

Wireless Sensor Data Security

A thesis submitted in partial fulfilment of the requirements for the degree of

Bachelor of technology

In

Electronics and communication engineering

By

MD NABIL SHAHRIAR

Roll no. 110EC0644

PARTHA SARATHI OJHA

Roll no. 110EC0176

Under the supervision of

Prof. Dr POONAM SINGH



Department of Electronics and Communication Engineering

National Institute of Technology

Rourkela -769008, India

May-2014

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

**NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA -769008, INDIA**

CERTIFICATE

This is to certify that the thesis entitled “**Wireless Sensor Data Security**” submitted by **MD Nabil Shahriar, 110EC0644** and **Partha Sarathi Ojha, 110EC0176** in partial fulfilment of the requirements for the Award of a degree of Bachelor of Technology in Electronics and Communication Engineering during session 2013-2014 at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any degree or diploma. In my opinion, the thesis is of the standard required for the award of a Bachelor of Technology degree in Electronics and Communication Engineering.

Prof. Dr Poonam Singh
(Supervisor)

Prof. Shrishailayya M. Hiremath
(Co-supervisor)

ACKNOWLEDGEMENT

We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this Project. Many thanks go to our supervisor, **Prof. Dr Poonam Singh**, who has given her full effort in guiding us in achieving the goal as well as her encouragement to maintain our progress in track.

We convey our special gratitude & acknowledge the valuable suggestions given by our co-supervisor **Prof. Shrishailayya M. Hiremath** as well as the panel, especially in our project presentation that has improved our presentation skills by their comments and tips.

A special thanks to **Mr. Mohd Suleman** and **Mr. Subhajit Sahu**, whose help, stimulating suggestions and encouragement, helped us to coordinate my project.

We would also like to acknowledge with much appreciation the crucial role of the staff and all **M.Tech students** in Mobile Communication Lab, for their regular suggestions and encouragements during our entire work.

Finally, but by no means least, thanks go to our **Mom, Dad** for their unbelievable support and encouragements. They are the most important people in our world and we dedicate this thesis to them.

Partha Sarathi Ojha

MD Nabil Shahriar

ABSTARCT

Wireless Sensor Network (WSNs) is a network of sensors deployed in places unsuitable for human beings and where constant monitoring is required. They work with low power, low cost smart devices having limited computing resources. They have a crucial role to play in battle surveillance, border control and infrastructure protection. Keeping in view the precious data they transmit, their security from active or passive attacks is very crucial. We came to know about LOCK model implementing novel Distributed Key Management Exclusion Basis (EBS) System is very efficient in providing with Network Security. Keeping in view the importance of Data Security we preferred to secure WSN data through Public Key Encryption methods like RSA. We also discussed and implemented Elliptic Curve Cryptography (ECC) and its advantages over RSA.

However our novel Spiral Encryption Technique implemented along with ECC algorithm, has shown how it helped in making the transmitted message more secure and less informative for the eavesdropper.

Key words: Wireless Sensor Network, Exclusion Basis System, LOCK, encryption, authentication, Public Key Encryption, RSA, Elliptic Curves, Galois Field, Modular Arithmetic

Content

CERTIFICATE	2
ACKNOWLEDGEMENT	3
ABSTRACT	4
LIST OF FIGURES	7
LIST OF TABLES	8
1. INTRODUCTION	9
2. SECURITY THREAT AND ATTACKS	11
2.1 Security	11
2.2 Threats	12
2.3 Attacks	12
3. NETWORK SECURITY	15
3.1 Key Management	15
3.2 Key Management Scheme in Sensor Network	16
3.2.1 Static Key Management	16
3.2.2 Dynamic Key Management	17
3.3 Sensor Network Model	17
3.4 Collusion Problem	18
3.5 Exclusion Basis System	18
3.6 Localized Combinatorial Keying	19
4. DATA ENCRYPTION	21
4.1 Software Based Encryption	21
4.1.1 Public Key Encryption	22
5. ELLIPTIC CURVE CRYPTOGRAPHY	25
5.1 Necessity and Advantages	26

5.2	Modular Arithmetic	27
5.3	Elliptic Curve Groups Over Real Numbers	27
5.4	Basic Mathematics Review	29
5.4.1	Point Addition in Elliptic Curve	31
5.4.2	Curve Addition in Elliptic Curve	31
5.5	Menezes-Vanstone Elliptic Curve Cryptosystem Algorithm	34
6.	WORK DONE	35
6.1	Spiral and De-spiral Algorithm	35
6.2	Overall Algorithm of Elliptic Curve Cryptography, Spiral and De-spiral combination	36
6.3	Result	37
6.3.1	Spiral Encryption	37
6.3.2	ECC Encryption	38
6.4	Result Analysis and Discussion	43
	CONCLUSION AND FUTURE WORK	44
	REFERENCE	45

LIST OF FIGURES

Figure 3.1: Key distribution to a node--k keys-- out of $P=k+m$ pool of keys

Figure 3.2: Clustered Sensor Network

Figure 3.3: LOCK Model Overview

Figure 4.1: Asymmetric key encryption

Figure 4.2: Eavesdropping

Figure 5.1: From [15]

Figure 5.2: $y^2 = x^3 - 4x + 0.67$

Figure 5.3: GF for $n>1$

Figure 5.4: GF for $n=1$

Figure 5.5: Point Addition

Figure 5.6: Point Doubling

Figure 6.1 Spiral Traversal

Figure 6.2 Overall combined ECC and Spiral algorithm

Figure 6.3 Spiral Encoding

Figure 6.4 ECC Co-ordinate

Figure 6.5: ECC Encryption

Figure 6.6: Sequence of user data being Encrypted Decrypted

Figure 6.7: Encrypted message sequence c1 component in "Twisted" form

Figure 6.8 Encrypted Message Sequence C2 in "Twisted" form

LIST OF TABLES

Table 5.1: Public key sizes for AES followed by the guidelines of NIST [15]

Chapter 1

1. Introduction

A cooperative network, collection of systematized nodes is known as wireless sensor network. Spatially distributed autonomous sensors are the main tool for the WSN which are used radically in the field of physical/environmental conditions monitoring i.e; pressure, temperature, sound etc. as well as to pass their data cooperatively to a main location by using the network. Each single node has its own processing capability (like a microcontrollers/CPU/DSP chips), consists of multiple types of memory (e.g., program/data/flash), have a RF transceiver, a power source (e.g., batteries/solar cells). As we're living in the 20th century, WSNs are on the verge of an accelerated mass scale deployment pace. That day is not far when everyone can access a wireless sensor network through internet worldwide. This can make Internet a giant corporeal network. With unlimited scope of abundant application, areas like environmental (e.g., earthquake monitoring, ocean & wildlife monitoring), medical, entertainment, military, smart spaces, crisis management, homeland defense, and transportation this new technology offers a new dimension to our evolving technologies. An even broader range of upcoming applications is expected to breakthrough in the areas of supervision of pollution, forest fires, potable water quality, and even human heart rates too.

Owing to the fact that a wireless sensor network is a real-time system, new solutions are need of the hour to suffice the fact that very little prior work can be implemented in the system. The prime reason is actually the set of initial conditions needed for implementing previous work has changed drastically. Lately in the previous studies it's been assumed that most scattered systems are wired with unlimited power source hence are not real-time. However the WSNs systems are real-time and wireless therefore there is limited power. They have dynamically changing sets of resources by making use of sensors & actuators as

interfaces. Utilization of resource constraint devices which places a strain for the effective training/ machine learning.

Chapter 2

Security, Threat and Attacks

2.1 Security

The security services in Wireless Sensor Networks aims to protect the information and resources from all kinds of attacks and misbehaviour. And it requires:

- a) *Authorization*: It ensures the information is provided only by the nodes which are authorized to the network services.
- b) *Secrecy*: It ensures that the delivered message is only understood by its recipient, not by anyone else.
- c) *Availability*: It ensures that the availability of the desired network even in presence of denial of service attacks.
- d) *Authentication*: It ensures the genuinity of the communication from one node to another, ensuring that a malicious node cannot camouflage itself as a trusted node.
- e) *Integrity*: Integrity is ensured by not letting the message modified by any malicious node while sending it from one node to another.
- f) *Non-repudiation*: It ensures that the denial of sending previously sent message is not possible for a node.
- g) *Robustness*: It denotes that the whole network is not compromised when few are under attack.

