair of nodes can communicate and find a common key because of the symmetric matrices that are used (Dai and Xu, 2010). Several algorithms in the WSN depend upon the (L) and (U) matrices to help the sensor nodes to build secure links. The algorithms that based on this scheme aim to achieve high resilience, reduces the communication overhead, and to increase the network connectivity.

In (Dai and Xu, 2010), The researchers have used the polynomial-based key pre-distribution scheme and the matrix-based to increase the resilience of the sensor nodes against node capture attacks and to achieve high network connectivity. In this approach, the researchers used the LU-Decomposition method to decompose a symmetric matrix of bivariate polynomials (i.e. $A = L \times U$, where (A) is a symmetric matrix, (L) is the lower triangular matrix, and (U) is the upper triangular matrix). The base station generates a set of bivariate symmetric polynomials randomly from a polynomial pool to establish a lower triangular matrix (L) and uses this matrix to construct an upper triangular matrix (U) and the original matrix, then it distributes the (L) and (U) elements to the sensor nodes to establish a secure communication. The researchers in (Banaie *et al.*, 2014) have also added the matrices to the polynomial-based scheme to increase the resilience of the WSN. In this approach, the base station generates random prime numbers and a set of bivariate polynomials to build symmetric matrices. Each sensor node receives a single row from each matrix based on its identity, then the sensor nodes use the pre-loaded information to establish a secure communication with the neighboring nodes.

In addition to achieve a high resilience, other approaches have focused on the communication overhead between the sensor nodes, as in (Tharani, Suganthi and Srinithi, 2014). They proposed a new key management scheme that allows the sensor nodes to generate shared keys by exchanging only one message containing one element from a column from a symmetric matrix and the node ID. Each sensor node receives a single row from the symmetric matrix (A) and one element from another symmetric matrix (G) according to its ID value, such that node (i) receives row (i) from the first matrix and one element from the second matrix (G). After the deployment of nodes, in order to calculate the shared keys, each node sends its ID and its column to the neighboring nodes in its range. Once the node receives the ID and the element of the other nodes, it calculates a column of the second matrix from the received elements, by raising it to k^{th}element. Then, each node calculates the secret key by multiplying its row by the other node column.

The researchers in (Choi, Kim and Youn, 2013) focused on network connectivity rather than the resilience and the communication overhead. In this approach, the researchers based on generating a number of keys to build a symmetric matrix. Next, the base station derives the values of the eigenvalues and eigenvectors from this matrix and distributes them as keys for the sensor nodes to ensure that any pair of sensor nodes can find a common key. Table 2.4 shows a summary for the matrix-based schemes represented in this section with the main reasons and results of each approach.

Table 2.4: Summary Table For Literature Review of The Matrix-Based Key Pre-Distribution Schemes

Author Name &The Year of Publication	The Technique Name	Main Reasons	Main Results
(Dai and Xu, (2010	Matrix-Based Key Pre-Distribution Scheme	• Increase the resilience and the connectivity of the sensor nodes.	• Achieved a strong resilience against node captured.
(Choi, Kim and Youn, 2013)	An Efficient Key Pre-Distribution Scheme	• Improve the WSN security.	• Reduced the communication and memory overhead required for secure connectivity. • Decreased the energy consumption of the sensor nodes to establish a secure link.
(Tharani, Suganthi and Srinithi, 2014)	Matrix-Based Key Pre-Distribution Scheme	• Reduce the communication overhead. • Increase the network connectivity. • Protect the information stored in the sensor nodes.	• Achieved greater network connectivity and scalability.
(Banaie et al., 2014)	A Matrix-Based Pairwise Key Management Scheme	• Increase the resilience of the WSN. • Decrease the storage complexity of the sensor nodes	• Increased the storage efficiency. • Increased the resilience of the sensor nodes. • Achieved a high network connectivity with a low memory overhead.

For several decades, the researchers have tried to find efficient key management schemes to protect the sensor nodes and the transmitted data between them. Some of these schemes appeared to solve specific problems that are found in older schemes. Table 2.5 shows the main weakness and strengths of the four schemes that have been presented in this chapter.

Table 2.5: Comparison Between The Different Types of The Key Pre-Distribution Schemes

The Pre-Distribution Scheme Type	Its Purposes	Its Strength	Its Weakness
The Probabilistic Key Pre-Distribution Scheme (Random Approach)	• Increase the scalability of the network. • Increase the network connectivity. • Reduce the storage complexity of the traditional approaches.	• Allows network scalability. • Reduces the storage complexity of the sensor nodes.	• Does not guarantee strong resilience. • Does not provide strong connectivity. • May require a large space to increase the network connectivity. • Cannot provide the desired authentication where more than a pair of nodes may use the same key.
Deterministic Key Pre-Distribution Scheme	• Solve the network connectivity of the probabilistic approaches.	• Has a good network connectivity. • Has a good network resilience. • Has low computation overhead. • Has low communication overhead.	• Low network connectivity. • limited scalability • poor authenticity.
Polynomial-Based Key Pre-Distribution Scheme	• Increase the resilience of the network. • Reduce the communication overhead.	• High resilience • low communication overhead. • t-security property.	• Low network connectivity. • Required a Large storage space.
Matrix-Based Key Pre-Distribution Scheme	• Increase the network connectivity.	• High network connectivity.	• low resilience.

The proposed approach uses both the Matrix–Based Key Pre-Distribution scheme and the Polynomial-Based Key Pre-Distribution scheme together with the BLU-Decomposition method. The Polynomial-Based scheme provides mutual authentication between the sensor nodes and a strong resilience of the WSN against node capture, while the Matrix-Based scheme provides a full connectivity between the sensor nodes in the same network.

Chapter Three

Research Methodology

This chapter presents a new key management scheme in WSN that allows the sensor nodes to establish a secure communication link between each other. In order to protect the transmission medium between the sensor nodes, the proposed approach based on the Polynomial-Based Key Pre-Distribution scheme, Matrix-Based Key Pre-Distribution scheme, and the Block LU-Decomposition algorithm. The main advantage of the proposed approach is the use of BLU- Decomposition algorithm, this algorithm allows the proposed approach to obtain a high resilience against node capture, full connectivity between the sensor nodes, and low memory overhead because it decreases the amount of pre-loaded information to the sensor nodes which, therefore, decreases the amount of the sensor nodes energy consumption.

The proposed approach goes through three main phases according to the key pre-distribution protocol:

Phase I: A large pool of polynomials is to be built by generating a large number of bivariate t-degree symmetric polynomials over a finite Field (F_q) (e.g. 2^{20}, 2^{64}...).

Phase II: The second phase comes before distributing the sensor nodes to the target area and this phase consists of five steps:

Step 1: Select a set of bivariate $t-$degree symmetric polynomials from the polynomial pool randomly to build a symmetric matrix.

Step 2: Divide the matrix into four blocks of the same size using the BLU-Decomposition algorithm.

Step 3: Apply the BLU-Decomposition algorithm to the matrix blocks to produce the block lower triangular matrix (BL) and the block upper triangular matrix (BU).

Step 4: Re-build the last block of the matrix (i.e. A_{22}) by multiplying the (L_{21}) from the (BL) matrix and (U_{12}) from the (BU) matrix.

Step 5: Distribute the (BL) and (BU) matrices elements (i.e. polynomials) of the first block (A_{11}) and the last block (A_{22}) and use the other blocks $(A_{12}\ and\ A_{21})$ as shared spaces (i.e. blocks) between the sensor nodes that belong to different spaces.

Phase III: This phase includes establishing the pairwise keys between the neighboring nodes after the deployment. Figure 3.1 shows the main research procedural phases of the proposed approach.

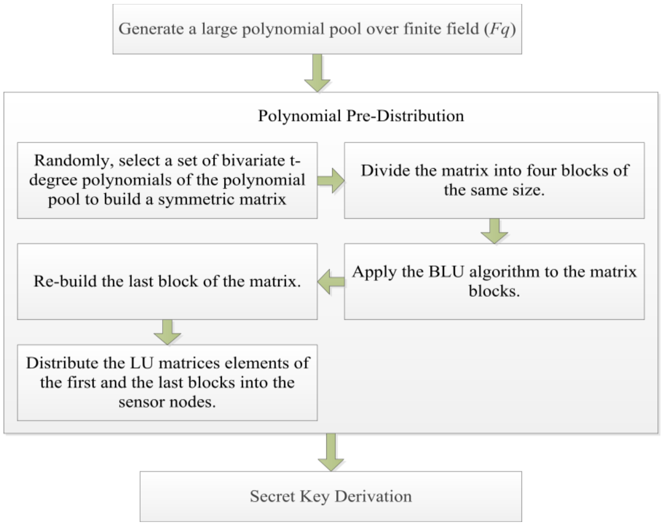

Generate a large polynomial pool over finite field (*Fq*)
Polynomial Pre-Distribution
Randomly, select a set of bivariate t-degree polynomials of the polynomial pool to build a symmetric matrix
Divide the matrix into four blocks of the same size.
Apply the BLU algorithm to the matrix blocks.
Re-build the last block of the matrix.
Distribute the LU matrices elements of the first and the last blocks into the sensor nodes.
Secret Key Derivation

Figure 3.1: The Main Research Procedural Phases of The Proposed Approach

3.1 Research Phases

This section illustrates each phase of the main phases of the proposed approach in terms of how they are formed along their main steps.

3.1.1 Phase I: Generating a Large Polynomial Pool

In this Phase, the base station generates a large number of bivariate $t-$degree symmetric polynomials (e.g. 2^{20}, 2^{64} or more) over a finite field $(\mathrm{F_q})$. Equation (2) shows the form of the bivariate $t -$ degree polynomial.

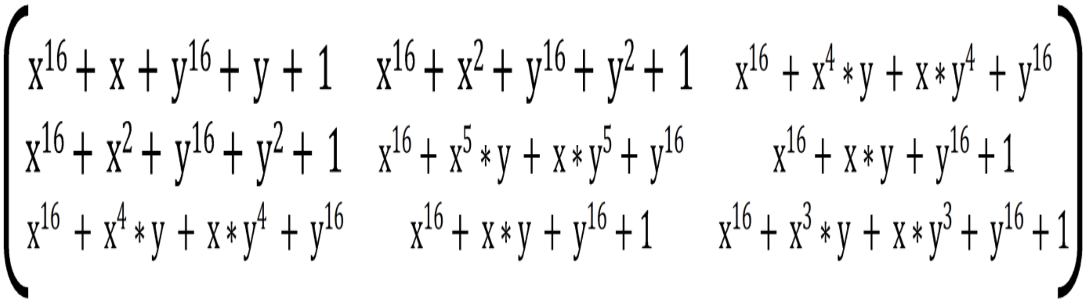
$$f\ (x\ ,y\ ) = \sum_{i\ ,j\ =0}^{t} a_{ij} x^{\ i} y^{\ j} = a_{t\ 0}x^{\ t} + a_{(t\ -1)1}x^{\ (t\ -1)}y\ + \ldots + a_{1(t\ -1)}xy^{\ (t\ -1)} + a_{0t}\ y^{\ t} \qquad (2)$$

The prime number (q) of the finite field should be large enough, where each sensor node can have a set of shared pre-loaded polynomials. The function $f(x, y)$ represents a symmetric polynomial such that $f(x, y) = f(y, x)$. Each polynomial has two variables (x, y) and $(t + 1)$ integer coefficients (i.e. a_{ij}) that are randomly selected from (F_q).

3.1.2 Phase II: Polynomial Pre-Distribution

Due to the random distribution of the sensor nodes into the target area, none of them can know or guess which of the sensor nodes can fall within its own range. Therefore, this phase focuses on the preparation of the desired sensor nodes keying information and distribute this keying information to them before the deployment of the sensor nodes to keep it secret from the adversary. This pre-loaded information is used later by the sensor nodes to generate shared keys in order to establish secure communication links between each other. Computing shared keys enable the sensor nodes to identify the authenticity of the neighboring nodes, hence protecting the transmitted data from any adversary. This phase begins when the base station starts generating bivariate polynomials from the polynomial pool until the sensor nodes become ready to receive their keying information, as follows:

Step 1: Select a set of bivariate symmetric polynomials of the polynomial pool to construct a symmetric matrix of $(\mathrm{N} \times \mathrm{N})$ dimension, where (N) is the number of the sensor nodes in the network. Figure 3.2 shows a (3x3) symmetric polynomial matrix over a finite field of order (2^{16}).



$$\begin{pmatrix} x^{16}+x+y^{16}+y+1 & x^{16}+x^{2}+y^{16}+y^{2}+1 & x^{16}+x^{4}*y+x*y^{4}+y^{16} \\ x^{16}+x^{2}+y^{16}+y^{2}+1 & x^{16}+x^{5}*y+x*y^{5}+y^{16} & x^{16}+x*y+y^{16}+1 \\ x^{16}+x^{4}*y+x*y^{4}+y^{16} & x^{16}+x*y+y^{16}+1 & x^{16}+x^{3}*y+x*y^{3}+y^{16}+1 \end{pmatrix}$$

Figure 3.2: Example of Polynomial Matrix over a Finite Field of Order (2^{16})

Step 2: Divide the symmetric matrix into (4) blocks of the same size of $(r \times r)$ dimension, Where $(r = \frac{N}{2})$.

Step 3: Apply the BLU algorithm that is presented in (Pathan, Dai and Hong, 2006) to the matrix blocks in step 2 to produce the (BL) and (BU) matrices. The matrix blocks can be written as follows:

- $A_{11} = BL_{11} * BU_{11}$
- $A_{12} = BL_{11} * BU_{12}$
- $A_{21} = BL_{21} * BU_{11}$
- $A_{22} = BL_{21} * BU_{12} + BL_{22} * BU_{22}$

Where the blocks (BL_{12}) and (BU_{21}) are equal to (0), the diagonal elements of the blocks (BL_{11}) and (BL_{22}) are equal to (1). Figure 3.3 illustrates how the BLU algorithm works assuming that there is a 4×4 symmetric matrix, where $r = 2$ and $N = 4$, then,

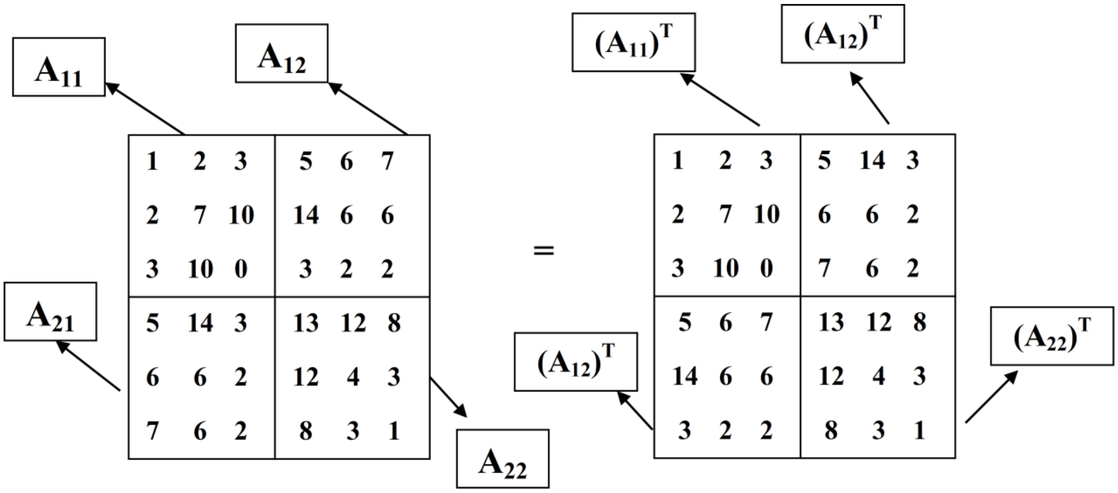
$$\left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ \hline l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{array}\right] * \left[\begin{array}{cc|cc} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ \hline 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{array}\right]$$

Figure 3.3: (4×4) Symmetric Matrix

Step 4: Re-build the last block (A_{22}) by multiplying the blocks $(\mathrm{B}L_{21})$ by (BU_{12}) to reduce the amount of information that can be distributed to the second half of the sensor nodes and at the same time to link the first half of the sensor nodes with the second half of them. So the last block can be written as shown in Equation (3).

$$A_{22} = BL_{21} * BU_{12} \qquad (3)$$

Step 5: Select the first block (A_{11}) and the last block (A_{22}) of the matrix and distribute the (BL) and (BU) matrices elements of them to the sensor nodes because they are symmetric. The other blocks (A_{12} *and* A_{21}) are not symmetric but each one of them represents the transpose of the other block, such that $A_{12} = (A_{21})^T$and vices versa as shown in Figure 3.4. So, the blocks (A_{12} *and* A_{21}) are used as shared blocks between the first and the last block.



$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 5 & 6 & 7 \\ 2 & 7 & 10 & 14 & 6 & 6 \\ 3 & 10 & 0 & 3 & 2 & 2 \\ \hline 5 & 14 & 3 & 13 & 12 & 8 \\ 6 & 6 & 2 & 12 & 4 & 3 \\ 7 & 6 & 2 & 8 & 3 & 1 \end{array}\right] = \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 5 & 14 & 3 \\ 2 & 7 & 10 & 6 & 6 & 2 \\ 3 & 10 & 0 & 7 & 6 & 2 \\ \hline 5 & 6 & 7 & 13 & 12 & 8 \\ 14 & 6 & 6 & 12 & 4 & 3 \\ 3 & 2 & 2 & 8 & 3 & 1 \end{array}\right]$$

Figure 3.4: The Transpose of The Matrix (A)

The distribution of the (BL) and (BU) elements to the sensor nodes is done through two important steps:

Step 1: Select one row of (BL) matrix and one column of (BU) matrix, such that if the row (i) of (BL) is distributed to sensor node (i), then the column (i) of (BU) must be distributed to node (i) as well to calculate the same shared key. Each sensor node calculates the shared key based on the other sensor nodes rows. Figure 3.5 illustrates why each sensor node must receive the same order of the row and the column and how the communication nodes can obtain the same key locally.

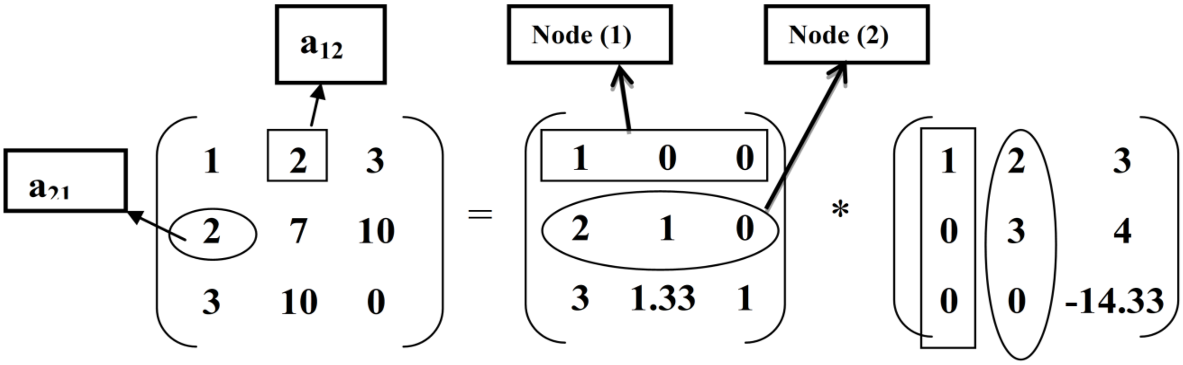


Figure 3.5: Calculate The Shared Key

Suppose that there are two sensor nodes, the first one takes the first row from BL_{r1} (1, 0, 0) and the first column from BU_{c1} (1, 0, 0). The second sensor takes the second row from BL_{r2} (2, 1, 0) and the second column from BU_{c2} (2, 3, 0). When they exchange their rows they can obtain the same key $a_{12} = a_{21}$, where $a_{12} = 1 * 2 + 0 * 3 + 0 * 0 = 2$, $a_{21} = 2 * 1 + 1 * 0 + 0 * 0 = 2$.

