



**A NODE AUTHENTICATION MECHANISM ON WIRELESS SENSOR
NETWORKS**

KAMERAN ALI AMEEN

MAY 2015

**A NODE AUTHENTICATION MECHANISM ON WIRELESS SENSOR
NETWORKS**

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY**

**BY
KAMERAN ALI AMEEN**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
MATHEMATICS AND COMPUTER SCIENCE
INFORMATION TECHNOLOGY PROGRAM**

MAY 2015


Title of the Thesis: **A Node Authentication Mechanism on Wireless Sensor Networks.**

Submitted by **Kameran Ali AMEEN**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.


Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.


Prof. Dr. Billur KAYMAKÇALAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.


Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV
Supervisor


Dr. Ahmed Chalak SHAKIR
Co-Supervisor




Examination Date: 07.05.2015

Examining Committee Members:

Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV (Çankaya Univ.)

Assist. Prof. Dr. Abdül Kadir GÖRÜR (Çankaya Univ.)

Assoc. Prof. Dr. Fahd JARAD (THK Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Kameran Ali, AMEEN

Signature :



Date

: 07.05.2015

ABSTRACT

A NODE AUTHENTICATION MECHANISM ON WIRELESS SENSOR NETWORKS

AMEEN, Kameran Ali

M.Sc., Department of Mathematics and Computer Science
Information Technology Program

Supervisor: Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV

Co-Supervisor: Dr. Ahmed Chalak SHAKIR

May 2015, 70 pages

A wireless sensor network (WSNs) is a network that consists of a large number of ultra-small autonomous devices which are resource-constrained called sensor nodes. Sensor networks may be deployed for a wide range of applications, including military sensing, environment monitoring and patient monitoring, etc. Sensors can collect, process and transmit data in a distributed and cooperative manner. They are usually deployed in an unattended environment (open area) and communicate with each other through wireless channels. Therefore, security is necessary to protect data from various types of attacks. The authentication process is one of the important ways to achieve security in WSNs. In this thesis, the pros of both symmetric and asymmetric key cryptography are used to achieve security. The proposed scheme secures communication by distributing the public key securely to both the base station and cluster head; as a result, this yields security of the message. The proposed scheme is efficient due to the application of data sequence over ECC, which in turn depends on the ECDLP. Therefore, it provides far more security and resistance against a number of attacks when compared with other existing schemes.

Keywords: Wireless Sensor Network, Authentication, Data Sequence, Security Issues, ECC, Scalar Multiplication.

ÖZ

TELSİZ SENSÖR ŞEBEKELERİNDE DÜĞÜM ONAYLAMA MEKANİZMASI

AMEEN, Kameran Ali

Yüksek Lisans, Matematik-Bilgisayar Anabilim Dalı

Bilgi Teknolojileri Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Yuriy ALYEKSYEYENKOV

Eş Tez Yöneticisi: Dr. Ahmed Chalak SHAKIR

Mayıs 2015, 70 sayfa

Telsiz sensör şebekesi (WSNs) sensör düğümleri olarak tanımlanan kısıtlı kaynağa sahip çok sayıda ultra- küçük bağımsız cihazlardan oluşan bir şebekedir. Sensor şebekeleri; askeri algılama, çevre izleme ve hasarların izlenmesi gibi geniş bir uygulama alanına sahiptir. Sensörler dağıtılmış bir şekilde ve işbirliği içerisinde verileri toplayabilmekte, işletebilmekte ve iletebilmektedir. Genellikle açık alanlara yerleştirilir ve telsiz kanallar vasıtasıyla birbirleriyle iletişim kurar. Bu itibarla, bilgileri çeşitli saldırılara karşı korumak için güvenlik gereklidir. Onaylama WSNlerde güvenliği sağlamanın önemli yollarından biridir. Bu tezde, güvenliğin sağlanması için hem simetrik hem de asimetrik kilit kriptografinin avantajları kullanılmaktadır. Önerilen proje ortak anahtarı hem baz istasyonuna hem de küme-başına güvenli bir şekilde dağıtarak haberleşmeyi emniyet altına almakta ve sonuç itibarıyla mesaj güvenliğini sağlamaktadır. Önerilen proje veri sırasının ECDLP' ne bağlı ECC üzerinde uygulanması nedeniyle verimlidir. Bu itibarla, diğer projelerle karşılaştırıldığında, bu yöntem çeşitli saldırılara karşı çok daha fazla güvenlik şartı ve direnci sağlamaktadır.

Anahtar Kelimeler: Telsiz Sensör Şebekesi, Onay, Veri Sırası, Güvenlikle İlgili Hususlar, ECC, Skalerle Çarpma.

ACKNOWLEDGEMENTS

At the outset, I would like to express my sincere gratitude to Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV for his supervision and for providing suggestions and tips to me, as well as for his immense knowledge throughout the development of this thesis. Also, I would like to express my sincere thanks towards Dr. Ahmed Chalak SHAKIR, the minor supervisor, who allocated his time and knowledge in the implementation of this project.

I express my sincere thanks to my wife for supporting me throughout the study period and to my family for their unlimited support, especially my mother, brothers and sisters for encouraging and supporting me all the time.

Distinct thanks go to the Iraq Ministry of Higher Education and Kirkuk University for their complete the study.

TABLE OF CONTENTS

STATEMENT OF NON PLAGIARISM.....	iii
ABSTRACT.....	iv
ÖZ.....	v
ACKNOWLEDGEMENTS.....	vi
TABLE OF CONTENTS.....	vii
LIST OF FIGURES.....	viii
LIST OF TABLES.....	x
LIST OF ABBREVIATIONS.....	xi

CHAPTERS:

1. INTRODUCTION AND LITERATURE SURVEY.....	1
1.1. Background.....	1
1.2. Applications of WSNs.....	7
1.2.1. Military	7
1.2.2. Health care	7
1.2.3. Environmental.....	8
1.2.4. General engineering.....	9
1.3. Literature Survey	9
1.4. Motivation.....	12
1.5. The Statement of the Problem	13
1.6. Organization of the Thesis.....	13
2. SECURITY REQUIREMENTS AND THE CRYPTOGRAPHY	
MECHANISM IN A WSN.....	15
2.1. Security in Wireless Sensor Network.....	15
2.2. Security Requirements in WSN.....	15
2.2.1. Authentication	15
2.2.2. Integrity.....	16

2.2.3.	Confidentiality.....	16
2.2.4.	Scalability.....	16
2.2.5.	Availability.....	16
2.3.	Kinds Attacks in Wireless Sensor Network.....	16
2.3.1.	Sybil attack.....	17
2.3.2.	Hello flood attack	17
2.3.3.	Black hole attack.....	18
2.3.4.	Clone attack.....	18
2.3.5.	Tunneling attack.....	18
2.3.6.	Modification and insertion attack.....	19
2.4.	Basic Cryptography Mechanism in WSN.....	19
2.4.1.	Symmetric key cryptography.....	20
2.4.2.	Asymmetric key cryptography.....	20
2.5.	Message Authentication Code	21
2.6.	Elliptic Curve Cryptography (ECC).....	21
2.6.1.	Overview of elliptic curve cryptography.....	21
2.6.2.	Fundamental mathematical terminology	22
2.6.3.	Elliptic curve discrete logarithm problem (Ecdlp).....	24
2.6.4.	Elliptic curve cryptography operations.....	24
2.6.5.	Scalar multiplication in elliptic curve cryptography.....	27
2.6.6.	Elliptic curve cryptography encryption and decryption.....	27
2.7.	Authentication in Wireless Sensor Network.....	28
3.	THE ENHANCED DATA SEQUENCE METHOD FOR ECC CRYPTOSYSTEM (EDS-M-ECC).....	30
3.1.	Introduction.....	30
3.2.	Elliptic Curve Definition.....	31
3.3.	Background on Data Sequence.....	32
3.4.	The Proposed Method Explaining	32
3.4.1.	Generating the data sequence process.....	32

3.4.2.	Description of applying the created sequence on the ECC process	35
3.5.	Application of the Proposed Method.....	37
3.6	Analysis of the Proposed Method.....	42
3.7	Outcome.....	45
4.	SYSTEM IMPLEMENTATION AND ANALYSIS OF RESULTS.....	46
4.1.	Introduction	46
4.2	The System Model.....	46
4.3	Simulation Setup.....	47
4.4	System Assumptions.....	47
4.5	The Proposed Scheme.....	48
4.5.1.	Pre-distribution phase	48
4.5.2.	Deployment phase	48
4.5.3	Public key distribution phase	50
4.5.4	Message authentication phase	60
4.6	Results and Analysis of the Proposed Scheme	65
4.6.1	Simulation results	65
4.6.2	Security analysis	66
5.	CONCLUSION.....	70
	REFERENCES.....	R1
	APPENDICES.....	A1
A.	CURRICULUM VITAE.....	A1

LIST OF FIGURES

FIGURES

Figure 1	Type size of the sensors.....	1
Figure 2	Types of WSN architecture.....	2
Figure 3	Architecture of MAC	4
Figure 4	Hash function.....	4
Figure 5	Symmetric key cryptography.....	5
Figure 6	Asymmetric key cryptography.....	5
Figure 7	Applying a WSN in military field.....	7
Figure 8	Applying a WSN in health monitoring field	8
Figure 9	Applying a WSN in an environmental field	8
Figure 10	Applying a WSN in engineering field	9
Figure 11	Sybil attack.....	17
Figure 12	Hello flood attack.....	17
Figure 13	Block hole attack	18
Figure 14	Tunneling attack.....	19
Figure 15	Modification and insertion attack.....	19
Figure 16	The role of security	20
Figure 17	Point addition if $J \neq K$	25
Figure 18	Point addition if $J = -J$	25
Figure 19	Point double	26
Figure 20	Point double if $y_J = 0$	26
Figure 21	Encryption using elliptic curve cryptography algorithm.....	28
Figure 22	Decryption using elliptic curve cryptography algorithm.....	28
Figure 23	Elliptic curve mathematical hierarchy.....	31
Figure 24	The points of the EC over the prime field.....	37
Figure 25	Encryption by EDS-M-ECC algorithm.....	41

FIGURES

Figure 26	Decryption by EDS-M-ECC algorithm.....	42
Figure 27	A Comparison between EDS-M-ECC [33] to send (hello).....	43
Figure 28	Hierarchical model of WSN.....	47
Figure 29	The <i>BS</i> setup.....	49
Figure 30	The sensors distribution.....	49
Figure 31	The <i>CHs</i> setup.....	50
Figure 32	The <i>BS</i> sends its public key to each <i>CH</i> algorithm.....	51
Figure 33	The <i>BS</i> sends its public key to each <i>CH</i> flowchart.....	52
Figure 34	The <i>BS</i> sends its public key to each <i>CH</i>	52
Figure 35	Decryption and authentication public key of the <i>BS</i> algorithm..	53
Figure 36	Decryption and authentication public key of the <i>BS</i> flowchart..	54
Figure 37	Each <i>CH</i> sends ACK to <i>BS</i>	54
Figure 38	The <i>CH</i> sends its public key to $L_{members}$ algorithm.....	57
Figure 39	The <i>CH</i> sends its public key to $L_{members}$ flowchart.....	57
Figure 40	Each <i>CH</i> sends its public key to $L_{members}$	58
Figure 41	Decryption and authentication public key of the <i>CH</i> algorithm..	58
Figure 42	Decryption and authentication public key of <i>CH</i> flowchart.....	59
Figure 43	L_{member} sends their data to its <i>CH</i> algorithm.....	61
Figure 44	Decryption and authentication message by <i>CH</i> algorithm.....	62
Figure 45	<i>CH</i> sends their data to <i>BS</i> algorithm.....	63
Figure 46	Decryption and authentication message by <i>BS</i> algorithm.....	64
Figure 47	L_{member} sends cipher message to <i>CH</i> then to <i>BS</i>	64
Figure 48	Time Execution of the 30 operations for the base point.....	66
Figure 49	Network lifetime.....	69

LIST OF TABLES

TABLES

Table 1	ECC key Sizes Compared with RSA Key Sizes.....	22
Table 2	Properties of Abelian Group.....	23
Table 3	Data Sequence Generate.....	33
Table 4	Points on the Elliptic Curve $E_{29}(-1, 16)$	38
Table 5	Data Sequence Form.....	39
Table 6	Encryption Message Before/After Applying DS.....	43
Table 7	A Comparison Between EDS-M-ECC and [33] to Send (hello).	43
Table 8	A Comparison EDS-M-ECC and [33] Based on no. of Digits....	44
Table 9	A Comparison EDS-M-ECC and [33] Based on no.of Bits.....	44
Table 10	Simulation Parameters.....	47
Table 11	Computation of Time Execution of Scalar Multiplication.....	65
Table 12	Comparison in Requirements Security.....	67
Table 13	The Attacks Security Comparison (Y=Yes,N=No).....	68

LIST OF ABBREVIATIONS

DARPA	Defense Advanced Research Projects Agency
DSN	Distributed Sensor Networks
BS	Base Station
WSN	Wireless Sensor Network
ECDLP	Elliptic Curve Discrete Logarithm Problem
MAC	Message Authentication Code
M	Message
SK	Share Key
SKC	Symmetric Key Cryptography
AKC	Asymmetric Key Cryptography
ECC	Elliptic Curve Cryptography
RSA	Rivest-Shamir-Adleman Algorithm
PKC	Public Key Cryptography
CA	Certificate Authority
ID	Identity
ECDH	Elliptic Curve Diffie-Hellman
SHA-1	Secure Hash Algorithm-1
HSN	Heterogeneous Sensor Network
L	Low Sensor
SCK	Self-Certified Keys Cryptosystem
LEACH	Low Energy Adaptive Clustering Hierarchy
CH	Cluster Head
EC	Elliptic Curve
DES	Data Encryption Standard
Ecdlp	Elliptic Curve Discrete Logarithm Problem
DSA	Digital Signature Algorithm

ECDSA	Elliptic Curve Digital Signature Algorithm
DS	Data Sequence
Q	Public Key
P	Base Point
k	Private Key
Pm	Plain Message
Cm	Cipher Message
ACK	Acknowledgment

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

1.1 Background

Thanks to the developments in the technology of manufacturing equipment particularly in the sensor networks, we have a smooth and promising way towards building networks. Thus, it becomes possible to deploy networks consisting several thousands of sensor networks [1]. Furthermore, these networks may be wireless sensor networks, which is one of the most basic technologies for the 21st century [2]. The first use of sensors occurred back in the early 1980s; the Defense Advanced Research Projects Agency (DARPA) had executed the Distributed Sensor Network (DSN). The military field had focused on wireless sensor networks until the 1990s, after which they came to be used in commercial fields and in different scopes [3]. Sensors, communications and processing are the three main elements whose combination is a small device. Their uses in many applications are endless in all levels of life, whether in the military or civilian applications. On the other hand, this technology presents enormous challenges, especially restrictions on energy, as well as challenges of power, the processor, memory, transmission range and the price of the sensor devices.

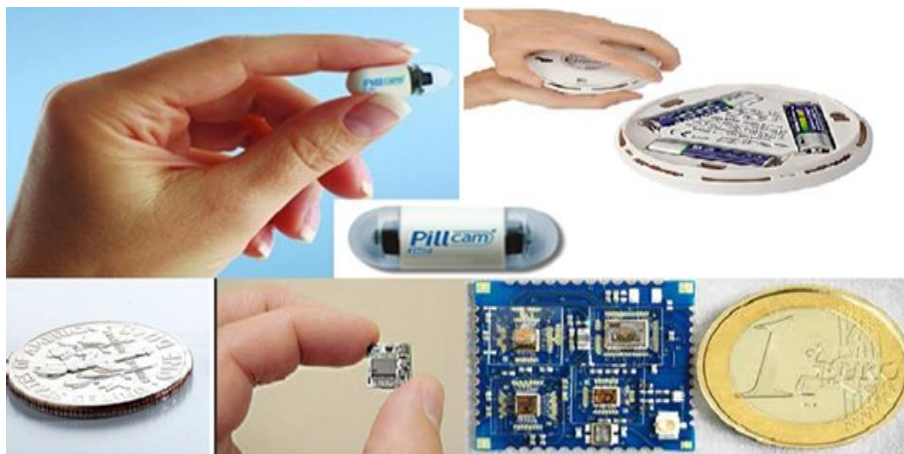


Figure 1 Type size of the sensors

These affect their characteristics and design. Thus, the price of sensors will be high when their quality increases. Fig.1 shows the different sizes of sensors [4].

A sensor network typically consists of a set of devices (resource-constrained) which are small and lightweight. In addition, these small, lightweight devices create a network through wireless links between them. This means that sensor nodes will be connected wirelessly with each other. A sensor has the ability to collect data from remote locations at one node or at the base station in order to achieve maximum life of the network [1, 5] in addition to implementing one function, such as sensing and processing and implementing many applications and objectives [6].

Wireless Sensor Networks (WSNs) are usually deployed in different environments, including hostile regions [6]. Self-organization is one property of sensor networks in hostile regions, which consequently means it has no fixed infrastructure. Moreover, there are two types of infrastructure network architecture, one of which is called a flat wireless sensor network and the other a hierarchical wireless sensor network. The former comprises the base station, which is one distinctive component of a WSN with more computational, energy and communication resources. Moreover, they act as a gateway sending the gathered data from the sensor nodes to the end user or administrator [7], and the sensors have the same characteristic and complexity because each sensor works as a sensing part and controller. However, in hierarchical networks, the network elements include a BS with low sensors, which are normal sensors with simple management and unlimited resources and a high sensor. Figure 2 shows the types of infrastructure network architecture [8].

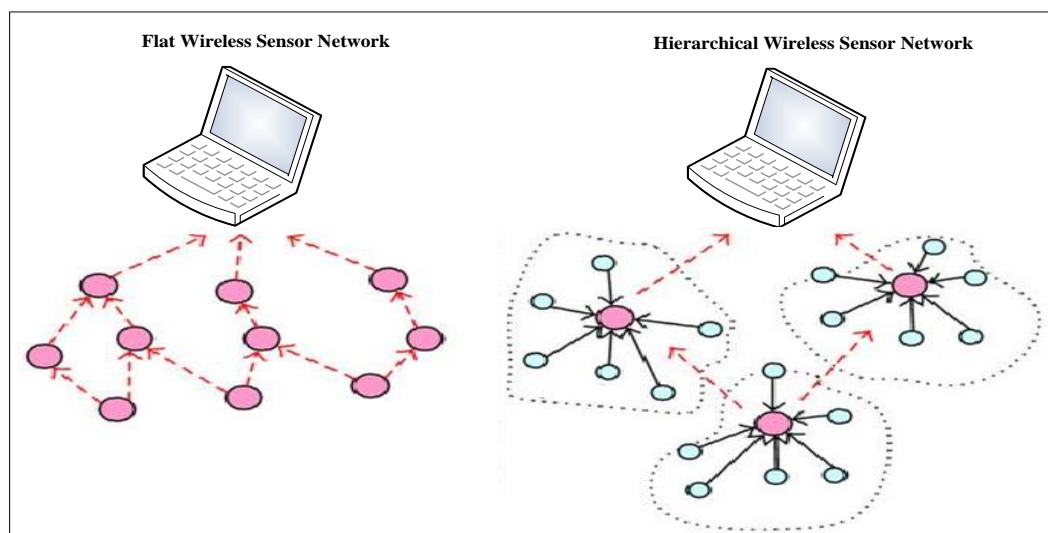


Figure 2 Types of WSN architecture

Furthermore, due to resource constraints, sensor networks cannot use complex security solutions and in wireless sensor networks, this must be taken into account. Additionally, traditional security solutions are unworkable for WSNs due to the unique characteristics of sensor networks. Nevertheless, WSNs require security solutions with high efficiency [9].

We can simply determine this as the study the mathematical techniques related to services of information security, such as confidentiality, integrity, and entity authentication. It is used to transform the original data into another form and transfer it via the network, which may or may not be safe. Thus, it cannot be read only by authorized parties [10]. In addition, there are many algorithms that are used to support security applications, such as message integrity and node authentication, which cannot define any better algorithm. There is a request to design security algorithms which are simple, flexible and scalable. However, designing such security algorithms is not an easy task. Stronger security algorithms may consume additional resources on sensor nodes, which can lead to a drop in performance of applications. Therefore, a balance between security and efficiency must be [6]. Security requirements essentially center on authentication, confidentiality and integrity. Authentication between nodes or nodes to the base station is important to ensure whether the identity of the sender for the receiver is a legal node or an illegal node. It ensures that only the authorized node can be a member for the communication. Integrity ensures that the data does not change during the communication. Confidentiality must be certain of the source and destination so as to ensure the data that is being sent [11]. Thus, without authentication between the parties involved in the communication, an attacker can easily modify the contents of the message or spread false information throughout the WSN. Therefore, the security design of these networks becomes an important and challenging design task [6].

Within the framework of WSNs, authentication can be formed from data authentication and nodes authentication.

There are three kinds of functions that can be used to produce an authenticator:

1. Message Authentication Code (MAC): This is a type of symmetric key cryptography that depends on fundamentals to produce a message authentication as a MAC value that is $MAC = (SK, M)$ and is used to exchange authentic messages; a sender and receiver must share the same key [9-12] as shown in Fig. 3.