2.2 Threats

Following categories are the classification of threats in Wireless Sensor Network (by Karlof et. al. [2]):

- a) *Bit-class vs. High end processor-class attacks:* In bit-class attacks, WSN is confronted by the intruder by using a few nodes with similar capabilities as that of network nodes. In High end processor-class attacks, considerably more damage is possible by an intruder by the use of high end processors, etc. to a network than a mischievous sensor node.
- b) *Stranger vs. insider attacks:* A stranger enemy has no admittance to most cryptographic resources in sensor network. Stranger attacks occurs by the nodes which are not a part of WSN. The insider attacks arise to the legitimate nodes of a WSN and then they behave in inadvertent or illicit ways.
- c) *Passive vs. active attacks:* Spying/observing of packets switched within a WSN is Passive in nature; the active attacks encompass some adjustments of the data flow or the making of a false inlet in a WSN.

2.3 Attacks

Invasive & Non-invasive are the two categories of attacks in wireless sensor networks. Side channel attacks (i.e.; power/timing/frequency) are caused due to Non-invasive attacks. On the other hand Invasive outbreaks are plentiful more common and the more significant of these are described in the subsequent sections.

Numerous outbreaks on sensor networks are enumerated as follows:

A) *Node Duplication Attack:*

In node duplication attack an enemy inserts a different node into a network which has been replicated after a present node. This replicated node can behave exactly like the old node or it can have some extra behaviour, such as conveying data of interest straight to the invader.

B) *Routing attack*

- **Discerning forwarding:** In discerning forwarding it impacts the net traffic by making the sensor network believe that all the contributing nodes in net are dependable to forward the message.
- **Sinkhole attacks:** In sinkhole attacks, enemy draws the traffic to a apprehended hub. Modest way of forming sinkhole is to place a spy hub where it can draw most of the traffic, perhaps nearer to the base station or spy node itself deceiving as a base station.
- **Sybil attacks:** In Sybil attack, multiple identities are presented by a single node to all other nodes in the WSN by which the other nodes may mislead. The efficacy of fault lenient structures are reduced by this attack. Topographical routing is additional malevolent factor is that a Sybil node can appear at more than one place concurrently.
- **Wormholes:** In wormhole attacks, an attacker tunnels messages over a low latency link by positioning an enemy nearer to base station and fully disrupts the traffic. It convinces the nodes that they are nearer to the base station which are truly multi hop away. Which constructs a sinkhole because enemy on the further side of the sinkhole offers a superior path to the base station.
- **Flooding:** Earlier, the malicious node can root a colossal traffic of unserviceable mails on the network. This is acknowledged as the flooding.

C) *Denial-of-Service (DoS) attack:*

Here the target machine is rendered and inaccessible by the legitimate users caused by the hacker. It's divided into two parts.

- **Active attack:** Malevolent nodes injure other nodes by initiating network outage by segregating while saving battery life is not a priority.
- **Passive attack:** Here selfish nodes practise the network but do not collaborate. They save battery lifespan in lieu of their own transport network, they don't aim to damage other nodes directly. They are like parasites.

D) Information in Transit Attack

Data transition period is prone to eavesdropping, modification, injection. By entrenched verification, reliability, confidentiality & repeat protocols these can be prevented. Tightly targeted attacks can be occurred by the attacker by mapping the routing layout of a network, enabling disruption into the chosen portions of a network for maximum damage.

Chapter 3

NETWORK SECURITY

The data is communicated, to forwarding nodes (or gateways), using wireless links. After further processing the data it communicates with the outer world over the base stations also known as command nodes. Access points to the network are provided by these Base Stations. Here user requests are initiated and network replies are acknowledged. Normally, gateways and base stations are high-end node. But it should be known that same or different node can perform the work of gateway and base station.

3.1 KEY MANAGEMENT

A key may be a symmetric bivariate polynomial or 128 bit string. Multitude of keys are needed to be coped for encryption and authentication of sensitive data exchanged. For authentication and encryption a multitude of keys might have to be coped then. The goal of key managing is to form protected communication among collaborative parties.

For secure and efficient redistribution and for secure channel communication keys generation, administrative keys are used. For securing direct/indirect communication between two nodes Communication keys are used and to maintain privacy and flexibility to attacks, or catastrophes. Network keys (both administrative and communication keys) are changed (re-keyed). Eschenauer and Gligor [3] introduced all contemporary seminal random key predistribution scheme. In order to enhance widespread applicability and robustness of the network the future of that scheme is motivated by the use of deployment knowledge [4] and key polynomials [5] known by the name of Static Key Managing.

Another outline based on distinctiveness based symmetric keying is proposed by Jolly et al. [6]. Dynamic schemes help to reach flexibility to attack in long-lasting networks and especially highlight group communication keys. Dynamic key management is preferred as a security scheme because of longevity of the network & rekeying facility, meant for the WSN's.

3.2 KEY MANAGEMENT SCHEME IN SENSOR NETWORK

3.2.1 Static Key Management

As the name suggests as soon as administrative keys are pre-deployed into the hubs, they will not be altered. They are assigned to hubs whichever arbitrarily or based on some prior deployment information. The concept of overlapping of administrative keys is used by Most static schemes to govern the suitability of neighbouring hubs to create a straight pair-wise communication key.

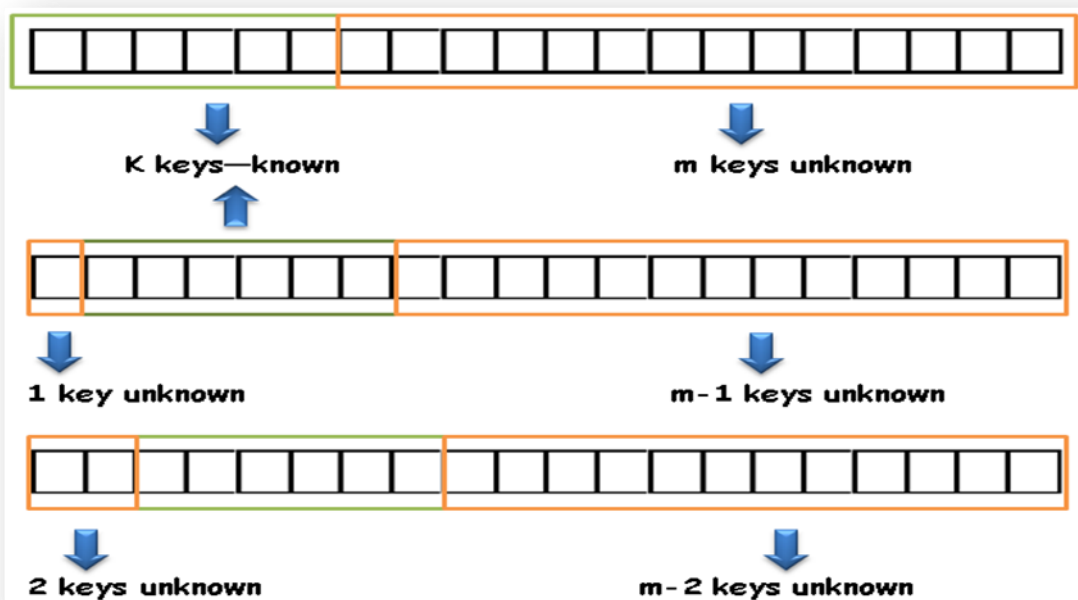


Figure 3.1: Key distribution to a node-- k keys-- out of $P = k + m$ pool of keys

3.2.2 Dynamic Key Management

Dynamic key management schemes periodically modifies administrative keys, on request or hub apprehension. The enhanced network survivability is the major benefit of dynamic keying. In Static keying the likelihood of network apprehension upsurges upon adding new hubs which is solved by the use of exclusion-based systems (EBSs).

3.3 SENSOR NETWORK MODEL

A clusters of sensors is formed on the basis of their capabilities & communication range. And it's organised by a cluster group head or gateway. It broadcasts messages to all cluster sensors. Because of the stationary nature of sensor & gateway hubs; their deployed position and message array of all hubs in the network are recognised. Each gateway can reach to all the other cluster sensors either directly or in multi-hop. Sensors perform by relaying & sensing. In case of further than single hop away from the gateway, hubs send their data through relaying; on the other hand sensing is accountable for piercing their surroundings to follow a target/ event & then relayed to the gateway. This model's architecture is illustrated in Fig. 3.2.