Step 2: The polynomials are distributed to the sensor nodes as shared polynomials, such that for each sensor node the base station replaces the bivariate t −degree polynomials $f(x, y)$ by univariate $t-$ degree polynomials by computing $f(i, y)$, where $(i = 1,\ 2, \dots,\ N)$ are considered as the IDs of the sensor nodes.

3.1.3 Phase III: Secret Key Derivation

Up to this stage, the sensor nodes are deployed into their field and equipped with the needed information to establish a secure communication with the neighboring sensor nodes. To illustrate the process, suppose that there are two sensor nodes want to communicate with each other, the first node is (N_a) contains the row (BL_i), the column (BU_i) and ID (I_a). The second node is (N_b) has the row (BL_j), the column (BU_j) and ID (I_b), they can exchange their information as illustrated in Figure 3.6.

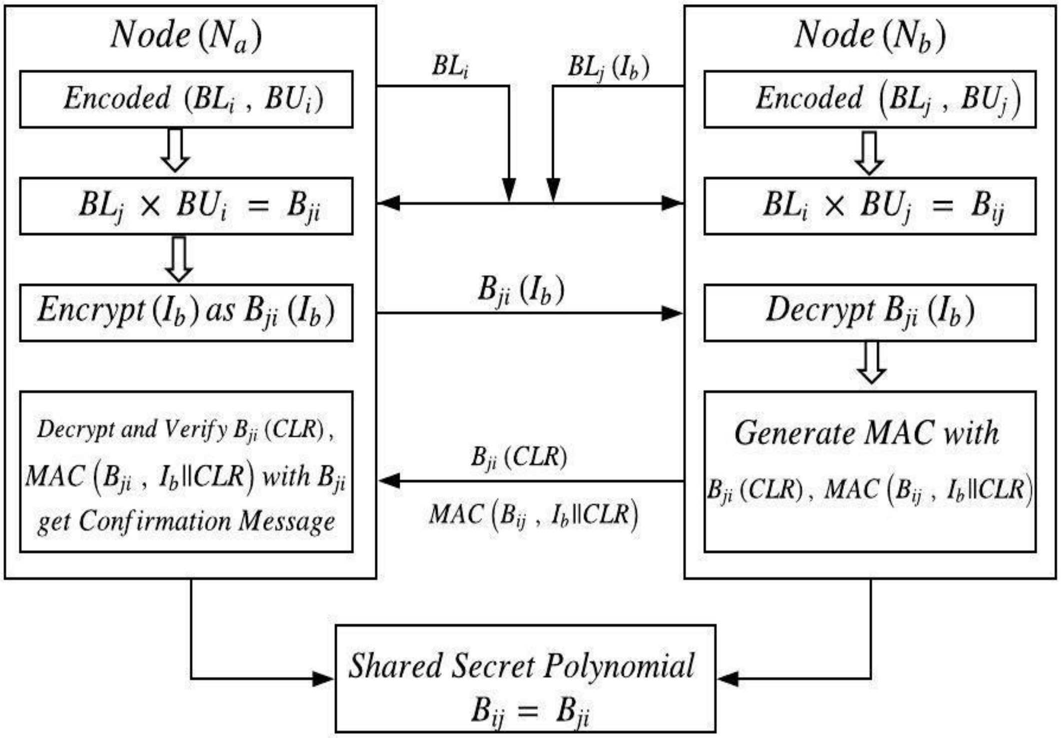

$Node\,(N_a)$
$Encoded\ (BL_i\ ,\ BU_i)$
$BL_j \times BU_i = B_{ji}$
$Encrypt\,(I_b)\,as\ B_{ji}\,(I_b)$
$Decrypt\ and\ Verify\ B_{ji}\,(CLR)$,
$MAC\,(B_{ji}\ ,\ I_b \| CLR)\ with\ B_{ji}$
$get\ Confirmation\ Message$
BL_i
$BL_j\,(I_b)$
$B_{ji}\,(I_b)$
$B_{ji}\,(CLR)$
$MAC\,(B_{ij}\ ,\ I_b \| CLR)$
$Node\,(N_b)$
$Encoded\ (BL_j\ ,\ BU_j)$
$BL_i \times BU_j = B_{ij}$
$Decrypt\ B_{ji}\,(I_b)$
$Generate\ MAC\ with$
$B_{ji}\,(CLR),\ MAC\,(B_{ij}\ ,\ I_b \| CLR)$
$Shared\ Secret\ Polynomial$
$B_{ij} = B_{ji}$

Figure 3.6: Exchange The Keying Information Between The Sensors Using DH Algorithm (Dai And Xu, 2010)

To exchange the secret keying information between the sensor node (N_a) and the sensor node (N_b), the DH protocol is used. So the derivation process of the shared key (i.e. polynomial) is performed as follows:

1) First, the sensor node (N_a) broadcasts its own ID (I_a) to its neighbors in its range. Let the sensor node (N_b) is one of its neighboring nodes. After receiving the ID of node (N_a), node (N_b) sends an acknowledgment message to node (N_a) to tell it that it has received its message and it is ready to establish a secure communication with it.
2) The node (N_a) sends its row (BL_i) to the sensor node (N_b). Then node (N_b) evaluates the shared polynomials as $f(I_b, I_a)$ using the ID of the node (N_a) and its own ID. After that, its multiply its column $(\mathrm{BU_j})$ by the received row to obtain a secret key as follows:

$$\mathrm{BL_i} * \mathrm{BU_j} = B_{ij} \tag{4}$$

3) Node $(\mathrm{N_b})$ sends its row $(\mathrm{BL_j})$ with its ID to the sensor node $(\mathrm{N_a})$.
4) Node (N_a) evaluates the shared polynomials as $f(I_a, I_b)$ using the ID of the node (N_b) and its own ID. After that, its multiply its column (BU_i) by the received row to obtain a secret key as follows:

$$\mathrm{BL_j} * \mathrm{BU_i} = \mathrm{B_{ji}} \tag{5}$$

5) The sensor $(\mathrm{N_a})$ sends the $(\mathrm{N_b})$ ID $(\mathrm{I_b})$ to $(\mathrm{N_b})$ encrypted using the secret polynomial $(\mathrm{B_{ji}})$.

6) The node (N_b) uses the polynomial (B_{ji}) to decrypt (I_b) that was sent from the sensor node (N_a). If (I_b) is decrypted, then the sensor nodes (N_a) and (N_b) are having the same polynomial and authenticated to each other.

7) Node (N_b) uses the (B_{ij}) polynomial to generate the message authentication code (MAC). It sends a key confirmation message encrypted using the (B_{ij}) to show that it has the same polynomial and it agrees with node (N_a) and $MAC\ (B_{ij}, I_b||\ CLR)$ to the node (N_a), where the message parameter of the MAC function combines the ID of the node (N_b) with the confirmation message (CLR) using the bitwise concatenation operator (||). Now both the nodes could generate a shared polynomial $(B_{ij} = B_{ji})$, and achieve node-to-node mutual authentication between each other.

Several symmetric encryption algorithms existed in the literature that can be used in WSN such as the Advanced Encryption Standard (AES) that was published by the National Institute of Standards and Technology (NIST) in 2000 and RC5 block cipher algorithm that is presented in (Menezes, Van Oorschot and Vanstone, 1996). However, these algorithms cannot be implemented using any hardware or software due to the key size, where the AES key must be at least (128) bits. In addition to that, the proposed approach does not focus on how to encrypt or authenticate the transmitted messages, it focuses on the keys that are used for both the encryption and the authentication process and how to keep these keys secure from the adversary as much as possible. Therefore, the proposed approach is based on a simple encryption algorithm that is presented in (Ursell, 2017). This algorithm is fast, the key size in this algorithm is less than (32) bits, it includes two functions; (1) Encoder function which is used to encrypt the plaintext, and the (2) Decoder function which is used to decrypt the cipher.

3.2 Evaluation Plan

The sensor nodes in the WSN suffer from the limited resources like the limited memory, limited energy, and the limited communication range. The key management scheme that is proposed in this thesis will be evaluated from different aspects according to the limitations of the sensor nodes. Each aspect will be represented by generating an equation using different formulas and rules of the probability theorems. To prove the efficiency of the proposed approach and to prove that it could make a great enhancement compared to the existing approaches from different aspects, the proposed approach will be compared to some of the existing approaches as shown in Table 3.1.

Table 3.1: The Comparison Plan

Limited Resource	Related Work	Reason to use
The Memory-Overhead	(Dai and Xu, 2010)	❖ Reduces the network-wide memory overhead by using an efficient encoding mechanism. ❖ Limits the number of polynomials that are required in each sensor node in order to establish secure communication with the neighboring nodes.
	(Harn and Hsu, 2015)	❖ Limits the storage space of each sensor node to have only $m(t + 1)$ coefficients, where (m) is the number of sensor nodes.
The Network Connectivity	(Mu and Li, 2014)	❖ Distributes (m) keys to the sensor nodes randomly.
	(Zhang, Li and Li, 2018)	❖ Distributes a set of shared polynomials and shared keys to the sensor nodes randomly in order to increase the network connectivity.
	(Dai and Xu, 2010)	❖ The approach used LU-Decomposition method that is similar to the BLU-Decomposition method that is used in the proposed approach.
The Resilience Against Node Capture	(Zhang et al., 2016)	❖ Forms between the polynomial pool and the keys pool which forces the adversary to crack both the polynomials coefficients and the keys simultaneously.
	(Dai and Xu, 2010)	❖ Merges between the LU-Decomposition method and the Polynomial-Based Key Pre-Distribution scheme like the proposed approach.
The Energy Consumption (Communication cost)	(Dai and Xu, 2010)	❖ The amount of transmitted keying information between the communicated nodes is limited to the number of sensor nodes in the network.

The evaluation of the memory overhead of the proposed approach based on two parts; (1) estimating needed storage space of each sensor node to store the pre-loaded information before and after applying an efficient encoding mechanism, and (2) the communication cost effect on the energy consumption of the proposed approach. The security analysis of the proposed approach will not only focus on the resilience against node capture, but it will also analyze the security performance in two additional terms; (1) the number of messages the adversary needs to intercept in order to break the proposed approach, and (2) the number of Sybil nodes the adversary needs to plant in the WSN to break the entire network.

3.3 Measurement; Formulas and Rules

As mentioned in the previous section, the evaluation of the proposed approach based on generating different equations. These equations will be compared to other equations that belong to other approaches in order to prove the efficiency of the proposed approach. This section shows the probabilities and theories that will be used to generate the desired equations to evaluate the performance of the proposed approach. The main rules and theorems that are used in the evaluation process are:

1) The summation notation. This method is used to find the summation of (n) terms as presented in (Hammack, 2013). The formula of this method is:

$$\sum_{i=1}^{n} a_i = a_1 + a_2 + \ldots + a_n \tag{6}$$

Where:

n :The number of terms, i: The index of the summation, a_i : The $\mathrm{i_{th}}$ term of the summation, 1: The lower bound of the summation, and n: The upper bound of the summation.

2) The Law of Complementary Events in Equation (7) represented in (Sahoo, 2013) is used to find all the possible outcomes that are not in a specific event.

$$P\left(A^c\right)=1-P\left(A\right) \tag{7}$$

Where:

(A): The event.

(A^c): The complement of the event (A).

3) The factorial rule in Equation (8) as presented in (Hammack, 2013) is used to find the number of different ways to arrange (n) objects, where (n) is a non-negative integer.

$$n! \tag{8}$$

4) The combination rule in Equation (9) is used to find the number of ways of selecting (r) objects from given (n) objects without interest to the order of the objects. (Guichard, 2016).

$$\binom{n}{r}=\frac{n!}{\left(n-r\right)!*r!} \tag{9}$$

5) Equation (10) represents the number of ways of dividing (n) distinct objects into (r) groups, and each group has (m) objects. This rule is called the combination rule with groups, where the number of the different ways of selecting objects regardless of the selection order is:

$$\binom{n}{r,m} = \frac{n!}{(m!)^r} \tag{10}$$

6) The probability of an event rule in Equation (11) is used to count the number of times a specific outcome can occurr compared to all possible outcomes (Sahoo, 2013). For example, the die has (6) possible sides (i.e. 1, 2, 3, 4, 5, 6), the number of times the number six can appear is $(\frac{1}{6})$ because there is only one (6) on the die and there are six possible outcomes.

$$pr(A) = \frac{\textit{The number of wanted outcome}}{\textit{The number of possible outcomes}} \tag{11}$$

7) The conditional probability in Equation (12) is used to calculate the probability of the event (A) under the condition that the event (B) has occurred. (Sahoo, 2013).

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \; P(B) \neq 0 \tag{12}$$

8) The Theorem of total probability in Equation (13) is used to calculate the probability of the event (B) of a sample space generated by a set of events $(A_n : n = 1, 2, ..., n)$. (L Hong, 2015). The formula of this rule is:

$$\Pr(B) = \sum_{i=1}^{n} \Pr(A_i)\, pr(B \mid A_i) \tag{13}$$

Where:

$A_n,\ n = 1,\ 2, ...,\ n$: It is a finite partition of a sample space, each event A_n is a measurable event.

9) The Multiplication rule in Equation (14) represents the probability that two events $(A) and\ (B)$ can occur at the same time, where (A) and (B) belong to the same sample space. (Guichard, 2016).

$$P(A \cap B) = P(A) * P(B \mid A) \tag{14}$$

10) The Additive rule in Equation (15) represents the probability that one of the two events $(A)\ or\ (B)$ can occur at the same time, where (A) and (B) belong to the same sample space. (Guichard, 2016).

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{15}$$

11) The chain of events rule in Equation (16) is used to calculate the probability that three events $(A,\ B,\ and\ C)$ can occur at the same time, where $(A,\ B,\ and\ C)$ belong to the same sample space. (Movellan, 2008).

$$P\left(A\cap B\cap C\right)=P(A)*P(B\,|\,A)*P(C\,|\,A,B) \tag{16}$$

12) Equation (17) represents the most important rule that is used to calculate the resilience aspect of the proposed approach. This rule is called the Cumulative Binomial Distribution Theorem. It is used to calculate the probability of getting exactly (k) successes in (n) independent trials. (Guichard, 2016).

$$\left(x+a\right)^{n}=\sum_{k=0}^{n}\binom{n}{k}x^{k}a^{n-k} \tag{17}$$

Where:

k: The number of successes.

n: The number of independent trials.

$n-k$: The number of failures.

$\binom{n}{k}$: The binomial coefficient. It is the number of ways of distributing (k) successes in a sequence of (n) trials.

x: The probability of success of an event.

a: The probability of failure of an event.

Table 3.2 summarizes the previous probabilities and theories in term of its name and a brief of its description.

Table 3.2: List of The Formulas

The Formula	Its Name	Its Description
$\sum_{i=1}^{n} a_i = a_1 + a_2 + \ldots + a_n$	Sigma Notation (i.e. Summation Notation).	The sum of (n) terms, where (i) is the index of the summation, (a_i) is the i_{th} term of the summation.
$P\left(A^c\right) = 1 - P\left(A\right)$	The Law of Complementary Events.	Means all the possible outcomes that are not in (A).
$n!$	The factorial Rule.	The number of ways to arrange (n) distinct objects.
$\binom{n}{r} = \frac{n!}{(n-r)!*r!}$	The combination Rule.	Selecting (r) objects from given (n) objects.
$\binom{n}{r,m} = \frac{n!}{(m!)^r}$	Combinations with groups .	The number of ways to divide (n) distinct objects into (r) groups and each group has (m) objects.
$pr\left(A\right) = \frac{The\ number\ of\ wanted\ outcome}{The\ number\ of\ possible\ outcomes}$	The probability of an Event	How many times an outcome (A) can occur compared to all possible outcomes.
$P\left(A \mid B\right) = \frac{P\left(A \cap B\right)}{P\left(B\right)}, P\left(B\right) \neq 0$	The conditional Probability.	The conditional probability of the event (A) under the condition that the event (B) has occurred.
$\Pr\left(B\right) = \sum_{i=1}^{n} \Pr\left(A_i\right) pr\left(B \mid A_i\right)$	Theorem of total probability.	The probability of event (B) of a union of a set of events $(A_1, \ldots A_n)$.
$P\left(A \cap B\right) = P\left(A\right) * P\left(B \mid A\right)$	The multiplication Rule.	The probability that two events $(A)\ and\ (B)$ can occur at the same time.

$P\left(A\cup B\right)=P\left(A\right)+P\left(B\right)-P\left(A\cap B\right)$	The additive Rule.	The probability to occur one of two events $(A)\ or\ (B)$.
$P\left(A\cap B\cap C\right)=P(A)*P(B\mid A)*P(C\mid A,B)$	The chain of Events Rule.	The probability that three events $(A,\ B,\ and\ C)$ can occur at the same time.
$(x+a)^n=\sum_{k=0}^{n}\binom{n}{k}x^k a^{n-k}$	The cumulative Binomial Distribution Theorem.	The probability of getting exactly (k) successes in (n) independent trials.

Chapter Four

Research Implementation

In this chapter, the implementation of the methodology of the proposed approach is explained and provided in its details. The proposed key management scheme is implemented using the MATLAB software as well as simulating the WSN environment. The simulation of the WSN environment is based on the random distribution of the sensor and the Sybil nodes to a virtual area representing the WSN as well as implementing its communication range, and its main functions (i.e. send and receive). The MATLAB software contains tools that provide a direct implementation of math rules (Kadry, 2014), and therefore it is the most appropriate tool that can be used in our study to program the proposed approach and simulate the resulting equations to evaluate the performance of the proposed approach.

The proposed approach has been applied to the simulation of the WSN to ensure that each phase in the proposed approach works correctly with the help of the proposed mathematical functions.

4.1 Research Phases

The proposed key management scheme goes through three main phases, and each of these phases consists of several steps. Most of the proposed approach steps are based upon the mathematical rules to generate the shared keys between the neighboring nodes, while other steps in this approach require the neighboring nodes to communicate with each other to transfer the keying information. The following three subsections illustrate and explain the implementation of these three phases and the steps of each phase.

4.1.1 Phase I: Generating a Large Polynomial Pool

The proposed key management scheme can operate in two modes; the offline mode by the setup server (Base station) and the online mode by the sensor nodes. Initially, the setup server initializes the keying information that is needed to create a symmetric matrix of symmetric polynomials. For this purpose, the **poly_gene** function was developed to generate a large pool of bivariate symmetric polynomials of degree (t) by using the format given in (Equation 2, ch3) (Zhang *et al.*, 2016).

```
function [P] = poly_gene (q, x, y)
```

The **poly_gene** function is called by the setup server during the initialization phase (offline mode). It receives three input parameters; (q) is the order of the finite field (F_q), (x) and (y) are the variables of the polynomials. The output of this function is a pool of bivariate symmetric polynomials with coefficients in the field (F_q), each polynomial with two variables (x, y). The flowchart in Figure 4.1 shows the main steps of the **poly_gene** function.