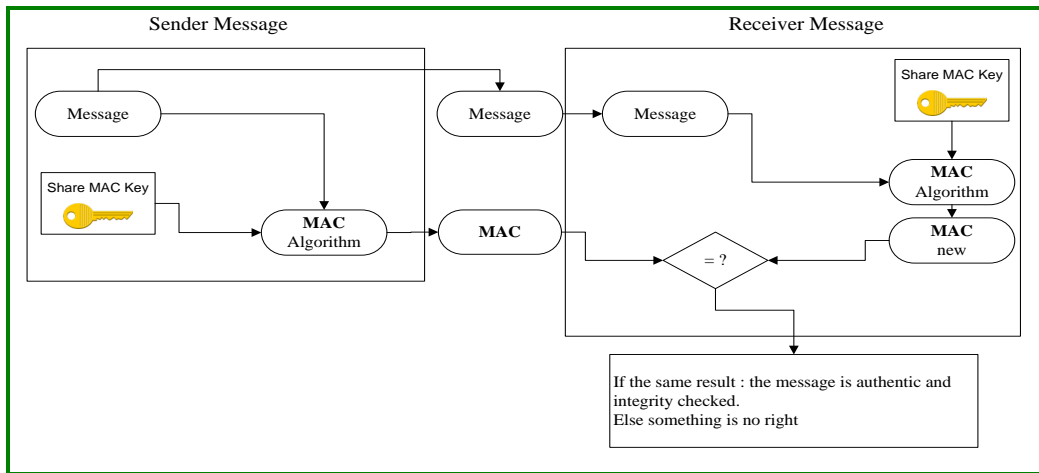


Figure 3 Architecture of MAC

2. Hash function: A hash function as known as a one-way function (which cannot be reversed) and requires entry of a message, of any length, ultimately being a fixed-length output. The output of the function can usually be called a message digest, a hash value, a checksum or a hash code [13], as shown in Fig. 4.

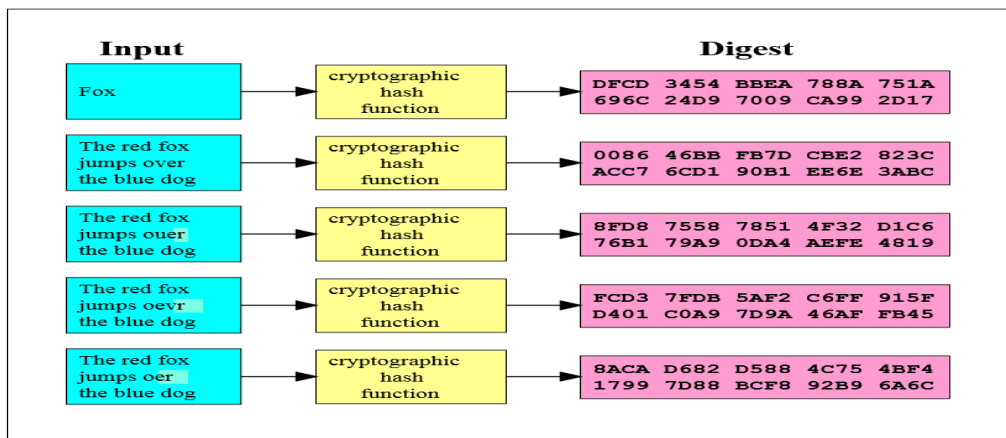


Figure 4 Hash function

3. Message Encryption: The process of encrypting the message by using symmetric key cryptography or asymmetric key cryptography provides security and has in itself a sufficient available amount of authentication. In addition, the respective SKC of the sender and receiver use the same key for encrypting and decrypting a message, as shown in Fig. 5. However, the respective AKC of the sender and receiver uses two keys, one for encryption and the other for decryption [12-14], as shown in Fig. 6.

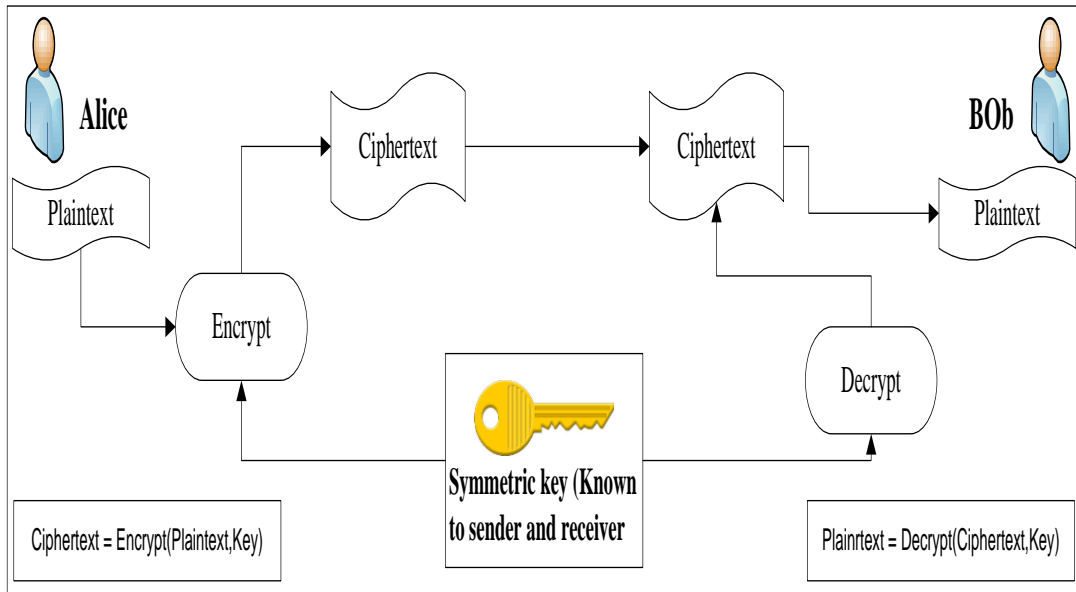


Figure 5 Symmetric key cryptography

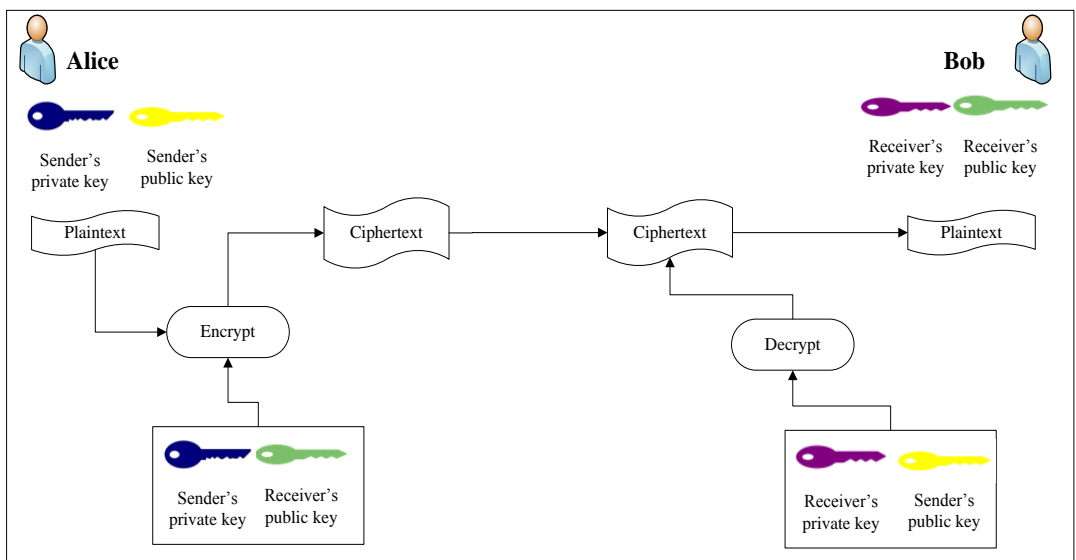


Figure 6 Asymmetric key cryptography

A sensor network is deployed in an open or unprotected field. It is able to collect and transfer data from numerous locations. Due the weakness of wireless communications, the wireless medium threatens the security of compiled data [9] thereby making the data prone to active and passive attacks. In addition, due to limited resources of WSNs, traditional algorithms used in wired networks cannot be used. It is therefore a challenge to incorporate basic security functions [15]. The important requirements in security are authentication, integrity and confidentiality,

and all three are required for almost every WSN in applications. These requirements are achieved by using the cryptography mechanics to obtain security, including SKC and AKC [16]. In addition, both of these have advantages and disadvantages. In our proposal, the scheme will exploit only from the pros of both.

Elliptic Curve Cryptography is a typical example of a public key algorithm implemented in wireless sensor networks. The Rivest-Shamir-Adleman algorithm is the competitor of the ECC due to its large computation and communication overhead and large key size. The ECC algorithm has a faster implement time, lower memory overhead and a smaller key size than the RSA algorithm, which makes it very suitable for data encryption in WSNs [8-11].

Public key cryptography (PKC) has a pair of keys (private and public) wherein the private key is secret while the second key is public. Additionally, an important issue of applying public key cryptography proves the authenticity of another party's public key to confirm that the public key is certainly owned by the person it claims to be a member. Therefore, the public keys of nodes must be authenticated in a WSN prior to communication [17-18]. In previous studies, authentication of public key nodes was by means of the Certificate Authority (CA); hence, they can use this public key to check signatures on certificates [18-19] that do not meet the needs of real-time applications. Moreover, signing a message consumes more time than MAC. The public key and certificate management in wireless sensor networks is another issue causing the scalability problem. It is not possible for nodes to store the public keys of all other nodes prior to deployment and this affects memory size [9]. In addition, public key authentication following the deployment phase can be achieved in a much more power-efficient manner [18]. Conversely, adversaries can easily spoofing any node by claiming its public key and acting as an invisible router to learn all the messages between the nodes [6]. Moreover, the authentication can ensure the reliability of the message by identifying its origin and ensuring the integrity and confidentiality of the message [20]. Our proposed scheme creates authentication by using SKC through ID with a share key based on MAC and PKC by applying a data sequence over the ECC.

1.2 Applications of WSNs

WSNs provide significant advantages over wired sensors due to network wired sensors being more time consuming in terms of construction and delays in deployment. Therefore, their use is favored in applications that require immediate processing of collections of information. Wireless sensors provide significant benefits when compared to wired sensors [21].

1.2.1 Military

Wireless sensors can be very rapidly deployed and without the need for the presence of pre-installed infrastructure [5-21]. These characteristics make WSN technology very promising for military systems. Military uses include tracking enemies, determining the location of 'an enemy's airplanes, tanks, transports or monitoring their movements. Moreover, WSNs can be used to evaluate losses during a battle and to detect nuclear and chemical attacks [5-9], as shown in Fig. 7.

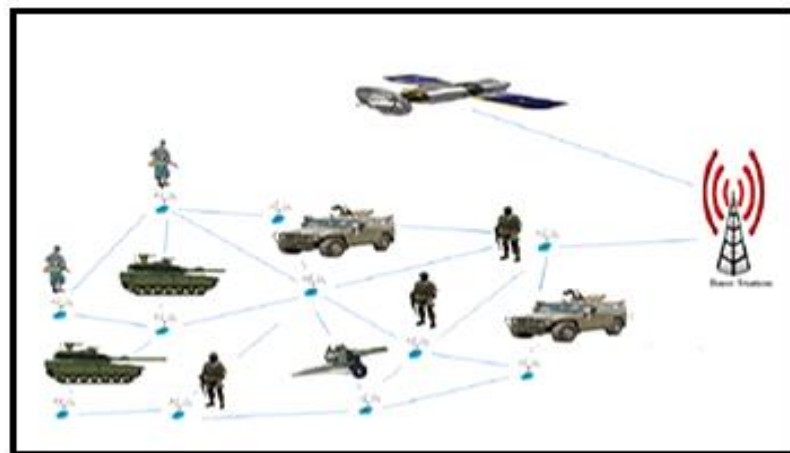


Figure 7 Applying a WSN in military field

1.2.2 Health care

WSNs are an emerging technology which creates new opportunities and is important in human endeavors [22]. Some of these include healthcare applications that offer interfaces for the elderly, offer disabled control and monitor movements of patients

and doctors in a hospital in addition to movement control in the calculation of a heartbeat [5], as shown in Fig. 8.

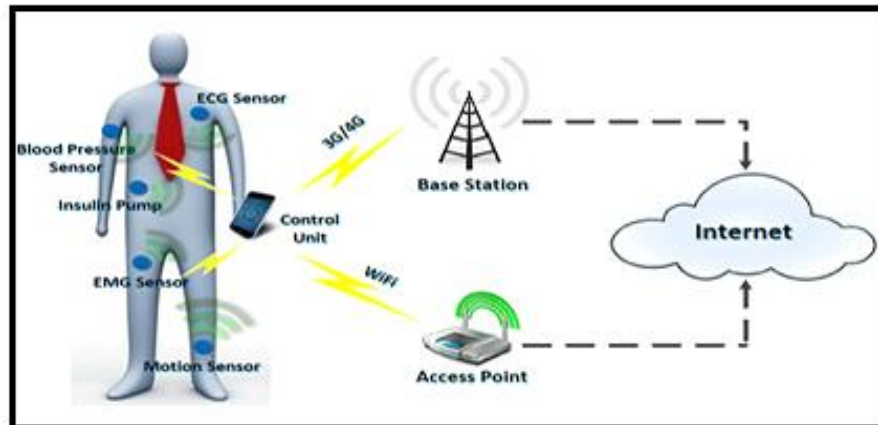


Figure 8 Applying a WSN in health monitoring field

1.2.3 Environmental

The monitoring of the environment is one of the earliest applications of the sensor network. Other examples include monitoring the weather and forecasting the weather, ocean surveillance, monitoring floods, hurricanes, monitoring volcanoes, earthquakes with assessments of harm caused. Moreover, a sensor network can track the movements of small and large animals and birds. In addition, they detect forest fires and study air pollution. Additionally, they are used in the detection of oil and gas fields under water [5-9-21] as shown in Fig. 9.

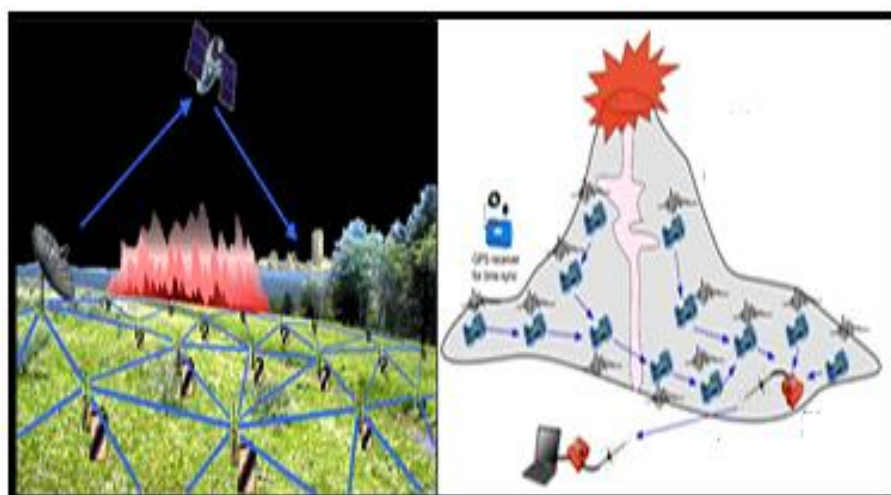


Figure 9 Applying a WSN in an environmental field

1.2.4 General engineering

Recently, there has been focused research to develop sensor technology that can be applied to bridges, smart buildings, intelligent buildings and factories. The purpose is the development of smart structures which are able to self-diagnose problems in themselves and determine objects or requirements to fix them [23]. In addition, sensor networks in smart buildings are used to monitor, control and improve living conditions, for example, by controlling the flow of air and the temperature of a building, which helps to reduce power consumption [4], as shown in Fig. 10.



Figure 10 Applying a WSN in engineering field

1.3 Literature Survey

WSN is a valuable technology to monitor and perceive data from the physical world. A sensor has many security challenges because of its properties such as memory constraints, low processing power and limited energy. Moreover, due to used wireless channels and deployment in open areas, a sensor is susceptible to many security attacks. This has given birth to a variety of authentication schemes [23]. Authentications include the authentication sensor node to cluster the head or base station before exchanges of data. Authentication is applied by using the ECC algorithm and it is suitable for WSN due to its greater number of advantages such as its superior strength per bit compared with other algorithms, such as RSA, thereby making ECC a very attractive option [24]. In this section, we will review the literature related to authentication schemes used in WSN environments.

In [20], the authors compared the ECC and RSA schemes for whichever is best to be used in order to achieve security in WSNs. Identification of suitable cryptography for WSNs is a very important challenge because sensors are resource constrained, including limited power, small memory, low bandwidth and limited energy. It has been determined that ECC is more useful than RSA, which results in lower memory usage, lower CPU consumption and smaller key size than RSA. ECC with 160 bits is twice as better than RSA with 1024 bits when code size and energy consumption are the factors taken into account.

In [25], the authors proposed an energy efficient authentication scheme based on multi-level μ Tesla. The idea of the proposed scheme is a broadcasted message by the BS being authenticated so that the compromised nodes can be refused efficiently. The proposed scheme will be free from reply attacks, DoS attacks and node capture attacks. An author displays a novel symmetric-key-based authentication scheme that appears as a low broadcast authentication overhead. This, therefore, helps to avoid the problem of flaws inherent in PKC based schemes.

In [26], the authors proposed an ECC-based node authentication scheme with a secure key establishment protocol, which brings together the Elliptic Curve Diffie-Hellman and Secure Hash Algorithm-1, in which they use the ECDH key agreement protocol to create sharing of private and public keys over an ensured medium with use of SHA-1 to provide message authentication for the key establishment phase. In addition, SHA-1 offers a guarantee of integrity between the sender and receiver. In addition, nodes are able to authenticate each other so as to achieve collaborative data processing. This is being proposed to understand the threats to WSNs and to attempt to show a unique secure node authentication scheme with a robust and efficient key management scheme.

In [27], the authors propose a novel routing-driven key management scheme. This scheme depends on a Heterogeneous Sensor Network model to prefer performance and security, which only provides shared keys to neighboring sensors that communicate with each other. This, therefore, dramatically reduces the communication and computation overheads of the key setup. These are the uses for ECC in the design of this scheme for sensor nodes and to further improve the key management scheme. The scheme shows only pre-loads of a few number keys on each L-sensor. This dramatically reduces the communication overhead, storage needs

and energy consumption. In addition, it is robust and flexible against any node compromise attack.

In [28], the authors propose the new algorithm, which indicates a modification of SHA-1 with the help of a pseudo random function. The regularly distributed pseudo random function is used instead of logical functions as it is in the original SHA-1. This, therefore, produces unique hash values for unique messages and offers a collision resistance requirement to the hash function. The greatest benefit of this algorithm is its use of unique numbers according to the input message. This means that the value of the function does not depend on constants as it does in the original SHA-1; however, this depends on the message only.

In [29], the authors search the energy consumption efficiency of SKC algorithms, which include both stream ciphers and block ciphers when security is applied to WSNs. To gauge the cost of computational energy for the encryption algorithm, they used the number of CPU cycles when comparing different symmetric key ciphers. After analysis, they concluded that it was the lightweight block cipher, referred to as byte-oriented substitution permutation network. Moreover, this is the most recommended cipher to achieve acceptable security and energy efficiency for WSNs among different symmetric key ciphers.

In [30], the authors proposed an efficient distributed user authentication scheme which is based on the self-certified keys cryptosystem (SCK) and modified to use ECC in order to create pair-wise keys. The proposed scheme imposes very light computational communication overheads and does not provide a variety of security features. However, it is efficient in terms of message exchanges and computational burdens. Thus, it can be easily implemented in real WSNs.

In [31], the authors proposed an improvement of dynamic user authentication, which is based on Wong et al.'s. Moreover, it does not specify weak points but improves its security. Additionally, legal users can freely change their passwords. Furthermore, it relies on an identity and hash password. The registration and password-changing phases are executed through a secure channel. The improved scheme can resist a number of additional attacks, such as resistance to forgery attacks and replay attacks. Moreover, it has many positives such as better efficiency, a reduction of the danger of leakage of a user's password and the capability of changing passwords. Furthermore, it does not add an additional computational cost if compared with

Wong et al.'s scheme; however, it does not provide mutual authentication between users.

In [32], the authors proposed the Low Energy Adaptive Clustering Hierarchy protocol. In networks of a homogenous model, all sensor nodes have equal capacity in terms of computation, power and communication. Moreover, each node is able to collect data and then send it to BS to reduce overhead. After the distribute sensor nodes phase, each node chooses its cluster head based on a number of parameters such as the strongest signal received by a *CH*. Then, a *CH* is selected from the deployed sensor nodes. The LEACH protocol evenly distributes the energy load in the network so as to reduce energy consumption by utilizing a rotation of *CHs*, which randomizes and turns off ordinary sensor nodes when it not required.