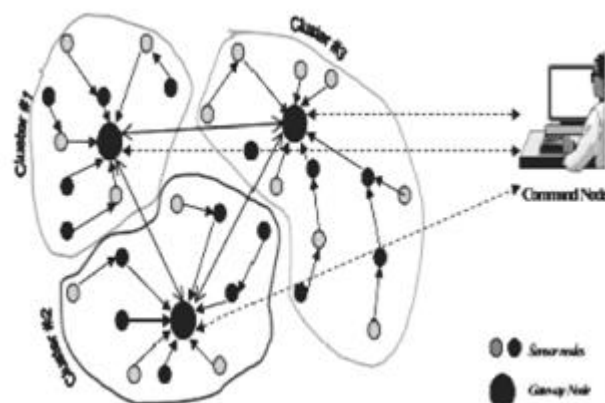


Figure 3.2: Clustered Sensor Network

3.4 COLLUSION PROBLEM

The security outline offered in [8] is founded on the EBS to discourse the collusion problem that implements position centred key consignment. The number of keys revealed via capturing collocated hubs can be minimized by this. In [8] the offered network prototype is alike to this model established by clusters and gateways. EBS structure is used to execute rekeying inside each cluster. Here gateways circulate keys to the hubs.

3.5 EXCLUSION BASIS SYSTEM (EBS)

EBS comprised of a combinatorial construction of the group key managing issue. Here each node is allocated k keys out of a pool of size P ($P = k + m$) keys. When nodes get captured, rekeying takes place but even when it's not captured then also it takes place periodically. Auxiliary keys are created & encoded via all the 'm' keys which are anonymous to the apprehended nodes, and lastly circulated to further nodes that mutually know the m keys. Small number of nodes may collude and all the network keys can collectively be revealed which is a drawback of the basic EBS. This happens when the m value is selected to be relatively small. It is conceivable that few compromised nodes can collude and reveal all the keys active in the network to an enemy. In that case that enemy will be able to reveal entirely encoded communications in the system. SHELL [6] utilizes the physical propinquity of nodes so that a hub would part maximum keys with nearby hubs. To elude the consignment of matching key combinations, key swap of is engaged.

Hence, it can be perceived that if numerous nodes conspire, it is probable to disclose all active keys. In demand of discoursing the collusion problem in [8], a proficient dynamic key management scheme has been offered by researchers.

3.6 LOCALIZED COMBINATORIAL KEYING (LOCK)

For the clustered sensor networks LOCK is an EBS-based dynamic key management scheme. The corporal network prototype shown as a three-tier WSN followed by cluster leader nodes (CLs) with the base station (BS) at the top & then consistent sensor nodes. In this scheme pre-deployment information cannot be presumed about the estimated positions of all the nodes. LOCK uses EBS administrative keys consisted of two layers. The upper layer is EBSb which empowers the base station for managing the cluster leaders as a set. The lower layer (level 0) comprises an EBSCi (Each cluster Ci). A cluster leader, Ci, is a member in both the upper EBSb as well as the lower EBSCi.

The EBSb administrative keys are used to produce (and refresh) group session keys used by the BS to converse with the CLs. whereas the administrative keys of every EBSCi are used to refresh & create) cluster session keys. As CL is being considered a regular member in its EBS; knows only those cluster administrative keys which the other node knows in the similar cluster.

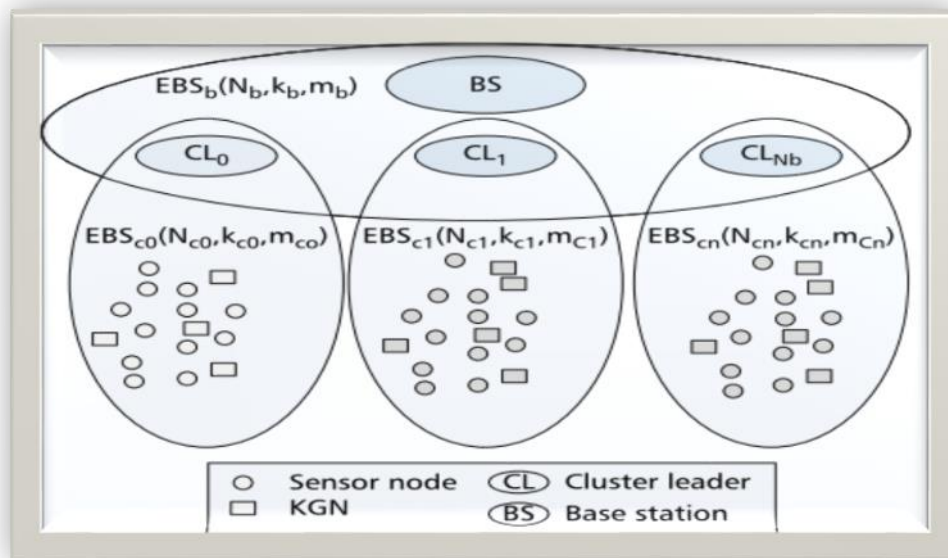


Figure 3.3: LOCK Model Overview

Accordingly, it's been assumed that if CL is captured, it will not provide any additional cluster keys than the captured ones to the attacker.

A backup key set (common with the base station and anonymous to their (or any further) cluster leader) is established by the sensor nodes of each cluster during the initialization phase. Key generation of EBSC keys is accomplished by a set of sensor nodes inside the cluster entitled as key generation nodes (KGNs), CL usually selects those. Key distribution is performed by the CL.

The likelihood of sharing a key polynomial between any two arbitrarily chosen nodes, P_s , is defined as follows:

$$P_s = \begin{cases} 1 & \text{if } k > m, \\ \left(1 - \prod_{i=0}^{k-1} \frac{m-i}{k+m-1} \right) & \text{if } k \leq m, \end{cases}$$

Where k is the number of polynomials known to a node and m is the number of polynomials unknown to that node.

Chapter 4

DATA ENCRYPTION

The alteration of data into a form which is practically meaningless and cannot be figured out without the use of some key. This transformed data is known as *cipher text*. Decryption is its reverse process i.e.; retrieval of original data (secret message) by use of another key. Encryption standard applications include secure storage of secrets in the keychain and in the creation of certificates and digital signatures

In Wireless Sensor Network; data security can be provided in two ways. One is Hardware based encryption and another is Software based encryption. But if we try to compare these two it happens that the security provided in the Wireless Sensor Network by only software based encryption is increased by few percent while combining both software and hardware encryption. And moreover in hardware encryption the transmitted data packets remains the same size. In this chapter we will describe the Software based encryption system.

4.1 SOFTWARE BASED ENCRYPTION SYSTEM

In WSN the data which is transmitted through a wireless channel can be easily fetched by an intruder. Basically in software based encryption system what happens that it encrypts the message by using keys which is only known to the receiver and then sends it over the wireless medium. But recently few new techniques evolved in which while encrypting the data, two sets of keys are used. One is Public Key and another is Private Key. This encryption system is called Public Encryption System.

4.1.1 Public Key Encryption System (PKE)

Encryption used in wireless security, is broadly classified into symmetric key encryption and asymmetric key encryption. PKE is a perfect example of asymmetric encryption [9].

In asymmetric key cryptography, for encrypting and decrypting a message different set of keys are used. In this one key is made public while the other is kept private. This arrangement is basically known as *Public Key Cryptography*. Main advantages over symmetric encryption are: The need for generating and assigning n number of secret keys for n people is eliminated and both authentication as well as cryptography can be implemented using this algorithm.

The first public key algorithm that went viral is known as *RSA encryption*. It is still the most popular method of encryption that is based on mathematical manipulation of two large prime numbers and their product. Its strength is supposed to be associated with the difficulty of factoring a very large prime number. The assortment of long-enough prime numbers by the recent, predictable speed of current digital computers for the generation of the RSA keys has the potential to make this algorithm secure indefinitely. However, there is no mathematical proof of this belief i.e.; breaking RSA encryption might be possible by a computationally fast factorization algorithm. With the current advancements and researches in the field of quantum computers factoring a large prime number will be a trivial issue.

Based on a different mathematical model, other public key algorithms equivalent to **RSA** in terms of complexity is **Elliptic Curve Encryption**. It is similar in use to RSA encryption method (though they have altogether a different mathematics behind), and will be discussed elaborately ahead.

Let us take an example to see how a public key algorithms (here it is RSA) handles the problem of key distribution, assuming that Maria wants to avail a safe and secure communication from Bob. This technique is demonstrated below in Figure 4.1.