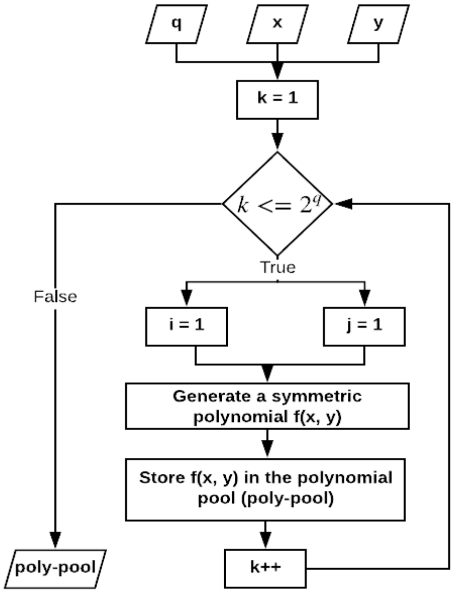

q
x
y
k = 1
$k <= 2^q$
True
False
i = 1
j = 1
Generate a symmetric polynomial f(x, y)
Store f(x, y) in the polynomial pool (poly-pool)
poly-pool
k++

Figure 4.1: The poly_gene Function Flowchart

Algorithm (1) shows the pseudocode to implement the **poly_gene** function.

```
Algorithm 1: Polynomial Pool Creation
Input: q, x, y.
Output: a pool of bivariate symmetric polynomials.
 %  The degrees of the variables
 $i \leftarrow [1,q]$;
 $j \leftarrow [1,q]$;
 $a \leftarrow$ size ($q$);                       % Create a matrix to store the coefficients of the polynomials
 poly_pool ← zeros (size ($2^q$))          % Create a matrix to store the generated polynomials
 k ←1                                     % The matrix index
For k to $2^q$ do
    $i \leftarrow 1$;
    $j \leftarrow 1$;
    poly_pool (k) ← $a(q)x^q + a(q-i)x^{(q-i)}y^j + a(q-(i++))x^{(q-(i++))}y^{j++} + \ldots$
                    $+a(q)y^q + a(q-i)y^{(q-i)}x^j + a(q-(i++))y^{(q-(i++))}x^{j++}$
    $k \leftarrow k+1$;
End for
Return poly_pool
```

At the end of this phase, a pool of bivariate symmetric polynomials over a finite field becomes available. The elements of the finite field of a specific order are constant, such that, $F_3 = \{0,1,2\}$ and $F_4 = \{0,1,2,3\}$. Therefore, the order of the finite field is chosen to be larger than the network size (i.e. the number of the sensor nodes). In addition to that, not all the generated polynomials in this pool will be taken, but only a subset of them will be taken. This pool is generated only to make it difficult for the adversary to guess the correct polynomials that would be selected to be distributed to the sensor nodes. The process of selecting a sample of polynomials of the polynomial pool and how these polynomials are distributed to the sensor nodes before the deployment is illustrated and explained in the next section.

4.1.2 Phase II: Polynomial Pre-Distribution

This phase focuses on the creation of a symmetric matrix of the polynomial pool by the setup server. This phase contains five steps; starts when the setup server starts selecting a sample of polynomials of the polynomial pool and finishes when the sensor nodes are distributed to the target area with their keying information. Therefore, several functions were developed and used to perform this phase. Three main functions were developed in this phase to perform the required steps as follows:

1) A function to select randomly a set of polynomials of the polynomial pool, then it creates a symmetric matrix from the selected polynomials.

```
function [A] = create_sym_matrix(P,N);
```

The **create_sym_matrix** function of type double receives two parameters; the polynomial pool (P) that is generated by the **poly_gene** function, and the number of the sensor nodes (N). It uses two built-in functions; **datasample** and **toeplitz** functions. The **datasample** function is responsible for selecting a sample of the polynomial pool, it receives three parameters; the first parameter (P) represents the polynomial pool, the second and the third parameters (N, N) represent the dimensions of the generated sample which is equal to the number of the sensor nodes (N). The **toeplitz** function is responsible for creating a matrix from the selected sample (i.e. the output of the **datasample** function). Then the **create_sym_matrix** function converts the matrix resulted from the **toeplitz** function to a symmetric matrix with the dimensions of $(N \times N)$. The output of this function is a symmetric matrix contains a set of bivariate symmetric polynomials, each of them has the format of (Equation 2, ch3). The flowchart in Figure 4.2 shows the main steps of the **create-sym-matrix** function.

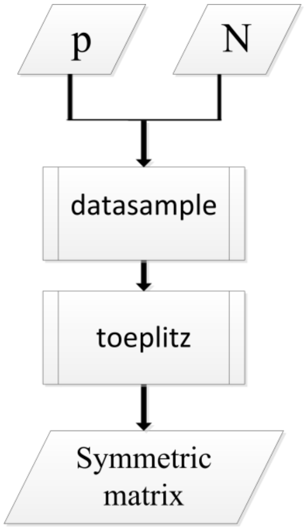


Figure 4.2: The create_sym_matrix Function Flowchart

Algorithm (2) shows the Pseudocode of the **create_sym_matrix** function.

```
Algorithm 2: Symmetric matrix creation
Input: P, N.
Output: Symmetric matrix of dimensions (N × N).
$ds \leftarrow \text{datasample}(P, N, N);$          % select a sample of the polynomial pool.
$A \leftarrow \text{toeplitz}(ds);$                  % Create a matrix from the selected sample.
Converts the matrix A to a symmetric matrix
Return A
```

2) Another function is developed to divide the created matrix four blocks of the same size and to find the (BL) and (BU) matrices of the matrix blocks.

```
function[BL,BU]= BLU(A,r)
```

The **BLU** function of type double receives tow parameters; the symmetric matrix (A) that is generated by calling the **create_sym_matrix** function, and the dimensions of the blocks (r). The **BLU** function uses a built-in function called

mat2cell. The **mat2cell** function receives three parameters; the first one represents the matrix (A), the second and the third parameters represent dimensions of the blocks (r). The **mat2cell** function is used to divide the passed matrix into four blocks of $(\mathrm{r} \times \mathrm{r})$ dimension each, where $\left(\mathrm{r} \approx \frac{\mathrm{n}}{2}\right)$. After that, the **BLU** function decomposes the divided matrix into two matrices; the (BL) and (BU) for each block, as illustrated in Figure 4.3. Finally, this function rebuilds the last block (A_{22}) of the matrix to reduce the number of distributed elements to the second half of the nodes by multiplying $(\mathrm{BL}_{21}$ by $\mathrm{BU}_{12})$.

The **BLU** function finishes when the passed matrix (A) is divided into blocks of size $(r \times r)$ each, and each block is decomposed into two matrices (BL) and (BU). Although each block of this matrix is manipulated as a separated matrix which contains different elements (i.e. polynomials), they are still related to each other and each node can establish shared keys with the neighboring nodes regardless if they are from the same block or from different blocks. This is the main advantage of the BLU-Decomposition method that is used in the proposed approach. Figure 4.3 illustrates the formation flow for the BLU function.

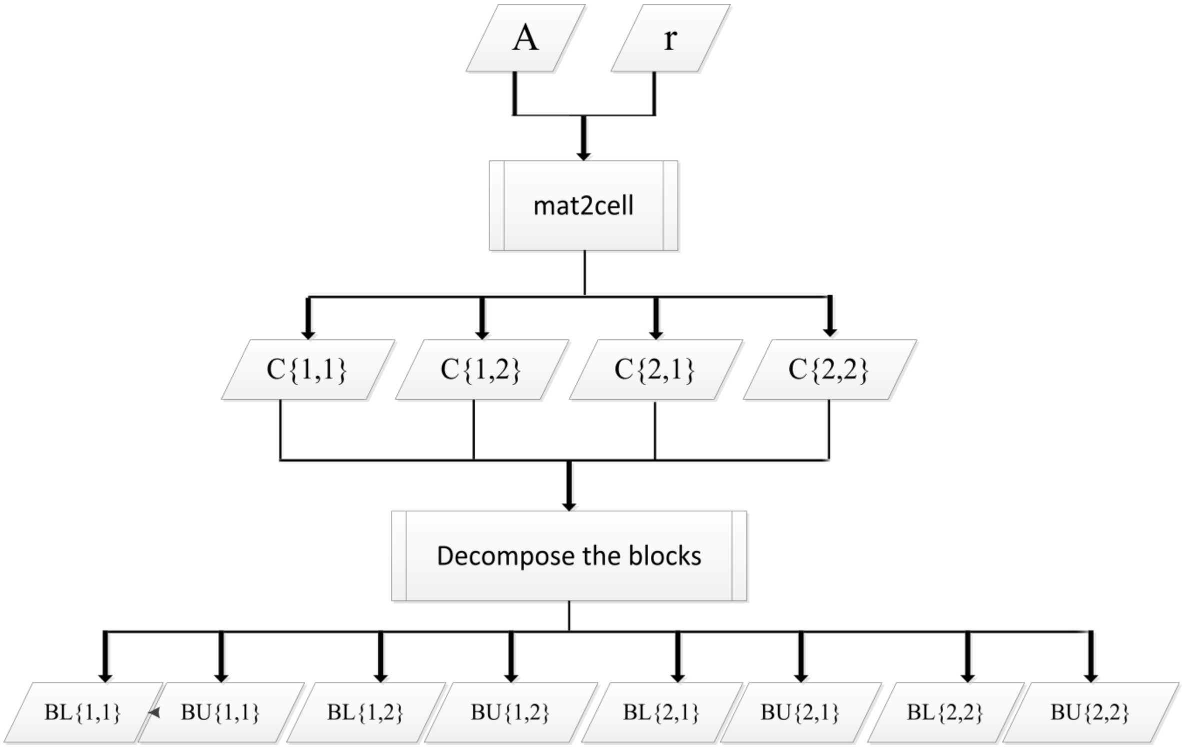


Figure 4.3: The BLU Function Flowchart

Algorithm (3) shows the pseudocode to implement the **BLU** function.

Algorithm 3: Divide and Decompose The Matrix

```
Input: A, r
Output: generate BL and BU of matrix A
 $L \leftarrow$ zeros (size (A));                 % create a matrix to store BL elements of matrix A
 $U \leftarrow$ zeros (size (A));                 % create a matrix to store BU elements of matrix A
 $C \leftarrow$ mat2cell (A, [r, r], [r, r]);     % Divide the matrix A into blocks of size $(r \times r)$
 $BL \leftarrow$ mat2cell (L, [r, r], [r, r]);    % Divide the matrix L into blocks of size $(r \times r)$
 $BU \leftarrow$ mat2cell (U,[r, r], [r, r]);     % Divide the matrix U into blocks of size $(r \times r)$
 $[BL\{1,1\}, BU\{1,1\}] \leftarrow$ Decompose $(C\{1,1\})$;   % Decompose the first block of the matrix
 $BU\{1,2\} \leftarrow BL\{1,1\}^{-1} * C\{1,2\}$;
 $BL\{2,1\} \leftarrow C\{2,1\} * BU\{1,1\}^{-1}$;
 $[BL\{2,2\}, BU\{2,2\}] \leftarrow$ Decompose $(C\{2,2\})$;   % Decompose the last block of the matrix
 $C\{2,2\} \leftarrow BL\{2,1\} * BU\{1,2\}$;                 % Re-build the last block
Return BL, BU
```

3) Finally, a function is needed to distribute the elements of the (BL) and (BU) matrices to the sensor nodes before distributing them to the target area. This function is called **dist-BLU-elements.**

```
function dist-BLU-elements(BL, BU, ID, N)
```

The **dist_BLU_elements** function receives four parameters; The outputs of the **BLU** function (i.e. (BL) and (BU) matrices), the number of sensor nodes (N), and the (ID) parameter. The (BL) and the (BU) parameters contain the elements (i.e. polynomials) that will be distributed to the sensor nodes. The (ID) is an array includes the IDs of the sensor nodes, where $(\text{the IDs} = 1,\ 2, \ldots,\ N)$. This function is responsible for distributing the matrices elements to the sensor nodes, such that the first half of the nodes receives the $\text{BL}\{1,\ 1\}$ and $\text{BU}\{1,\ 1\}$ matrices elements while the second group receives the $\text{BL}\{2,\ 1\}$ and $\text{BU}\{1,\ 2\}$ elements. The flowchart in Figure 4.4 Show the main steps of the implementation of the **dist_BLU_elements** function.

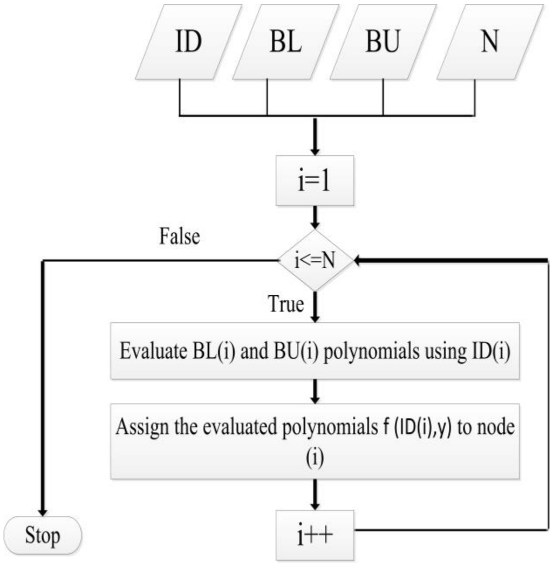

ID
BL
BU
N
i=1
False
i<=N
True
Evaluate BL(i) and BU(i) polynomials using ID(i)
Assign the evaluated polynomials f (ID(i),y) to node (i)
Stop
i++

Figure 4.4: The dist_BLU_elements Function Flowchart

Algorithm (4) shows the Pseudocode of the **dist-BLU-elements** function.

Algorithm 4: Distribute matrices elements

```
Input: BL, BU, ID, N
Output: Each sensor node has one row and one column.
For  i=1 to N do
    $x \leftarrow ID(i)$;
    BL (i, :) $\leftarrow$ f (x, y);        % Evaluate the $i_{th}$ row polynomials in BL using ID (i).
    BU (:, i) $\leftarrow$ f (x, y);        % Evaluate the $i_{th}$ column polynomials in BU using ID (i).
    Assign BL( i , : ) and BU(: , i) to node (i);
    i $\leftarrow$ i+1 ;
End for
```

This phase finishes when the sensor nodes are distributed to the target area and each sensor node has its own keying information to establish its shared keys with the neighboring nodes. The next section illustrates the third phase of the proposed approach in terms of how the neighboring nodes communicate with each other and how they transfer their keying information to establish their shared keys.

4.1.3 Phase III: Secret Key Derivation

This phase starts when the sensor nodes are distributed to the targeted area and equipped with the desired keying information. The following function was developed to distribute the sensor nodes and the Sybil nodes randomly into the simulated area.

```
function[sensor_neighbors,Sybil_neighbors]=distribute(SEN,SYN,R)
```

The **distribute** function requires three parameters; the number of sensor nodes (SEN), the number of the Sybil nodes (SYN), and the communication range (R). This function returns two matrices; the first matrix contains all sensor nodes in the network and the neighboring nodes for each sensor node (Sensor_neighbors). The second matrix is for the Sybil nodes, where it contains all Sybil nodes in the network and the neighboring legitimate nodes for each Sybil node (Sybil_neighbors). This function is responsible for distributing the sensor nodes and the Sybil nodes into the simulated area and examines the neighboring nodes for each Sybil and sensor node by calculating the Euclidean distance between the nodes and comparing the results of the calculations with the communication range (R). Algorithm (5) shows the pseudocode of the **distribute** function.

Algorithm 5: Simulate WSN environment

```
Input: SEN,  SYN, R
Output: Simulate the WSN environment.
$sensor\_neighbors \leftarrow \mathrm{size}(\mathrm{SEN});$      % Create a matrix for sensor nodes neighbors of size
SEN.
$Sybil\_neighbors \leftarrow \mathrm{size}(\mathrm{SYN});$      % Create a matrix for Sybil nodes neighbors of size SYN.
SN ← Zeros (SEN) ;                % Create a matrix for Sensor nodes.
SY ← Zeros (SYN) ;                % Create a matrix for Sybil nodes.
For  i=1 to SEN do
    % dimensions of sensor node (i)
    Randomly determine x coordinate of $\mathrm{SN}(\mathrm{i});$
    Randomly determine y coordinate of $\mathrm{SN}(\mathrm{i});$
    i ← i +1 ;
End for
For  i=1 to SYN do
    % dimensions of Sybil node (i)
    Randomly determine x coordinate of $\mathrm{SY}(\mathrm{i});$
    Randomly determine y coordinate of $\mathrm{SY}(\mathrm{i});$
    i ← i +1 ;
End for
For  i=1 to SEN do
    For j=2 to SEN do
        Calculate the Euclidean distance between node i and node j;
        If (distance between node i and node j <=  R)
            $sensor\_neighbors(\mathrm{i, j}) \leftarrow 1;$      % set a link between node i and node j
        End if
        else if (distance between node i and node j > R)
            $sensor\_neighbors(\mathrm{i, j}) \leftarrow 0;$
        End if
        j ← j +1 ;
    End for
    i ← i +1 ;
End for
For  i=1 to SYN do
```

```
    For j=2 to SYN do
        Calculate the Euclidean distance between node i and Sybil node j
        If (distance between node i and Sybil node j <=  R)
            $Sybil\_neighbors(i, j) \leftarrow 1;$      % set a possible link between node i and Sybil node
        End if
        else if (distance between node i and Sybil node j > R)
            $Sybil\_neighbors(i, j) \leftarrow 0;$
        End if
        $j \leftarrow j+1;$
    End for
    $i \leftarrow i+1;$
End for
Distribute the sensor nodes and the Sybil nodes into the simulated area;
Return Sensor_neighbors, Sybil_neighbors;
```

In this phase, the neighboring sensor nodes start communicating with each other in order to generate the shared keys. In the proposed approach, the communication process between the sensor nodes includes three main functions; Send, Receive, and Broadcast function, where the key derivation process depends on the keying information that is exchanged between the neighboring nodes.

The communication process between the sensor nodes goes through two important steps: acquaintance and information transfer steps. In the acquaintance step, each node just tries to open a communication with its neighboring nodes (Sybil node and sensor nodes). In the information transfer step, the communicating nodes start to transfer their keying information to establish their shared keys. Each step has its own functions. The main functions that are used in these steps are:

1) The **broadcast** function is the first function that is used by the sensor nodes in the proposed approach. After the sensor nodes are distributed to the target area, they start broadcasting their own IDs to discover their neighbors.

```
function broadcast(SID, N)
```

The **broadcast** function is used in the first step of the communication process with two parameters; the first parameter represents the ID of the sender node (SID), and the other parameter represents the number of the sensor nodes in the network (N). This function is used by each node to examine the other sensor nodes that exist in its communication range in the WSN. For example, if the sensor node (i) wants to know its neighbors and who can establish shared keys with it, it first broadcasts its own ID, and each node exists in its range will receive the ID of the sensor node (i) by calling another developed function called **receive_msg**. The flowchart in Figure 4.5 shows the main steps of the implementation of the **broadcast** function.