In [33], the authors proposed an algorithm to generate a data sequence and apply it over an ECC-encrypted message over the finite field. First, the message (point) is transformed into an affine point of the EC. Then, the message is encrypted through the algorithm in Fig. 25 which is a pair of points. Finally, a pair of points is converted into a data sequence (meaning binary) and then broadcasts it. Upon receiving the message, there is a series of bits. First, they are converted into points through a series of procedures, after which a plain message is retrieved through the algorithm in Fig. 26. Consequently, an attacker cannot interpret the message, which is a series of bits. The proposed scheme provides sufficient strength against crypto analysis compared with RSA.

In [34], the authors proposed a scheme that uses a pre-shared secret key between nodes and the base station, which is obtained from the ECDH key exchange algorithm and is based on a modified SHA-1, which helps to calculate the message authentication code for given messages. The scheme provides both authenticity and integrity of the messages with only one hash value.

1.4 Motivation

By virtue of rapid development in the field of WSNs, WSNs have been used in different fields, such as military and civilian applications, and they have become an important aspect of our daily lives. WSNs consist of a large number of resource-limited devices called sensors. They are capable of probing the environment and

reporting any collected data to BS via wireless channels. Since a wireless channel is easily vulnerable to various attacks, the avoidance of attacks has become a security issue in wireless sensor networks. Node authentication is one of the most important security services which these applications require. It is considered to be merely the method to achieve this. There are many mechanisms used in the authentication, but all the mechanisms have weaknesses. With the rapid growth of cryptographic mechanisms, recent results show that, asymmetric key cryptography (ECC) and symmetric key cryptography (MAC) are suitable for WSNs.

1.5 The Statement of the Problem

WSN uses wireless channels to establish connections of nodes and it is prone to various attacks. An adversary can easily access data and make objections, changes or replay messages. Therefore, the critical problem is the distribution of keys that are used for cryptography between nodes. An important issue with regard to applying PKC is to verify the authenticity of another party's public key so as to insure that the public key is indeed held by the person to which it belongs. Otherwise, an enemy can act as an invisible router and learn or modify any message between nodes. Therefore, cryptography is urgent and represents one way to obtain the security requirements, therefore involving SKC and PKC.

1.6 Organization of the Thesis

This thesis contains five chapters and all the necessary information to understand and read the thesis. Every preceding chapter is a prerequisite for understanding subsequent chapters.

Chapter 1 is an introduction to the general background of the WSN, an overview of applications of WSNs, reviews literature survey, motivation, and a statement of the problem.

Chapter 2 includes the importance of security, security requirements, kinds of attacks, basic cryptography mechanisms and a concentration on the ECC, which is the best choice for cryptography due to its high efficiency.

Chapter 3 includes the enhanced data sequence method for the ECC cryptosystem, which is used to distribute the public key of BS and CH in addition to using it to provide authenticated messages between nodes.

Chapter 4 includes the implementation the proposed scheme. The results and the security and efficiency of the proposed scheme are analyzed with other schemes.

Chapter includes the conclusions.

CHAPTER 2

SECURITY REQUIREMENTS AND THE CRYPTOGRAPHY MECHANISM IN A WSN

2.1 Security in Wireless Sensor Network

The idea of data security leads to the development of encryption. Encryption is the science of keeping data safe. In many WSN applications, nodes are deployed in open areas or hostile areas. Moreover, the medium used is wireless, whose nature of transmission enables anyone to interrupt and compromise the data in these channel communications or to inject counterfeit data. Thus, they are prone to various attacks. Therefore, implementing security is more important in order to ensure secrecy of data versus attacks. Because of the restricted resources of WSN, it cannot allow the use of complex security, unlike traditional networks [9-35-36].

2.2 Security Requirements in WSN

In general, the security services in a WSN should protect data and resources over the network from various attacks as well as protect data from the misbehavior of nodes during communication. The main important security requirements in a WSN include:

2.2.1 Authentication

Applying authentication allows nodes in the WSN to ensure the identities of the peer nodes that are communicating with each other so as to eliminate any bogus messages. The guarantee of receiving messages comes from legal node [11-35].

2.2.2 Integrity

Integrity is a basic need for communication due to the receiver having to guarantee the receiving of messages not being altered or changed via a malicious node or enemy. Integrity accomplishes confidentiality since without integrity, there is still the possibility of altering or changing data and integrity prevents this possibility [11-35].

2.2.3 Confidentiality

Confidentiality is an issue in network security to protect data due to its transfer on the channel between the nodes, and between BS and nodes by hiding messages. Then, it prevents an adversary, which is an unauthorized node, from understanding the content of messages [9-11-35].

2.2.4 Scalability

Scalability is defined as adding new nodes to network and increasing the size of the network without influencing the security level and node characteristics [7].

2.2.5 Availability

The guarantee of the ability of a sensor network is to provide services to authorized parties even though there are internal or external attacks against network. Moreover, it ensures that the developing security mechanism and limits on the network performance are not affected [9-37].

2.3 Kinds Attacks in Wireless Sensor Network

A WSN is deployed in an open area. It is vulnerable to security attacks such as the tapping and extraction of the content of a message. Moreover, there is the possibility of presenting fake messages, injecting new messages or modifying messages during communication due to the nature of the transmission medium [35].

2.3.1 Sybil attack

An attacker is a fake node which can impersonate the identity of more than one node within the network. This affects the confidentiality, data authenticity and data integrity. Moreover, the forgery of an identity can compromise the node and routing mechanism used in the network. This appears to be a part of a legitimate path routing. In addition, it can send bogus information into the network. Furthermore, it can eavesdrop on any passing packets and it can attempt to extract any secret key and identity information [35-36-38], as shown in Fig. 11.

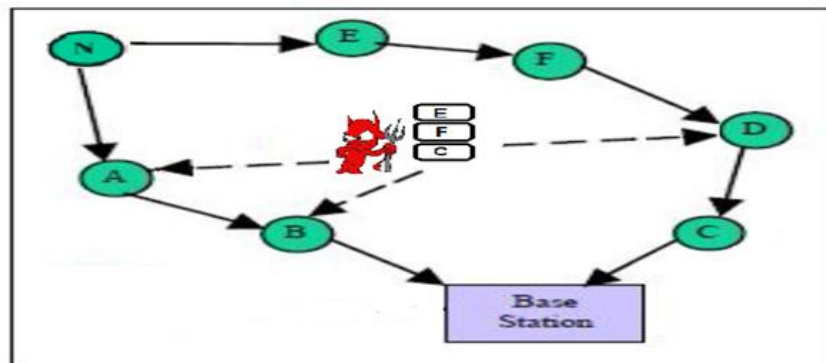


Figure 11 Sybil attack

2.3.2 Hello flood attack

An attacker pretends to be a BS, and it sends a hello message with a high transmission range to all network's members. It acts as a counterfeit BS to broadcast their messages to it rather than an legal BS. Moreover, an intruder disguises the BS and it acts as a neighbor to many nodes in the network, thereby routing the network badly [35-36], as shown in Fig. 12.

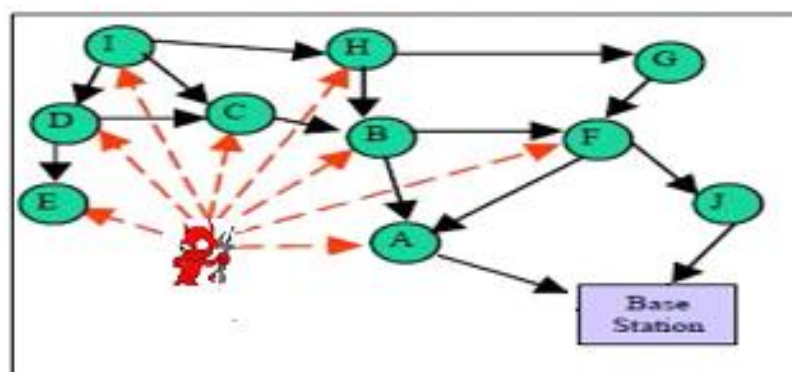


Figure 12 Hello flood attack

2.3.3 Black hole attack

An intruder node is placed in the center and endeavors to attract traffic over an illegal node in order to control it and establish a sinkhole, usually near the base station where it attracts all of the traffic. In addition, when it receives the messages, it may refuse to pass on some of the messages or modify them [35-36], as shown in Fig. 13.

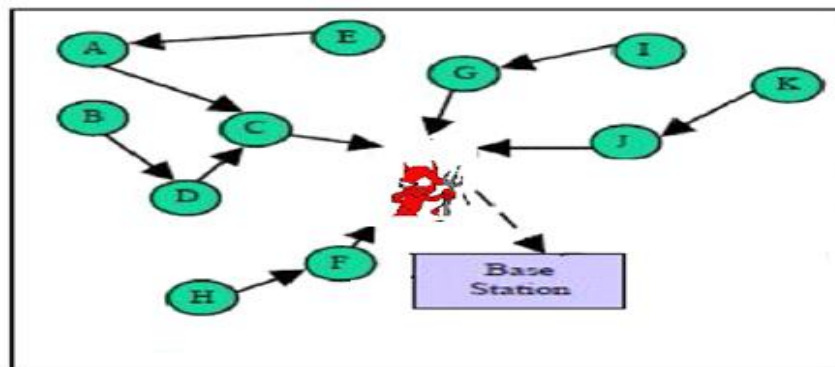


Figure 13 Block hole attack

2.3.4 Clone attack

An attacker captures a node physically and copies its information to another node, which is called as cloned node. Moreover, the intruder can install itself and acquire information from the system network. Additionally, it can enter bogus information into the system network [39].

2.3.5 Tunneling attack

An attacker is a node that uses a tunnel to put its character between two legal nodes, and then mix the routing protocol with a better communication resource than normal nodes, thereby creating better communication channels between them. Furthermore, the network topology changes because routing information goes wrong. Moreover, an attacker can change or modify the packets within the network [35-36], as shown in Fig. 14.

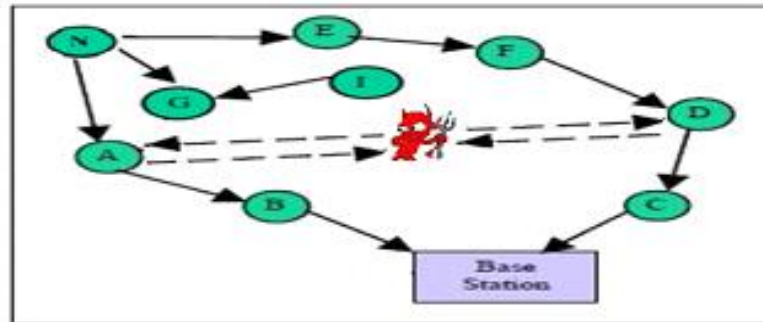


Figure 14 Tunneling attack

2.3.6 Modification and insertion attack

These kinds of attacks from an adversary can alter or change the integrity of the exchanged messages between nodes or they can generate counterfeit messages in the network and hence especially affect the security network [8-35], as shown in Fig. 15.

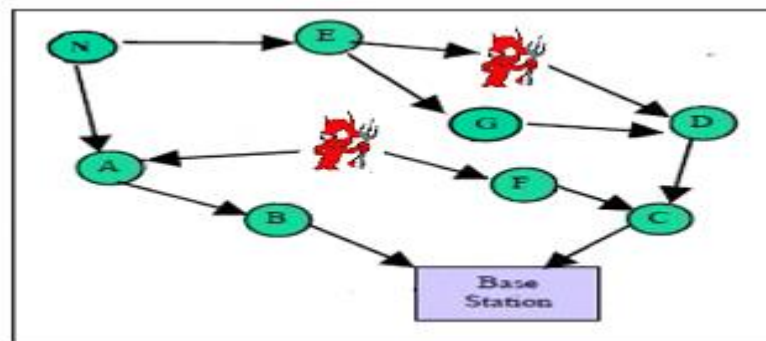


Figure 15 Modification and insertion attack

2.4 Basic Cryptography Mechanism in WSN

Nobody is able to deny the importance of security in data communications and networking. Furthermore, cryptography is the only way to make the network more secure, immune against attacks and threats. Therefore, network security is achieved by means of cryptography techniques. Furthermore, the design and analysis of mathematical techniques enables secure communications by offering random sampling mechanisms and interactive proofs so as to allow the receiver's node to verify the delivered data by the user nodes in the presence of malicious adversaries, called cryptography. Fig. 16 shows the role of security [12-40-41]. The cryptography mechanism can be classified into two types:

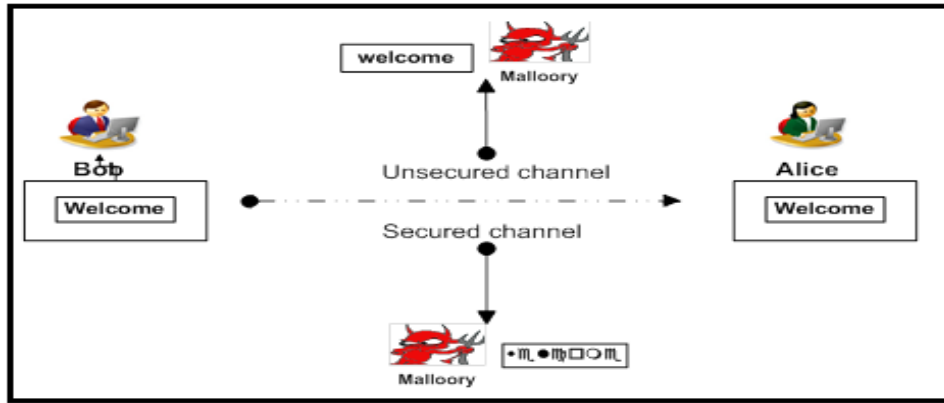


Figure 16 The role of security

2.4.1 Symmetric key cryptography

Symmetric key cryptography, also known as private key cryptography, is very old cryptography and dates back to 500 BC. Additionally, in a symmetric key, both parties being associated with the encryption/decryption key pair means that symmetric encryption uses the same single secret key for encrypting and decrypting data, as shown in Fig. 5. Moreover, this system has the advantage of encryption which is perceived to have an extensive history, a large amount of data, high rates of data throughput, a small key size, ease of analysis and lower computational efforts and requirements compared to asymmetric key cryptography on the resource constrained sensor nodes. However, a problem arises such that the key must remain secret for both parties only and there are many key pairs to be managed. Finally, there are many examples of SKC such as Data Encryption Standard (DES) [23-40].

2.4.2 Asymmetric key cryptography

In 1975 Diffie, Hellman and Merkle introduced a solution to the aforementioned shortcomings of SKC which utilizes two keys: the private key and public keys. As shown in Fig. 6 each node has its private and public keys and not only does every node know its own private key, the public key of every node is known to each other. When Alice sends a message to Bob, she ciphers the plaintext by its private and Bob's public key. Moreover, when Bob receives the message, he decrypts the message by its private key and Alice's public key to extract the plaintext. Hence, an

enemy cannot acquire the plaintext without the private key because it is secret. Finally, there are many examples of AKC, such as RSA and ECC [23-40].

2.5 Message Authentication Code

MAC is a type of SKC that is used to achieve the necessary security, such as message authentication, node authentication and data integrity between communication parties, because it works against forgery of the message and prevents the probabilities of these codes being changed via an attacker in order to influence an action to the attacker's advantage, as shown in Fig. 3. The message sent is divided into two pieces, the first of which is the clear message and the other an encryption by a shared MAC key, which acts as the MAC. Furthermore, the receiver adds the same share MAC key to clear the message, which is the first part of message. This is followed by a new MAC which then matches it with the second piece of the received message to ensure the message is transmitted from a legal and not intruder node [9-12-42].

2.6 Elliptic Curve Cryptography (ECC)

2.6.1 Overview of elliptic curve cryptography

During the last three the decades, the elliptic curve has become a very important topic of research in a number of theoretical and related fields, such as cryptography. ECC was invented independently by Victor Miller and Neil Koblitz in 1985. By using elliptic curves to design public key cryptographic systems [12], their research proved that they can be used for security requirements such as authentication, confidentiality, data integrity and more [41]. Furthermore, it has become more agreeable an alternative to RSA [41]. The research in the field of ECC is mostly concentrated on its focus on application-specific systems due to systems having restricted resources, such as storage and processing speed [43].

In addition, ECC offers more jobs, such as RSA with the security level being achieved with far smaller keys in EC systems than is possible with RSA peers [12]. Moreover, ECC has attracted much attention as a security solution for wireless

networks due to its small key size and very small computational overhead [3-41]. For instance, a 160-bit ECC key size provides the same level of security as a 1024-bit RSA key size, as shown in Tab. 1. Hence, the ECC private key is faster than the RSA private key. Moreover, it relies on the assumed difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP) for its security [44]. Moreover, ECC offers a wide range of applications, such as key exchange, privacy via encryption, sender authentication and integrity of the message via a digital signature algorithm (DSA) [45].

Table 1 ECC key Sizes Compared with RSA Key Sizes

ECC key size Depend on No. of (bits)	RSA key size Depend on No. of (bits)
160	1024
224	2048
256	3072

2.6.2 Fundamental mathematical terminology

We must know the core of basic mathematical terminology of ECC in order to understand that it is very good. Some basic mathematical terminologies are offered in the following sections:

1. Groups

A group is said to be an abelian group where it comprises a binary operation, symbolized by \bullet , which is associated with each ordered pair (w, x) of elements in G and an element $(w \bullet x)$ in G , [41-44], as shown in Tab. 2.

Table 2 Properties of Abelian Group

Property	Describe
Closure	If w and $x \in G$, then $w \cdot x$ is also belong to G
Associative	When $w \cdot (x \cdot z) = (w \cdot x) \cdot z$ for all w, x and z in G
Identity element	$w \cdot e = e \cdot w = w$ for all elements $w \in G$.
Inverse element	There is an element w^{-1} in G such that $w \cdot w^{-1} = w^{-1} \cdot w = e$.
Commutative	$w \cdot x = x \cdot w$ for all w and $x \in G$

2. Cyclic Group

A group G is said to be cyclic if there exists an element s in G such that every element y of G is an integral power of s ; that is, y is of the form s^k for some integer number k . Then, the element s is said to be the generator of group G . For any element y in G , we express the integral power of y where, $y^0 = e, y^1 = y, y^2 = y * y, y^3 = y * y * y$. A cyclic group G is generated by s being represented via $\langle s \rangle$ [44].

3. Finite fields

A field F is referred by F and comprises a set of elements with two operations, namely addition, which is symbolized by $(F,+)$ and multiplication, which is symbolized by $(F,*)$, such that for all w, x, z in F , the following hold [12-44].

1. $+: F + F \rightarrow F$.
2. $*: F * F \rightarrow F$.
3. $(F,+)$ is an abelian group with (additive) identity symbolized by 0.
4. $(F, \setminus \{0\}, *)$ is an abelian group with (multiplicative) identity symbolized by 1.
5. $(w+x)*z = w*z + x*z$ for all $w, x, z \in F$.

Furthermore, the fields consist of two categories and are commonly used in cryptography applications. These include

1. Prime field F_p , where P is prime number.
2. Binary field $F_p = 2^m$, where, m is large number.

2.6.3 Elliptic curve discrete logarithm problem (Ecdlp)

The security of many cryptographic mechanisms relies on the intractability of the discrete logarithm problem. Moreover, in this context, we assume P and Q are points on an elliptic curve, k is an integer number and, $Q = k.P$ for some k , thus, the k is called the elliptic curve discrete logarithm problem of Q to P and due to this difficulty of discovery, the k is what makes the elliptic curves an area worth exploring for cryptography. Therefore, the only way to recover k from $k.P$ is to try every possible repeated summation, such as $(P + P), (P + P + P), (P + P + P + \dots P)$ until the result equals to $k.P$ [41].