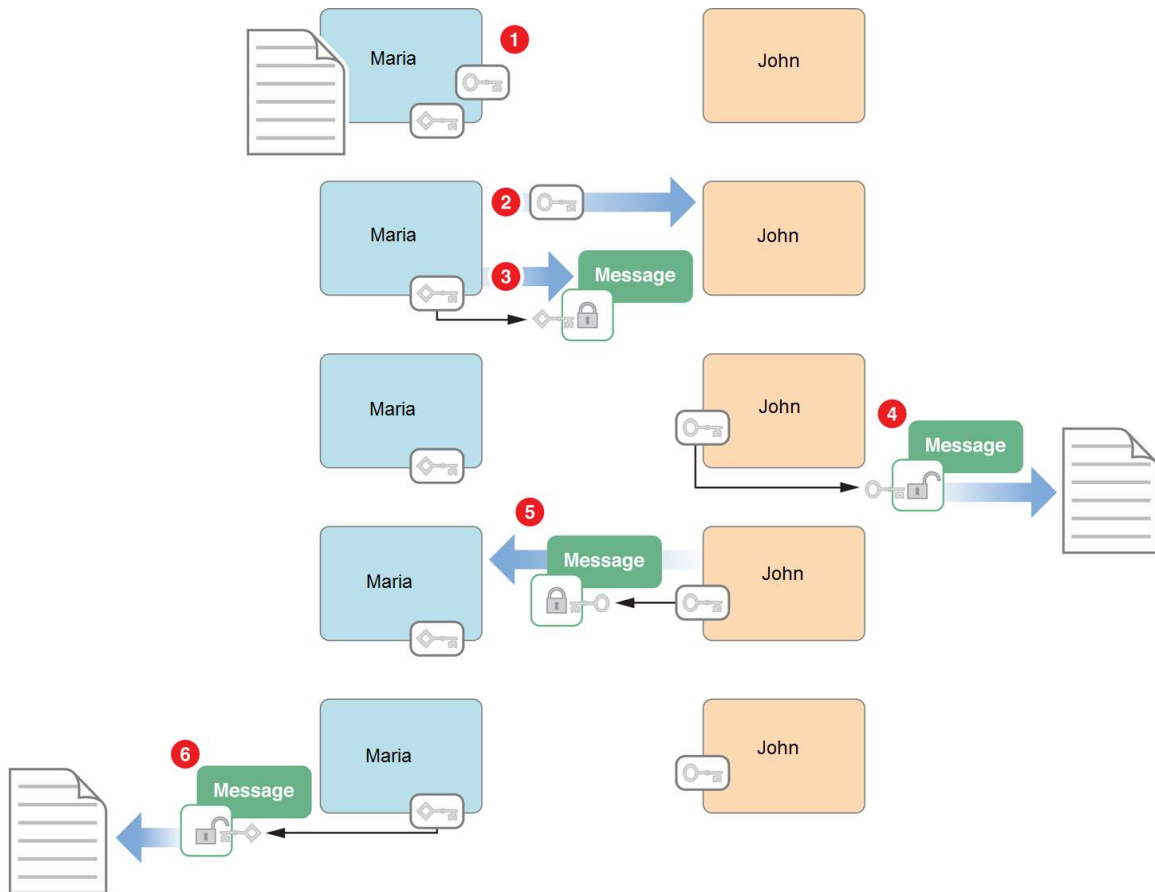


Figure 4.1: Implementation of asymmetric key encryption

The protected message exchange demonstrated in Figure 4.1 is described here:

1. Maria using public key algorithm generates two encryption key; a reserved key, which she has to keep undisclosed, and a public key. Meanwhile Maria has a message for John.
2. Now Maria sends her public key to John. As estimating private key from public key is next to impossible, sending public key won't compromise her private key.
3. Maria now proves her uniqueness to John (a process known as *authentication*). Only thing she has to do is to encrypt entire or a part of the message using her private key and transmit to John.

4. Now John decrypts the message with Maria's public key. As the message was encrypted by private key of Maria, it must be decrypted back to the original message only with the use her Public key which proves her authenticity.
5. John encrypts his message using Maria's public key and sends it to Maria. Since only Maria has the private key to decrypt the message, it remains hidden to eavesdropping.
6. Maria decrypts the message with her private key.

Issues concerning national security as well as protection of Intellectual properties of corporates, heavily rely on data encryption and authentication. For this a crew of extremely smart people are assigned the task of creating, testing and breaking secure systems exploring and fixing each and every possibility of loopholes. From this it is very easy to conclude the huge complexity of actual real time secure communication and authentication

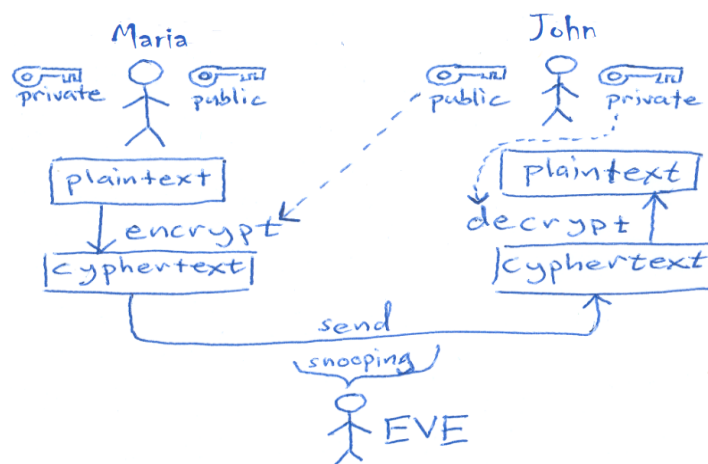


Figure 4.2: Eavesdropping

Encrypting the message with the private key for authentication can be meddled by a **man-in-the-middle attack**, in which someone (Eve) with mischievous intent (Figure 4.2) replaces intercepted Maria's original message with her own, so as to mislead John to use Eve's public key. Eve then re-encrypts each of Maria's messages after decrypting it with Maria's public key by altering it with her own private key and. Then John decrypts it with Eve's public key under the perception that that the keys came from Maria.

Chapter 5

ELLIPTIC CURVE CRYPTOGRAPHY

It was proposed by Victor Miller [10] and Neal Koblitz [11] individually in 1985. ECC provided an efficient alternative to other public-key encryption algorithms, such as RSA. Public key based all ECC schemes, implement the discrete log problem for elliptic curves which are quite difficult to solve. Presently, many standardizing bodies such as ANSI, IEEE [12], ISO and NIST [13] have adopted ECC and accepted commercially. Now it has been a trend that conventional public key cryptographic systems are gradually being replaced with ECC systems. With evolving computational power, the key size of the conventional systems also has to show similar dramatic increase.

Elliptic curves have been studied by mathematicians for over a hundred years. They have been deployed in diverse areas.

Number theory: Proves Fermat's Last Theorem in 1995 [16]. The equation has no nonzero integer solutions for $x^n + y^n = z^n$ when the integer n is greater than 2.

Modern physics: String theory: The notion of a point-like particle is replaced by a curve-like string.

Elliptic Curve Cryptography: An efficient public key cryptographic system.

5.1 Necessity and Advantages

For an equivalent amount of security ECC systems is at par with RSA. It leads to less important operations, lesser transistors for hardware implementation and ,less time consuming Encryption. For example: 155-bit ECC uses 11,000 transistors while a 512-bit RSA implementation uses 50,000[14].

Although same level of security, data sizes, encrypted message sizes and computational power are present in both ECC and RSA, former has smaller keys (Table 5.1 and Fig 5.1) and less bandwidth requirements than the other cryptographic algorithms (RSA).

ECC KEY SIZE (Bits)	RSA KEY SIZE (Bits)	KEY SIZE RATIO	AES KEY SIZE (Bits)
163	1024	1:6	
256	3072	1:12	128
384	7680	1:20	192
512	15360	1:30	256

Table 5.1: Public key sizes for AES followed by the guidelines of NIST [15]

RSA vs. ECC

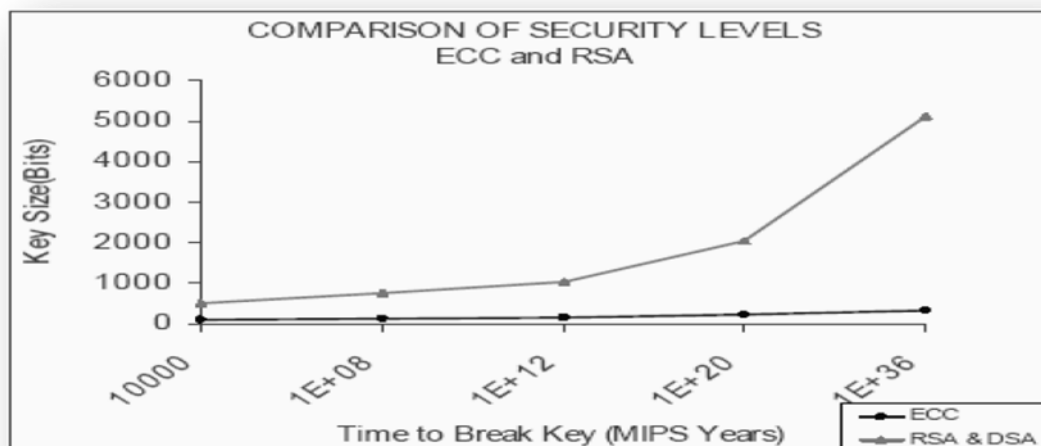


Figure 5.1: From [15]