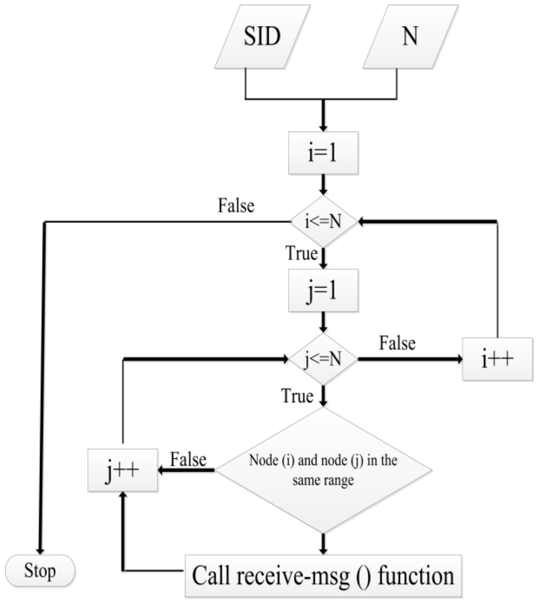

SID
N
i=1
False
i<=N
True
j=1
False
j<=N
i++
True
j++
False
Node (i) and node (j) in the same range
Stop
Call receive-msg () function

Figure 4.5: The broadcast Function Flowchart

Algorithm (6) shows the pseudocode of the implementation of the **broadcast** function.

```
Algorithm 6: Broadcast ID
Input: SID, N
Output: The neighboring nodes discover each other.
For i=1 to N do
    For j=2 to N do
        If (node j exists in node i range)
            receive_msg (ID(j))     % Call  receive_msg function to the neighbors of node(i)
        End if
        j ← j +1 ;
    End for
    i ← i +1 ;
End for
```

2) The following two functions (**receive_msg** and **receive_info**) are developed to complete the communication process between the neighboring nodes. It is not enough for the sender nodes to send their information, because the communication process is completed when the other participant receives the transmission information.

```
function receive_msg(msg)
function receive_info(SID, row)
```

The **receive_msg** function is used in the acquaintance and in the information transfer steps. It requires only one parameter that represents either the ID of the sender node or an encrypted information from the received nodes. In this function, the received nodes send an acknowledgment message to the sender node using another function called **send_msg**. Algorithm (7) shows the pseudocode of the **receive_msg** function.

```
Algorithm 7: Receive message
Input: msg
Output: acknowledge or authenticate the transmission medium.
Enqueue the received message in the receive buffer.
In the broadcast step:
    Enqueue the send buffer of the sender (acknowledgment message)
    send_msg (acknowledgment message)
In the information transfer step:
    Decrypt the received message.
    If (the receiver could decrypt the message)
        If (decrypted ID== the receiver ID)
            Start the authentication phase between the sender and the receiver.
        End if
        Else if (decrypted ID!= the receiver ID)
            Break the communication phase between the sender and the receiver.
        End if
    End if
```

The **receive_info** function is used in the information transfer step. This function receives two parameters; the ID of the sender node and its row. The multiplication and the evaluation of the polynomials processes happen in this function. Algorithm (8) shows the pseudocode of the **receive_info** function.

```
Algorithm 8: Receive the transmitted information
Input: SID, row
Output: Calculate shared keys between neighboring nodes or Break the connection.
Enqueue the received message in the receive buffer.
If (the sender and the receiver do not have  a shared key)
    Evaluate all univariate polynomials using the IDs of the receiver and the sender;
    Apply the Dot multiplication method on the polynomials;
    Enqueue the send buffer of the sender(receiver  id, the receiver row);
    send_info (receiver  id, the receiver  row);
End if
Else if (the receiver has a key with the sender)
    Encrypt the sender ID.
    Enqueue the send buffer of the sender (encrypted id).
    send_msg (encrypted id).
End if
```

3) The following two functions (**send_msg** and **send_info**) are developed to start transferring information between neighboring nodes. This function is like the **receive** function has two formats.

```
function send_msg(msg)
function send_info(SID, row)
```

The **send_msg** function is used in the broadcast step. This function is called by the sensor nodes who received the ID of the sender node. The message in this function contains an acknowledgment message from the receiver node to the sender node to inform the sender node that exists in its range. This function is used to finish the broadcast step by calling another function **send_info** function. The flowchart in Figure 4.6 shows the main steps of the implementation of the **send_msg** function.

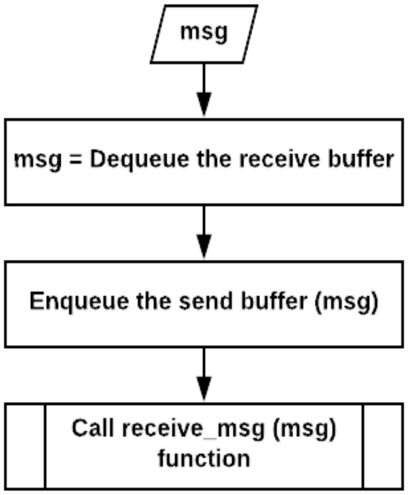

msg
msg = Dequeue the receive buffer
Enqueue the send buffer (msg)
Call receive_msg (msg) function

Figure 4.6: The send_msg Function Flowchart

Algorithm (9) shows the pseudocode of the **send_msg** function.

```
Algorithm 9: Send an acknowledgment message
Input: msg
Output: acknowledge or authenticate the transmission medium.
 msg ← Dequeue the receive buffer;
Enqueue the send buffer message (msg)
Call receive_msg (msg).
```

The **send_info** function requires two parameters; the ID of the sender node and its row elements. The transmitted information in this function is the keying information of the communicating nodes. This function calls another developed function called **receive_info** because every send operation needs the receive operation to complete the communication process. The flowchart in Figure 4.7 shows the main steps of the implementation of the **send_info** function.

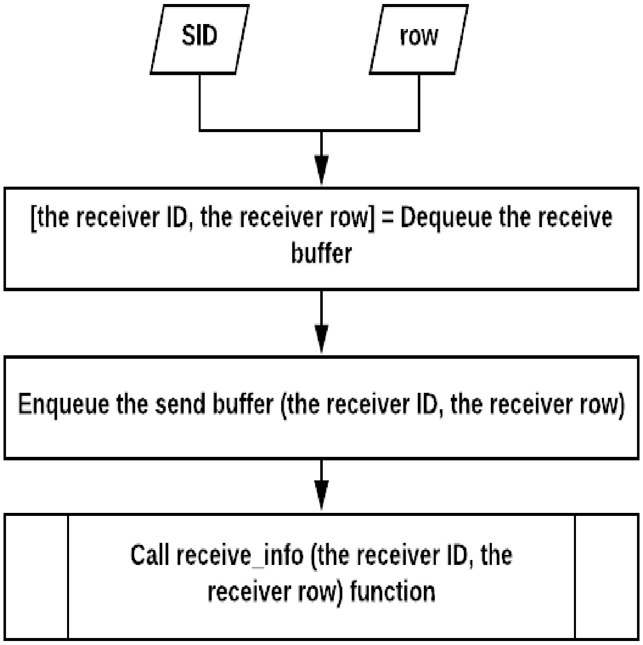

SID
row
[the receiver ID, the receiver row] = Dequeue the receive buffer
Enqueue the send buffer (the receiver ID, the receiver row)
Call receive_info (the receiver ID, the receiver row) function

Figure 4.7: The send_info Function Flowchart

Algorithm (10) shows the pseudocode of **send_info** function.

Algorithm 10: Send information

```
Input: SID, row
Output: Calculated shared keys between neighboring nodes or Break the connection.
[the receiver ID, the receiver row ] $\leftarrow$ Dequeue the receive buffer;
Enqueue the send buffer (the receiver ID, the receiver row);
Call receive_info (the receiver ID, the receiver row);
```

The proposed approach as an algorithm receives two important parameters; the number of the sensor nodes (SEN) and the number of the Sybil nodes in the network (SYN) (i.e. the size of the network). All developed functions in the proposed algorithm depend on these parameters such as the **distribute** function which is used to distribute the sensor nodes and the Sybil nodes into the simulated area, the **create_sym_matrix** function which is used to create a symmetric matrix according to the number of the sensor nodes, **dist_BLU_elements** function which is used to evaluate the polynomials in both (BL) and (BU) matrices and distribute these polynomials to the sensor nodes before the deployment,…etc. As an output, it aims to provide a full connectivity between the neighboring nodes, where each sensor node has a shared key with all its legal neighbors. Algorithm (11) shows the pseudocode of implementation of the proposed algorithm.

Algorithm 11: The proposed Algorithm

```
Input: SEN, SYN
Output: WSN  has a full connection.
SEN;                        % Number of the sensor nodes
SYN;                        % Number of the Sybil nodes
BS ← 1;                     % Number of the base stations
$A \leftarrow zeros$ (SEN);     % Create a matrix to store the selected polynomials from the polynomial pool
$BL \leftarrow zeros\ (size\ (A))$;
$BU \leftarrow zeros\ (size\ (A))$;      % Create matrices to store the results of the BLU function
$q \gg SEN$ ;                      %  The order of $F_q$ should be too larger than SEN.
$P \leftarrow$ zeros ($2^q$);              % Create a matrix to store the polynomial pool
$ID \leftarrow zeros\ (SEN)$;              % Create a matrix o store the IDs of the sensors
% Create an array to store the sensor nodes IDS;
For i=1 to SEN do
    $ID[i] \leftarrow i$;
    $i \leftarrow i+1$;
End for
$[P] \leftarrow$ poly_gene$(q, x, y)$;                        %  Build a polynomial pool.
$[A] \leftarrow$ create_sym_matrix$(P, SEN, SEN)$;   %  Build a symmetric matrix of polynomials
$[BL, BU] \leftarrow$ BLU$(A)$;                               %  Apply the BLU-Decomposition to the matrix.
distribute (SEN, SYN);                                  %  Simulate the WSN environment.
dist_BLU_elements (BL,  BU,  ID) ;         % Distribute the keying information to the sensors.
% start key derivation process between the sensor nodes.
For i=1 to SEN do
    broadcast (ID (i)) ;
    $i \leftarrow i+1$;
End for
Return  WSN has a full connectivity
```

Chapter Five

Implementation Results

In this chapter, the performance of the proposed approach has been evaluated and analyzed to show its effectiveness. The evaluation is based on generating different equations that represent the performance of the proposed approach in different sides. The equations were generated using different formulas and rules of probability and calculus theorems. The WSN environment and the generated equations were simulated using MATLAB software tools, the generated equations were simulated to be compared to other equations that represent the performance of other approaches.

5.1 Simulation Environment

The simulation is the basic method that has been used to evaluate and analyze the performance of the proposed approach using the MATLAB software tool. Figure 5.1 shows a simulation of the wireless sensor network environment (i.e. sensor nodes, base station, and Sybil nodes), where the sensor nodes and the Sybil nodes have been distributed randomly in an area of $(1000m \times 1000m)$dimensions, and the communication range of the sensor nodes is assumed (200m).

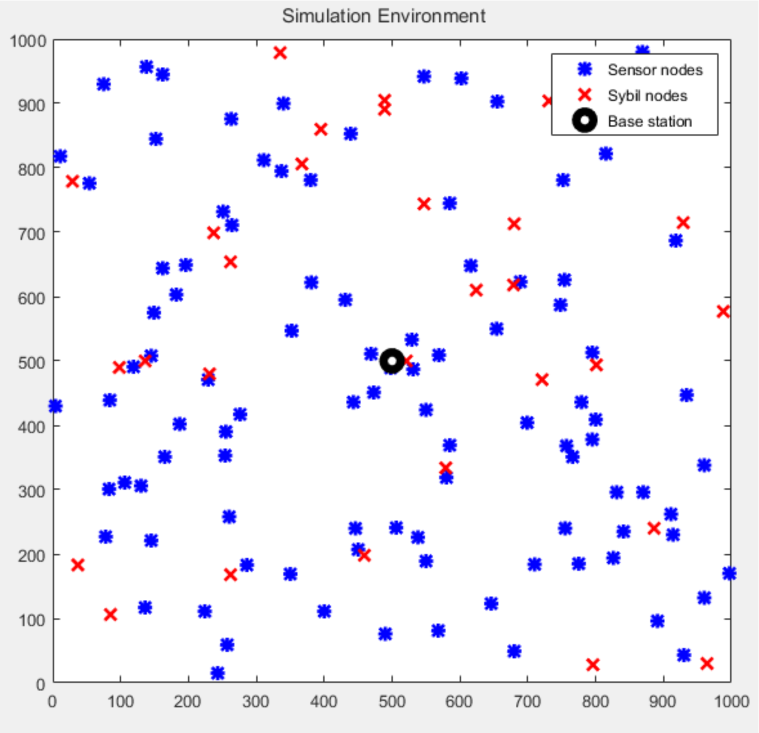


Figure 5.1: WSN Simulation Environment

Table 5.1 shows the parameters that have been used to simulate the WSN environment. All the values of the parameters were constant in each experiment except the number of the sensor nodes and the Sybil nodes.

Table 5.1: WSN Simulation Parameters

Parameter	Its value
Deployment Area	1000m × 1000m
Deployment Distribution	Randomly
Number of Sensor Nodes	Based on the experiment
Number of Base Stations	1
Number of Sybil Nodes	Based on the experiment
Position of Base Station	(500, 500)
Communication Range	200m

5.2 Performance Analysis

This subsection focuses on analyzing the performance of the proposed approach from different aspects such as: (1) the memory overhead, (2) the security, (3) the energy consumption, and (4) the network connectivity. The evaluation and the analysis are based on the simulation results.

5.2.1 The Memory Overhead

Due to the random distribution of the sensor nodes in the target area as well as their communication range and their memory limitation, (i.e. each node can communicate only with the nodes within its range), the shared keys cannot be pre-loaded to the sensor nodes before the deployment. Therefore, the sensor nodes are deployed into their field and equipped with the desired information to establish a secure communication. The pre-loaded information should be restricted to the limited memory in the sensor nodes. This metric estimates the required storage space of each sensor node to store the pre-loaded information.

The proposed approach uses two important methods: (1) the Block LU-Decomposition algorithm, and (2) building a symmetric matrix of symmetric polynomials. The Block LU-Decomposition algorithm divides the matrix into four blocks of the same size (i.e. $\frac{N}{2} \times \frac{N}{2}$). Therefore, each node receives a half of one row from (BL) matrix and a half of one column from (BU) matrix, where (N) is the number of sensor nodes in the network. The elements of these matrices are $t-$degree bivariate polynomials, each polynomial has $(t+1)$ coefficients from a finite field (F_q). Equation (18) represents the storage space (i.e. number of bits) that is occupied by each one of the polynomials in each sensor node.

$$P\_M = (t+1)L \tag{18}$$

Where:

L: The number of bits to represent the coefficients of each polynomial.

$(t+1)$: The number of the coefficients of each polynomial.

t : The degree of the polynomials.

Since each sensor node receives a half of one row from (BL) matrix and a half of one column from (BU) matrix and according to the storage overhead for each polynomial (P_M) that is presented in the previous equation, the memory overhead (M) to store the keying information in each sensor node (i) is given in Equation (19).

$$M_i = N * [(t+1) * L] \tag{19}$$

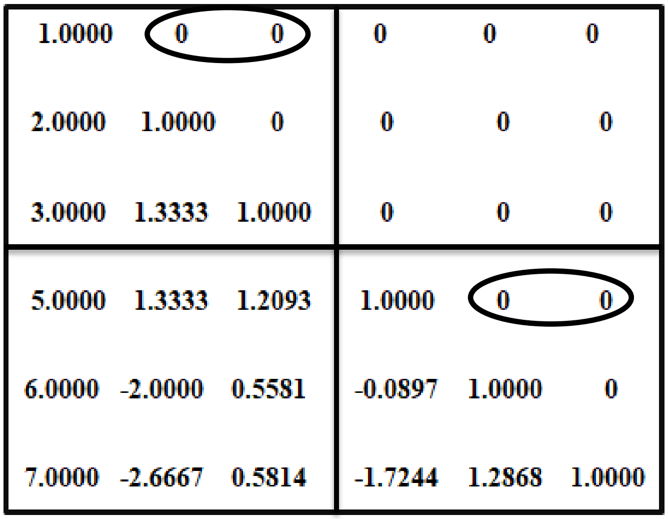
Where:

N: The number of the polynomials in the sensor node (i) (i.e. the number of sensor nodes in the network).

The overall memory consumption in the entire network is equal to the sum of memory usage in all sensor nodes. So the memory overhead of the entire network is given in Equation (20).

$$
\begin{aligned}
M &= \left[N * \left[(t+1) * L\right]\right] * N \\
&= N^{2} * \left[(t+1) * L\right]
\end{aligned}
\qquad (20)
$$

To reduce the memory overhead of the entire network, the elements of (BL) and (BU) matrices can be partitioned into two parts because they contain many zeros as shown in Figure 5.2; the first part for the non-zero elements and the second part represents the largest number of the zero elements in (BL) and in (BU) matrices.



1.0000	0	0	0	0	0
2.0000	1.0000	0	0	0	0
3.0000	1.3333	1.0000	0	0	0
5.0000	1.3333	1.2093	1.0000	0	0
6.0000	-2.0000	0.5581	-0.0897	1.0000	0
7.0000	-2.6667	0.5814	-1.7244	1.2868	1.0000

Figure 5.2: 6x6 (BL) Matrix

So each node receives the non-zero elements and one value represents the largest number of zero elements in the (BL) and (BU) matrices. This mechanism is called the encoding mechanism. The new memory overhead (New_M_i) to store the keying information in each sensor node (i) using the encoding mechanism will be as follows:

$$
New\_M_i = \boldsymbol{h_i P\_M} + 2\boldsymbol{z} \qquad (21)
$$

Where:

h_i : The number of non-zero elements in the sensor node (i).

z : The number of bits to store the largest number of zeros in zero element part in (BL) and in (BU) matrices.

The new memory overhead (New_M) of the entire network after using the encoding mechanism became as given in Equation (22).

$$\begin{aligned} New\_M &= \sum_{i=1}^{N} New\_M_i + 2zN \\ &= (t+1)L\left[\frac{N}{2}\left(\frac{N}{2}+1\right)\right] + N * \left(2*ceil\left(\log_2\left(\frac{N}{2}-1\right)\right)\right) \end{aligned} \qquad (22)$$

Where:

ceil: Ceiling function which rounds up the integer number to the next larger integer number, $\log_2\left(\frac{N}{2}-1\right)$: The largest number of zeros in the zero part in the row or in the column.

Figure 5.2 shows (BL) matrix with dimensions (6×6) after applying the BLU-Decomposition method. Using the ceil function, the largest number of zeros in the blocks (L_{11}, L_{22}) is (2), where $2 = \frac{6}{2} - 1$.

The encoding mechanism focuses on reducing the number of zero elements in each sensor node to one value represents the largest number of zeros in (BL) matrix or in (BU) matrix. Equation (23) represents the network memory saving (M_{saving}) after applying the encoding mechanism.

$$M_{saving} = \left[N^{2} * (t+1)L\right] - \left[(t+1)L\left[\frac{N}{2}\left(\frac{N}{2}+1\right)\right] + 2N * ceil\left(\log_{2}\left(\frac{N}{2}-1\right)\right)\right] \qquad (23)$$

Figure 5.3 shows the memory-wide overhead (i.e. network memory overhead) with and without the encoding mechanism. The experiment is performed using a different number of sensor nodes (i.e. [10-100]), $(t = 20)$, and the value of (L) is assumed one bit.