2.6.4 Elliptic curve cryptography operations

ECC includes two main operations, which are addition and doubling, as follows:

1. Point addition in ECC

Point addition is the addition of two points J and K if $J \neq K$ and $J \neq -K$ to obtain another elliptic curve point R on an elliptic curve. We suppose that J and K are two points on an elliptic curve, as shown in Fig. 17. If a line is drawn through J and K , this line will intersect EC point $(-R)$. Then, we produce a reflection of this point about the x-axis, which is R , and the result is $R = J + K$. Point addition is calculated by following Eq. (2.1) and Eq. (2.2).

$$P1 = (x_J, y_J), P2 = (x_K, y_K). \quad \text{,if } J \neq K \text{ and } J \neq -K \quad (2.1)$$

$$R(x_R, y_R) = P1 + P2.$$

$$S = \left[\frac{(y_J - y_K)}{(x_J - x_K)} \right] \bmod p \quad (2.2)$$

$$x_R = (x^2 - x_J - x_K) \bmod p$$

$$y_R = -y_J + S(x_J - x_R) \bmod p$$

Additionally, in the case of J being the negative of J meaning $J = -J$, it means there is no intersection with the elliptic curve, then the result is O (mean infinity), which is the identity of addition, as in Eq. (2.3) [43-44-45-46], as shown in Fig. 18.

$$J + (-J) = O \quad , \text{if } J = -J \quad (2.3)$$

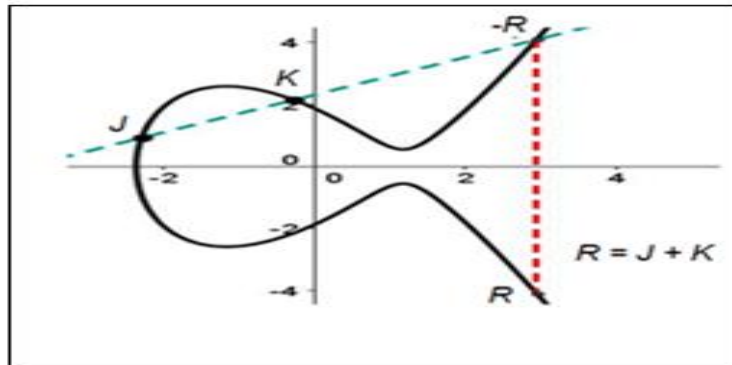


Figure 17 Point addition if $J \neq K$

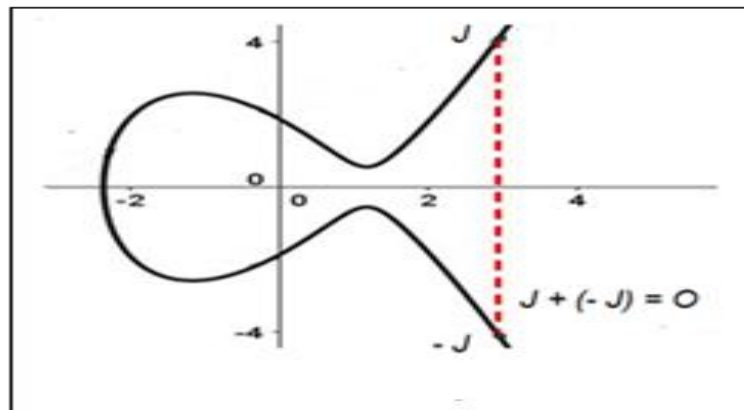


Figure 18 Point addition if $J = -J$

2. Point doubling in ECC

Point doubling is the adding of elliptic curve points J with itself if $J \neq -J$ and $y_J \neq 0$ so as to obtain another elliptic curve point R on an elliptic curve. Suppose we have a J point on an elliptic curve, as shown in Fig. 19. If we draw a tangent line at point J , then this line intersects EC at point $(-R)$. Then, point R is the reflection of this point

about the x-axis, which is the result $R = J + J$. Point doubling is computed by the following Eq. (2.4) and Eq. (2.5).

$$P1 = (x_J) \quad , \text{if } J \neq -J \text{ and } y_J \neq 0 \quad (2.4)$$

$$R(x_R, y_R) = 2.P$$

$$S = \left[\frac{(3x_J^2 + a)}{(2y_J)} \right] \bmod p \quad (2.5)$$

$$x_R = (x_J^2 - 2x_J) \bmod p$$

$$y_R = -y_J + S(x_J - x_R) \bmod p$$

Additionally, in case of $y_J = 0$ it means there is no intersected with elliptic curve then the result is O (mean infinity) as in Eq. (2.6) [43-44-45-46], as shown in Fig. 20.

$$2.J = O \quad , \text{if } J = 0 \quad (2.6)$$

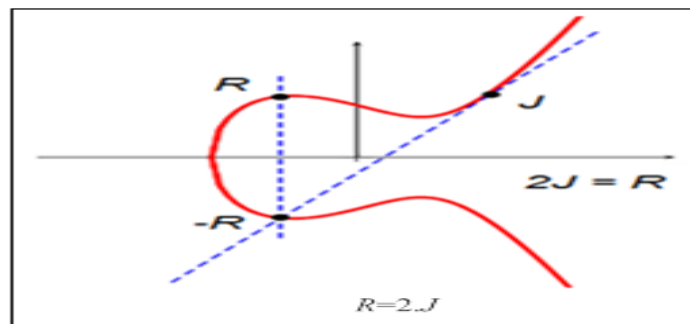


Figure 19 Point double

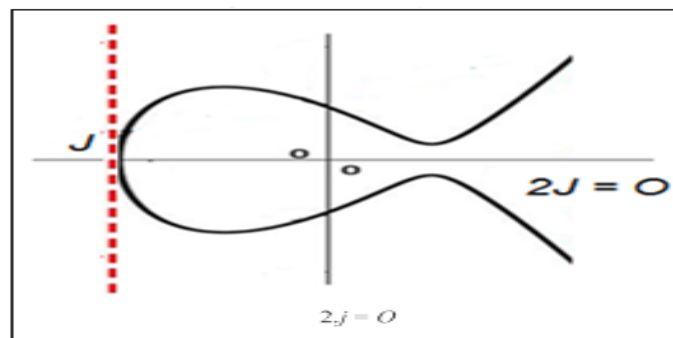


Figure 20 Point double if $y_J = 0$

2.6.5 Scalar multiplication in elliptic curve cryptography

The scalar multiplication operation is carried out using repeated addition. We consider a point P on the elliptic curve which is multiplied by a scalar k through the elliptic curve equation to obtain another point P . Scalar multiplication can be achieved using point addition and point doubling operations. For instance, in order to find $Q = k.P$, if $k=23$, then $k.P \rightarrow 23P = 2(2(2(2P) + P) + P) + P$. This method is known as the ‘double and add’ method as it involves both point addition and point doubling operations in order to find the result of the scalar multiplication [43-46].

2.6.6 Elliptic curve cryptography encryption and decryption

There are several ways to apply elliptic curves to the encryption/decryption mechanism. We assume working with a group F_p finite field and an elliptic curve EC. Initially, the users randomly choose a base point (P), lying on EC and an elliptic group $F_p(a,b)$ as parameters. In order to encrypt/decrypt a message in this system, the first task is to encode the plain message to be sent as point P_m . Each user A selects a private key d . Next, a public key is computed by Eq. (2.7).

$$Q = d.P, \text{ by ECC.} \quad (2.7)$$

To encrypt message and send P_m to user B , user A selects a random positive integer k and computes the cipher message C_m = consisting of the pair of points as $C_m = \{k.P, P_m + k.Q_B\}$, where, A has used B 's public key Q_B . The sender A transfers the points as $\{C_1, C_2\}$ where $C_1 = k.P$, and $C_2 = \{P_m + k.Q_B\}$ to the receiver. Finally, to decrypt the cipher message C_m , B multiplies the first point in the pair via B 's secret key and subtracts the result from the second point (C_2) as Eq. (2.8).

$$\{P_m + (k.Q_B) - d(k.P) = P_m + k(d.P) - d(k.P)\} = P_m = \text{plainmessage} \quad (2.8)$$

This can be illustrated as the algorithm in Fig. 21 and Fig. 22. [44].

-
1. Input: Elliptic curve domain parameters (E, P, n) . E consists of (a, b) , P is base point, n is the order of the elliptic curve and Q is public key. Plain message P_m .
 2. Output: Cipher message C_m .
 3. Represent the plain message as a point (x, y) P_m in $E(F_p)$.
 4. Choose $K_1 \in F[1, n-1]$.
 5. Calculate $C_1 = K_1 \cdot P$.
 6. Calculate $C_2 = P_m + (d \cdot Q)$.
 7. Return (C_1, C_2) , that is pair of points.
-

Figure 21 Encryption using elliptic curve cryptography algorithm

-
1. Input: Elliptic curve domain parameters (E, P, n) . E consists of (a, b) , P is base point, n is the order of the elliptic curve, and d is private key. Cipher message is C_m .
 2. Output: Plain message P_m .
 3. Calculate $P_m = C_2 - (d \cdot C_1)$.
 4. Return (P_m) that is the plain message.
-

Figure 22 Decryption using elliptic curve cryptography algorithm

2.7 Authentication in Wireless Sensor Network

Node deployment in a WSN is achieved in an unreachable area where the wireless channel is exploited for node communication. Hence, it is prone to many security threats. These threats can be avoided significantly by using authentication techniques. It is a mechanism in which, the identity of a node can be identified as a valid node within a network followed by the data authenticity being achieved. The reliability of the message is guaranteed via authentication that identifies its origin. In the WSN, the attackers not only alter the data, they are also able to inject false data. In addition, the integrity and confidentiality of a message are verified by data

authentication via a method known as MAC (Message authentication code). Initially, this is appended to the data after which is sent. Only valid sensor nodes are able to decrypt the MAC via some determinable methods (such as symmetric and asymmetric key cryptography) [35-47].

CHAPTER 3

THE ENHANCED DATA SEQUENCE METHOD FOR ECC CRYPTOSYSTEM (EDS-M-ECC)

3.1 Introduction

Lately, several studies show that ECC has many advantages, especially in wireless communication, because of its small keys sizes when comparing it with other PKC algorithms, such as RSA. Moreover, the security in RSA is based on the difficulty of the integer factorization problem, while ECC is based on the stiffness of different problems, namely the Elliptic Curve Discrete Logarithm Problem [12]. The proposed method developed and promoted the previous method, which is used to secure the output of ECC by converting the size of the ECC point to 6 bits. In this chapter, we improve the method in [33] by relying on a base 6 sequence when converting each digit to the sequence instead of base 2, hence driving to reduce the number of bits. Thus, the application of this study, which depends on the ECC under the finite field with the idea of a data sequence, offers better performance in terms of energy consumption, memory size and the real time being implemented by the comparison with another method.

In [33], the authors exploit the method of data sequences to save more power. The proposed method relies on [33] and has developed its message size to become 6 bits instead of 8 bits. As a result, the sent message's (M) size is reduced. We apply this method in the node authentication scheme in WSNs depending on the data sequence over ECC based on MAC. This is to distribute the public key of BS securely for each CH and distribute the public key of each CH securely for each cluster member. Moreover, we use this method to provide authenticated messages between nodes.

3.2 Elliptic Curve Definition

An elliptic curve (EC) is a plane curve defined by an Eq. (3.1).

$$y^2 \bmod p = x^2 + ax + b \bmod p \quad (3.1)$$

where, p is a prime number and x, y, a, b are elements of the finite field $GF(P)$. The chosen a, b should satisfy Eq. (3.2).

$$4a^3 + 27b \bmod p \neq 0 \quad (3.2)$$

Fig. 23 elucidates the mathematical hierarchy of the elliptic curve scalar multiplication that includes three levels: scalar arithmetic at the top, point arithmetic at the medium and field arithmetic at the lowest. EC scalar multiplication algorithms placed at the top level, which is the basic operation in of the system are used in cryptography systems and so used in both ECDH and ECDSA, to exchange the keys, use ECDH and for authentication of public key use ECDSA. In addition, the medium level involves operations of adding and doubling the points in the elliptic curve, which is not normal adding and doubling. This is explained in Chapter 2. Finally, the lowest level includes square, add, multiple, and inverse operations over the finite field $GF(P)$ to release the medium level [48].

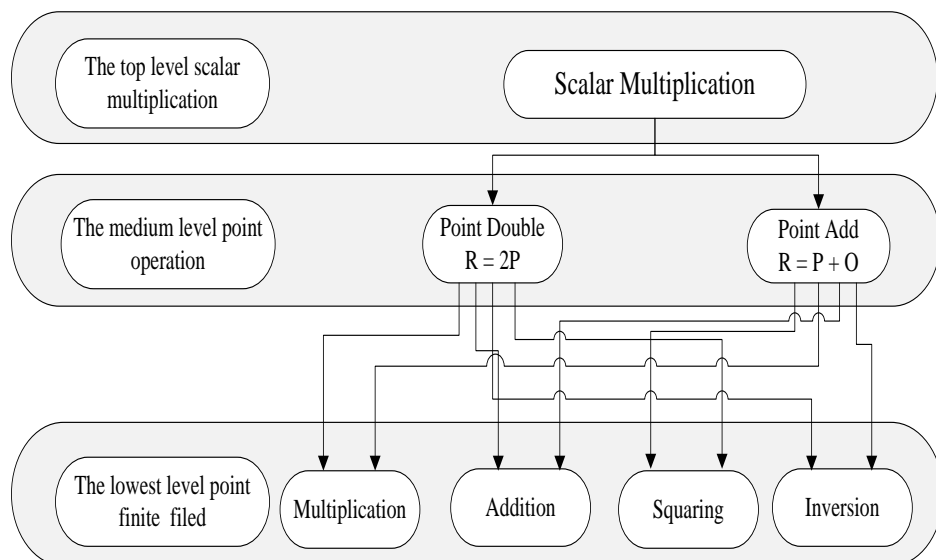


Figure 23 Elliptic curve mathematical hierarchy

3.3 Background on Data Sequence

Recently, data sequences have become more beneficial and are being used widely in cryptosystems. Furthermore, they have become essential in computer science. There are many examples of data sequences, such as stacks, queues, text files and strings which are represented as sequences or series of characters or integer numbers or real numbers or binary numbers, etc. Moreover, it is a group of objects ordered with the same properties under any type and are countable. The sequence consists of a special name for the two ends. Sequences have many styles,; for example, one object as [] or one object as [a] or more object [a₁.....a_n]. As well as, it writes in a row like (x₁, x₂, x₃,) or formula (x₀, x₁, x₂,) or (x₀, x₂, x₄,). The borders of the sequence are represented by objects, the first of which in the sequence is x₁, and the second is x₂, the third is x₃ and so on, such as the prime sequence, even sequence, natural number sequence and odd numbers. The final example of a sequence begins with two numbers and both are 1, which represent the first and second elements of the sequence. Moreover, the third object resulting from the addition of the first two elements, 1+1=2, and the fourth object results from the addition of the second and third elements, 1+2=3, and so on. Finally, the sequence of 7 objects ordered as : 1, 1, 2, 3, 5, 8, 13, 21. [49-50].

3.4 The Proposed Method Explaining

The idea of the proposed method is to merge the data sequence over the output of ECC to produce a stream of vectors called *S_i* to give a stream of bits of constant length such as a one-way function such as the Hash function (SHA-1). Thus, it confuses the attacker. The proposed method includes a number of processes to be implemented as follows:

3.4.1 Generating the data sequence process

To gain Tab. 3, which is a data sequence, it must take the following steps:

1. Suppose that *P* is a point generator or base point and *n* is the order of *P*.

Table 3 Data Sequence Generate

#	The data sequence of $n=30$ & $m=2$		Corresponding
	Digit ₁	Digit ₀	
1	0	0	0
2	0	1	1
3	0	2	2
4	0	3	3
5	0	4	4
6	0	5	5
7	1	0	6
8	1	1	7
9	1	2	8
10	1	3	9
11	1	4	10
12	1	5	11
13	2	0	12
14	2	1	13
15	2	2	14
16	2	3	15
17	2	4	16
18	2	5	17
19	3	0	18
20	3	1	19
21	3	2	20
22	3	3	21
23	3	4	22
24	3	5	23
25	4	0	24
26	4	1	25
27	4	2	26
28	4	3	27
29	4	4	28
30	4	5	29

2. Suppose the sequence for $i= 0$ to n which is values.
3. Convert every digit of the sequence in base 6 (this differs from [33] which is in base 3).
4. Suppose m to be the number of digits such as: $n=30$ thus, we find m is 2 (this differs from [33] in which m is 4. as shown in Tab. 3.

In this example, the confirmation of how the sequence 40 is equal to 24 can be explained as follows: $40 = (0 * 6^0) + (4 * 6^1) = 0 + 24 \rightarrow n = 24$.

5. The mapping of Tab. 3 into the following matrix, (this matrix differs from [33] in the number of columns which is 4).

$$M = \begin{bmatrix} a_{0,0} & a_{0,m-1} \\ a_{1,0} & a_{1,m-1} \\ \vdots & \vdots \\ a_{n,0} & a_{n,m-1} \end{bmatrix}$$

This matrix is clarified much in Eq. (3.3).

$$M = (n + 1) \times m \quad (3.3)$$

6. Do a circular right shift for each row of M by 1 digit. For instance.

$$\begin{bmatrix} a_{i,0} & a_{n,m-1} \end{bmatrix} \rightarrow M = \begin{bmatrix} a_{n,m-1} & a_{i,0} \end{bmatrix}$$

where, this sequence differs from [33] in the number of columns which is 4.

Finally, the form of the sequences will be as follows:

$$S: [S_0 = [a_{0,m-1} \quad a_{0,0}], S_1 = [a_{1,m-1} \quad a_{1,0}], \dots, S_n = [a_{n,m-1} \quad a_{n,0}]]$$

3.4.2 Description of applying the created sequence on the ECC process

In this step, the procedure is elucidated in step1 and is applied on the Elliptic curve cryptography (ECC) as in [33]. However, it is applied according to the differences that the proposed method achieves as followsing:

1. Encrypting the message transferal

We suppose that we have some elliptic curve EC over a finite field $GF(p)$ and that E and a point $P \in E$ are known publicly, and the merge system $m \rightarrow Pm$; which merges the plain text on an elliptic curve. After that, when Alice decides to communicate secretly with Bob [33], they must carry out the following steps:

- a. Bob must select a random integer a , and publicize the point $a.P$ that will be a public key (where a remains secret and unknown to everyone but Bob) as in [33].
- b. Afterwords as in [33], Alice chooses her own random integer l , where l is private key and calculates two points: where P_1 is her public key as shown in Eq. (3.4), and P_2 is the cipher text in the form of P_i , where P_i is a character which will represent a point in E as shown in Eq. (3.5).

$$P_1(x_1, y_1) = l.P \quad (3.4)$$

$$P_2(x_2, y_2) = P_i + l(a.P) \quad (3.5)$$

- c. Calculate $S(x_1, y_1)$ and $S(x_2, y_2)$ in where S is a corresponding sequence value in gained from process 3.4.1 step 4 and 5 after applying the circular right shift by 1digit. Thus, the cipher message Cm takes the form of Eq. (3.6).

$$Cm = (S(x_1, y_1), S(x_2, y_2)) \quad (3.6)$$

Cm will be 4 digits such as 00 01, but in [33] it is 8 digits like as 0000 0001.

d. Alice converts the cipher text Cm to octal binary representing a form (of three bits for each digit) such as :
 $00 \rightarrow 000000, 01 \rightarrow 000001, 43 \rightarrow 100101, 35 \rightarrow 010101$ while in [33] 0000 is converted to 00000000 , 0001 is convert to 00000001 , etc. Linally, Alice sends the series of bits to Bob.

2. Decrypting the received message

To retrieve the plain text from the cipher text Cm , where, Bob knows the sequence of S_i , his own private key a and ECC parameters. He receives the encrypted message which is a series of bits such as in [33] and does the following:

a. Bob converts the binary bits into digits, for instance:

$000000 \rightarrow 00, 000001 \rightarrow 01, 100101 \rightarrow 45$. But in [33] Bob receives 16 bits and converts them to 8 digits as following: 0000000000000001 becomes 00000001 .

b. Bob converts the cipher message Cm into a group of $3m$ (digits), while in [33] is converted into $2m$.

c. Bob gets a group of m digits from the sequence in step (b).

d. Bob does a circular left shift to the sequence of m digits by 1 for two digits, while in [33]. He does a circular left shift for 4 digits.

e. He converts the sequence of digits to decimal form, and saves the value in k , For instance:

$(011\ 001)$ is separated as: $(011 = 3)$ and $(001=1)$. Then it can be represent as (31) and be in the form of the following: $(1 * 6^0) + (3 * 6^1) = 1 + 18 = 19 \rightarrow k = 19$.

f. Then find a point (x_1, y_1) by $(k + 1)$.

g. Repeat the same steps described in the decryption part for the next element of the sequence of step (c) for the retrieval of $S(x_2, y_2)$.

h. In order to get P_i from $P_i + l(a.P)$, Bob uses his private key a and calculate $a(l.P)$ from the first part of the pair, then subtracts it from the second part such as in Eq. (3.7).

$$P_i + l(a.P) - a(l.P) = P_i + laP - laP = P_i \quad (3.7)$$

and then separate the merging to retrieve the plain text as in [33].

3.5 Application of the Proposed Method

The proposed method depends on the following Eq. (3.8) to generate EC points and as shown in Fig. 24 of the Elliptic Curve Cryptography.

$$Y^2 = X^2 - X + 16 \pmod{29} \quad (3.8)$$

EC points: (0,4),(0,25),(1,4),(1,25),(2,14),(2,15),(5,7),(5,22),(6,9),(6,20),(7,2),(7,27), (10,7),(10,22),(13,5),(13,24),(14,7),(14,22),(16,6),(16,23),(18,1),(18,28),(21,11),(21, 18),(22,12),(22,17),(23,3),(23,26),(28,4),(28,25).