5.2 MODULAR ARITHMETIC

Simple operations in Modular Arithmetic

$$5 + 7 \bmod 5 = 12 \bmod 5 = 7 \bmod 5$$

To first reduce the operands moduli operator can be applied individually..

$$7 + 18 \bmod 6 = 1 + 0 \bmod 6 = 25 \bmod 6$$

Multiplication operation can also be performed like this

$$3 * 5 \bmod 13 = 15 \bmod 13 = 28 \bmod 13$$

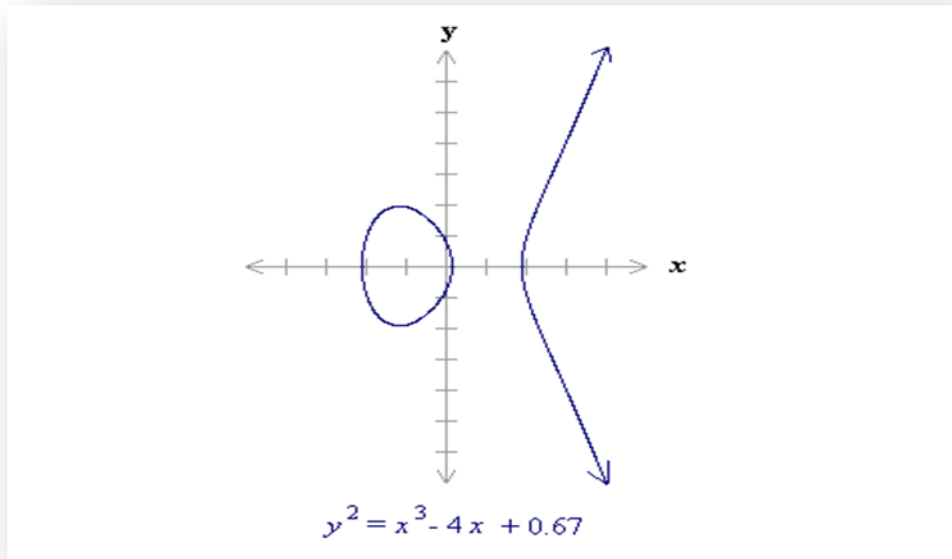
5.3 ELLIPTIC CURVE GROUPS OVER REAL NUMBERS

Over real numbers an elliptic curve is set of points (x, y) that satisfy an elliptic curve equation of the form:

$$y^2 = x^3 + ax + b, \text{ where } x, y, a \text{ and } b \text{ are real numbers.}$$

Each choice of the numbers a and b yields a different elliptic curve. For example, $a = -4$ and $b = 0.67$ gives the elliptic curve with equation $y^2 = x^3 - 4x + 0.67$; the graph of this curve is shown below: (Fig 5.2) Remember that an Elliptic curve is not an Ellipse.

Elliptic curve $y^2 = x^3 + ax + b$ can be used to form a group if it does not have any repeated roots. In other words $4a^3 + 27b^2$ should not be zero. An elliptic curve group over real numbers consists of the points present in itself as well as a special point O called the point at infinity.

Figure 5.2: $y^2 = x^3 - 4x + 0.67$

The curves are defined in their standard Weierstrass form $E: y^2 = x^3 + ax + b$ and are defined over a finite field F_p , where $p > 3$ and is prime and $a, b \in F_p$. A large prime-order subgroup of the group $E(F_p)$ of F_p -rational points on the curve E is employed in the protocol. All solutions $(x, y) \in (F_p)^2$ to the curve equation along with a point at infinity, the neutral element is contained in this group of rational points. The number of F_p -rational points is being denoted by $\#E(F_p)$ and the prime order of the subgroup by n . A fixed generator of the cyclic subgroup that may be selected at random is called as base point and denoted by $G \in E(F_p)$.

5.4 BASIC MATHEMATICS REVIEW

Field: A collection of numbers on which addition and multiplication are defined, and follow these rules [17] is known as a field :

Additive Commutativity: $C + D = D + C$

Multiplicative Commutativity: $C * D = D * C$

Additive Associativity: $C + (E + D) = (C + E) + D$

Multiplicative Associativity: $C * (E * D) = (C * E) * D$

Distributive: $C * (E + D) = (C * E) + (C * D)$

Additive Identity: $C + 0 = C$

Multiplicative Identity: $C * 1 = C$

Additive Negation: $C - C = 0$

Multiplicative Inversion: $C / C = 1$ (C has to be nonzero)

Galois Field: Galois fields exist only of size p^n , for a prime no. p and natural no. n

- When n is equal to 1 (i.e. a field of size that of prime number), (Fig 5.4) and modular arithmetic over p
- When n greater than 1 (Fig 5.3) modular arithmetic not possible.[18]

GF(2²) or GF(4)

		0	1	2	3			0	1	2	3			0	1	2	3	
	+							*						/				
0		0	1	2	3	0		0	0	0	0	0		0	.	0	0	0
1		1	0	3	2	1		0	1	2	3	1		1	.	1	3	2
2		2	3	0	1	2		0	2	3	1	2		2	.	2	1	3
3		3	2	1	0	3		0	3	1	2	3		3	.	3	2	1

Figure 5.3: GF for n>1

- For every $A \neq 0, A \in M$ there exists an element $A^{-1} \in M$ such that $A^{-1} \times A = A \times A^{-1} = 1$.

A finite (or Galois) field has a finite set of elements. In the field F^2 can numbers be represented by $\{0, 1\}$ and numbers in F_{2^n} can be represented as n-bit binary numbers.

5.4.1 Point Addition Operation in Elliptic Curve

Let Z be the points on an elliptic curve. Let it be defined over the field F_2 , with the addition of the point O_E

- Lines at infinity will intersect Z at O_E thrice.
- Vertical lines will intersect Z twice, and at O_E
- All lines in F_2 intersect Z in three places.

. Addition occurs as follows [18]. Let C, D be in Z .

- First, make sure a line is drawn between C and D .
- Where C and D intersect Z for the third time, draw another vertical line.
- $C + D$ is the point where the above vertical line will intersect Z a second time

5.4.2 Curve Addition in Elliptic Curve

Consider two points $P(-2.35, -1.86)$ and $Q(-0.1, 0.83)$ on the Elliptic Curve as shown below. (Fig 5.5) Any straight line through these points will surely pass through the third point on the curve $-R(3.89, 5.62)$.

Take the reflection of the point about y axis to get the resultant point $R(3.89, -5.62)$. This operation is called **Point Addition**.

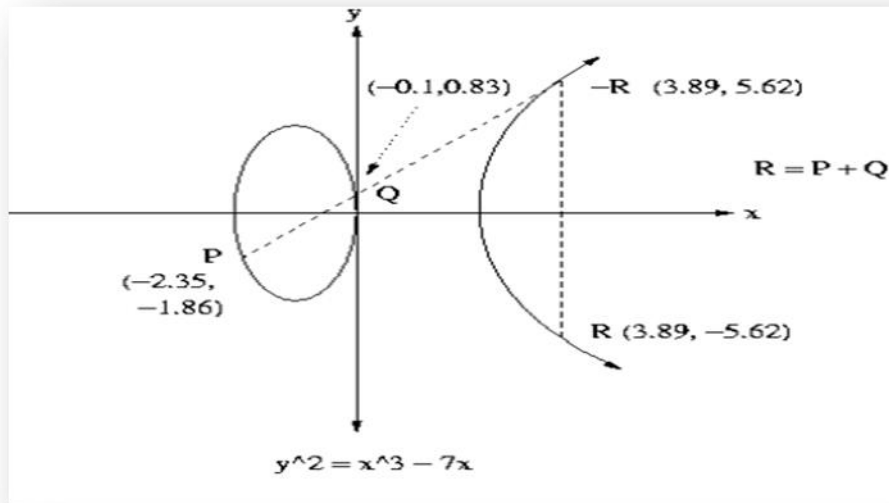


Figure 5.5: Point Addition

When P and Q converges to a single point we get a tangent to the curve and after taking reflection of the intersected point along y axis we get the final point. This operation is **Point Doubling** as shown below (Fig 5.6).