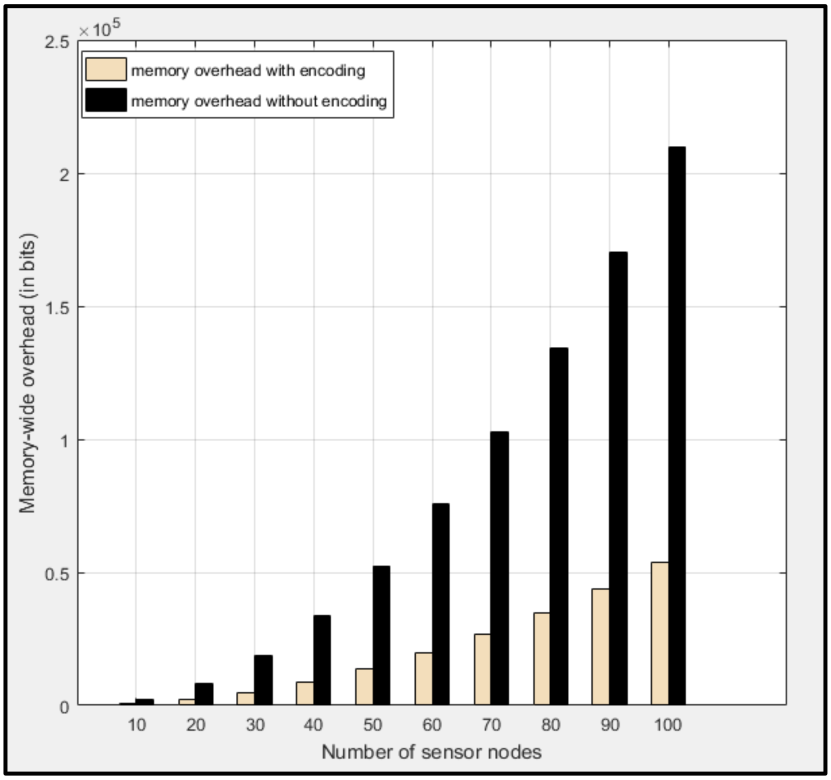


Figure 5.3: Network Memory Overhead With Encoding and Without Encoding mechanism

From Figure 5.3, it is clear that the network memory overhead with the encoding mechanism is much better than network memory without the encoding mechanism which gives the network a chance to have a large number of sensor nodes. Table 5.2 shows a comparison between the memory overhead with and without encoding the zero elements in the zero element part and the results of the memory saving when $(t = 20)$.

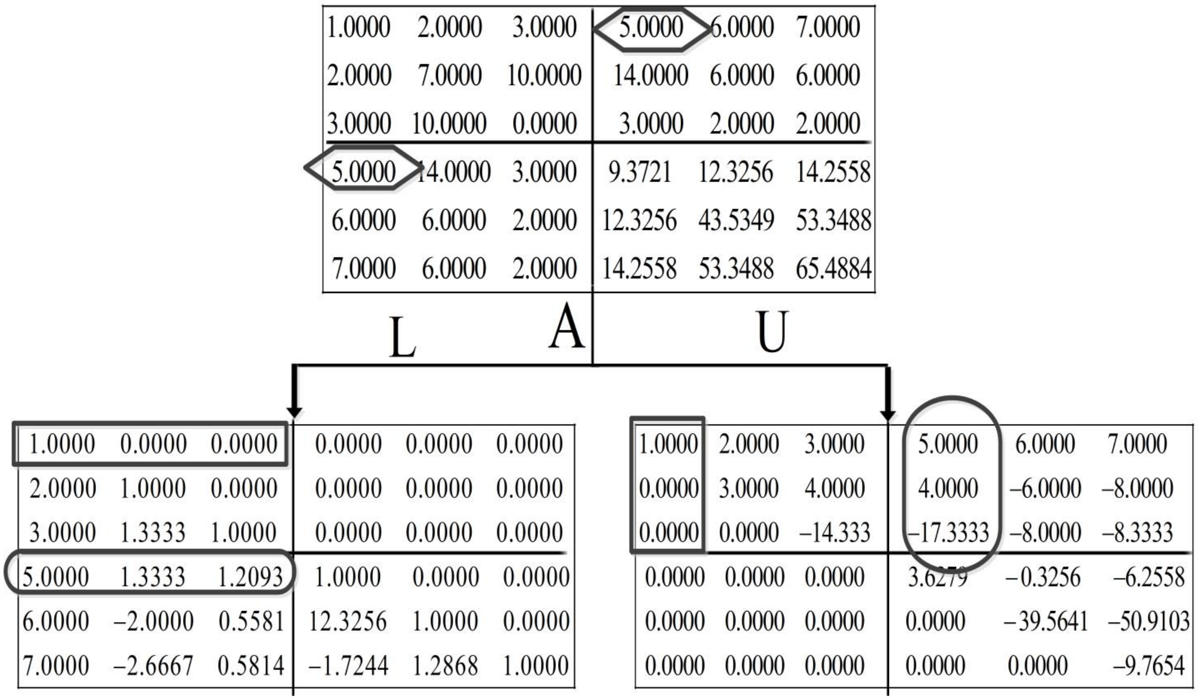
Table 5.2: Results of Memory Saving

N	**Memory required without encoding mechanism (bit)**	**Memory required with encoding mechanism (bit)**	**Memory saving (bit)**
20	$0.08 * 10^5$	$0.01 * 10^5$	$0.07 * 10^5$
40	$0.33 * 10^5$	$0.04 * 10^5$	$0.28 * 10^5$
60	$0.75 * 10^5$	$0.1 * 10^5$	$0.65 * 10^5$
80	$1.34 * 10^5$	$0.18 * 10^5$	$1.16 * 10^5$
100	$2.1 * 10^5$	$0.27 * 10^5$	$1.82 * 10^5$

5.2.2 Network Connectivity

Due to the limited communication range of the sensor nodes, not all nodes have the ability to establish a direct communication with the base station. So they have to keep transmitting their readings between each other until these readings reach the base station. In order to achieve that, each sensor nodes should be able to establish a secure transmission medium with the neighboring sensor nodes.

Although the proposed approach uses the BLU-Decomposition method which divides the original matrix into four blocks, each sensor node still has the ability to establish a secure communication link with the neighboring nodes. In the proposed approach, the matrix is divided into four sub-matrices (i.e. blocks); two of them are distributed to the sensor nodes while the other two blocks are used as shared blocks, such that the sensor nodes are divided into two groups of the same size (i.e. approximately $\frac{N}{2}$) and each group receives unique elements belonging to one of the two blocks. According to (Dai and Xu, 2010) if the sensor nodes have elements that belong to the same block, then they can absolutely establish shared keys due to the fact that the LU-Decomposition has a fully connected graph. But what if they have elements belong to different blocks?. If the sensor nodes have elements that belong to different blocks, they can use the third and the fourth blocks as a shared block because each one contains the transpose of the other one. To illustrate the process, suppose that there are two sensor nodes from different blocks want to communicate with each other, the first node is (N_a) from the first block (A_{11}) and it contains the row $(BL_{11}\ (1,\ 0,\ 0))$, the column $(BU_{11}(1,\ 0,\ 0))$ and ID (I_a). Since the last block is re-build by multiplying $(BL_{21}\ by\ BU_{12})$ then, the second node is (N_b) from the last block (A_{22}) has the row $(BL_{21}(5,\ 1.3333,\ 1.2093))$, the column $(BU_{21}(5,\ 4,\ -17.3333))$ and ID (I_b). After they exchanged their rows and each one multiplied its column by the other node's row to generate the shared polynomial (5) as shown in Figure 5.4.



1.0000	2.0000	3.0000	5.0000	6.0000	7.0000
2.0000	7.0000	10.0000	14.0000	6.0000	6.0000
3.0000	10.0000	0.0000	3.0000	2.0000	2.0000
5.0000	14.0000	3.0000	9.3721	12.3256	14.2558
6.0000	6.0000	2.0000	12.3256	43.5349	53.3488
7.0000	6.0000	2.0000	14.2558	53.3488	65.4884

1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2.0000	1.0000	0.0000	0.0000	0.0000	0.0000
3.0000	1.3333	1.0000	0.0000	0.0000	0.0000
5.0000	1.3333	1.2093	1.0000	0.0000	0.0000
6.0000	−2.0000	0.5581	12.3256	1.0000	0.0000
7.0000	−2.6667	0.5814	−1.7244	1.2868	1.0000

1.0000	2.0000	3.0000	5.0000	6.0000	7.0000
0.0000	3.0000	4.0000	4.0000	−6.0000	−8.0000
0.0000	0.0000	−14.333	−17.3333	−8.0000	−8.3333
0.0000	0.0000	0.0000	3.6279	−0.3256	−6.2558
0.0000	0.0000	0.0000	0.0000	−39.5641	−50.9103
0.0000	0.0000	0.0000	0.0000	0.0000	−9.7654

Figure 5.4: Matrix (A) After Applying The BLU-Decomposition Method

The full connectivity between the sensor nods in the proposed approach allows node-to-node mutual authentication, which limits the impact of the Sybil nodes on the network work. Figure 5.5 shows a scenario when a Sybil node and a legitimate node trie to communicate with a legitimate node. Because the Sybil node could not establish a common key with the legitimate node it fails to establish a communication with it. If two nodes cannot establish a communication between each other, then the send and the receive functions cannot be completed between them. On the other hand, the legitimate node could generate a common key with the other legitimate node. So that, they could establish a communication between each other.

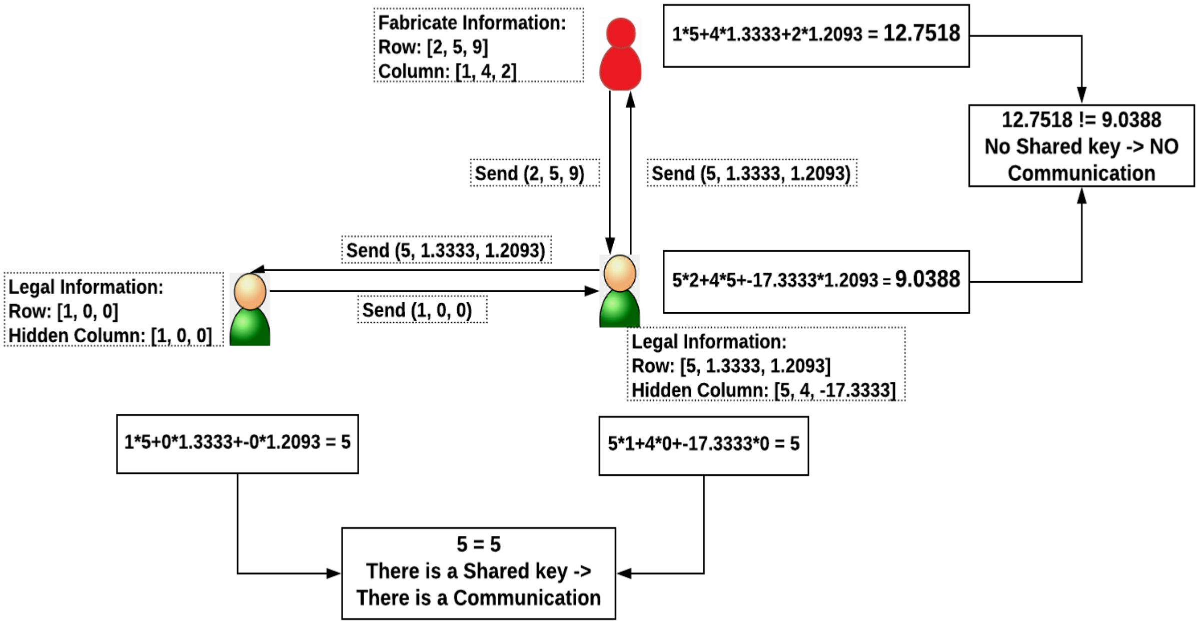


Figure 5.5: A Scenario of Sybil Node Attack

5.2.3 The Security

This part discusses the security aspects of the proposed approach from the following perspectives: (1) Resilience against node capture, (2) The number of Sybil nodes required to break the entire network, and (3) The required number of intercepted messages to break the proposed scheme.

5.2.3.1 Resilience Against Node Capture

The sensor nodes are distributed randomly into the target area and equipped with secret information in order to establish their own transmission channels. However, when the adversary captures a node, he can discover its keying information and any other information that is stored in the captured node memory. The compromised nodes have the ability to negatively affect the non-compromised nodes if they have a direct communication with them or they receive random keys from the same space. In order to limit the impact of the compromised nodes on the transmission channels of the non-compromised nodes, it is important to employ a key management scheme that allows each pair of sensor nodes to establish their own transmission channels that only depends on their own keying information.

The resilience metric measures the resistance of the key management scheme to the compromised nodes. In other words, it measures the extent to which the sensor nodes that have been compromised can affect the entire network. The LU-Decomposition method provides high resilience to the network because each pair of sensor nodes derive their shared keys based on their own information where each node receives a unique row from

(L) matrix and unique column from (U) matrix. The proposed approach enhances the LU-Decomposition approach by using the BLU-Decomposition method, where it divides the original matrix into four blocks. This approach limits the number of elements each node should possess to $(N/2)$, where (N) is the number of the sensor nodes in the network which definitely limits the impact of the compromised nodes over the non-compromised nodes.

The evaluation of the resilience of the proposed approach is done by calculating the fraction of compromised communication among non-compromised nodes taking into account there are three different blocks are used. This fraction is calculated by evaluating the probability of compromising the shared keys between non-compromised sensor nodes after (x) sensor nodes have been compromised. Table 5.3 shows the list of symbols that were used in the resilience evaluation.

Table 5.3: List of Resilience Equation Symbols

The Symbol	Its Definition
x	The number of sensor nodes that have been compromised.
ω	The number of polynomials in the polynomial pool (i.e. $P_1, P_2, \dots \dots P_\omega$).
τ	The number of polynomials in each sensor node (i.e. the number of polynomials in each block).
C_X	The event that (x) nodes are compromised.
K	The shared polynomial between any non-compromised nodes in the network
A_i:	The events that the shared polynomial (K) is calculated by the (τ) polynomials, where $i = 1, 2, 3, \dots \dots \binom{\omega}{3,\tau}$. These events are exclusive, where each pair of nodes calculates its shared polynomial depends on the other node's row and its own column.

Using the combination rule with groups (Guichard, 2016), and the probability of an event (Sahoo, 2013), the probability that the shared polynomial (K) is calculated by any (τ) polynomials taking into consideration that these polynomials could belong to one of the matrix blocks is given in Equation (24).

$$\Pr\left(K \in P_1, P_2, \ldots .P_\tau\right) = \frac{1}{\binom{\omega}{3,\tau}} \tag{24}$$

Using the conditional probability theorem (Sahoo, 2013), the probability that the shared polynomial (k) is compromised when (x) nodes are compromised taking into account that the adversary needs to expose the events (A_i) that are followed to calculate the shared polynomial is given in Equation (25).

$$\Pr\left(K_{compromised} | C_x\right) = pr(A_1 \cup A_2 \cup ... \cup A_i \cup \ldots A_{\binom{\omega}{3,\tau}} | C_x) \tag{25}$$

Using the theorem of total probability (Liang Hong, 2015), Equation (25) can be written as follows:

$$pr(K_{compromised} \, | C_X) = \sum_{i=1}^{\binom{\omega}{3,\tau}} pr(A_i \, | C_X) \tag{26}$$

Using the Chain of an Event rule (Movellan, 2008) and due to the fact that all (A_i) events are equally and they have the same probability to be broken then,

$$\sum_{i=1}^{\binom{\omega}{3,\tau}} pr(A_i \,|C_X) = \binom{\omega}{3,\tau} * pr(A_1 \,|C_X)$$

$$= \binom{\omega}{3,\tau} * \left[\frac{pr\left(K \in p_1, p_2, ..., p_\tau\right) \cap \left(p_1, p_2, ..., p_\tau \text{ are compromised}\right) \cap C_X}{pr\left(C_X\right)} \right] \tag{27}$$

$$= \binom{\omega}{3,\tau} * \left[pr\left(K \in p_1, p_2, ..., p_\tau\right) * \left(p_1, p_2, ..., p_\tau \text{ are compromised|C}_X\right) \right]$$

From Equations (24, 25, 26, 27), and according to the fact that the network is being broken when the number of compromised nodes becomes more than $(t + 1)$ as presented in (Du *et al.*, 2005) the probability of compromising the shared polynomial (K) using the cumulative Binomial distribution theorem is:

$$pr\left(K_{compromised} \,|C_X\right) = \sum_{i=t+1}^{x} \binom{x}{i} \left(\frac{(\tau!)^3}{\omega!} \right)^i * \left(1 - \frac{(\tau!)^3}{\omega!} \right)^{x-i} \tag{28}$$

5.2.3.2 The Required Number of Intercepted Messages to Break The Proposed Scheme

The scheme can be broken if the adversary is able to obtain all the pair keys in the network or fabricate invalid pair keys of the sensor nodes in order to communicate with the legitimate nodes, either by monitoring the transmission medium between the legitimate nodes or by compromising the legitimate nodes and use its shared keys (Yang, Al-Anbuky and Liu, 2014). This part of the analysis focuses on the case if the adversary is able to obtain the shared keys by monitoring the transmission medium because the other way (i.e. compromising the legitimate nodes and use its shared keys) has been considered in the previous part. Since the derivation of the key pairs of the sensor nodes between the neighboring nodes based on the broadcast transmission of the keying information between the communicated nodes, the adversary has the ability to sniff this information through its malicious nodes (i.e. Sybil nodes) that are distributed randomly between the legitimate nodes.

The proposed approach depends on decomposing the original matrix that contains all the pair keys of the sensor nodes into two different matrices; (BL) and (BU) matrices, and each sensor node receives a unique row and column respectively from these matrices according to its ID. Each shared key is calculated based on the IDs of the pair nodes, their transmitted rows, and their hidden columns. Meanwhile, certain information is broadcasted between them as messages while the others are kept secret. Therefore, to break the proposed approach the adversary has to know the original matrix. In other words, the adversary should know all the elements of (BL) and (BU) matrices and the IDs of the pair nodes. To illustrate this, assume that there are two nodes (i) and (j) have a shared key $(a_{ij} = a_{ji})$, this key is obtained by multiplying the other node row by the hidden column after evaluating all the shared polynomials using the IDs values of both nodes. Therefore, the adversary needs to know their hidden columns in order to obtain their shared key as shown in Figure 5.6. According to that, it is impossible for the adversary to break the proposed approach (i.e. knows all the shared keys) through intercepting the transmitted keying information because the adversary needs to know the elements of the (BU) matrix (i.e. hidden elements) even if the adversary could obtain all the (BL) matrix elements. Therefore, the adversary needs to compromise the sensor nodes in order to break the proposed approach which will be discussed in the next section.