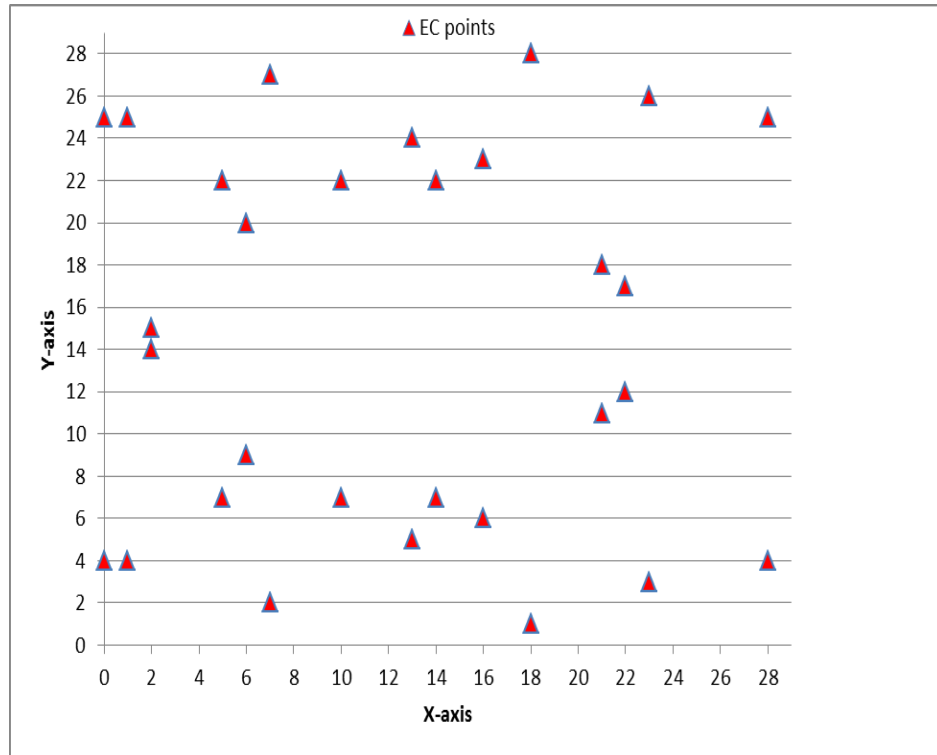


Figure 24 The points of the EC over the prime field

In addition, the chosen base point P is select as (5, 7). The set of all points in the curve is shown in Tab. 4.

Table 4 Points on the Elliptic Curve $E_{29}(-1, 16)$

#	Elliptic Curve point	Corresponding alphabet	#	Elliptic Curve point	Corresponding alphabet
1P	(5,7)	a	16P	(0,25)	p
2P	(28,4)	b	17P	(1,25)	q
3P	(18,1)	c	18P	(7,2)	r
4P	(22,12)	d	19P	(16,6)	s
5P	(6,20)	e	20P	(14,7)	t
6P	(13,5)	f	21P	(10,22)	u
7P	(2,14)	g	22P	(23,26)	v
8P	(21,11)	h	23P	(21,18)	w
9P	(23,3)	i	24P	(2,15)	x
10P	(10,7)	j	25P	(13,24)	y
11P	(14,22)	k	26P	(6,9)	z
12P	(16,23)	l	27P	(22,17)	,
13P	(7,27)	m	28P	(18,28)	.
14P	(1,4)	n	29P	(28,25)	/
15P	(0,4)	o	30P	(5,22)	space

The number of created points for Eq. (3.8) is 30 points where 26 of them are assigned to the English alphabetic, and the residual points are given to some other special characters as (‘,’ , ‘.’ , ‘/’ , and ‘space’). Recall that the number of points can be increased to more characters in the case of changing the elliptic curve equation as is the case in [33]. For instance, we assume that Alice wants to encrypt and send a message like "hello" to Bob. Hence, Alice must implement the following steps:

1. Creating the data sequence

- a. P is a point generator with order $n = 30$ and $m = 2$.
- b. Convert a sequence 0 to n to the form such as Tab. 5, which can be represented as a matrix called $M_s [30 \times 2]$.

Table 5 Data sequence Form

<i>Ms</i> =	00
	01
	02
	03
	04
	05
	10
	11
	12
	13
	14
	15
	20
	21
	22
	23
	24
	25
	30
	31
	32
	33
	34
	35
	40
	41
42	
43	
44	
45	

c. Makes circularly right shifting each row of *Ms* by one digit.

d. The sequence formed will be similar after rotate R-shifting.

[00], [10], [20], [30], [40], [50], [01], [11], [21], [31], [41],[51], [02], [12], [22],
[32], [42], [52], [03], [13], [23], [33],[43], [53], [04], [14], [24], [34], [44], [54].

2. Encryption message mechanism

The encryption message mechanism can be described and explained through the following example:

Firstly, assume that the private key *l* of Alice is 13, and the private key *a* of Bob is 24. The plain text is character 'h'. Thus, to encrypt 'h', Alice must achieve the following stages:

a. Calculate the public key of Bob by Eq. (3.9).

$$Q_B = a.P \quad (3.9)$$

$Q_B = a.P = 24(5,7) = (2,15)$, P is base point.

b. Transform the plain text 'h' to the equivalent ECC point as in Tab. 4. Plain text ('h') = (21,11).

c. Compute lP_B as $l.Q_B = 13(2,15) = (28,4)$.

d. Encrypting of the character 'h'.

Cipher message $C_m = Plainmessage(P_i) + l.(Q_B) = (21,11) + (28,4) = (10,7)$.

e. Then, compute the public key of Alice (Q_A).

$l.P = 13(5,7) = (7,27)$, where, P is base point.

Thus, the encrypted version of the message is: $C_m = (S(7,27), S(10,7))$.

where, $x_1 = 7, y_1 = 27, x_2 = 10, \text{and } y_2 = 7$.

g. Alice applies process 1 to create the sequence S : $S(7,27) = 20, S(10,7) = 13$.

The message converted to "2013".

h. Make R-shift for 2013 by one digit and to become 0231.

Lastly, convert the transmitted message "0231" into a series of bits: (000 010 011 001).

3. Decrypting message mechanism

When receiving the encrypted message Cm by Bob. Bob must take the following steps to recover the plain text character 'h'.

a. To decrypt the (000010) convert it into digits as (000 = 0) and (010=2).

b. Bob recives two digits, which is (20).

c. Do circular left shifting to the sequence by one digit, which yields (20).

d. Then, Bob converts (20) to decimal form and save the value in k in Eq. (3.10).

$$k = (digit_1 \times 6^0) + (digit_1 \times 6^1) \quad (3.10)$$

thus, $k = (0 \times 6^0) + (2 \times 6^1) = 0 + 12 = 12$.

- e. Find the point from the pre-calculated and stored point $k = (k + 1) = (x_1, y_1), P = 12 + 1 = 13$.

Thus, $(x_1, y_1) = 13(5, 7) = (7, 27)$ which represents $k.P$ and it is the public key for Alice. In the same manner, to recover other points the data sequence in process 1 is attained. Then, the encrypted version $((7, 27), (10, 7))$ is recovered to extract P_i .

- f. Bob must calculate his private key and Alice's public key as follows:

$$P_i = \text{Cipher message } C_m - a.(lQ_A) = (10, 7) - 24(7, 27) = (10, 7) - (28, 4) = (21, 11) = 'h'$$

.This can be illustrated as the algorithm in Fig. 25 and Fig. 26.

-
1. Input: EC parameters, d is integer.
 2. Output: Cipher message M.
 3. Select $d \in F_p [1, n-1]$, d is the private key of sender.
 4. Define A as encode plain message as a point(x, y) on the EC according to table3.
 5. Compute $F = d . Q_{receiver}$, where d is private key.
 6. Compute $B = A + F \text{ mod } p$. Adding operation done by them location in table3 and B is integer.
 7. Define Y as conversion of B according to table1 to data sequence.
 8. Make R-shift by one element for Y.
 9. Define W as conversion of Y to octal binary, which is 6-bit.
 10. Compute $Q_{sender} = d . P$, by ECC
 11. Define F1 as conversion of Q_{sender} to integer according to the table 4.
 12. Define X as conversion of F1 to data sequence in table 3.
 13. Make R-shift by one element to X.
 14. Define Z as conversion of X to 6 bits octal binary.
 15. Append Z with W as $(Z||W)$.
 16. Return M.
-

Figure 25 Encryption by EDS-M-ECC algorithm

-
1. Input: EC parameters, d is integer. $M=12$ -bit where M is received message.
 2. Output: plain message.
 3. Separate and save M from 1-6 bits in A .
 4. Separate and save M from 7-12 bits in B .
 5. Define F as conversion each A and B to 2digit according table 4.
 6. Make L- shift by one element for A and B .
 7. Compute $N_A = (\text{digit}_1 * 6^0) + (\text{digit}_2 * 6^1)$.
 8. Compute $K1 = N_A + 1$.
 9. Compute $W_A = K1.P$.
 10. Compute $N_B = (\text{digit}_1 * 6^0) + (\text{digit}_2 * 6^1)$.
 11. Compute $K2 = N_B + 1$.
 12. Compute $S = K2.P$, S which is cipher message (character).
 13. Compute $H = W_A$, $1, 1$ is private key of receiver.
 14. Compute $D = S - H$. Subtracting operation done by them location in table 4.
 15. Convert D to character according to table 4, which is plain message.
-

Figure 26 Decryption by EDS-M-ECC algorithm

3.6 Analysis of the Proposed Method

The proposed method offers a better performance in terms of memory size, in comparison with [33]. This reduction of memory usage can easily be depicted for both methods (proposed and [33]) as the following example. To send the “hello” message by the proposed method according to the Tab. 6 the following series of bits should be sent ‘hello’:

As “0000100110010000100000010000100010100000100010100000101000010”

while in [33] the series of bits that will be sent is:

“000001010000010000000101000000100000010101000101000001010100010100001010001010000110” as shown in Fig. 27, Tab. 7.

Table 6 Encryption Message Before/After Applying DS

Character	Point P_i	Encryption message before applying data sequence $C_m = (lP, P_i + lQ_B)$	Encryption message after applying data sequence 4 digit	Encryption message after covert into binary
h	(21, 11)	((7, 27), (10, 7))	0231	000010011001
e	(6, 20)	((7, 27), (2, 14))	0201	000010000001
l	(16, 23)	((7, 27), (1, 4))	0212	000010001010
l	(16, 23)	((7, 27), (1, 4))	0212	000010001010
o	(0, 4)	((7, 27), (1, 25))	0242	000010100010

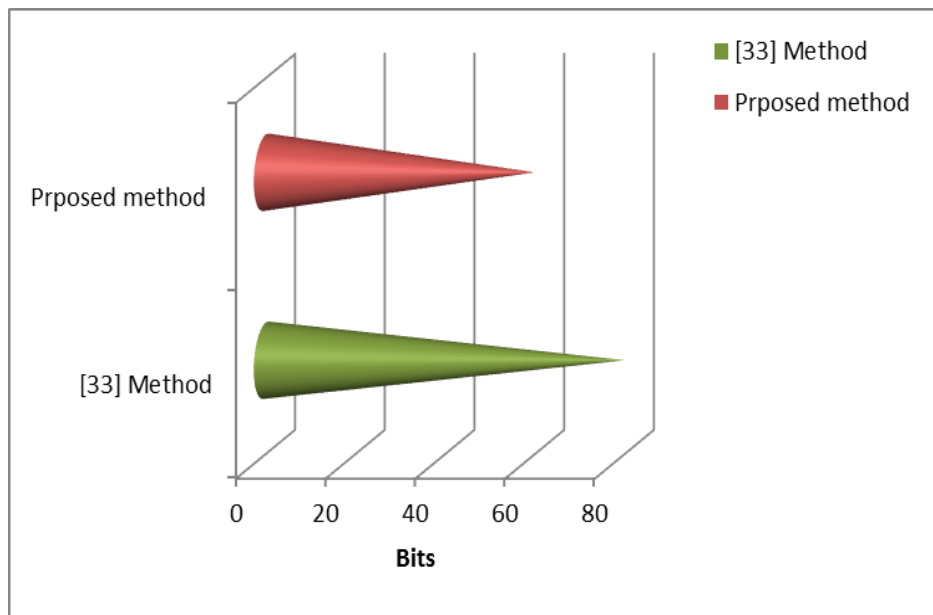


Figure 27 A Comparison between EDS-M-ECC and [33] to send (hello)

Table 7 A Comparison Between EDS-M-ECC and [33] to send (hello)

Plain text	No. of bits for sending 'hello' in the proposed method	No. of bits for sending 'hello' in [33]
hello	$5 \times 12 = 60$	$5 \times 16 = 80$

Finally, the comparison between the proposed method and [33] in Terms of no. of digits and no. of bits are shown in Tab. 8 and Tab. 9 respectively.

Table 8 A Comparison Between EDS-M-ECC and [33] Based on no. of Digits

Character	Encryption message after applying DS $C_m = (S(x_i, y_i), S(x_i, y_i))$ proposed method		Encryption message after applying DS $C_m = (S(x_i, y_i), S(x_i, y_i))$ in [33]	
	h	0231	4-digits	00110010
e	0201	00110002		
l	0212	00111011		
l	0212	00111011		
o	0242	00111012		

Table 9 A Comparison Between EDS-M-ECC and [33] Based on no. of Bits

Character	Point P_i	Encryption message before applying DS $C_m = (l.P, P_i + l.Q_B)$	(12 bits) for sending in the proposed method	(16 bits) for Sending in [33]
h	(21, 11)	((7, 27), (10, 7))	000010011001	00000101 00000100
e	(6, 20)	((7, 27), (2, 14))	000010000001	00000101 00000010
l	(16, 23)	((7, 27), (1, 4))	000010001010	00000101 01000101
l	(16, 23)	((7, 27), (1, 4))	000010001010	00000101 01000101
o	(0, 4)	((7, 27), (1, 25))	000010100010	00000101 01000110

3.7 Outcome

The EDS-M-ECC developed the previous method in [33] and uses the idea of a data sequence to protect the output of ECC. This development concentrates on the reduction of the number of bits for the sent encrypted message. The EDSM-ECC proves that it is much smaller than comparable methods in terms of the number of bits. Furthermore, we apply this method in the node authentication scheme in WSNs, which depends on data sequences over ECC based on MAC. This is to distribute the public key of the *BS* securely for each *CH* and distribute the public key of each *CH* securely for each cluster member and to provide authenticated messages between nodes.

CHAPTER 4

SYSTEM IMPLEMENTATION AND ANALYSIS OF RESULTS

4.1 Introduction

A WSN is a set of a large number of unheeded devices that are severely constrained in terms of power, cost and memory size. These devices in the WSN use radio links to communicate with each other. Sensor nodes are able to collect data from their surroundings and deliver their data to the *BS* for statistical analysis. In this chapter, a node authentication mechanism on a WSN based on ECC with the concept of data sequence that provides a safe channel and authentication between the nodes is implemented with recorded results. The MATLAB program is used to simulate the proposed scheme. The proposed scheme provides securely distribute public key of the *BS* and *CH*, and 'authentication of the message. Thus, it meets security requirements, and provides efficient resistance to any threat when compared with other schemes.

4.2 The System Model

The proposed scheme depends on the model of HWSN, as shown in Fig. 28. This model consists of a *BS* and two types of sensor, a small number of *CHs*, and a large number of *Ls*. Each *L* has limited power and small memory. In contrast, each *CH* has high power and large memory. Moreover, the *BS* has higher resource sensor nodes than the *CH* and *L*. In the proposed network, data are centralized and collected securely by using cryptographic mechanisms from the leaf nodes and are aggregated by the *CH*. Then, the *CH* sends the data to the *BS*. Notice that the *CH* can directly communicate with the *BS*, whereas the *L* can communicate with the *CH* via wireless channels.

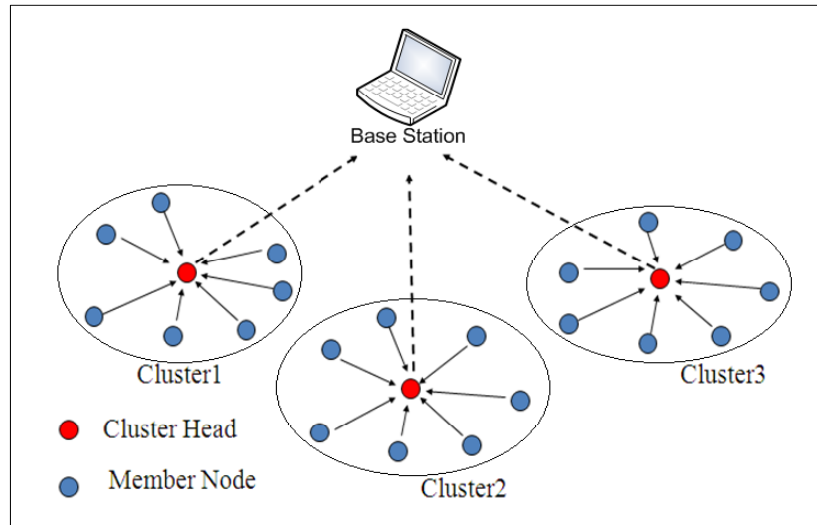


Figure 28 Hierarchical model of WSN

4.3 Simulation Setup

MATLAB version *R2013a* was used. The simulation parameters that are used in the proposed scheme are shown in Tab. 10.

Table 10 Simulation Parameters

Parameter	value
Deployment area (m*m)	100*100 m ²
No of nodes	100
BS location	100*120 m ²

4.4 System Assumptions

The following assumptions are taken into consideration for the proposed system:

1. The *BS* is trusted and fixed, has unlimited energy, powerful processing and is equipped to be tamper-resistant.
2. *CH* and *L* are not equipped to be tamper-resistant because of the cost constraints.
3. Every sensor is static.
4. Every sensor is assigned a unique node ID.

5. Each *CH* is pre-loaded with the share key of the *BS*, which is the same share key that exists in the *BS* prior to deployment.
6. The *BS* is responsible for generating the private key for each *CH*, while each *CH* is responsible for generating the private key for all L_{members} .
7. Network connectivity is checked, cluster formation is created, and each *CH* and *L* stores its private and public key via ECC.

4.5 The Proposed Scheme

In this section, the working principle of the proposed scheme which consists of four phases, is presented as follows:

4.5.1 Pre-distribution phase

1. The *BS* is safe, fixed and preloaded with its identity ID_{BS} , ID of all sensors, ECC parameters, share key, as well as being equipped to be tamper-resistant. Moreover, it is preloaded with Tab. 3.
2. Each *CH* is pre-loaded with its identity, the identity of the *BS*, ECC parameters, base point, Tab. 3, share key, and its own share key of the *BS*.
3. Each *L* is pre-loaded with its identity, the identity of all *CHs*, ECC parameters, base point, and Tab. 3.

4.5.2 Deployment phase

This phase occurs immediately after nodes deployment in the defined area assuming that 100 (4 of *CH* and 96 of *L*) nodes are randomly distributed inside a predefined area of size $100 * 100m^2$ as shown in Fig. 29 and Fig. 30.