Arithmetic for this is given as follows

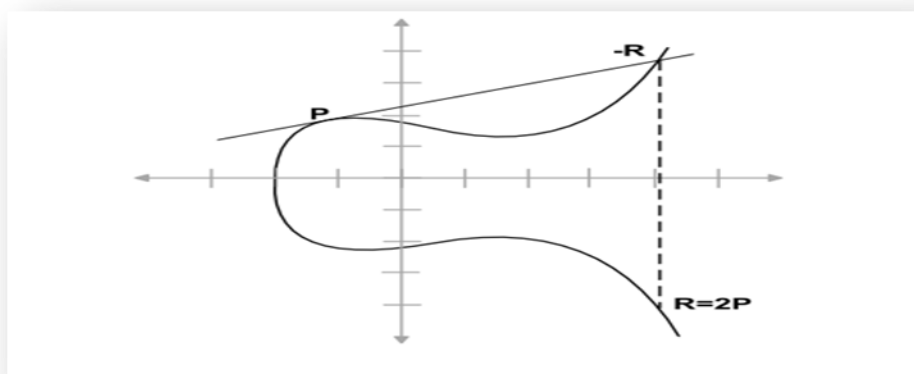


Figure 5.6: Point Doubling

For addition the generic algorithm [20] is:

Given $E : y^2 = x^3 + ax + b$, $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$, both on E

$$P_1 + P_2 = \begin{cases} O_E & \text{if } x_1 = x_2, \quad y_1 = -y_2 \\ (x_3, y_3) & \text{otherwise} \end{cases}$$

Where

$$(x_3, y_3) = (\gamma^2 - x_1 - x_2, \quad \gamma(x_1 - x_3) - y_1)$$

And

$$\gamma = \begin{cases} \frac{3x_1^2 + a}{2y_1} & \text{if } P_1 = P_2 \\ \frac{y_2 - y_1}{x_2 - x_1} & \text{otherwise} \end{cases}$$

Scalar Multiplication is defined as a process of recursive additions.

- Given an Elliptic Curve Z , a point M in Z , and scalar t .
- $tM = M + M + M + \dots$ t times.

Simplified steps for the above operation is:

- Double
- Add P

This scalar multiplication is the strength of ECC. As shown above, while computing tM means we have to add point M exactly $t-1$ times to itself. This would result in another point Q on the elliptic curve. The reverse process / the inverse operation, of recovering k when the points P and $Q = tM$ are given, is known as the Elliptic Curve Discrete Logarithm Problem (ECDLP) which is practically very difficult for the unauthenticated intruder.

5.5 Implementation of Menezes-Vanstone Elliptic Curve Cryptosystem

Maria has a message M to be sent to John which is split into chunks (m_1, m_2) . Following is a convention which both follow: ---

- p – Any prime number greater than 3 preferably large
- F_p – A Galois field of size p
- E – $y^2 = x^3 + ax + b$ (a, b in F_p) ---- an elliptic curve in field F
- P – Base Point(A randomly selected point on E) to generate subgroup H
- H – A subgroup of E of the same size as of E

Public Key: John's public key. Actually it is circulated to the whole world.

- β : John's public key is computed as $\beta = aP$. It is a point in H .

Secret: Maria also has a secret not to be shared with any one.

- k : An arbitrarily selected number by Maria which varies each time a message is sent.

Private Key: John's private key. It is secure with him. Nobody else does know it.

- a : John's private key is a arbitrarily selected natural number.

Encryption: Maria has a secret m . Before encryption she separates it into m_1 and m_2 (message splitting)

- Maria computes $(y_1, y_2) = k\beta$
- Maria computes $c_0 = kP$ ← Note that c_0 is a point.
- Maria computes $c_1 = y_1 m_1 \text{ mod } p$.
- Maria computes $c_2 = y_2 m_2 \text{ mod } p$.
- Maria transmits encrypted message $c = (c_0, c_1, c_2)$ to John.

Owing to the message overhead present memory requirement of c is double the original message m .

Decryption: John desires to retrieve the message m from c .

- John computes $a c_0 = (y_1, y_2)$
- John recovers message m by computing $m = (c_1 y_1^{-1} \text{ mod } p, c_2 y_2^{-1} \text{ mod } p)$

6.2 OVERALL ALGORITHM OF ELLIPTIC CURVE CRYPTOGRAPHY, SPIRAL AND DE-SPIRAL COMBINED

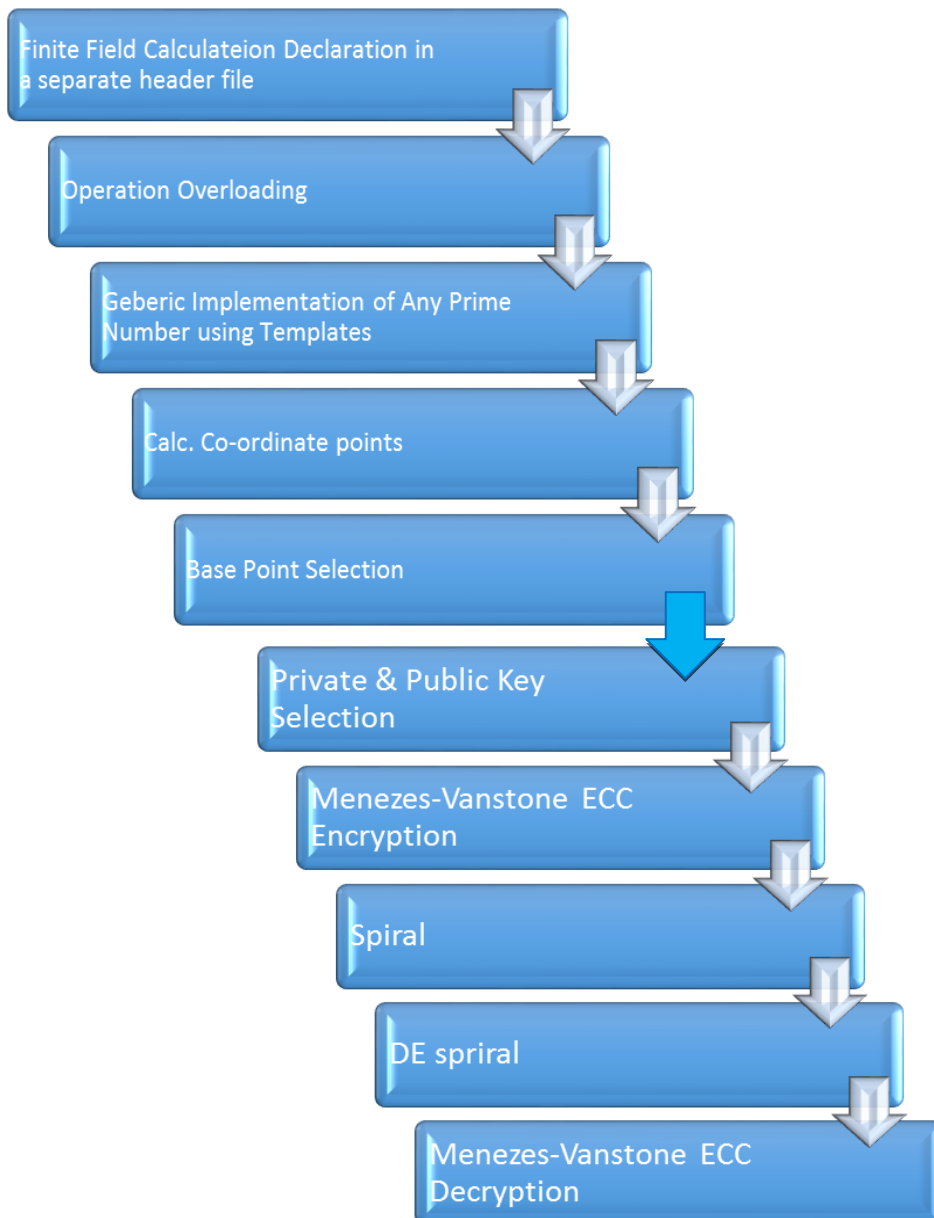


Figure 6.2 Overall combined ECC and Spiral algorithm

6.3 RESULT

6.3.1 Spiral Encryption

```

"C:\Users\PARTHA\Documents\c progs\pro_code.exe"
-----
ORIGINAL SEQUENCE
1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  2
3  24  25  26  27  28  29  30
-----
values in 2D array-----
1 6 11 16 21
2 7 12 17 22
3 8 13 18 23
4 9 14 19 24
5 10 15 20 25

Rest values for 1D array-----
26 27 28 29 30
ENCRYPTED DATA
-----
5  1  6  11  16  21  22  23  24  25  20  15  10  5  4  3  2  7  12  17  18  19
14  9  8  13  26  27  28  29  30
-----
DECRYPTED DATA
-----
1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  2
3  24  25  26  27  28  29  30
-----