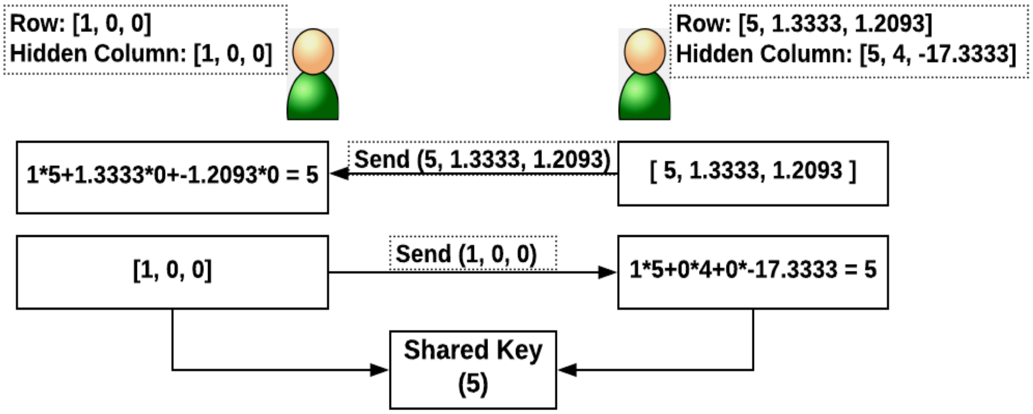


Figure 5.6: Calculate a Shared Key Based on The Transmitted Rows and The Hidden Columns

5.2.3.3 The Number of Sybil Nodes Required to Break The Entire Network

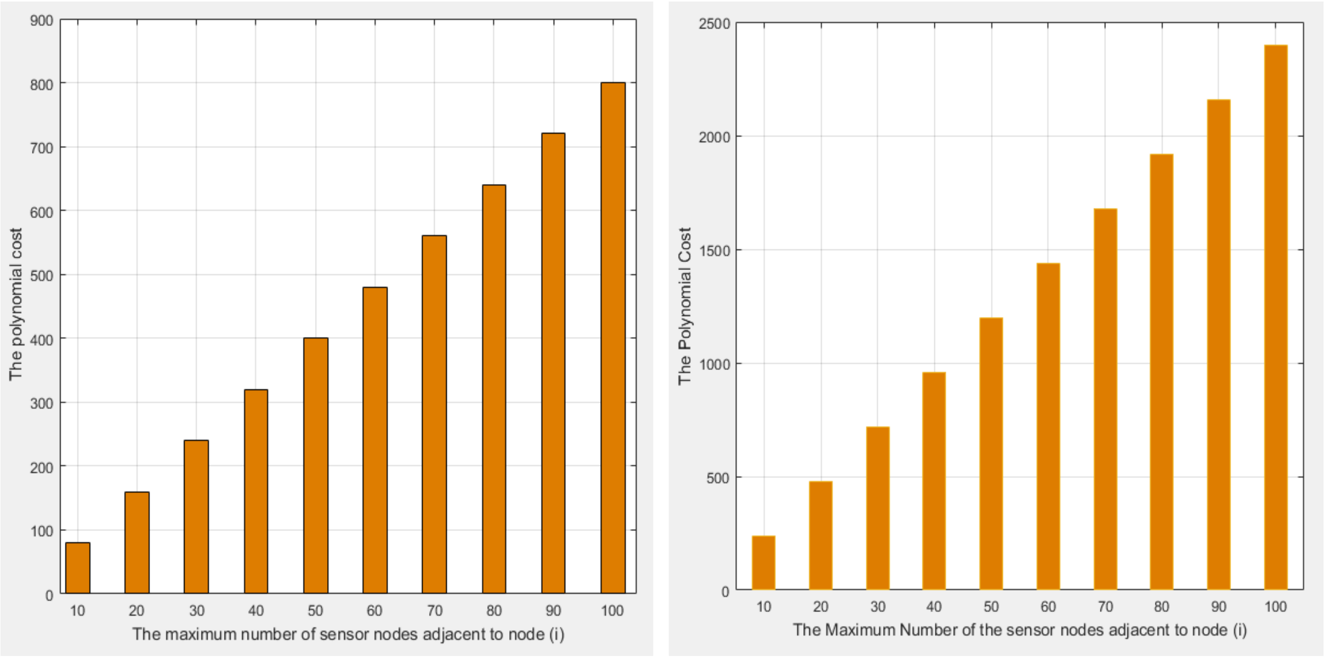
Due to the broadcast transmission of the messages between the sensor nodes, each Sybil node in the communication range of the sender sensor can receive all its messages, modify them then forward them to another node which may lead to degrade the performance of the network or break it (i.e. violate the privacy, integrity and the secrecy of the transferred data or change the topology of the network). It is challenging to identify the malicious nodes (Sybil nodes) because they can communicate with the legitimate nodes using fake identities (Debasis *et al.*, 2017). For communication purposes, any communicating nodes should compute a shared key (i.e. node-to-node mutual authentication) which prevents the Sybil nodes from fabricating invalid pair keys.

As discussed in the previous section the Sybil nodes cannot guess the shared keys through monitoring the transmission and intercepting the messages that contain part of the keying information. In addition to that, the proposed approach provides full connectivity between the sensor nodes which allows node-to-node mutual authentication. If the adversary compromises a legitimate node, he can only know all its shared keys, then he can intercept on the messages that are sent and received by this node only. However, the adversary cannot intercept the communication at the other links that are not related to the compromised node, because each key is unique. So the adversary should compromise a number of sensor nodes in order to break the network which was discussed in the resilience against node capture part in section (5.2.3.1).

5.2.4 Energy Consumption

The lifetime of the wireless sensor network depends on the energy consumption of its sensor nodes. The sensor nodes suffer from the limited energy due to its small size. The energy consumption is affected by the computation and the communication. Therefore, in the proposed approach, the initialization phase has no communication between the sensor nodes, because all the computational procedures are done at the base station (i.e. generate a set of bivariate polynomials over a finite field (F_q), build the symmetric matrix, apply the BLU- Decomposition to the matrix…etc..). As a result, the analysis of the energy consumption of the sensor nodes focuses on the phases after the sensor nodes are distributed into the target area.

In terms of the computational overhead, each sensor node needs to evaluate the polynomial using the other sensor node’s ID and multiplies its own column by the rows received from the other sensor nodes to calculate a shared key. The dominant computational procedure in the proposed approach in each sensor node is the multiplying procedure. The time complexity of the addition operation can almost be ignored in comparison with the time complexity of the multiplication process. The addition operation has less time complexity compared with the time complexity of the multiplication process, where the time complexity of the addition operation is $O(N)$, While the time complexity of multiplying two polynomials of the same size is $O(N^2)$. The time complexity of the polynomial multiplication can be reduced to $O(NlogN)$ depending on the multiplication algorithm that is used (Jia, 2017), where (N) is the degree of the polynomial.



(a) (b)

Figure 5.7: Polynomial Computational Cost When (a) t = 4; (b) t = 8

Figure 5.7 shows the computation overhead of the multiplication process in the sensor node (i) depending on the number of its adjacent nodes when (t = 4, and t = 8). The purpose of this experiment is to show that the polynomial cost is affected by two factors;

the number of the adjacent nodes of the node (i), and the degree of the polynomials (t). Figure 5.7 also shows that the greater the number of adjacent nodes are, the greater the number of multiplications in the sensor node. In addition, it also shows that the polynomial costs is increased when the degree of the polynomials increases, such that, in Figure 5.7(a) when the degree of the polynomial was (4) and the maximum number of the adjacent nodes reached to (100), the polynomial cost was (800) while in Figure 5.7(b) when the degree of the polynomial was (8) the polynomial cost was (2400). The computational overhead of the proposed approach compared with the asymmetric key cryptographic-based schemes (i.e. public key schemes) is very low. The public key algorithms, like the RSA algorithm, are based on the modular exponentiation which needs around ($\geq 2^{1024}$) multiplications (Paar and Pelzl, 2010). The public key schemes are identified as NP-complete problems that cannot be solved in a polynomial time (Wu *et al.*, 2015).

In terms of the communication consumption, the communication consumption is based on the amount of the transmitted data between the communicating nodes (Choi, Kim and Youn, 2013). In the proposed approach, the IDs of the sensor nodes and some pre-loaded information (i.e. one row) is the only information that is transmitted between any two sensor nodes to establish a secure transmission channel.

5.3 Results Comparison and Discussion

In this section, the performance of the proposed approach is compared to other approaches to prove its effectiveness. The comparison is based on simulating the equations that are generated from the previous section and compare the obtained values with the values of other equations from other approaches.

5.3.1 Memory Overhead

To prove the efficiency of the proposed approach from the memory overhead aspect, the memory-wide overhead of the proposed approach is compared with two approaches; the first one in (Dai and Xu, 2010) depends on using a pool of bivariate polynomials like the proposed approach while the other one in (Harn and Hsu, 2015) depends on using only one shared polynomial between all sensor nodes. The researchers of the first approach distribute bivariate $t-$ degree polynomials to the sensor nodes using the LU-Decomposition method. The network-wide memory overhead of this approach is given in Equation (29).

$$M_{Dai\ and\ Xu} = (t+1)L\left[\frac{N(N+1)}{2}\right] + 2ceil\left(\log_2(N-1)\right)*N \qquad (29)$$

Where:

N : The number of sensor nodes in the network.

L : The number of bits required to represent the coefficients of each polynomial.

$(t+1)$: The number of the coefficients in each polynomial.

The authors of the second approach in (Harn and Hsu, 2015) proposed a non-interactive group key pre-distribution scheme using multivariate polynomials over a finite field with $(m-1)$ variables, where (m) is the number of sensor nodes within the same group. The storage requirement for each polynomial in each sensor node in this approach is given in Equation (30).

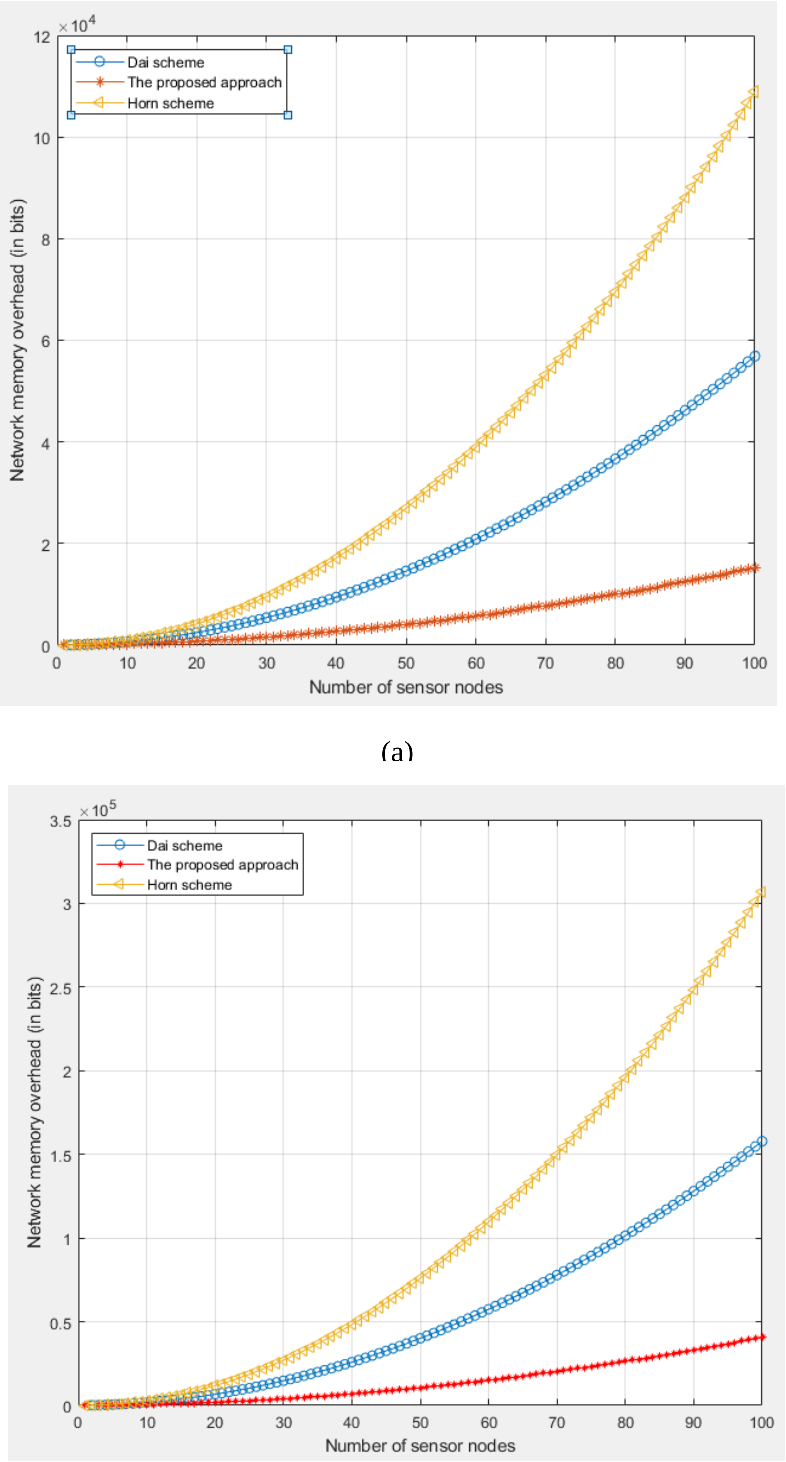
$$M_{Harn\ et\ al}=(m-1)*(t+1) \tag{30}$$

Where:

$(t+1)$: The number of the coefficients in each polynomial.

$(m-1)$: The number of variables of each polynomial

Figure 5.8 shows the memory overhead comparisons between the proposed approach and the other two approaches that are mentioned in this part. The approach in (Harn and Hsu, 2015) is based on using only one polynomial with (m) variables. The other approach in (Dai and Xu, 2010) is based on using a pool of polynomials, each polynomial with two variables (x, y). The advantage of these approaches is that they limited the number of polynomials that are required in each sensor node, where the number of polynomials is equal to the number of the sensor nodes in the network. In the experiment, the value of (L) is assumed one bit for all approaches, the degree of the polynomials is $(10,\ 30)$, and the size of the network is (100) nodes.



(a)

(b)

Figure 5.8: Network-Wide Memory Overhead Comparisons When (a) t = 10; (b) t = 30

The simulation results in Figure 5.8 shows that the proposed approach has the least memory overhead compared to the other two approaches, such that when the number of the sensor nodes exceeded (30) nodes as shown in Figuers 5.8(a)(b), the memory overhead of the network in the other two approaches started increasing while in the proposed approach the memory overhead for the network started increasing when the number of the sensor nodes reached to (40) nodes. This is because the other two approaches depend on pre-distribute (N) elements which is equal to the number of the sensor nodes in the network. While in the proposed approach each sensor node receives half of these elements (i.e. $\frac{N}{2}$ elements) which gives an advantage to the proposed approach.

5.3.2 Network Connectivity

The comparison in this part based on comparing the approaches that based on the idea of the random distribution of the keys to the sensor nodes prior to deployment and the approaches that based on the idea that the keys should be generated by the sensor nodes after the deployment. For comparison, three approaches of existing approaches have been used. The first approach presented in (Mu and Li, 2014). In this approach, the base station generates a large pool of keys then distributes (m) keys to each sensor node from the key pool. After that, each node performs (ι) times hash function operation (i.e. $H^{\iota}\,(K_i)$) on its keys to generate (m) new keys called derivative keys. The probability that two neighboring nodes can establish communication pairwise keys for a given pool size (w) and with key ring size (m) in each sensor node is given in Equation (31).

$$P_c = 1 - \frac{\binom{w-m}{m}}{\binom{w}{m}} \tag{31}$$

The researchers in the second approach in (Zhang, Li and Li, 2018) proposed a new key pre-distribution mechanism based on the polynomial pool key pre-distribution and the random probabilistic key pre-distribution schemes. In this approach, the base station builds a key pool of a set of polynomial shares. Then it stores (s) keys from the key pool in each sensor node. As each node is preloaded with (g) polynomials and (s) keys, the local connectivity of this approach is given in Equation (32).

$$P_L = 1 - \frac{\binom{m-g}{g}}{\binom{m}{g}} \frac{\binom{(w-s)(N-2)}{s}\binom{(w-s)(N-2)-s}{s}}{\binom{(w-s)(N-1)}{s}\binom{(w-s)(N-1)}{s}} \tag{32}$$

Where:

g: The number of polynomials selected for each sensor node.

s: The number of keys preloaded in each sensor

m: The number of polynomials in the polynomial pool.

N: The number of sensor nodes in the WSN.

w: The storage (bits) per node in the number of keys.

The third approach is the approach that is presented in (Dai and Xu, 2010). The authors of this approach based on the LU-Decomposition method to establish a secure transmission medium between the sensor nodes. This method is similar to the BLU-Decomposition method that is used in the proposed approach.

Figure 5.9 shows the relation between the amount of the pre-loaded keying information and the network connectivity for the proposed approach and the three approaches that are mentioned in this part of the comparison section. The number of nodes is (1000) nodes and the degree of the polynomials is (19).

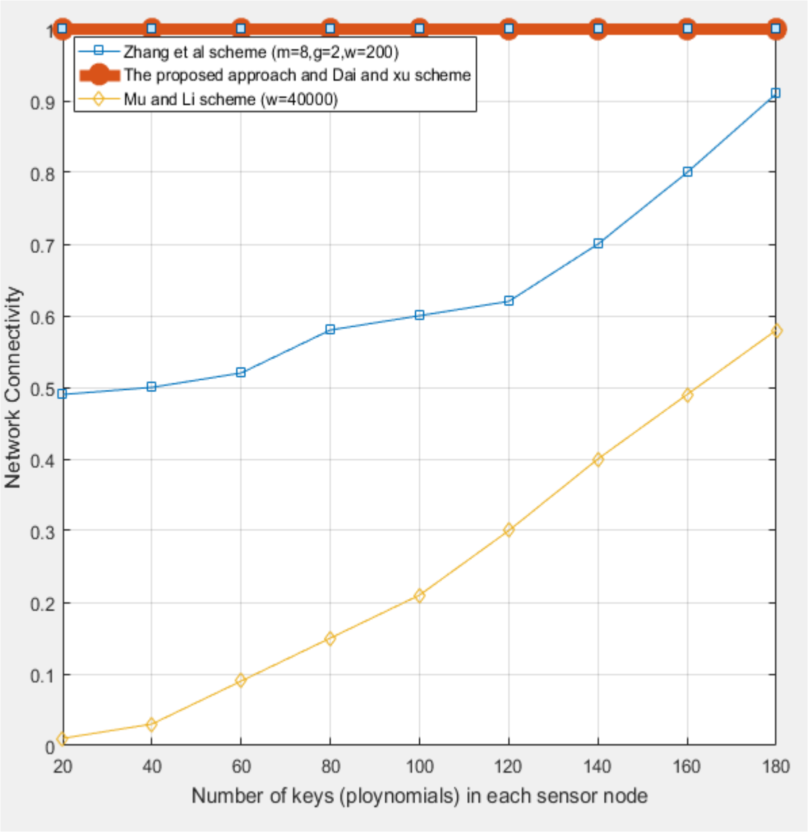


Figure 5.9: Network Connectivity Comparison Between The Proposed Approach and Existing Schemes

Due to the random distribution of the keys to the sensor nodes before the deployment in the approaches in (Mu and Li, 2014; Zhang, Li and Li, 2018), the simulation results shows that when the number of pre-loaded keys increases, the network connectivity increases. For example, when the number of keys increased from (60 to 100) the network connectivity in (Mu and Li, 2014; Zhang, Li and Li, 2018) increased from (0.1 to 0.2) and the approach in (Mu and Li, 2014; Zhang, Li and Li, 2018) the network connectivity increased from (0.51 to 0.6). This increase may impact negatively on the memory space of the sensor nodes. While the proposed approach and the approach in (Dai and Xu, 2010) has 100% connectivity because the shared keys are generated after the deployment and depending on the keying information that is preloaded to the sensor nodes.

5.3.3 Resilience Against Node Capture

To evaluate the proposed approach in this part, two approaches of the related work were used. The first one is presented in (Zhang et al., 2016). This approach based on Polynomial- Based Key Pre-Distribution scheme that is presented in (Dai and Xu, 2010) and Basic Random Key Pre-Distribution scheme that is presented in (Eschenauer and Gligor, 2002), where each sensor node receives a set of $t-$degree polynomials from a polynomial pool and another set of keys from a key pool. The distribution mechanism in this approach forces the adversary to crack both the polynomials coefficients and the keys simultaneously which gives it a high resilience. In this approach, the probability of compromise a secure link (i.e. P_{link}) is given in Equation (33).

$$P_{link} = P_K * P_P$$

$$= 1 - \left(1 - \frac{n}{S_k}\right) * \left[1 - \sum_{j=0}^{t} \binom{x}{j} \left(\frac{N}{S_p}\right)^x \left(1 - \left(\frac{N}{S_p}\right)^{x-j}\right)\right] \quad (33)$$

Where:

P_k: The probability of compromising the key pool.

P_p: The probability of compromising the polynomial pool.

x: The number of compromised nodes.

N: The number of polynomials in each sensor node.

n: The number of keys in each sensor node.

S_p: The number of polynomials in the polynomial pool.