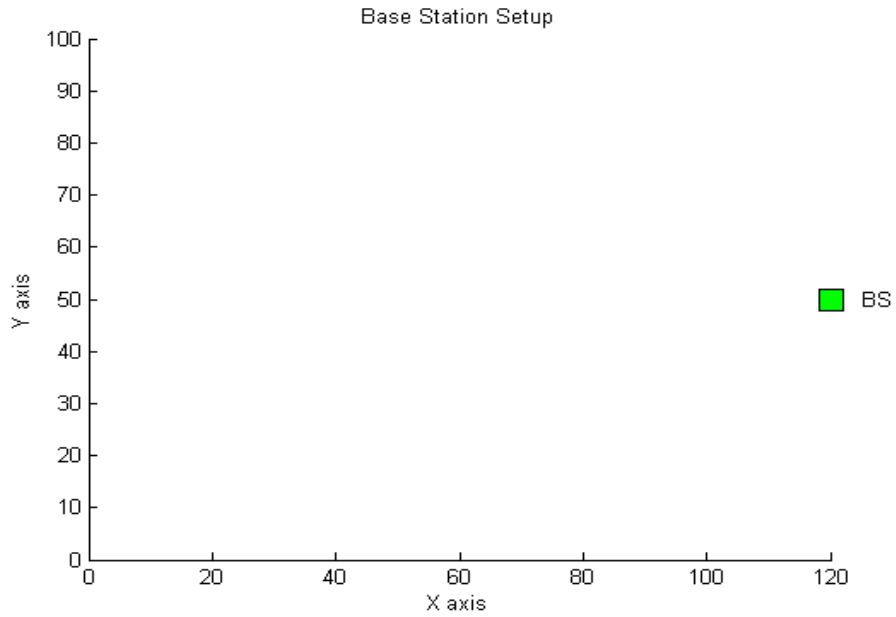


Figure 29 The *BS* setup

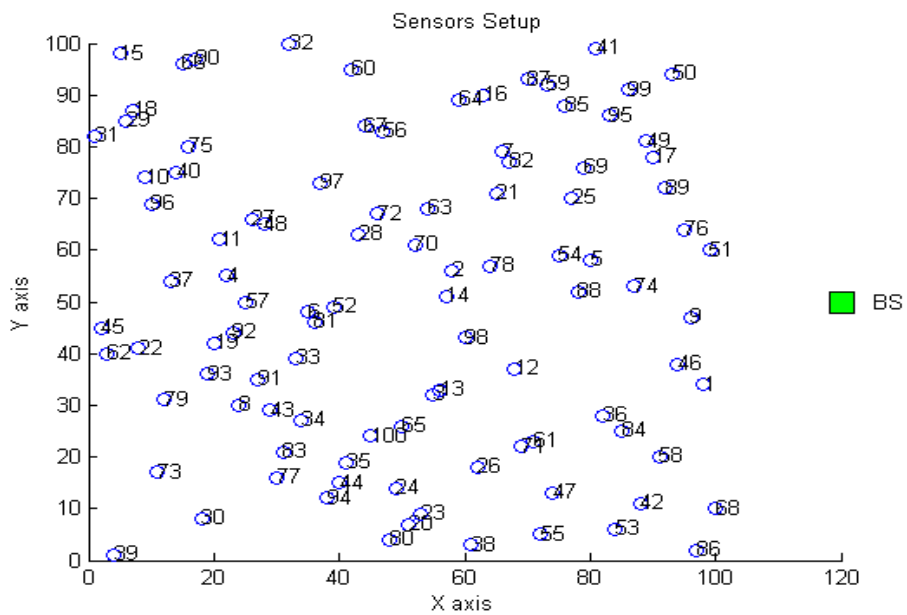


Figure 30 The sensors distribution

After the process of deploying sensors (CH, L) is finished, the cluster formation is created [51-52-53] as shown in Fig. 31. Hence, each CH acquires IDs of its $L_{members}$. Then, each CH obtains the private key from the BS and each L_{member} obtains the private key from the CH . Therefore, each sensor can compute its public key by Eq. (4.1).

$$Q = k.P, \text{ by ECC} \quad (4.1)$$

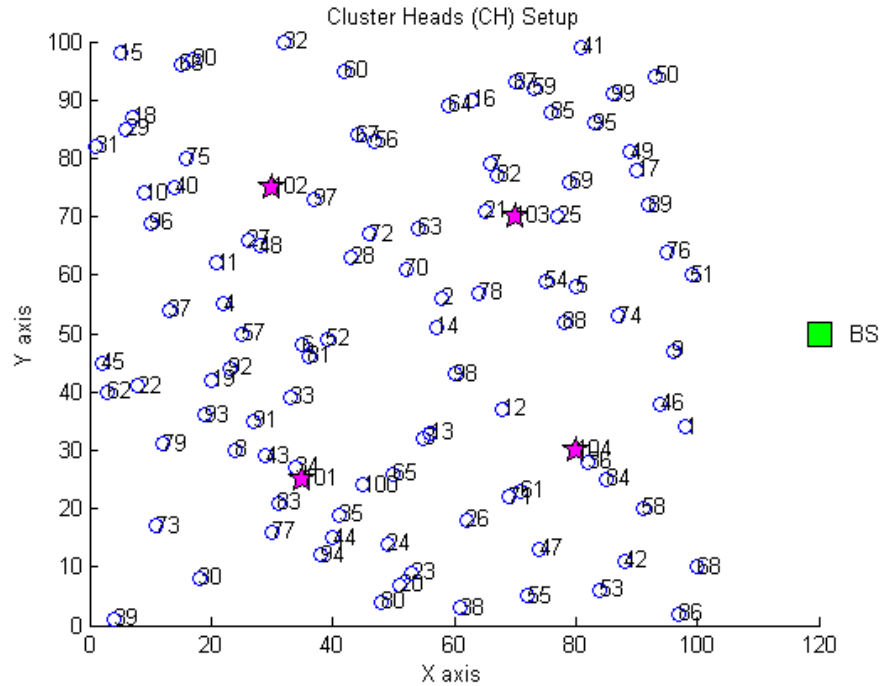


Figure 31 The CHs setup

4.5.3 Public key distribution phase

The proposed scheme relies on ECC and it is considered as one type of PKC to encrypt and decrypt messages (BS to CH or $L_{members}$ to CH) so as to achieve authentication between nodes. Each node should have a pair of keys (private and public). The private key of the node is kept secret while publishing the public key. Thus, the sender must have the public key of the recipient for encryption and decryption. The critical issue is the authentication of the public key in order to confirm that the public key is certainly owned by the person to whom it is claimed to belong. Otherwise, an adversary can impersonate a node and learn every message [6-18].

To increase the security of the network, the public key is encrypted and sent. The main goal of this phase is to distribute the public key of the BS and CHs securely. This phase consists of two procedures, which are explained in the following.

A. Distribution the public key of BS for CHs

Firstly, the BS creates Tab. 4, which acts as a fixed key ring of the base point. The BS computes a public key for itself using Eq. (4.2). Then, it calculates the MAC for its public key with the share key using the data sequence over ECC. Next, it is appended with its public key after converting it to a data sequence producing the cipher message, which is 12 bits. Finally, it is sent to each CH according to Eq. (4.3) and as the algorithm in Fig. 32 and as the flowchart in Fig. 33 and Fig. 34.

$$Q_{BS} = d.P, \text{ by ECC} \quad (4.2)$$

$$BS \rightarrow CH : MAC(\text{share key}, Q_{BS}) || Q_{BS} \quad (4.3)$$

-
1. Input: EC domain parameters, Share key and d are integer.
 2. Output: cipher message M.
 3. Compute $Q_{BS} = d.P$ by ECC, d is private key of BS.
 4. Define W as a location of Q_{BS} according to table 4, where W is integer.
 5. Represent A as a conversion of W according to table 3 to data sequence.
 6. Make R-shift by one element for A.
 7. Represent C as a conversion of A to octal binary (6-bit).
 8. Compute D = share key mode p.
 9. Compute F = D. Q_{BS} by ECC, where F is a point.
 10. Define E as a location of F according to table 4, where E is integer.
 11. Represent Y as conversion of E according to table3 to data sequence.
 12. Make R-shift by one element for Y.
 13. Represent Z as conversion of Y to octal binary (6-bit). Z is responsible MAC.
 14. M = (Z || C).
-

Figure 32 The BS sends its public key to each CH algorithm

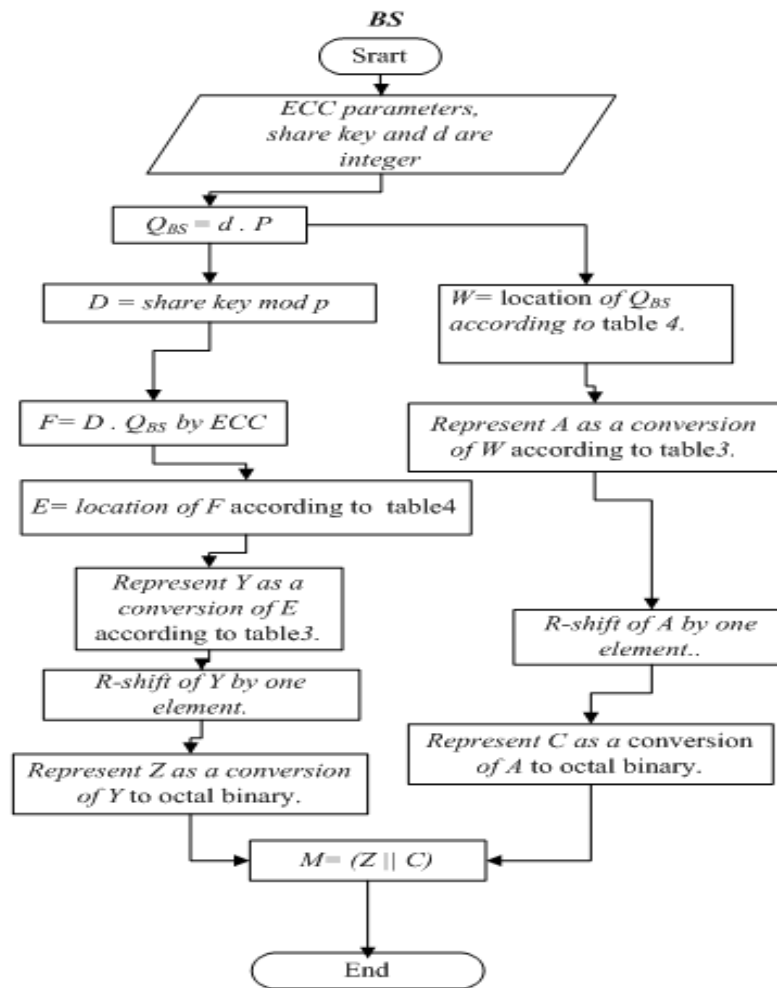


Figure 33 The BS sends its public key to each CH flowchart

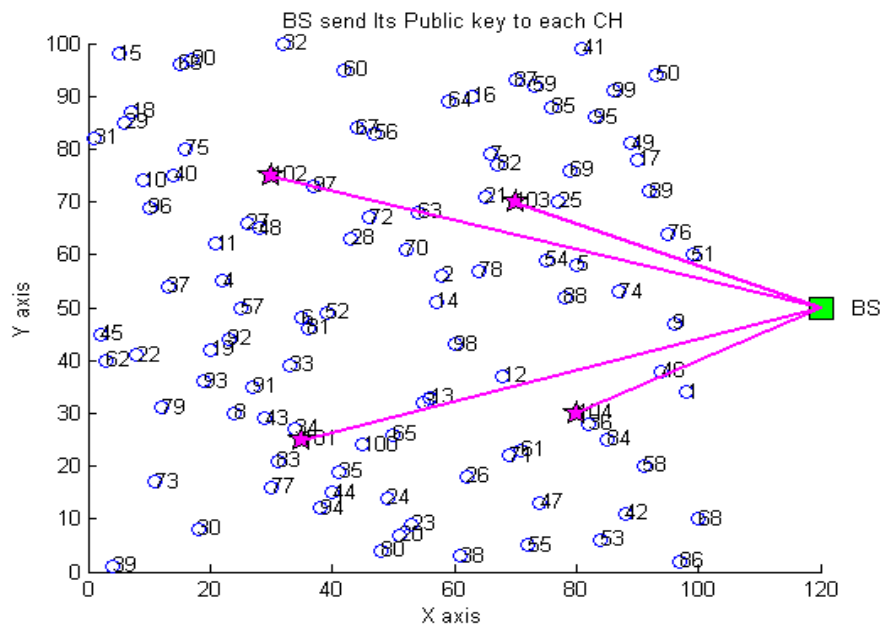


Figure 34 The BS sends its public key to each CH

Each *CH* receives a message from the *BS*, which consists of 12-bit. Initially, each *CH* generates Tab.4, which acts as a fixed key ring of the base point. Each *CH* decrypts the message to recover the public key of *BS*. Then, it calculates the MAC_{new} in the same manner. If $MAC_{new} = MAC_{received}$, then, save Q_{BS} . This indicates that the public key is authenticated to the *BS*. At that time, each *CH* sends an ACK to the *BS*, which has a hash value of ID_{CH} . Finally, each *CH* deletes the key rings, which is Tab. 4. Notice that only each *CH* can compute MAC_{new} because it has a share key according to the algorithm in Fig. 35 and as the flowchart in Fig. 36 and Fig. 37.

-
1. Input: EC domain parameters, Share key and d are integer, $M=12$ -bit where M is received message.
 2. Separate and save M from 1-6 bits in $M1$.
 3. Separate and save M from 7-12 bits in $M2$.
 4. Represent F as conversion of $M2$ to two digit.(3bits = 1 digit)
 5. Make L- Shift by one element for F .
 6. Compute $N = (digit_1 * 6^0) + (digit_2 * 6^1)$.
 7. Compute $K=N+1$.
 8. Define Q_{BS} as location of K according to table 4.(K is the index of the point)
 9. Define Z by algorithm in fig. 32 from step 8 to13.
 10. If ($Z=M1$) where $M1$ acts as the $MAC_{received}$ and $Z= MAC_{new}$.
 - 10.1 Save Q_{BS}
 - 10.2 Send ACK.
 - 10.3 End.
 11. Else
 - 11.1. Reject M .
 12. End.
-

Figure 35 Decryption and authentication public key of the *BS* algorithm

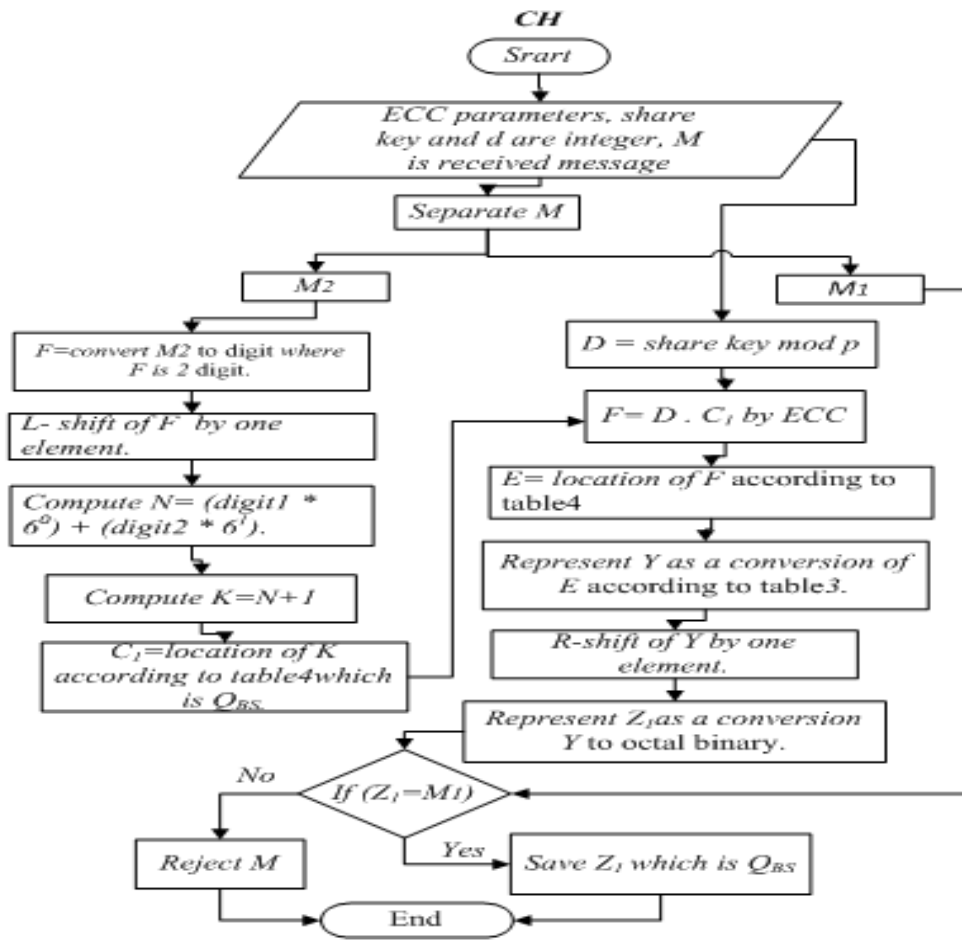


Figure 36 Decryption and authentication public key of BS flowchart

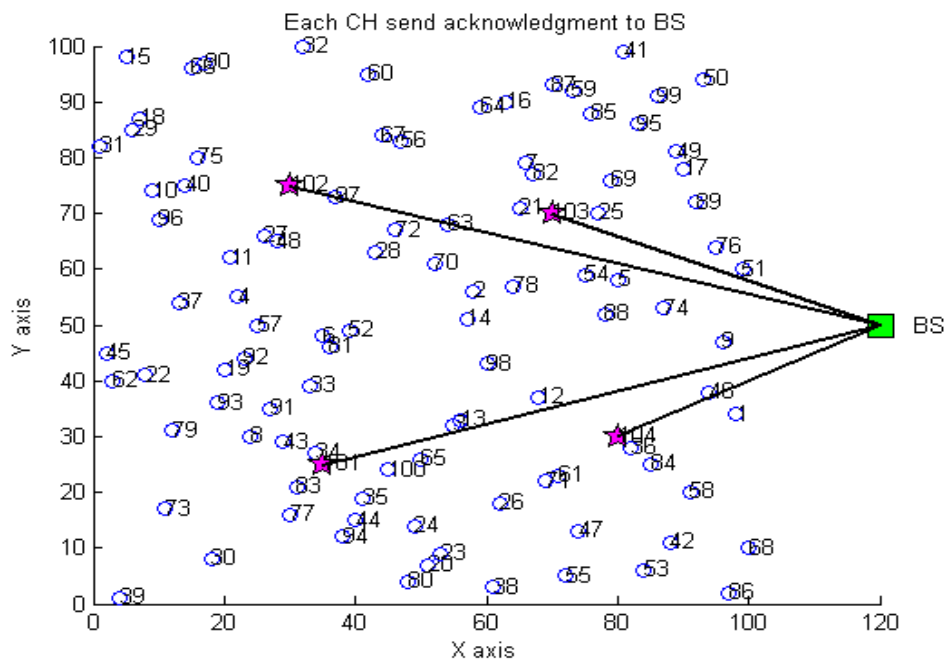


Figure 37 Each CH sends ACK to BS

This procedure can be clarified using example, which depends on the proposed method. The *BS* sends its public key to each *CH*.

Example. 29 is a prime number that satisfies the conditions of an EC equation such that $a = -1$, $b = 16$ as Eq. (4.4). The *BS* generates Tab. 4, which acts as a fixed key ring of 30 with the P (5, 7). The *BS* has Tab. 3, which is the data sequence and share key=198.

$$y^2 \bmod 29 = x^3 - x + 16 \bmod 29 \quad (4.4)$$

First, the *BS* selects a random number d , which is less than the order of $GF(p)$. Hence, we shall assume that the random number $d=13$ is a private key. Next the *BS* computes Q_{BS} by Eq. (4.5).

$$Q_{BS} = d_{BS} \cdot P, \text{ by ECC} \quad (4.5)$$

$$Q_{BS} = 13 (5, 7) = (7, 27).$$

Then, applying as the algorithm in Fig. 32 from (4 to 7) yields:

$(7, 27) = 13 = 20 = 02 = 000010$, which is the second part of a cipher message.

The *BS* computes the mode of a share key by Eq. (4.6).

$$D = \text{share key} \bmod p, \text{ by ECC} \quad (4.6)$$

$$D = 198 \bmod 29 = 24.$$

Then, the *BS* computes the MAC by Eq. (4.7).

$$MAC = (D \cdot Q_{BS}) \bmod p, \text{ by ECC} \quad (4.7)$$

$$MAC = 24 (7, 27) \bmod 29 = (28, 4).$$

$(28, 4) = 2 = 01 = 10 = 001000$ which is the first part of a cipher message.

Finally, append a first part with a second part. At this time, the transmitted message is 001000000010.

Each *CH* receives a cipher message and it must attain the following steps to recover the public key of *BS* (7, 27).

First, the *CH* generates Tab. 4, which acts as a fixed key ring of 30 with the base point (5, 7). Then, a separate cipher message in two parts yields:

1 to 6 bits is the first part = 001000, as a $MAC_{received}$.

7 to 12 bits is the second part = 010000 as a public key of *BS*.

Then, each *CH* achieves the following steps to recover Q_{BS} .

Each *CH* takes the second part = 000010 = 02 = 20 = [(digit₁ (0) * 6⁰) + (digit₂ (2 * 6¹)] = 12. k=12+1 =13.

Then, it finds the location of 13 according to Tab. 4, which is a public key of the *BS*.

Each *CH* recomputes the MAC_{new} using the same manner that the *BS* computes it to verify of Q . If $MAC_{new} = MAC_{received}$ then *CH* saves Q_{BS} (7, 27).

B. Distribution public key of *CH* for $L_{members}$

Firstly, the *CH* creates the Tab. 4, which acts as a fixed key ring of the base point. The *CH* computes the public key for itself by Eq. (4. 8). Then, it calculates the MAC for its public key with its ID using the data sequence over ECC. Next, it is appended with its public key after converting it to the data sequence producing the cipher message, which is 12-bit. Finally, it is sent to $L_{members}$ and deletes the key rings, which is Tab. 4, according to Eq. (4.9) , the algorithm in Fig.38.and as the flowchart in Fig. 39 and Fig. 40.

$$Q_{CH} = d.P , \text{ by ECC} \quad (4.8)$$

$$CH \rightarrow L : MAC(ID_{CH}, Q_{CH}) \parallel Q_{CH} \quad (4.9)$$

1. Input: EC domain parameters, ID_{CH} , d are integer.
2. Output: cipher message M .
3. Compute $Q_{CH} = d \cdot P$, by ECC, where d is private key of CH .
4. Define Q_{CH} by algorithm in fig.32 from, step 4 to 7.
Instead of Q_{BS} , use $Q_{CH} \cdot C$
5. Compute $D = ID_{CH} \bmod p$.
6. Compute $F = D \cdot Q_{CH}$, by ECC, where F is point.
7. Define E as location F according to table 4 by the order where E is integer.
8. Represent Y as conversion E according to table 3 to data sequence.
9. Make R-shift by one for Y .
10. Represent Z as conversion Y to octal binary (6-bit). Z is responsible MAC.
11. $M = (Z \parallel C)$.