```

Figure 6.3 Spiral Encoding

6.3.2 ECC Encryption

```

"C:\Users\PARTHA\Documents\c progs\ecc.exe"
A little Elliptic Curve cryptography example
by Jarl Ostensen, 2007

The elliptic curve:  $y^2 \text{ mod } 263 = (x^3 + 1x + 1) \text{ mod } 263$ 

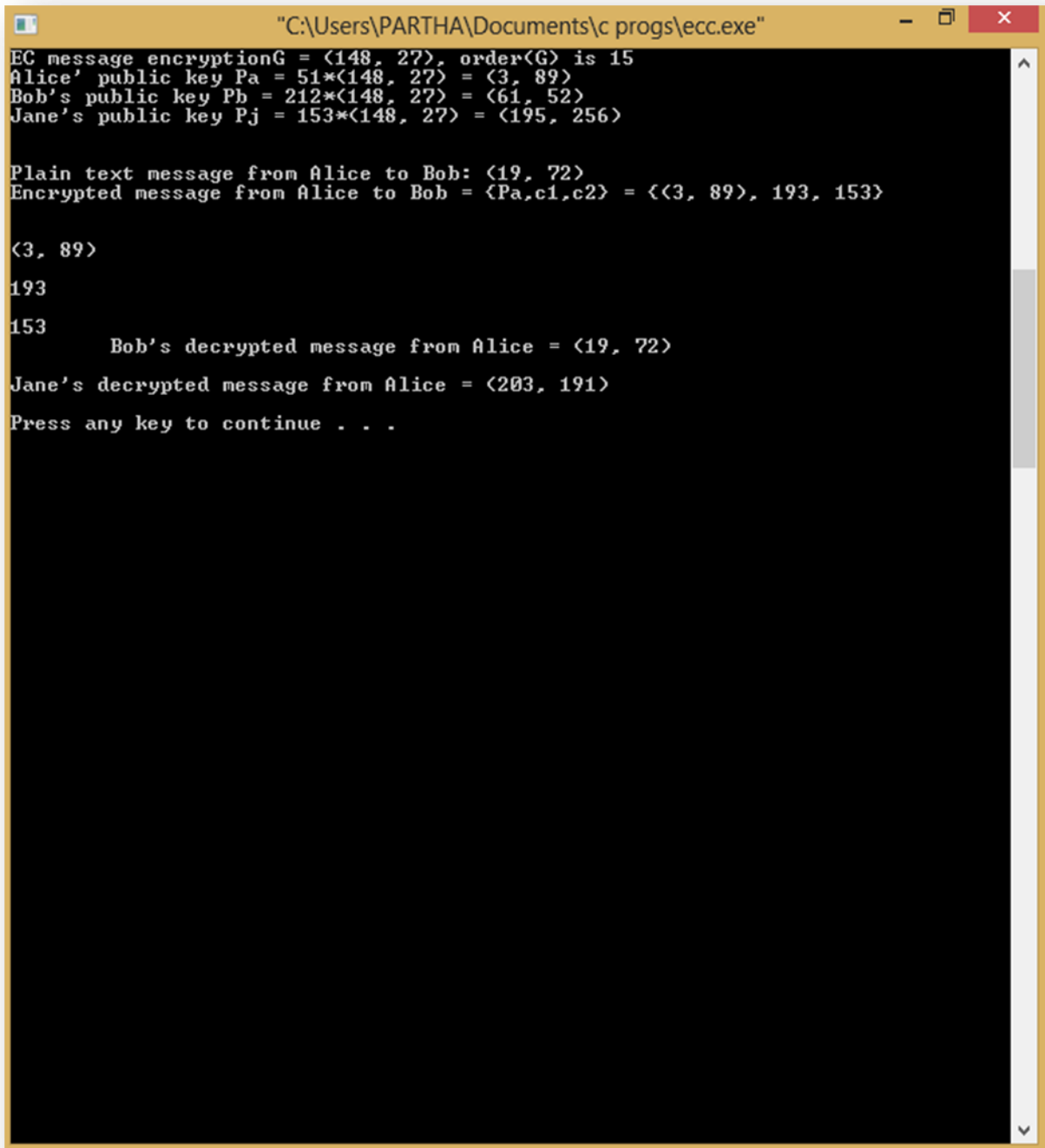
Points on the curve (i.e. the group elements):
<0, 1> <0, 262> <1, 23> <1, 240> <2, 96> <2, 167>
<3, 89> <3, 174> <4, 73> <4, 190> <6, 111> <6, 152>
<7, 80> <7, 183> <8, 28> <8, 235> <10, 119> <10, 144>
<14, 38> <14, 225> <15, 32> <15, 231> <21, 128> <21, 135>
<22, 64> <22, 199> <23, 57> <23, 206> <24, 35> <24, 228>
<27, 81> <27, 182> <29, 111> <29, 152> <30, 87> <30, 176>
<31, 34> <31, 229> <33, 27> <33, 236> <38, 101> <38, 162>
<39, 45> <39, 218> <40, 55> <40, 208> <44, 65> <44, 198>
<45, 35> <45, 228> <47, 81> <47, 182> <48, 60> <48, 203>
<49, 123> <49, 140> <51, 64> <51, 199> <57, 25> <57, 238>
<58, 5> <58, 258> <59, 6> <59, 257> <60, 123> <60, 140>
<61, 52> <61, 211> <66, 34> <66, 229> <67, 119> <67, 144>
<68, 74> <68, 189> <69, 108> <69, 155> <72, 85> <72, 178>
<74, 4> <74, 259> <75, 25> <75, 238> <77, 116> <77, 147>
<78, 53> <78, 210> <81, 0> <82, 27> <82, 236> <83, 114>
<83, 149> <87, 61> <87, 202> <88, 91> <88, 172> <89, 79>
<99, 184> <103, 131> <103, 132> <108, 83> <108, 180> <110, 130>
<110, 133> <111, 77> <111, 186> <112, 44> <112, 219> <115, 59>
<115, 204> <116, 125> <116, 138> <117, 18> <117, 245> <118, 106>
<118, 157> <123, 23> <123, 240> <127, 2> <127, 261> <131, 25>
<131, 238> <132, 70> <132, 193> <134, 29> <134, 234> <137, 107>
<137, 156> <138, 29> <138, 234> <139, 23> <139, 240> <141, 109>
<141, 154> <142, 26> <142, 237> <143, 37> <143, 226> <144, 69>
<144, 194> <145, 115> <145, 148> <146, 97> <146, 166> <147, 94>
<147, 169> <148, 27> <148, 236> <150, 129> <150, 134> <152, 124>
<152, 139> <154, 123> <154, 140> <157, 100> <157, 163> <159, 102>
<159, 161> <162, 104> <162, 159> <166, 34> <166, 229> <169, 107>
<169, 156> <170, 130> <170, 133> <173, 90> <173, 173> <174, 109>
<174, 154> <175, 20> <175, 243> <180, 122> <180, 141> <181, 59>
<181, 204> <182, 110> <182, 153> <184, 43> <184, 220> <185, 127>
<185, 136> <186, 119> <186, 144> <188, 70> <188, 193> <189, 81>
<189, 182> <190, 64> <190, 199> <192, 15> <192, 248> <194, 35>
<194, 228> <195, 7> <195, 256> <196, 116> <196, 147> <199, 2>
<199, 261> <200, 2> <200, 261> <206, 70> <206, 193> <207, 117>
<207, 146> <208, 24> <208, 239> <210, 79> <210, 184> <211, 109>
<211, 154> <216, 4> <216, 259> <217, 79> <217, 184> <218, 108>
<218, 155> <219, 118> <219, 145> <220, 107> <220, 156> <221, 33>
<221, 230> <222, 58> <222, 205> <223, 50> <223, 213> <224, 9>
<224, 254> <226, 99> <226, 164> <228, 111> <228, 152> <230, 59>
<230, 204> <236, 4> <236, 259> <239, 108> <239, 155> <240, 31>
<240, 232> <242, 72> <242, 191> <245, 48> <245, 215> <246, 130>
<246, 133> <249, 98> <249, 165> <251, 75> <251, 188> <253, 116>
<253, 147> <254, 29> <254, 234> <259, 14> <259, 249> <260, 51>
<260, 212>

some point P = <1, 23>, 2P = <87, 61>
some point Q = <1, 240>, P+Q = <0, 0>
P += Q = <0, 0>
P += P = 2P = <87, 61>

EC message encryption G = <148, 27>, order(G) is 15