S_k: The number of keys in the key pool.

The second approach is presented in (Dai and Xu, 2010). In this approach, the researchers merged the LU-Decomposition method with the polynomial-based key pre-distribution scheme that is presented in (Blundo *et al.*, 1998; Du *et al.*, 2005) to produce a new key management scheme. Each sensor node receives (N) $t-$degree bivariate polynomials, where (N) is the number of sensor nodes in the network. In this approach, the fraction of compromised network communication between non-compromised nodes after (x) nodes have been compromised is given in Equation (34).

$$\begin{aligned} pr(K_{compromised} \mid C_x) &= P\left(P_1, P_2, \ldots, p_\tau \text{ are compromised}\right) \\ &= \sum_{i=t+1}^{x} \binom{x}{i} \left(\frac{(\omega-\tau)!\tau!}{\omega!} \right)^{i} * \left(1 - \frac{(\omega-\tau)!\tau!}{\omega!} \right)^{x-i} \end{aligned} \tag{34}$$

Figure 5.10 shows the fraction of compromising a secure link between the non-compromised sensor nodes when the adversary captures (x) sensor nodes in the proposed approach and the approaches in (Dai and Xu, 2010; Zhang *et al.*, 2016). These approaches were used because they are similar to the proposed approach such that the first approach that is presented in (Zhang *et al.*, 2016) forces the adversary to compromise two spaces in order to compromise a node (i.e. the key space and the polynomial space). In the second approach in (Dai and Xu, 2010), although the authors use one space, it has a high resilience because of using the LU-Decomposition method in this approach, where the sensor nodes can generate their shared keys based on the pre-loaded information. Therefore, the adversary needs to compromise a high number of sensor nodes in order to expose the space that contains the shared keys. Figure 5.10 shows several experiments; when ($t = 149$) and when ($t = 99$)

The variable (p) in Figure 5.10 represents the probability of the network connectivity between the sensor nodes for a network of (40000) nodes. The approach in (Zhang *et al.*, 2016) has been evaluated using two different values for (p) (i.e. 0.97, 0.33) because the value of the resilience against node capture is changed according to the network connectivity between the sensor nodes. While in the proposed approach and in the Dai and Xu scheme, the value of the resilience against node capture cannot be affected by the probability of the network connectivity because the network connectivity is 100%. So, the probability of the network connectivity (P) for them is assumed (1). Table 5.4 shows the list of parameters that were used in the previous experiments.

Table 5.4: List of Resilience Comparisons Parameters

The proposed approach

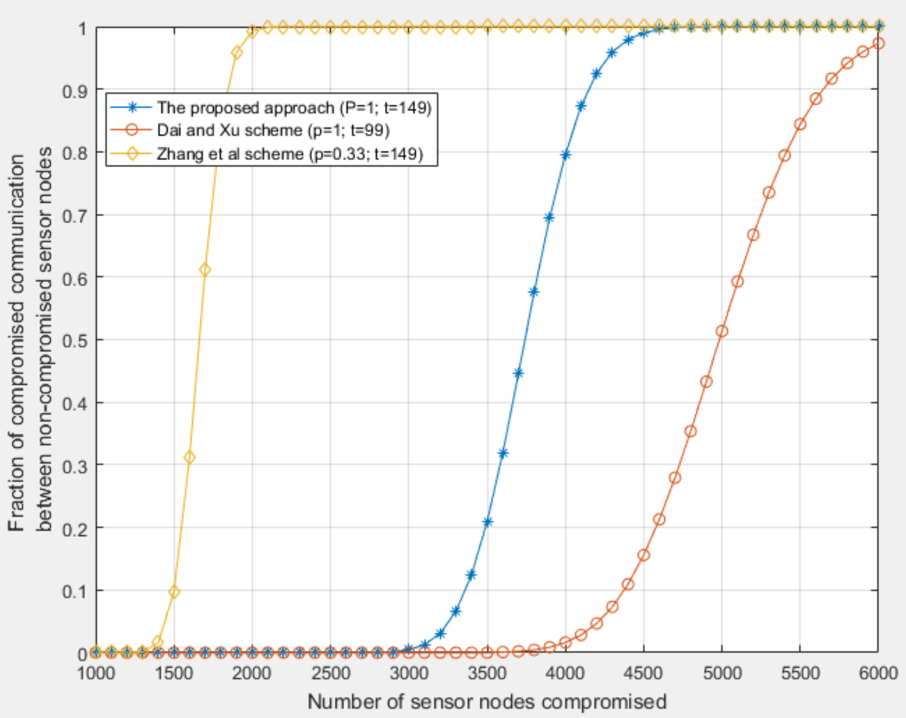
t	τ	ω	N	P
99	4	10	40000	1
149	4	10	40000	1

Dai and Xu scheme

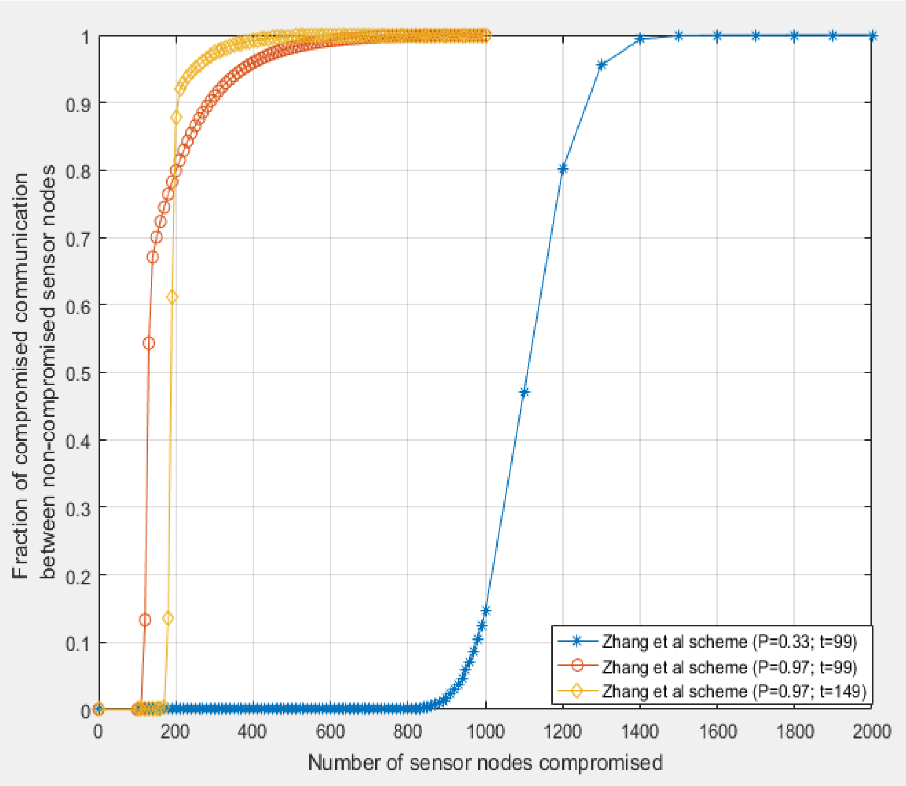
t	τ	ω	N	p
99	8	10	40000	1
149	8	10	40000	1

Zhang et al scheme

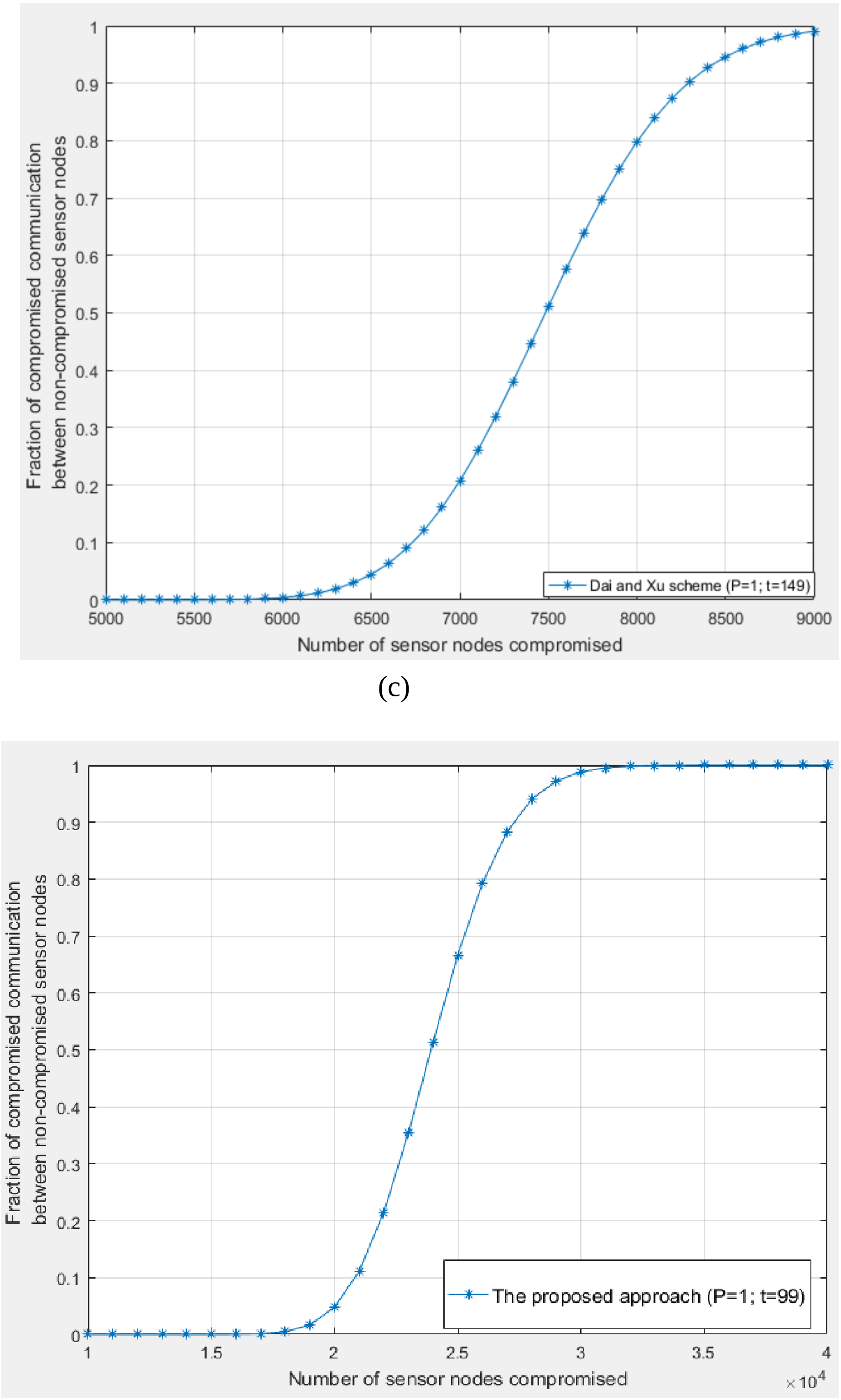
t	n	S_k	S_p	N	N	p
99	800	100000	10	8	40000	0.97
149	1200	100000	10	8	40000	0.97
99	5	1000	22	2	40000	0.33
149	5	1000	22	2	40000	0.33


Fraction of compromised communication between non-compromised sensor nodes
Number of sensor nodes compromised
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
6000
The proposed approach (P=1; t=149)
Dai and Xu scheme (p=1; t=99)
Zhang et al scheme (p=0.33; t=149)

(a)


Fraction of compromised communication between non-compromised sensor nodes
Number of sensor nodes compromised
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Zhang et al scheme (P=0.33; t=99)
Zhang et al scheme (P=0.97; t=99)
Zhang et al scheme (P=0.97; t=149)

(b)



(c)

(d)

Figure 5.10: Resilience Comparisons Between The Proposed Approach and Other Approaches

The simulation results in Figure 5.10 show the following facts: (1) in the proposed approach, the least keys (polynomials) capacity (t) in each sensor node, yields lower probability of disclosed communication links between non-compromised nodes as shown in Figures 5.10(a)(d), such that, the case in which $(t = 149)$ in Figure 5.10(a) the adversary needs to compromise at least (3100) nodes while in Figure 5.10(d) when $(t = 99)$ the adversary needs to compromise at least (15000) nodes to start compromising the other nodes in the network which is suitable for the memory space of the sensor nodes. In Dai and Xu and Zhang et al approaches, the least keys (polynomials) capacity (t) in each sensor node, yields higher probability of disclosed communication links between non-compromised nodes as shown in Figures 5.10(a)(b)(c), such that, when $(t = 149)$ in Figure 5.10(c) the adversary needs to compromise at least (6100) nodes to start compromising the other nodes in the network. In Figure 5.10(a) when $(t = 99)$ the adversary needs to compromise at least (4100) nodes to start compromising the other nodes in the network which is not suitable for the memory space of the sensor nodes. In Figure 5.10(b), when $(P = 0.33,\ t = 99)$ the adversary needs to compromise at least (900) nodes to start compromising the other nodes while when $(P = 0.33,\ t = 149)$ in Figure 5.10(a) the adversary needs to compromise at least (1499) nodes to start compromising the other nodes. So, at this point, the proposed approach is better than the other two approaches. The proposed approach achieves better performance compared to the other approaches because the adversary needs to break three spaces (i.e. blocks) not one space like Dai and Xu approach or two spaces like Zhang et al approach to start compromising the other nodes in the network. Table 5.5 shows the comparisons between the proposed approach and the other two approaches based on the polynomials capacity.

Table 5.5: Comparisons Based on The Polynomials Capacity

The Approach	**$t = 99$, Figure Number**		**$t = 149$, Figure Number**	
Proposed Approach	15000 (d)		3100 (a)	
Dai and Xu Approach	4100 (a)		6100 (c)	
Zhang et al Approach	P = 0.33 900 (b)	P = 0.97 100 (b)	P = 0.33 1499 (a)	P = 0.97 199 (b)

(2) The second fact that can be concluded from Figure 5.10 is that the probability of the network connectivity of the proposed approach and Dai and Xu scheme does not affect the resilience against node capture because they have full connectivity, unlike Zhang et al approach. In Zhang et al approach, the least probability of network connectivity (P) between the sensor nodes, yields lower probability of disclosed communication links between non-compromised nodes as shown in Figure 5.10(b). So, the case in which $(p = 0.33)$ is better than the case in which $(P = 0.97)$. In Figure 5.10(b) when $(P = 0.33, t = 99)$ the adversary need at least to compromise (900) nodes to start compromising the other nodes while in Figure 5.10(b) when the probability of the network connectivity is increased $(P = 0.97, t = 99)$ the adversary needs less nodes to compromise (100) nodes to start compromising the other nodes. So, at this point, the Dai and Xu scheme and the proposed approach has better performance than Zhang and et al approach.

5.3.4 Communication Overhead

The evaluation of the communication overhead of the proposed approach is done using MATLAB based simulation program of the Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol that is presented in (Heinzelman, Chandrakasan and Balakrishnan, 2000). It is one of the dominant hierarchical routing protocols that is used in the WSN. In the simulation, (100) sensor nodes are distributed randomly in (100×100)

region, (5%) of them are cluster head, the base station is located in the middle $(i.e.\ 50 \times 50)$. The parameters that are used in the simulation are summarized in Table 5.6.

Table 5.6: LEACH Protocol Simulation parameters

Parameter	Its value
Network dimensions	100×100
Number of sensor nodes	100 nodes
Initial energy of each sensor node	0.5 J
Energy for data aggregation	5nJ/bit/signal
E_{elec}	50nJ/bit
ε_{amp}	100 PJ/bit/m2
Location of base station	$(50\text{m}, 50\text{m})$

The communication overhead is determined as follows: For transmission, the amount of energy that is consumed by a sensor node that wants to send (k) bits of data to the other nodes with a distance (d) is given in Equation (35).

$$E_{TX}\left(k,d\right)=E_{elec}*k+\varepsilon_{amp}*k*d^{4} \tag{35}$$

Where:

E_{elec}: It is the Radio electronics energy. It is the energy that is required to run the transmitter or the receiver circuit, ε_{amp}: is the energy that is consumed by the amplifier device of the sender in order to transmit (k) bits of data, and d: is the distance (in meter) between the sender and the receiver.

The amount of energy that is consumed by the sensor node to receive (k) bits of data is given in Equation (36).

$$E_{RX}\left(k\right)=E_{elec}*k \tag{36}$$

Figure 5.11 shows the energy consumption of the sensor node (i) when the number of its neighbors increases in terms of the communication overhead, where the sensor nodes consume about 97% from the energy when they communicate and transmit data between each other. In the experiment, the degree of the polynomials is set to (49), and the number of bits required to represent the coefficients of the polynomials is assumed (1) bit.

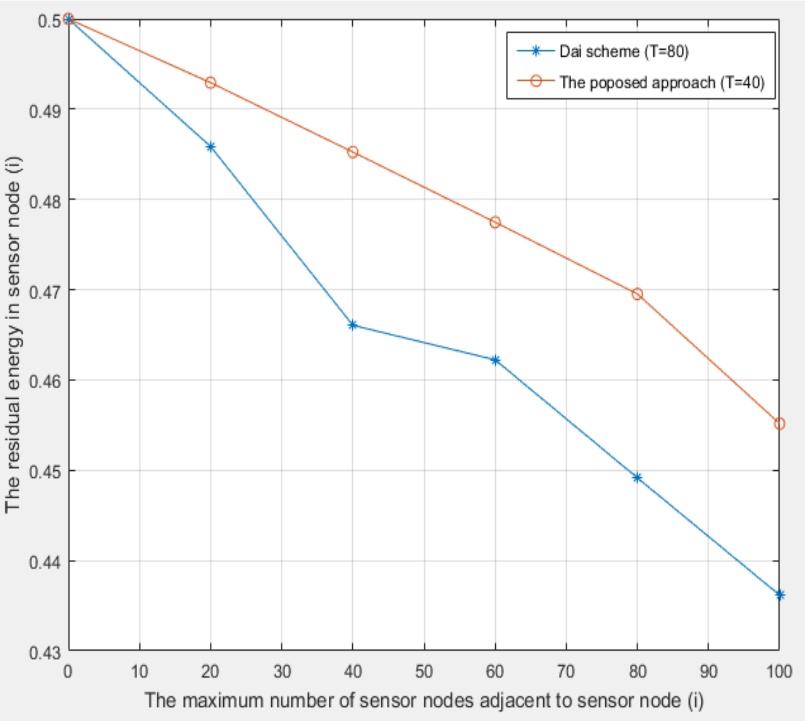


Figure 5.11: Comparison of The Residual Energy of The Schemes When (t = 49) and (L = 1)

For a fair comparison, the same amount of memory was used to store the keying information in each sensor node. In the proposed approach, the sensor nodes consume less energy when they communicate and exchange their keying information compared with the approach in (Dai and Xu, 2010) because the size of the elements that are transmitted between the sensor nodes is less. In Dai and Xu scheme, each sensor node receives (N) elements equal to the number of the sensor nodes in the network while in the proposed approach, each sensor node receives $(N/2)$ elements. Generally, the energy consumption due to the communication is determined not only by the amount of transmitted data but also by the distance between the sender and the receiver node, where the far communicators require a high energy for transmission and receiving.

Chapter Six

Conclusion and Future Work

Conclusions

The wireless sensor networks have received much attention in the civil and military applications such as environmental monitoring and border protection. The working environments of some of these applications, such as the military applications, are hostile and cannot be accessed physically. Therefore, the sensor nodes are distributed randomly into the target area. In addition to that, the transmitted messages between the sensor nodes in these applications require to be secret. Therefore, the sensor nodes require a secure medium to transmit their secret messages.