Figure 38 The CH sends its public key to $L_{members}$ algorithm

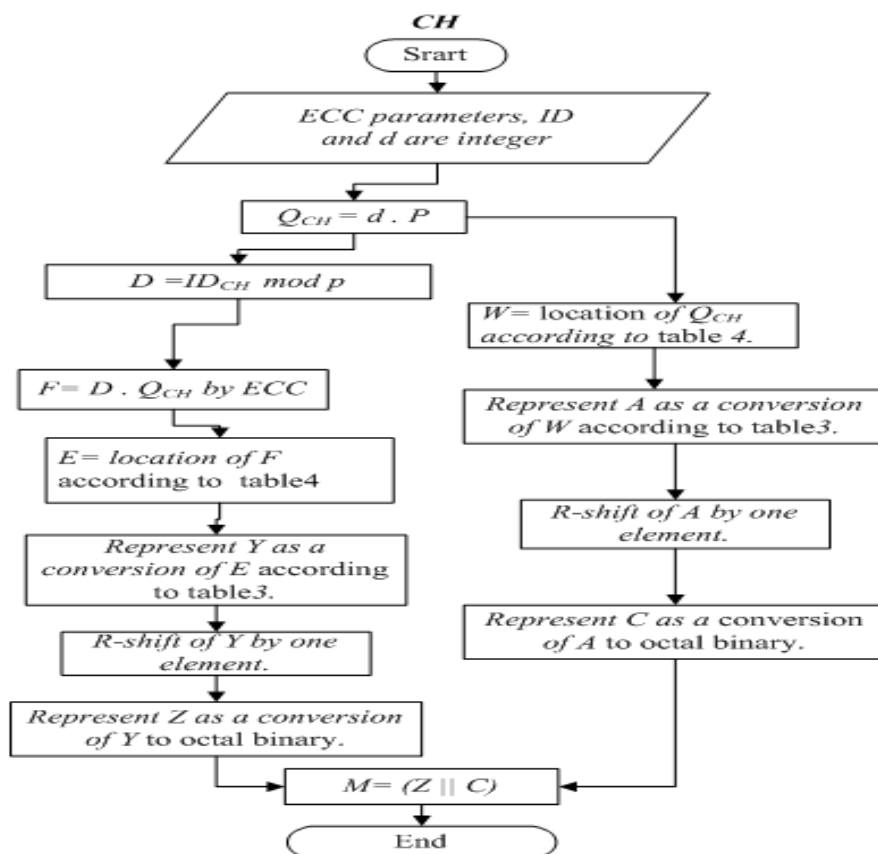


Figure 39 The CH sends its public key to $L_{members}$ flowchart

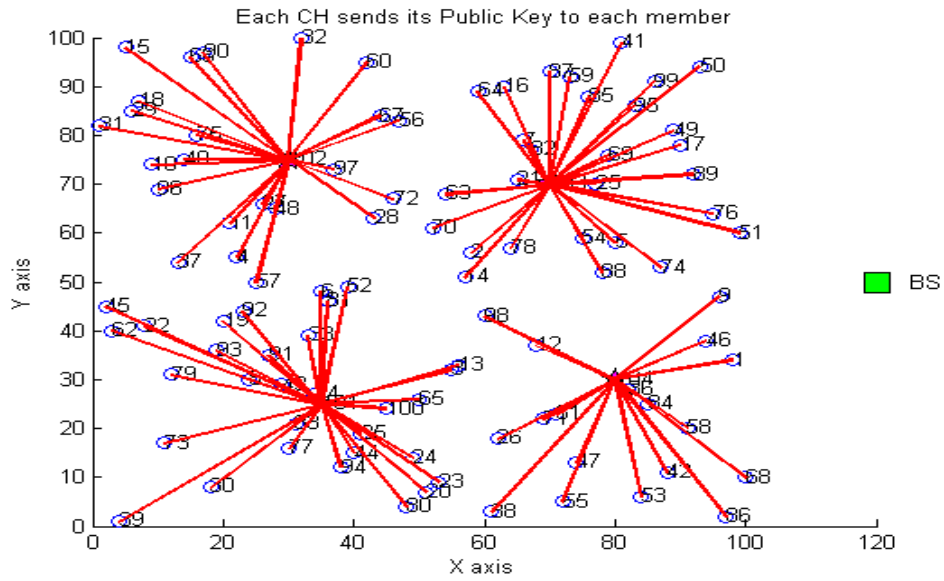


Figure 40 Each CH sends its public key to $L_{members}$

Each L_{member} receives a message from the CH , which consists of 12-bit. Initially, the L_{member} generates Tab. 4, which acts as a fixed key ring of the base point. Each L_{member} decrypts the message to recover the public key of the CH . Then, it calculates the MAC_{new} in the same manner. If $MAC_{new} = MAC_{received}$, then, save Q_{CH} . This indicates that the public key is authenticated to the CH . Finally, each L_{member} deletes the key rings, namely Tab. 4. Notice that, only $L_{members}$ can compute MAC_{new} because, it has the ID of the CH correct according to the algorithm in Fig. 41 and as the flowchart in Fig. 42.

-
1. Input: EC domain parameters, ID_{CH} and d are integer, $M=12$ -bit where M is received message.
 2. Separate and save M from 1-6 bits in $M1$.
 3. Separate and save M from 7-12 bits in $M2$.
 4. Steps from 4 to 7 as algorithm in fig.35.
 5. Define Q_{CH} as location of K according to table 4, (K is the index of the point)
 6. Define Z by algorithm in fig.38 from step 5 to 10.
 7. If ($Z=M1$) where $M1$ acts as the $MAC_{received}$ and $Z=MAC_{new}$
 - 7.1. Save Q_{CH} .
 - 7.2. End.
 8. Else
 - 8.1. Reject M .
 9. End.
-

Figure 41 Decryption and authentication public key of the CH algorithm

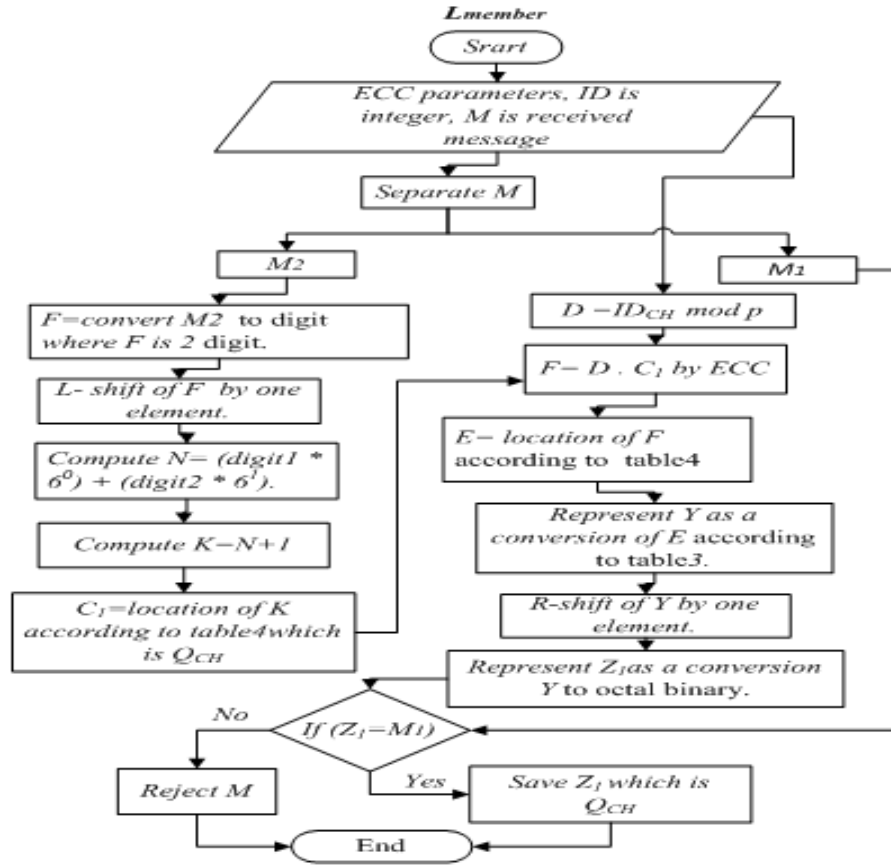


Figure 42 Decryption and authentication public key of CH flow chart

This procedure can be clarified with example, which depends on the proposed method. Each CH sends its public key to $L_{members}$.

Example. 29 is a prime number that satisfies the conditions of the EC equation. The CH generates Tab. 4, which acts as a fixed key ring of 30 with P (5, 7). The CH preloads Tab. 3, which is the data sequence and the ID of the CH is 100.

First, the CH selects a random number d which is less than the order of $GF(p)$. Hence, we shall assume that the random number $d=24$ is a private key. Next the CH computes Q_{CH} using Eq. (4.10).

$$Q_{CH} = d_{CH} \cdot P, \text{ by ECC} \quad (4.10)$$

$$Q_{CH} = 24 (5, 7) = (2, 15).$$

Then, applying as the algorithm in Fig. 32, from (4 to 7) yields:

$(2, 15) = 24 = 53 = 35 = 100101$, which is the second part of a cipher message.

The CH computes the mode of its ID by Eq. (4.11).

$$D = ID_{CH} \bmod p, \text{ by ECC} \quad (4.11)$$

$$D = 100 \bmod 29 = 13.$$

Then, the *CH* computes the MAC by Eq. (4.12).

$$MAC = (D \cdot Q_{CH}) \bmod p, \text{ by ECC} \quad (4.12)$$

$$MAC = 13 (2, 15) \bmod 29 = (28, 4).$$

$(28, 4) = 2 = 01 = 10 = 001000$ which is the first part of a cipher message.

Finally, a first part is appended to a second part. At this time, the transmitted message is 001000100101.

$L_{members}$ received the cipher message and it must attain the following to recover the public key of the *CH*, which is $(2, 15)$.

First, the *CH* generates Tab. 4, which acts as a fixed key ring of 30 with the base point $(5, 7)$. Then, the cipher message is separated into two parts and yields:

1 to 6 bits is the first part = 001000, where is $MAC_{received}$.

7 to 12 bits is the second part = 100101, as a public key of *CH*.

Then, each L_{member} achieves the following steps to recover Q_{CH} .

Each L_{member} takes the second part = 100101 = 35 = 53 = $[(digit_1 (5) * 6^0) + (digit_2 (3 * 6^1))] = 23$. $k = 23 + 1 = 24$.

Then, find the location of 24 according to Tab. 4, which is the public key of *CH*.

Each L_{member} recomputes MAC_{new} using the same method that *CH* computes it to verify of Q . If $MAC_{new} = MAC_{received}$ then $L_{members}$ saves $Q_{CH}(2, 15)$.

4.5.4 Message authentication phase

In the hierarchical sensor network model, data transmission consists of two steps. Firstly, each L_{sensor} sends their data to *CH*. Secondly, *CH* sends it to *BS*. Due to the use of wireless channels to communicate between the nodes, those channels become easily vulnerable to various attacks. Thus, each sensor encrypts the message prior to transmission in order to avoid attacks and to achieve the security requirements. This phase consists of two procedures, which are explained as follows:

A. L_{member} sends their data to its CH .

When, L_{member} sends a message to CH , L_{member} generates Tab. 4, which acts as a fixed key ring of the base point. L_{member} encrypts the plain message with Eq. (4.13). Then, L_{member} calculates MAC for it with ECC and converts it to a data sequence, after which it converts the cipher message to a data sequence and adds it with ID_L . Then it appends it with the result of MAC and ID_L , and then sends it to CH . Finally, L_{member} deletes the key rings according to Eq. (4.14) and as the algorithm in Fig. 43.

$$Cm = Pm + (d_L \cdot Q_{CH}) \quad (4.13)$$

$$L \rightarrow CH : MAC(ID_{CH}, cipher\ message) \parallel cipher\ message + ID_L \parallel ID_L \quad (4.14)$$

-
1. Input : EC domain parameters, d , ID_{CH} and ID_L are integer, Plain message is ρ character.
 2. Output: cipher message M .
 3. Represent D as conversion of location plain message according to table 4.
 4. Compute $A = d_L \cdot Q_{CH}$, by ECC, where d is private key.
 5. Define E as location of A according to table 4.
 6. Compute $G = D + E$.
 7. Represent cipher text as conversion of G to point according to table 4.
 8. Compute $W = ID_{CH} \text{ mode } p$.
 9. Compute $F = W \cdot \text{Cipher message}$, by ECC, where F is a point.
 10. Define Z by algorithm in fig.32 from step10 to13. Z is responsible MAC.
 11. Define O by algorithm in fig.32 from step10 to13 for cipher text.
 12. Compute $U = O + ID_L$.
 13. $M = Z \parallel U \parallel ID_L$.
-

Figure 43 L_{member} sends their data to its CH algorithm

Upon receipt of the cipher message from L_{member} , each CH generates Tab. 4, which acts as a fixed key ring of the base point. CH performs a number of processes to find the encryption message. Then, each CH is verified from MAC_{received} by recalculating MAC_{new} . If it is equivalent, it indicates that the legal L_{member} has sent a message to CH . Notice that only CH can compute MAC_{new} because it has the ID of L_{member} . Each

CH decrypts the message to find a plain message using Eq. (4.15). Finally, each *CH* deletes the key rings according to the algorithm in Fig. 44.

$$P_m = C_m - (d_{CH} \cdot Q_L) \quad (4.15)$$

-
1. Input: EC domain parameters, d , ID_{CH} and ID_L are integer, plain message is 'character' where M is received message, and k is length of M .
 2. Separate and save M from 1- 6 bits in $M1$.
 3. Separate and save M from 7-12 bits in $M2$.
 4. Separate and save M from 13- k bits in $M3$.
 5. Compute $R1=M2 - M3$, where $M3$ is ID_L .
 6. Define R by algorithm in fig.35 from step4 to 7. Instead of $M2$, use R .
 7. Define cipher text as location of K according to table 4, which is a point.
 8. Define Z by algorithm in fig.43 from steps 8to10.
 9. If ($Z=M1$) where $M1$ acts as the $MAC_{received}$ and $Z= MAC_{new}$.
 - 9.1. Define cipher text as location of K according to table 4, which is integer.
 - 9.2 Compute $A= d_{CH} \cdot Q_L$, by ECC, where d is private key.
 - 9.3. Define J as location of A according to table 4, which is integer.
 - 9.4. Compute $I= Cipher\ text - J$.
 - 9.5. Plain message = 'Character' location of I according to table 4.
 - 9.6. End.
 10. Else
 - 10.1. Reject M .
 11. End.
-

Figure 44 Decryption and authentication message by *CH* algorithm

B. *CH* sends their data to *BS*.

When *CH* sends a message to *BS*, *CH* generates Tab. 4, which is acting as a fixed key ring of the base point. *CH* encrypts the plain message using Eq. (4.16). Then *CH* calculates MAC for it using ECC and converts it to a data sequence, after which it converts the cipher message to a data sequence and adds it with ID_{CH} . Then it is appended to the result of MAC and ID_{CH} . Then it is sends it to *BS*. Finally, *CH* deletes the key rings according to Eq. (4.17) and as the algorithm in Fig. 45.

$$Cm = Pm + (d_{CH} \cdot Q_{BS}) \quad (4.16)$$

$$CH \rightarrow BS : MAC(\text{share key}, \text{cipher message}) || \text{cipher message} + ID_{CH} || ID_{CH} \quad (4.17)$$

-
1. Input: EC domain parameters, share key, d, and ID_{CH} are integer; Plain message is 'character.
 2. Output: cipher message M.
 3. Represent D as conversion of location plain message according to table 4.
 4. Compute $A = d_{CH} \cdot Q_{BS}$, by ECC. d is private key.
 5. Define E as location of A according to table 4.
 6. Compute $G = D + E$.
 7. Represent cipher message as conversion of G to point according to table 4.
 8. Compute $W = \text{Share key mode } p$.
 9. Compute $F = W \cdot \text{Cipher message}$, by ECC, where f is a point.
 10. Define Z by algorithm in fig.32 from step10 to13. Z is responsible MAC.
 11. Define O by algorithm in fig.32 from step10 to13 for cipher text.
 12. Compute $U = O + ID_{CH}$.
 13. $M = Z || U || ID_{CH}$.
-

Figure 45 CH sends their data to BS algorithm

Upon receipt of the cipher message from CH, BS generates Tab. 4, which is acting as a fixed key ring of the base point. BS performs a number of processes to find the encryption message. Then, the BS verifies $MAC_{received}$ by recalculating MAC_{new} . If it is equivalent, it indicates that the legal CH has sent a message to BS. Finally, BS decrypts the message to find a plain message using Eq. (4.18), as the algorithm in Fig. 46 and Fig. 47.

$$Pm = Cm - (d_{BS} \cdot Q_{CH}) \quad (4.18)$$

1. Input: EC domain parameters, Share key, ID_{CH} and d are integer, plain message is 'character' where M is received message and k is length of M .
2. Separate and save M from 1-6 bits in $M1$.
3. Separate and save M from 7-12 bits in $M2$.
4. Separate and save M from 13- k bits in $M3$.
5. Compute $R1=M2-M3$. where $M3= ID_{CH}$.
6. Define R by algorithm in fig.35 from step4 to 7. Instead of $M2$, use R .
7. Define cipher message as location K according to table 4, which is a point.
8. Define Z by algorithm in fig.45 from steps 8 to10.
9. If ($Z=M1$) where $M1$ acts as the $MAC_{received}$ and $Z= MAC_{new}$.
 - 9.1. Define cipher message as location of K according to table 4, which is integer.
 - 9.2 Compute $A= d_{BS} \cdot Q_{CH}$, by ECC, where d is private key.
 - 9.3. Define J as location of A according to table3, which is integer.
 - 9.4. Compute $I= Cipher\ message - J$.
 - 9.5. Plain message= 'Character' location I according to table 4.
 - 9.6. End.
10. Else
 - 10.1. Reject M .
11. End.

Figure 46 Decryption and authentication message by *BS* algorithm

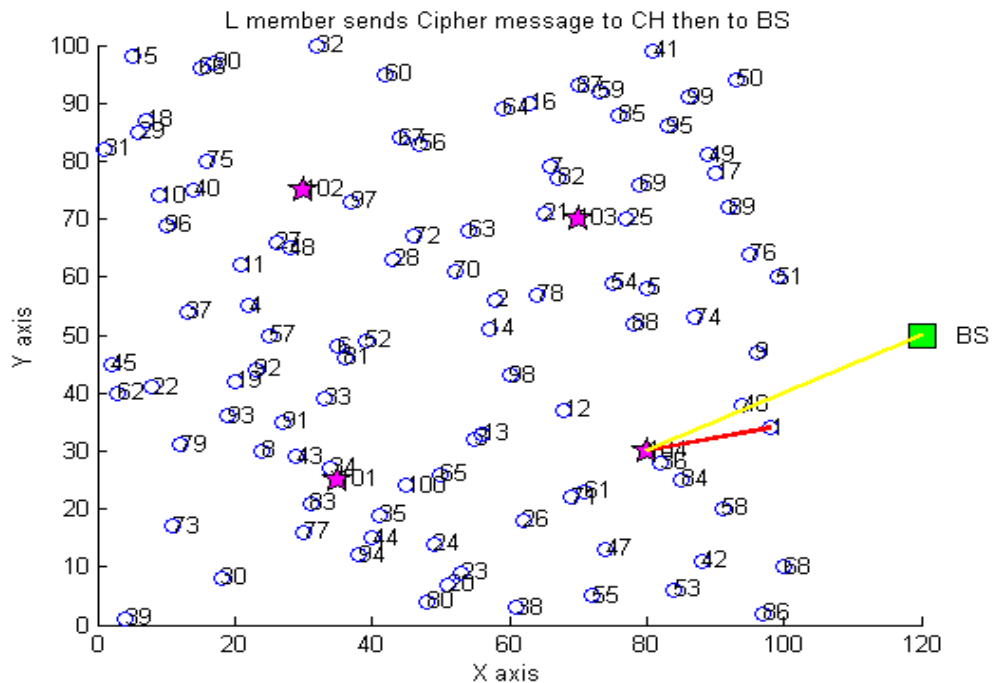


Figure 47 L_{member} sends a cipher message to *CH* then to *BS*

4.6 Results and Analysis of the Proposed Scheme

In this section, the results are evaluated and the security issue is analyzed. A comparison between the proposed scheme and the other schemes is discussed.

4.6.1 Simulation results

The proposed scheme has been simulated with the MATLAB *R2013a* program, which is implemented on a 1.80 GHz Intel Core i3 CPU with 4 GB of RAM running Windows 7 Ultimate. The performance timings have been recorded in seconds. The statistical analysis is summarized as shown in Tab. 11 and Fig. 48.