```

Figure 6.4 ECC Co-ordinate



```
"C:\Users\PARTHA\Documents\c progs\ecc.exe"
EC message encryptionG = <148, 27>, order<G> is 15
Alice' public key Pa = 51*<148, 27> = <3, 89>
Bob's public key Pb = 212*<148, 27> = <61, 52>
Jane's public key Pj = 153*<148, 27> = <195, 256>

Plain text message from Alice to Bob: <19, 72>
Encrypted message from Alice to Bob = <Pa,c1,c2> = <<3, 89>, 193, 153>

<3, 89>
193
153
    Bob's decrypted message from Alice = <19, 72>
Jane's decrypted message from Alice = <203, 191>
Press any key to continue . . .
```

Figure 6.5: ECC Encryption

```

"C:\Users\PARTHA\Documents\c progs\ecc.exe"
2
33
1
1
55
4
4
1
90
2
22
1
67
2
12
7
8
12
1
Plain text message from Alice to Bob: <2, 33>
Encrypted message from Alice to Bob = <Pa,c1,c2> = <<3, 89>, 48, 103>
    Bob's decrypted message from Alice = <2, 33>
Jane's decrypted message from Alice = <229, 230>
Plain text message from Alice to Bob: <1, 55>
Encrypted message from Alice to Bob = <Pa,c1,c2> = <<3, 89>, 24, 84>
    Bob's decrypted message from Alice = <1, 55>
Jane's decrypted message from Alice = <246, 208>
Plain text message from Alice to Bob: <4, 4>
Encrypted message from Alice to Bob = <Pa,c1,c2> = <<3, 89>, 96, 140>
    Bob's decrypted message from Alice = <4, 4>
Jane's decrypted message from Alice = <195, 259>
Plain text message from Alice to Bob: <1, 90>
Encrypted message from Alice to Bob = <Pa,c1,c2> = <<3, 89>, 24, 257>
    Bob's decrypted message from Alice = <1, 90>
Jane's decrypted message from Alice = <246, 173>
Plain text message from Alice to Bob: <2, 22>
Encrypted message from Alice to Bob = <Pa,c1,c2> = <<3, 89>, 48, 244>
    Bob's decrypted message from Alice = <2, 22>
Jane's decrypted message from Alice = <229, 241>
Plain text message from Alice to Bob: <1, 67>
Encrypted message from Alice to Bob = <Pa,c1,c2> = <<3, 89>, 24, 241>

```

Figure 6.6: Sequence of user data being Encrypted Decrypted


```

"C:\Users\PARTHA\Documents\c progs\pro_code.exe"
enter the size
12
Enter the c1-----
48
24
96
24
48
24
48
168
25
239
145
102
-----
ORIGINAL SEQUENCE
48 24 96 24 48 24 48 168 25 239 145 102
-----
values in 2D array-----
48 24 48
24 48 168
96 24 25
Rest values for 1D array-----
239 145 102
ENCRYPTED DATA
-----
3 48 24 48 168 25 24 96 24 48 239 145 102
-----
DECRYPTED DATA
-----
48 24 96 24 48 24 48 168 25 239 145 102

```

```

Untitled - Notepad
File Edit Format View Help
c1
48
24
96
24
48
24
48
168
25
239
145
102
c2
103
84
140
257
244
241
157
17
35
217
70
35

```

Figure 6.7: Encrypted message sequence c1 component in “Twisted” form

```

"C:\Users\PARTHA\Documents\c progs\pro_code.exe"
enter the size
12
Enter the c2-----
103
84
140
257
244
241
157
17
35
217
70
35

-----
ORIGINAL SEQUENCE
103 84 140 257 244 241 157 17 35 217 70 35
-----

values in 2D array-----
103 257 157
84 244 17
140 241 35

Rest values for 1D array-----
217 70 35

ENCRYPTED DATA

-----
3 103 257 157 17 35 241 140 84 244 217 70 35
-----

DECRYPTED DATA

-----
103 84 140 257 244 241 157 17 35 217 70 35
-----

```

```

Untitled - Notepad
File Edit Format View Help
c1
48
24
96
24
48
24
48
168
25
239
145
102

c2
103
84
140
257
244
241
157
17
35
217
70
35

```

Figure 6.8 Encrypted Message Sequence C2 in “Twisted” form

6.4 Result Analysis & Discussion

As source message is chunked into two parts m_1 and m_2 , in Encrypted message we get Public Key of Alice, C_1 and C_2 (Fig 7.4). In reality the spiral code was implemented on a single pressure sensor sending 16 Byte of Hexadecimal data to a remote server in each interval of 30 seconds. The maximum value that can be represented by a Byte is FF(255). Hence for appropriate Finite Field Calculation in the field 263 was chosen (any prime greater than 255) for modular arithmetic.

In Fig 7.5 the input data is shown. Actually this data is obtained by splitting the data into two parts such that the combined data is a valid sensor output. Hence the combination of two consecutive values over here is less than 263.

C_1 and C_2 for a sequence of data were stored in two separate arrays (Fig 7.6 & Fig 7.7) and Spiral algorithm was applied on it individually making received data more complex for the intruder.

Conclusion & Future Work

By doing this project we came to know about Wireless Sensor Network, their essential day to day application, threats and attacks against their security as well as some security protocols and algorithms. Among network security protocols we came to know about Dynamic Key Management being implemented in LOCK model and how it is superior to SHELL model although both use EBS scheme. However wireless data being always prone to attacks / Eavesdropping software cryptography has always a crucial role to play. We came to know about Public Key Encryption, RSA and elaborately discussed about data security and authentication. However ECC being state of art widely used Encryption Technique and more efficient than RSA in terms of memory/ power requirements we implemented that that in our project but with the modification of our twisting Spiral Algorithm.

Our future work will include the real world application of this cryptography technique. For the sake of understanding we have used small prime number for modular arithmetic but for all practical purposes we need to handle really huge prime numbers requiring separate Maths Library Functions. We would also like to research in the field data signature and authentication.

Reference

- [1] Ritu Sharma, Yogesh Chaba, Yudhvir Singh, “Analysis of Security Protocols in Wireless Sensor Networks”, Int. J. Advanced Networking and Applications 707 Volume: 02, Issue: 03, Pages: 707-713 (2010).
- [2] Chris Karlof and David Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures”, Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols. 2003.
- [3] L. Eschenauer and V. Gligor, “A Key Management Scheme for Distributed Sensor Networks,” Proc. 9th ACM Conf. Comp. & Community Sec., Nov. 2002, pp. 41-47.
- [4] W. Du et al., “A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge,” Proc. IEEE INFOCOM '04, Mar. 2004.
- [5] D. Liu and P. Ning, “Improving Key Pre-Distribution with Deployment Knowledge in Static Sensor Networks,” ACM Trans. Sensor Networks, 2005, pp 204–39.
- [6] G. Jolly et al., “A Low-Energy Key Management Protocol for Wireless Sensor Networks,” Proc. IEEE Symp. Comp. and Community., June 2003, p. 335
- [7] M. Eltoweissy et al., “Group Key Management Scheme for Large- Scale Wireless Sensor Network,” J. Ad Hoc Networks, Sept. 2005, pp. 796–802.
- [8] M. Younis, K. Ghumman, and M. Eltoweissy, “Location aware Combinatorial Key Management Scheme for Clustered Sensor Networks,” to appear, IEEE Trans. Parallel and Distrib. Sys., 2006.
- [9] <https://developer.apple.com/library/mac/documentation/security/conceptual/cryptoservices/CryptographyConcepts/CryptographyConcepts.html>
- [10] V. Miller, “Uses of Elliptic Curves in Cryptography”, Advances in Cryptology, CRYPTO '85, Proceedings, Lecture Notes in Computer Science 218, Springer-Verlag, 1986, 417-426
- [11] N. Koblitz, “Elliptic Curve Cryptography”, Mathematics of Computation, 48 (1987), 203-209.
- [12] IEEE P1363, “Standard Specifications for Public Key Cryptography,” 2000.
- [13] Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-2, National Institute of Standards and Technology. 2000.
- [14] Amit N. Gathani, Implementation of Elliptic Curve Cryptography in Embedded System, 2001.
- [15] “The Basics of ECC”, <http://www.certicom.com>
- [16] G. Faltings (July 1995): The Proof of Fermat's Last Theorem by R. Taylor and A. Wiles. Notices of the AMS 42 (7): 743–746. ISSN 0002-9920. July 1995.
- [17] G. R. Blakley, Notes on Arithmetic of Some Commutative Rings and Fields, October 1993.
- [18] Elliptic curve Cryptography by Joel Allardyce, Nitish Goyal, April 15, 2004
- [19] Elliptic Curve Cryptography and Its Application, Moncef Amara and Amar Siad, 2011 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA)
- [20] Song Y. Yan, Number Theory for Computing, 2nd ed, Springer Verlag, Berlin, Germany, 2002.