In this thesis, a new key management scheme was proposed in the wireless sensor network to improve the resilience of the network against node capture, to reduce the memory overhead and the energy consumption of the sensor node and at the same time to keep the connectivity between the sensor nodes complete. The proposed approach based on the key pre-distribution protocol, the Block LU-Decomposition algorithm and the Polynomial-Based key pre-distribution scheme. The key pre-distribution protocol was used to distribute the keying information to the sensor nodes before the deployment of the sensor nodes to increase the resilience of the network, increase the network connectivity, and to decrease the memory overhead. The Polynomial-Based key pre-distribution scheme has been used to make it difficult for the adversary to learn the shared keys due to its *t*-security property.

The main contribution of the proposed approach was the BLU-Decomposition algorithm. This algorithm gave the proposed approach several advantages:

1) It allows the proposed approach using more than one space to obtain the shared keys for the sensor nodes which make it harder for the adversary to break the proposed approach.
2) It limits the amount of the pre-distributed information to the sensor nodes compared to the random key pre-distribution approaches. This advantage reduces the memory overhead of the sensor nodes because the base station does not need to increase the amount of the pre-distributed information to increase the network connectivity. The limited amount of the pre-distributed information to the sensor nodes limits the amount of the transmitting information between them that is needed to calculate the shared keys. Since the sensor nodes consume 97% from the energy when they communicate and transmit data between them, this extra feature reduces the energy consumed by the sensor nodes while transmitting their keying information as compared to the other approaches.
3) The shared keys were obtained based on the keying information that was pre-distributed to the sensor nodes before the deployment of the sensor nodes in the field which allows the proposed approach to have full connectivity. And because each sensor node receives unique elements (polynomials), each shared key is unique. In other words, the same key cannot be repeated in more than one pair of nodes. Due to this advantage, the adversary failed to break the proposed approach while intercepting the transmitted messages between the sensor nodes

because these messages include part of the keying information that is desired to calculate the shared keys.

The encoding mechanism used in the proposed approach reduces the memory overhead of the sensor nodes. This mechanism reduces the number of zero elements that are distributed to the sensor nodes before the deployment to only one element representing the number of the zeros. In other words, each node needs to receive only one element representing the number of zeros rather than receives (N/2) zero elements. This element representing the number of zero elements that are included in the (BL) and (BU) matrices. The encoding mechanism and the BLU-Decomposition algorithm enhanced the scalability of the network (i.e. the size of the network). They decreased and limited the amount of the keying information that should be distributed to the sensor nodes. This feature allows the sensor nodes to communicate with more nodes in its range which gives the network the ability to double the number of the sensor nodes that can be employed.

The MATLAB software simulation has been used in the evaluation to compare the results of the proposed approach with the results of other approaches. The simulation results show that due to the use of the BLU-Decomposition algorithm, the proposed approach has high resilience, low memory overhead due to the reduction in the amount of the pre-distributed elements to the sensor nodes, and low energy consumption compared to other existing schemes. In addition to that, the proposed approach could provide a full connectivity between the sensor nodes regardless of the number of keys which provided node-to-node mutual authentication.

Future Work

As a future work, it is suggested to extend the proposed approach to include the key revocation and renewal features.

The sensor nodes can die or new sensor nodes can be added to existing sensor nodes and distributed into the target area. Each time a sensor node dies or a new one is added the proposed approach has to be repeated from the beginning. This can consume the energy of the existing sensor nodes which may lead to break down the network rapidly. Therefore, it is important for the sensor nodes to have the ability to update their information either to communicate with new nodes or revoke their shared keys with the died or compromised nodes without repeating the proposed approach from the beginning. The key revocation and renewal features help the sensor nodes to update their information in each period of time without repeating the proposed approach from the beginning.

References

Alshanty, A. and Erşan, I. (2016) 'Trusted Third Party Authentication Protocol Development for Clustered Wireless Sensor Networks', International Journal of Communications, Network and System Sciences, 09(11), pp. 451–470. doi: 10.4236/ijcns.2016.911037.

Anita, E. M., Geetha, R. and Kannan, E. (2015) 'A Novel Hybrid Key Management Scheme for Establishing Secure Communication in Wireless Sensor Networks', Wireless Personal Communications. Springer US, 82(3), pp. 1419–1433. doi: 10.1007/s11277-015-2290-9.

Anzani, M., Javadi, H. H. S. and Moeni, A. (2018) 'A deterministic Key Predistribution Method for Wireless Sensor Networks Based on Hypercube Multivariate Scheme', Iranian Journal of Science and Technology, Transaction A: Science. Springer International Publishing, 42(2), pp. 1–10. doi: 10.1007/s40995-016-0054-3.

Banaie, F. et al. (2014) 'MPKMS : A Matrix-based Pairwise Key Management Scheme for Wireless Sensor Networks', Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2014), 1(1), pp. 20–21.

Banaie, F. et al. (2015) 'A Polynomial-Based Pairwise Key Pre-distribution and Node Authentication Protocol for WSNs', Telkomnika (Telecommunication Computing Electronics and Control), 13(4), pp. 1113–1120. doi: 10.12928/TELKOMNIKA.v13i4.3122.

Bandara, H. and Ranasinghe, D. (2005) 'Effective GPU Strategies for LU Decomposition', Imap.Hipc.Org, 12. Available at: http://imap.hipc.org/hipc2011/studsym-papers/1569512927.pdf.

Benvenuto, C. J. (2012) Galois Field in Cryptography. Available at: https://sites.math.washington.edu/~morrow/336_12/papers/juan.pdf.

Blom, R. (1985) 'A N OPTIMAL CLASS OF SYMMETRIC KEY', in In Workshop on the Theory and Application of of Cryptographic Techniques, pp. 335–338.

Blumenthal, M. (2007) 'Encryption : Strengths and Weaknesses of Public-key Cryptography', in CSRS, pp. 1–7. doi: PA 19085 CSC 3990 – Computing Research Topics.

Blundo, C. et al. (1998) 'Distribution for Dynamic', Information and Computation, 146(1), pp. 1–23.

Çamtepe, S. A. and Yener, B. (2004) 'Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks', in In European Symposium on Research in Computer Security. Springer Berlin Heidelberg, pp. 293–308.

Chan, H., Perrig, A. and Song, D. (2003) ‘Random key predistribution schemes for sensor networks’, in Proceedings - IEEE Symposium on Security and Privacy, pp. 197–213. doi: 10.1109/SECPRI.2003.1199337.

Chen, J. et al. (2009) ‘Implementation of Block Algorithm for LU Factorization’, In Computer Science and Information Engineering, 2(4), pp. 569–573. doi: 10.1109/CSIE.2009.814.

Choi, S. J., Kim, K. T. and Youn, H. Y. (2013) ‘An energy-efficient key predistribution scheme for secure wireless sensor networks using eigenvector’, International Journal of Distributed Sensor Networks, 9(6), pp. 216–754. doi: 10.1155/2013/216754.

Chowdhury, A. R. and Dasbit, S. (2015) ‘LMAC: A lightweight message authentication code for wireless sensor network’, in 2015 IEEE Global Communications Conference, GLOBECOM 2015, pp. 1–6. doi: 10.1109/GLOCOM.2014.7417118.

Dai, H. and Xu, H. (2010) ‘Key Predistribution Approach in Wireless Sensor Networks Using LU Matrix’, Sensors Journal, IEEE, 10(8), pp. 1399–1409. doi: 10.1109/JSEN.2009.2039130.

Daoub, H. . (2012) ‘THE FUNDAMENTAL THEOREM ON SYMMETRIC POLYNOMIALS Hamza Elhadi S. Daoub’, THE TEACHING OF MATHEMATICS, (82), pp. 55–59.

Debasis, K. et al. (2017) ‘Detection of Sybil Nodes in Wireless Sensor Networks’, Indian Journal of Science and Technology, 10(3). doi: 10.17485/ijst/2017/v10i3/110641.

Demmel, J. ., Higham, N. . and Schreiber, R. . (1995) ‘1. Introduction’, Numerical linear algebra with applications, 31(2), pp. 173–190.

Divya, C. et al. (2014) ‘Analysis and Design of Various Key Pre- Distribution Schemes’, International Journal of Innovative Research in Computer and Communication Engineering, 2(4), pp. 3899–3905.

Du, W. et al. (2005) ‘A pairwise key pre-distribution scheme for wireless sensor networks’, in Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003, pp. 42–51. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-3042783638&partnerID=40&md5=d855436220fe5395318a85b38e23c997.

Eschenauer, L. and Gligor, V. D. (2002) ‘A key-management scheme for distributed sensor networks’, in Proceedings of the 9th ACM conference on Computer and communications security - CCS ’02, pp. 41–47. doi: 10.1145/586110.586117.

Giri, N. . and Mahadevan, G. (2013) ‘Key Distribution in Wireless Sensor Networks using Finite Affine Plane’, International Journal of Computer Applications, 62(19), pp. 35–38. doi: 10.1109/WAINA.2011.39.

Guichard, D. (2016) An Introduction to Combinatorics and Graph Theory. Available at: https://www.whitman.edu/mathematics/cgt_online/cgt.pdf%0Ahttp://www.freetechbooks.com/an-introduction-to-combinatorics-and-graph-theory-t1079.html.

Gundimeda, N. K. (2014) ANALYSIS OF KEY PREDISTRIBUTION SCHEMES IN WIRELESS SENSOR By. Doctoral dissertation, Memorial University of Newfoundland.

Hammack, R. (2013) Book of Proof. Available at: http://www.people.vcu.edu/~rhammack/BookOfProof/.

Harn, L. and Hsu, C. F. (2015) 'Predistribution Scheme for Establishing Group Keys in Wireless Sensor Networks', IEEE Sensors Journal, 15(9), pp. 5103–5108. doi: 10.1109/JSEN.2015.2429582.

Heinzelman, W. R., Chandrakasan, A. and Balakrishnan, H. (2000) 'Energy-Efficient Communication Protocol for Wireless Microsensor Networks', in Proceedings of the 33rd Hawaii International Conference on System Sciences, pp. 1–10.

Hoffmann, J. (2011) 'Types with potential: Polynomial resource bounds via automatic amortized analysis', in Types with potential: Polynomial resource bounds via automatic amortized analysis. Available at: http://books.google.com/books?hl=en&lr=&id=vqVkbeazN_kC&oi=fnd&pg=PR1&dq=Types+with+Potential+:+Polynomial+Resource+Bounds+via+Automatic+Amortized+Analysis&ots=EgGa_25w4U&sig=81J-00JF2Br-1BF7ir_SX4_DpyU.

Hong, L. (2015) 'An Optimized Service Broker Routing Policy for Datacenter Selection Based on Differential Evolution Algorithm by Ala ' a Khalifah Aldomi Supervisor A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF REQUIRMENT FOR THE DEGREE OF Table of Contents', in arXiv preprint arXiv:1502.00124.

Hong, L. (2015) 'Law of total probability and Bayes' theorem in Riesz spaces', in arXiv preprint arXiv:1502.00124. Available at: http://arxiv.org/abs/1502.00124.

Hsu, C. et al. (2014) 'A new secure authenticated group key transfer protocol', Wireless Personal Communications, 74(2), pp. 457–467. doi: 10.1007/s11277-013-1298-2.

Huckle, T. K., Waldherr, K. and Schulte-Herbrüggen, T. (2013) 'Exploiting matrix symmetries and physical symmetries in matrix product states and tensor trains', Linear and Multilinear Algebra, 61(1), pp. 91–122. doi: 10.1080/03081087.2012.663371.

Ilakkiya, T. D., Jayakumar, C. and Shobana, T. . (2013) 'A Secure Key Pre-distribution Scheme in Wireless Sensor Networks using Elliptic Curve Diffie-Hellman Key Exchange', in International Conference on Innovations In Intelligent Instrumentation, Optimization And Signal Processing "ICIIIOSP-2013", pp. 34–38.

Jia, Y. B. (2017) Polynomial Multiplication and Fast Fourier Transform. Available at: http://web.cs.iastate.edu/~cs577/handouts/polymultiply.pdf.

Joshi, P., Verma, M. and Verma, P. R. (2015) ‘Secure authentication approach using Diffie-Hellman key exchange algorithm for WSN’, in 2015 International Conference on Control Instrumentation Communication and Computational Technologies, ICCICCT 2015, pp. 527–532. doi: 10.1109/ICCICCT.2015.7475336.

Kadry, S. (2014) ‘Learning Basic Mathematics Using MATLAB Learning Basic Mathematics Using MATLAB’, International Journal of Information Technology and Management, 14(2).

Kallam, S. (2015) Diffie-Hellman:Key Exchange and Public Key Cryptosystems. Available at: http://cs.indstate.edu/~skallam/doc.pdf.

Khandke, M. . et al. (2013) ‘International Journal of Advanced Research in Authentication and Key Distribution Schemes for Wireless Sensors Network’, International Journal of Advanced Research in Computer Science and Software Engineering, 3(7), pp. 1343–1350.

Khuraijam, S. K. and Radhika, K. R. (2013) ‘A Novel Symmetric Key Encryption Algorithm Based on RC5 in Wireless Sensor Network’, International Journal of Emerging Technology and Advanced Engineering, 3(6), pp. 373–376.

Li, N. (2010) ‘Research on diffie-hellman key exchange protocol’, ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings, 4(4), pp. 634–637. doi: 10.1109/ICCET.2010.5485276.

Mahmood, Z., Ning, H. and Ghafoor, A. U. (2017) ‘A polynomial subset-based efficient multi-party key management system for lightweight device networks’, Sensors, 17(4), p. 670. doi: 10.3390/s17040670.

Mansour, I., Chalhoub, G. and Lafourcade, P. (2015) ‘Key Management inWireless Sensor Networks’, Journal of Sensor and Actuator Networks, 4(3), pp. 251–273. doi: 10.3390/jsan4030251.

Menezes, A. J., Van Oorschot, P. . and Vanstone, S. A. (1996) Applied cryptography, Boca Raton,FL:CRC press. Available at: https://www.taylorfrancis.com/books/9781439821916.

Movellan, J. R. (2008) Introduction to Probability Theory and Statistics, Analysis of Genetic Association …. Available at: http://link.springer.com/chapter/10.1007/978-1-4614-2245-7_1.

Mu, K. and Li, L. (2014) ‘An Efficient Pairwise Key Predistribution Scheme for Wireless Sensor Networks’, Journal of Networks, 9(2), pp. 277–282. doi: 10.4304/jnw.9.2.277-282.

Paar, C. and Pelzl, J. (2010) Understanding Cryptography, Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media. doi: 10.1007/978-3-642-04101-3 7.

Pathan, A. S. K., Dai, T. T. and Hong, C. S. (2006) ‘An efficient LU decomposition-based key pre-distribution scheme for ensuring security in wireless sensor networks’, Proceedings - Sixth IEEE International Conference on Computer and Information Technology, CIT 2006. doi: 10.1109/CIT.2006.44.

Polok, L. and Smrz, P. (2017) ‘Pivoting Strategy for Fast LU decomposition of Sparse Block Matrices’, in Proceedings of The 25th High Performance Computing Symposium, pp. 1–12.

Rani, T. . and Kumar, C. . (2012) ‘Establishment of Secure Communication in Wireless Sensor’, Computer Science & Engineering, 2(2), pp. 35–39.

Rehman, S. U. et al. (2012) Comparison Based Analysis of Different Cryptographic and Encryption Techniques Using Message Authentication Code (MAC) in Wireless Sensor Networks (WSN), International Journal of Computer Science Issues. Available at: http://arxiv.org/abs/1203.3103.

Sahoo, P. (2013) Probability & Mathematical Statistics, Journal of Chemical Information and Modeling. doi: 10.1017/CBO9781107415324.004.

Sanchez, D. and Baldus, H. (2005) ‘A Deterministic Pairwise Key Pre-distribution Scheme for Mobile Sensor Networks’, in Security and Privacy for Emerging Areas in Communications …, pp. 277–288. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1607585%5Cnfile:///Users/insel/Dropbox/Papers3/Library.papers3/2005/Sanchez/Security and Privacy for Emerging Areas in Communications … 2005 Sanchez.pdf%5Cnpapers3://publication/uuid/49CBFB4C-59D2-4BF.

Singh, S. K., Singh, M. P. and Singh, D. K. (2010) ‘Routing protocols in wireless sensor networks’, International Journal of Computer Science & Engineering Survey, 1(2), pp. 36–83. doi: 10.3390/s91108399.

Sziklai, P. (2013) Applications of Polynomials Over Finite. Available at: http://web.cs.elte.hu/~sziklai/nagydoktori/nd.pdf.

Tandel, R. I. (2016) ‘Leach Protocol in Wireless Sensor Network : A Survey’, International Journal of Computer Science and Information Technologies, 7(4), pp. 1894–1896.

Tharani, T. A., Suganthi, N. and Srinithi, R. (2014) ‘Matrix based Key Pre – Distribution Scheme for Wireless Sensor Networks’, International Journal of Computational Intelligence and Informatics, 4(2), pp. 140–144.

Tiwari, P. et al. (2015) ‘Wireless Sensor Networks: Introduction, Advantages, Applications and Research Challenges Introduction to Wireless Networks’, HCTL Open International Journal of Technology Innovations and Research ISBN, 14, pp. 1–11. Available at: http://ijtir.hctl.org.

Ursell, U. (2017) encoder.m decoder.m - File Exchange - MATLAB Central, MathWork. Available at: https://www.mathworks.com/matlabcentral/fileexchange/61435-encoder-m-decoder-m (Accessed: 11 July 2018).

Wu, W. et al. (2015) ‘A public key cryptosystem based on data complexity under quantum environment’, Science China Information Sciences, 58(11), pp. 1–11. doi: 10.1007/s11432-015-5408-5.

Yang, M., Al-Anbuky, A. and Liu, W. (2014) ‘An Authenticated Key Agreement Scheme for Wireless Sensor Networks’, Journal of Sensor and Actuator Networks, 3(3), pp. 181–206. doi: 10.3390/jsan3030181.

Yum, D. H. and Lee, P. J. (2012) ‘Exact formulae for resilience in random key predistribution schemes’, IEEE Transactions on Wireless Communications, 11(5), pp. 1638–1642. doi: 10.1109/TWC.2012.031212.110887.

Zhang, J., Li, H. and Li, J. (2018) ‘Key establishment scheme for wireless sensor networks based on polynomial and random key predistribution scheme’, Ad Hoc Networks. Elsevier B.V., 71, pp. 68–77. doi: 10.1016/j.adhoc.2017.12.006.

Zhang, Y. et al. (2016) ‘A hybrid key management scheme for WSNs based on PPBR and a tree-based path key establishment method’, Sensors (Switzerland), 16(4). doi: 10.3390/s16040509.

Zhao, L. and Ye, L. (2014) ‘Pair-Wise Key Predistribution Using the Deployment Knowledge in WSN’, in Proceedings of the 2nd International Conference on Soft Computing in Information Communication Technology, pp. 98–102. doi: 10.2991/scict-14.2014.23.

Zheng, J. and Jamalipour, A. (2009) ‘Introduction to Wireless Sensor Networks’, Wireless Sensor Networks: A Networking Perspective, 1(1), pp. 1–18. doi: 10.1007/978-1-4939-2468-4_1.

Zhu, L. et al. (2016) ‘An Improved Random Key Predistribution Scheme for Wireless Sensor Networks Using Deployment Knowledge Lina’, International Journal of Security and Its Applications, 10(5), pp. 225–234. doi: 10.1007/s00607-009-0036-9.

Zia, T. a and Zomaya, A. Y. (2011) ‘A Lightweight Security Framework for Wireless Sensor Networks’, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2(3), pp. 53–73.