Table 11 Computation of Time Execution of Scalar Multiplication

No. of Scalar Multiplications	Time execution to calculate Scalar Multiplication in the proposed based scheme	Calculations
2P	0.00057s	2P
5P	0.00025s	$2(2P)+P$
8P	0.00055s	$2(2(2P))$
11P	0.00060s	$(2(2(2P)+P))+P$
14P	0.00052s	$2(2(2P)+P)+P$
17P	0.00033s	$2(2(2(2P)))+P$
20P	0.00034s	$2(2(2(2P)+P))$
23P	0.00058s	$2(2(2(2P)+P)+P)+P$
26P	0.00024ms	$2(2(2((2P)+P)))+P$
29P	0.00063ms	$2(2(2((2P)+P)+P))+P$

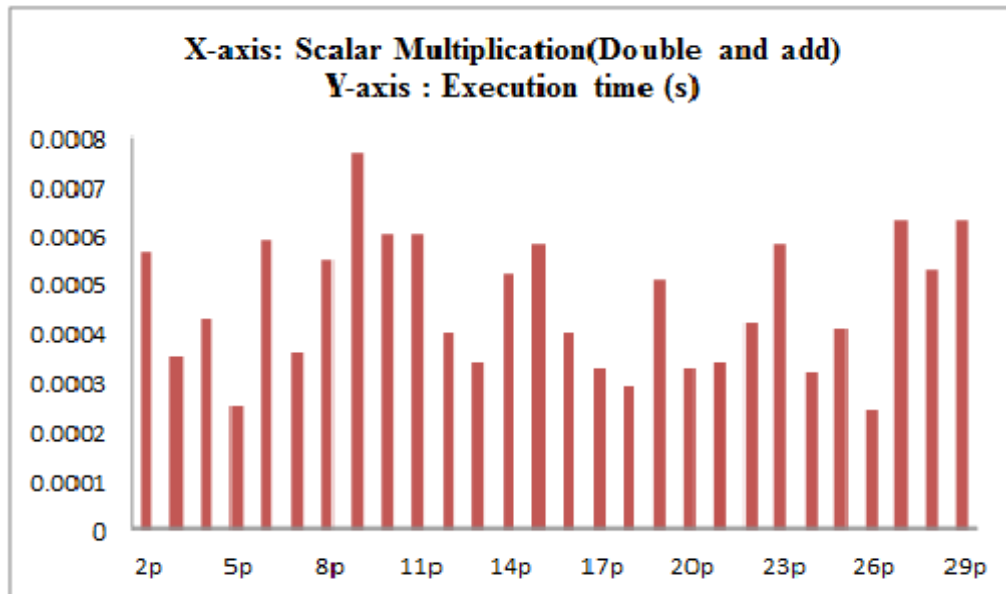


Figure 48 Time Execution of the 30 operations for the base point

4.6.2 Security analysis

In this section, the proposed scheme is analyzed as two aspects: security requirements and security attacks; then a comparison is made with the other schemes.

1. Security requirement analysis

The security requirements are important part in WSNs and the proposed scheme is compared with some previous schemes, in the basis of satisfies of the security requirements and is shown in Tab. 12.

Table 12 Comparison in Requirements Security

Requirements security	Proposed scheme	Scheme [54] Zinaida B et al.	Scheme [30] Canming J et al.	Scheme [31] Huei R. T et al.
Authentication	One-way	One-way	One-way	One-way
Confidentiality	Satisfies	Not Satisfies	Not Satisfies	Not Satisfies
Integrity	Satisfies	Not Satisfies	Not Satisfies	Not Satisfies
Cryptographic mechanism	PKC and SKC(MAC) based on ECC	PKI based on ECC	Self-certified key	hash function and XOR

1. **Authentication:** The proposed scheme satisfies the authentication requirement. Because, every message is encrypted with the ID of the node that depends on PKC and MAC to verify the identity of legitimate nodes during exchanges of messages.
2. **Confidentiality:** This remains an important point even though a message is authenticated so as to maintain the secrecy of messages. The proposed scheme satisfies it due to the fact that each message will not offer any usable information and because everything is encrypted by applying the data sequence over ECC.
3. **Integrity:** Message integrity ensures the receiver that the received message is not altered in transit by an attacker. The proposed scheme fulfills this requirement. The attacker is not active due to using a MAC such that it is encrypted and attached to a message.

2. Security attacks analysis

In this section, some attacks that can take place in WSNs have been identified, and the proposed scheme is compared with some previous schemes in terms of resistance against the attacks as shown in Tab. 13.

Table 13 The Attacks Security Comparison (Y=Yes,N=No)

Type attacks	Proposed scheme	Scheme[55] Boushra M et al.	Scheme[56] Sajid H et al.
Sybil attack	Y	N	Y
Block hole attack	Y	Y	Y
Eavesdropping attack	Y	N	N
Tunneling attack	Y	Y	Y

1. Eavesdropping attack: This type of attack can be avoided in our proposed scheme because it cannot understand the content of the messages. Moreover, it 'does not know the encryption key. Hence, an attacker cannot extract any valuable information from any messages.
2. Black khole attack: This attack can be avoided in our proposed scheme because it is based on a forger message sent to the base station. This is impossible because the message is encrypted prior sending and the *CH* can only communicate with the base station depending on a share key.
3. Sybil attack: An attacker uses a fake ID and uses it when he wants to send a message to the nodes. This attack can be prevented because the ' identity of the node is verified prior to decrypting the message. Each *CH* uses a share key for communication with the *BS*; furthermore, the *CH* has the IDs of its member. Thus, it can ignore an illegal ID.

4. Tunneling attack: An attacker puts himself between two legal nodes and starts to send fake or modified messages to the nodes. These attacks can be avoided in the proposed scheme because it uses MAC to verify the message; then the legal node rejects the message.
5. Insertion and modification attack: This attack attempts to change the messages exchanged between nodes. This type of attack can be avoided in the proposed scheme by using the PKC to encrypt messages. Moreover, an attacker without a valid ID or share key would not be able to compute MAC.

Finally, although the proposed scheme achieves many operations in addition to the authentication, it gives a logical duration of node live when making a comparison with the LEACH, as shown in Fig. 49.

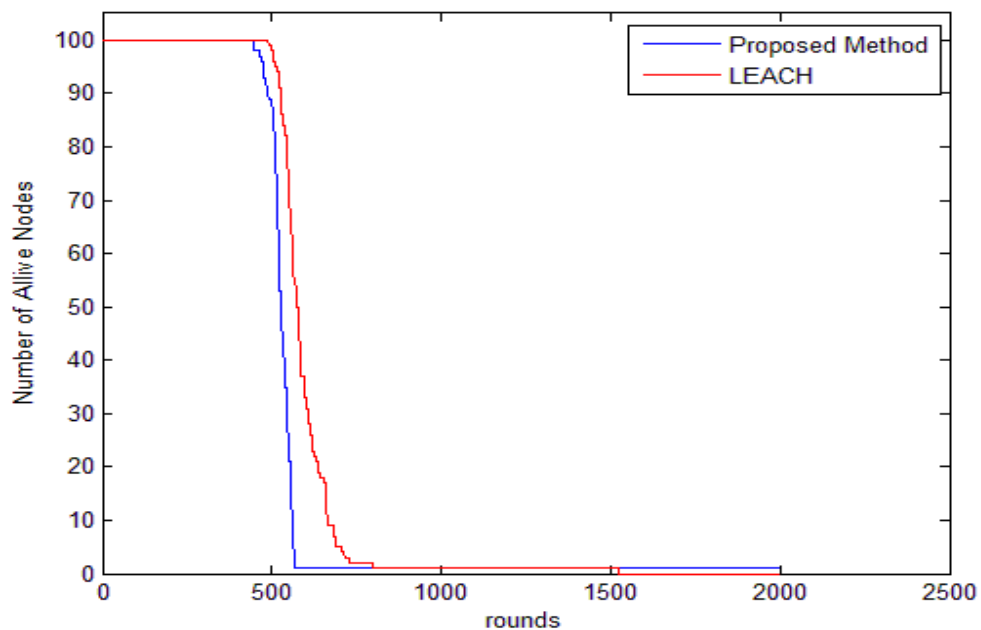


Figure 49 Network lifetime

CHAPTER 5

CONCLUSION

Several WSN applications require communication between nodes to be secure and free from a variety of attacks. Since the wireless channel is easily vulnerable to attacks, the probability of modifying messages is high. Security of this channel is achieved by the proposed scheme through the secure distribution of the public key of the base station and cluster head which depends on [57] and by decreasing the number of each point to 6 bits. The public key can be retrieved only with the regular sensor nodes, whereas harmful nodes cannot be retrieved. Moreover, providing authentication messages between the nodes depends on [57]. In the proposed scheme, the public key of the cluster head is distributed after deployment. This is the opposite of [27] in order to decrease the memory size. Its security depends on the ECDLP. In terms of attacks, the security analysis showed that the proposed scheme is efficient and provides security requirements better than other schemes.

REFERENCES

1. **Javier L., Jianying Z., (2008)**, "*Wireless Sensor Network Security*", IOS Press, Netherland, pp. 77,115.
2. **Sanjit K., Subasish M., Prasant K. P., (2010)**, "*A Survey on Application of Wireless Sensor Network Using Cloud Computing*," International Journal of Computer Science & Engineering Technologies (E-ISSN: 2044-6004), vol. 1, pp. 50-55.
3. **Liljana G., Srdjan K., Veljko M., Ivan S., Roman T., (2011)**, "*Application and Multidisciplinary Aspects of Wireless Sensor Networks*", Concepts, Integration, and Case Studies, Springer, London, pp. 78-81,89-90.
4. **Daniele P., Martin H., (2005)**, "*Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing*," Circuits and Systems Magazine, IEEE, vol. 5, pp. 19-31.
5. **Ian. F. A., Su W., Sankarasubramaniam Y., Cayirci E., (2002)**, "*Wireless Sensor Networks: A Survey*," Computer networks, vol. 38, pp. 393-422.
6. **Yun Z., Yuguang F., Yanchao Z., (2008)**, "*Securing Wireless Sensor Networks: A Survey*," Communications Surveys & Tutorials, IEEE, vol. 10, pp. 6-28.
7. **Khadija R., Nujhat N., Al-Sakib P., (2010)**, "*An Enhanced Tree-Based Key Management Scheme for Secure Communication in Wireless Sensor Network*," in High Performance Computing and Communications (HPCC), 12th IEEE International Conference on, pp. 671-676.
8. **Benamar K., Djilalli M., Mohammed F., Abdellah M., (2012)**, "*An Efficient Key Management Scheme for Hierarchical Wireless Sensor Networks*," Wireless Sensor Network, pp. 155-161.

9. **Rehana Y., (2012)**, Ph. D. Thesis, "*An Efficient Authentication Framework for Wireless Sensor Networks*", Dept. School of Computer Science, School of Computer Science College of Engineering and Physical Sciences, University of Birmingham, United Kingdom, pp. 4-5,27-29,104-105.
10. **Azzedine B., (2008)**, "*Algorithms and Protocols for Wireless Sensor Networks*", John Wiley & Sons, New Jersey, vol. 62, pp. 481,486.
11. **Jaykumar S. P., Vijaykumar M. C., (2014)**, "*Security Vulnerability and Robust Security Requirements Using Key Management in Sensor Network,*" International Journal of Grid & Distributed Computing, vol. 7, pp. 23-28.
12. **Darrel H., Alfred M., Scott V., (2004)**, "*Guide to Elliptic Curve Cryptography*", Springer, New York, pp. 1-3,3-6,11-13,182-189.
13. **Wang X., Liu J., Zhao C., Wang Y., (2012)**, "*Hash Function Construction and Analysis for Wireless Sensor Network,*" Computer Science and Network Technology (ICCSNT), International Conference on Dec, pp. 921-924.
14. **William S., (2006)**, "*Cryptography and Network Security,*" Pearson Education India, pp. 320-323.
15. **Rasmita R., Itun S., (2011)**, "*A Survey on Authentication Protocols for Wireless Sensor Network,*" International Journal of Engineering Science and Technology, vol. 3, pp. 4253-4256.
16. **Munivel E., Ajit G. M., (2010)**, "*Efficient Public Key Infrastructure Implementation in Wireless Sensor Networks,*" in Wireless Communication and Sensor Computing, ICWCSC, International Conference on, pp. 1-6.
17. **Jianqing M., Shiyong Z., Yiping Z., Yu W., (2006)**, "*PEAN: A Public Key Authentication Scheme in Wireless Sensor and Actor Network,*" in Computer and Information Technology, CIT'06, The Sixth IEEE International Conference on, pp. 230-230.
18. **Wenliang D., Ronghua W., Peng N., (2005)**, "*An Efficient Scheme for Authenticating Public Keys in Sensor Networks,*" in Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 58-67.

19. **Ping G., Jin W., Jiezhong Z., Yaping C., (2013)**, "*Authentication Mechanism on Wireless Sensor Networks: A Survey*," The 2nd International Conference on Information Technology and Computer Science, vol. 25, pp. 425-431.
20. **Madhumita P., (2014)**, "*Security in Wireless Sensor Networks Using Cryptographic Techniques*," American Journal of Engineering Research vol. 03, pp. 50-56.
21. **Feng Z., Leonidas J. G., (2004)**, "*Wireless Sensor Networks: An Information Processing Approach*", Morgan Kaufmann, Elsevier, pp. 9-10,14-15.
22. **Kazem S., Daniel M., Taieb Z., (2007)**, "*Wireless Sensor Networks: Technology, Protocols, and Applications*", John Wiley & Sons, New Jersey, PP. 10-11.
23. **Ayman T., Ayman K., Ali C., Imad E., (2014)**, "*Authentication Schemes for Wireless Sensor Networks*," in Mediterranean Electrotechnical Conference (MELECON), IEEE, pp. 367-372.
24. **Suresh J. S., Manjushree A., Eswaran P., (2014)**, "*Differential Power Analysis (DPA) Attack on Dual Field ECC Processor for Cryptographic Applications*," International Journal of Engineering Science and Innovative Technology, pp. 1-5.
25. **Wafa B. J., Aref M., Habib Y., (2010)**, "*An Efficient Source Authentication Scheme in Wireless Sensor Networks*," in Computer Systems and Applications (AICCSA), IEEE/ACS International Conference on, pp. 1-7.
26. **Pratik R., Nachiketa T., (2013)**, "*An Efficient Node Authentication Scheme Based on Elliptic Curve Cryptography for Wireless Sensor Networks*," International Journal of Computer Science & Engineering Technology, vol. 4, pp. 561-566.
27. **Xiaojiang D., Mohsen G., Yang X., Hsiao H. C., (2009)**, "*Transactions Papers a Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks*," Wireless Communications, IEEE Transactions on, vol. 8, pp. 1223-1229.

28. **Abduvaliyev A., Sungyoung L., Young K. L., (2009),** "*Modified SHA-1 Hash Function (mSHA-1)*," in ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications, pp. 1320-1323.
29. **Lingling S., Zhigang J., Zihui W., (2012),** "*The Application of Symmetric Key Cryptographic Algorithms in Wireless Sensor Networks*," Physics Procedia, vol. 25, pp. 552-559.
30. **Canming J., Bao L., Haixia X., (2007),** "*An Efficient Scheme for User Authentication in Wireless Sensor Networks*," in Advanced Information Networking and Applications Workshops, AINAW'07, 21st International Conference on, pp. 438-442.
31. **Huei R. T., Rong H. J., Wu Y., (2007),** "*An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks*," in Global Telecommunications Conference, GLOBECOM'07, IEEE, pp. 986-990.
32. **Wendi R. H., Anantha C., Hari B., (2000),** "*Energy-Efficient Communication Protocol for Wireless Microsensor Networks*," in System Sciences, Proceedings of the 33rd Annual Hawaii International Conference on, vol. 2, pp. 1-10.
33. **Amounas F., El Kinani E., (2012),** "*ECC Encryption and Decryption with a Data Sequence*," Applied Mathematical Sciences, vol. 6, pp. 5039-5047.
34. **Abror A., Sungyoung L., Young K. L., (2009),** "*Simple Hash Based Message Authentication Scheme for Wireless Sensor Networks*," in Communications and Information Technology, ISCIT, International Symposium on, pp. 982-986.
35. **Shantala P., Vijaya K. B. P., Sonali S., Rashique J., (2012),** "*A Survey on Authentication Techniques for Wireless Sensor Networks*," International Journal of Applied Engineering Research, vol. 7, pp. 1-4.
36. **Gaurav G., Nikita V. M., (2014),** "*Securing Multipath Routing Protocol Using Authentication Approach for Wireless Sensor Network*," in Communication Systems and Network Technologies (CSNT), Fourth International Conference on, pp. 729-733.

37. **Abdullah A., Rumana A., (2012)**, "*Secure Sensor Node Authentication in Wireless Sensor Networks*," International Journal of Computer Applications, vol. 46, pp. 10-17.
38. **Vamsi P. R., Kant K., (2014)**, "*A Lightweight Sybil Attack Detection Framework for Wireless Sensor Networks*," in Contemporary Computing (IC3), Seventh International Conference on, pp. 387-393.
39. **Manali D. S., Shrenik N. G., Narendra M. S., (2014)**, "*Lightweight Authentication Protocol Used in Wireless Sensor Network*," in Circuits, Systems, Communication and Information Technology Applications (CSCITA), International Conference on, pp. 138-143.
40. **Selvam R., Senthilkumar A., (2014)**, "*Cryptography Based Secure Multipath Routing Protocols in Wireless Sensor Network: A Survey*," in Electronics and Communication Systems (ICECS), International Conference on, pp. 1-5.
41. **Garima V., Amandeep K., (2014)**, "*A Review on Distributed System Security Using Elliptic Curve Cryptography*," International Journal of Scientific and Research Publications, vol. 4, pp. 1-6.
42. **Elaine B., William B., William B., William P., Miles S., (2012)**, "*NIST Special Publication*," NIST Special Publication, Department of Commerce, United States of America , pp. 37-38.
43. **Rahat A., Suresh C. M., (2011)**, "*A Review on Elliptic Curve Cryptography for Embedded Systems*," International Journal of Computer Science & Information Technology, vol. 3, pp. 84-103.
44. **Pradip K. S., Chhotray R. K., Gunamani J., Sabyasachi P., (2013)**, "*An Implementation of Elliptic Curve Cryptography*," in International Journal of Engineering Research and Technology, vol.2, pp. 1-8.
45. **Reaz M. B. I., Jalil J., Husian H., Hashim F. H., (2011)**, "*FPGA Implementation of Elliptic Curve Cryptography Engine for Personal Communication Systems*," Scientific Research and Essays, vol. 6, pp. 6214-6221.

46. **Shweta L., Monika S., (2013),** "*An Efficient Elliptic Curve Digital Signature Algorithm (ECDSA)*," in Machine Intelligence and Research Advancement (ICMIRA), International Conference on, pp. 179-183.

47. **Lalitha T., Saravana K. R., Hamsaveni R., (2014),** "*Efficient Key Management and Authentication Scheme for Wireless Sensor Networks*," American Journal of Applied Sciences, vol. 11, pp. 969-977.

48. **Min T., Jizhi W., Shujiang X., Yinglong W., (2009),** "*SPA Resistant Algorithms for Elliptic Curve Cryptography over $GF(2^m)$* ," in Information Engineering and Computer Science, ICIECS, International Conference on, pp. 1-4.

49. **John O., Cordelia H., Rex P., (2006),** "*Discrete Mathematics Using a Computer*", Springer-Verlag, London, pp. 6-12.

50. **László L., Balázs S., (2006),** "*Limits of Dense Graph Sequences*," Journal of Combinatorial Theory, vol. 96, pp. 933-957.

51. **Qing Y., Qiaoliang L., Sujun L., (2008),** "*An Efficient Key Management Scheme for Heterogeneous Sensor Networks*," in Wireless Communications, Networking and Mobile Computing, WiCOM'08, 4th International Conference on, pp. 1-4.

52. **Ravi K. K., Sushant K. C., (2013),** "*Hierarchical Key Agreement Protocol for Wireless Sensor Networks*," International Journal on Recent Trends in Engineering & Technology, vol. 9, pp. 25-33.

53. **Kamanashis B., Vallipuram M., Elankayer S., Muhammad U., (2013),** "*An Energy Efficient Clique Based Clustering and Routing Mechanism in Wireless Sensor Networks*," in Wireless Communications and Mobile Computing Conference (IWCMC), 9th International, pp. 171-176.

54. **Zinaida B., Felix C. G., Dogan K., (2004),** "*User Authentication in Sensor Networks*," Jahrestagung der Gesellschaft für Informatik, Workshop on Sensor Networks, Ulm, Germany, pp. 385-389.

55. **Boushra M., Hatem B., Abdelmadjid B., (2008),** "*TLA: A Tow Level Architecture for Key Management in Wireless Sensor Networks*," in Sensor

Technologies and Applications, SENSORCOMM'08, Second International Conference on, pp. 639-644.

56. **Sajid H., Firdous K., Ashraf M., (2007),** "*An Efficient Key Distribution Scheme for Heterogeneous Sensor Networks,*" in Proceedings of the International Conference on Wireless Communications and Mobile Computing, pp. 388-392.

57. **Aso A. M., Kameran A. A., Ahmed. C. S., Yuriy A., (2014),** "*The Enhanced Data Sequence Method for ECC Cryptosystem,*" Applied Mathematical Sciences, vol. 8, pp. 5553-5564.

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: AMEEN, Kameran Ali

Date and Place of Birth: 05 January 1977, Kirkuk

Marital Status: Married

Phone: +90 5535474303

Email: kameranaliameen@gmail.com



EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Çankaya University, Information Technology	2015
B.Sc.	Kirkuk University, Computer Science	2008
High School	Almosala for Boys	1997

FOREIN LANGUAGES

Advanced Arabic, Advanced English, Beginner Turkish

PUBLICATIONS

1. Kameran A. A., *ea tl.*, (2014), "The Enhanced Data Sequence Method for ECC Cryptosystem," Applied Mathematical Sciences, vol. 8, pp. 5553-5564.

HOBBIES

Football, Tourism, Swimming, Books,