

Integration Of Digital Signature And ElGamal

Scheme For Secure Data Transmission

In Digital Transactions

A project report submitted in partial fulfillment of the requirement for the award of

degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Submitted by

P.Harisankar Narayan	19341A1284
R.Ramakrishna	19341A1295
G.Sai Sandeep	19341A1298
V.Srikara Sai Ramesh	19341A12C4
I.Bala Raju	20345A1207

Under the esteemed guidance of

Mr. Y.Surya Prakash

Asst. Professor, Dept of Information Technology

GMR Institute of Technology

An Autonomous Institute Affiliated to JNTUK-Kakinada

(Accredited by NBA, NAAC with 'A' Grade & ISO 9001:2015 Certified Institution)

GMR Nagar, Rajam – 532127,

Andhra Pradesh, India

2022 – 2023

Department of Information Technology

CERTIFICATE

This is to certify that the thesis entitled “**INTEGRATION OF DIGITAL SIGNATURE AND ELGAMAL SCHEME FOR SECURE DATA TRANSMISSION IN DIGITAL TRANSACTIONS**” submitted by **P.HARISANKAR NARAYAN (19341A1284), R. RAMAKRISHNA (19341A1295), G.SAI SANDEEP (19341A1298), V. SRIKARA SAI RAMESH (19341A12C4), I.BALA RAJU(20345A1207)** has been carried out in partial fulfillment of the requirement for the award of degree of **Bachelor of Technology in Information Technology** of **GMRIT, Rajam** affiliated to **JNTUK** is a record of bonafide work carried out by them under my guidance & supervision. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree.

Signature of Supervisor

Mr. Y.Surya Prakash
Asst. Professor,
Department of IT
GMRIT, Rajam

Signature of HOD

Dr. Ajit Kumar Rout
Professor & HOD
Department of IT
GMRIT, Rajam

The report is submitted for the viva-voice examination held on

Signature of External Examiner

ACKNOWLEDGEMENT

It gives us an immense pleasure to express deep sense of gratitude to our guide **Mr. Y.Surya Prakash**, Asst. Professor, Department of Information Technology of whole hearted and invaluable guidance throughout the project work. Without his sustained and sincere effort, this project work would not have taken this shape. He encouraged and helped us to overcome various difficulties that we have faced at various stages of our project work.

We would like to sincerely thank our Head of the Department **Dr. Ajit Kumar Rout**, for providing all the necessary facilities that led to the successful completion of our project work.

We would like to take this opportunity to thank our beloved Principal **Dr. C.L.V.R.S.V. Prasad**, for providing all the necessary facilities and a great support to us in completing the project work.

We would like to thank the project coordinator **Mr. Ch. Anil Kumar**, all the faculty members and the non-teaching staff of the Department of Information Technology for their direct or indirect support for helping us in completion of this project work.

Finally, we would like to thank all of our friends and family members for their continuous help and encouragement.

P.HARISANKAR NARAYAN	19341A1284
R.RAMAKRISHNA	19341A1295
G.SAI SANDEEP	19341A1298
V.SRIKARA SAI RAMESH	19341A12C4
I.BALA RAJU	20345A1207

ABSTRACT

Security can be considerably a major concern when it comes to high-end data transmissions like satellite parameter communication, defence security codes communication etc. Sensitive information of any type can leads to various attacks from intruders who is having malicious thoughts. These attacks can be significantly harmful to loss of users data (sensitive or non-sensitive data). This problem is look thoughtfully for a long time and resolved by integrating Digital Signature and ElGamal scheme(Elliptic Curve Cryptography). This Integration provides a Cognitive Feature for Authorization. The encryption is done using the standard ElGamal scheme with a well-built reputation for itself in secure data transmission. It provides a secure communication channel between the two ends by authenticating the sender with Digital Signature. One of the thrust area of this project is Digital Transactions.

Keywords: Cryptography, Elliptic Curve Cryptography, attacks, cognitive feature, Digital Signature, ElGamal.

INDEX

ACKNOWLEDGEMENT	i
ABSTRACT	ii
LIST OF FIGURES	iii
LIST OF TABLES	v
LIST OF SYMBOLS & ACRONYMS	vi
CHAPTER 1 INTRODUCTION	1
1.1 CRYPTOGRAPHY	2
1.2 ELLPTIC CURVE CRYPTOGRAPHY	4
1.3 AES AND DES	6
1.4 DIGITAL SIGNATURES	8
1.5 KEY EXCHANGE ALGORITHMS	10
1.6 DIFFIE-HELLMAN	10
1.7 RIVEST–SHAMIR–ADLEMAN	12
1.8 FLASK APPLICATION	13
1.9 WEB TECHNOLOGIES	15
1.9.1 HTML	15
1.9.2 CSS	15
1.9.3 JAVA SCRIPT	15
CHAPTER 2 LITERATURE SURVERY	16
CHAPTER 3 REQUIREMENT SPECIFICATION	30
3.1 SOFTWARE REQUIREMENTS	31
3.1.1 OPERATION SYSTEM	31
3.1.2 SDK	31
3.2 HARDWARE REQUIREMENTS	31
3.3 NON-FUNCTIONAL REQUIREMENTS	32
3.4 PYTHON LIBRARIES TO BE INSTALLED	32

CHAPTER 4 METHODOLOGY	33
4.1 IMPLEMENTATION OF ENCRYPTION USING ECC	35
4.1.1 GENERATING SYSTEM PARAMETERS	37
4.1.2 ENCODING THE MESSAGE TO NUMERICAL VALUES	38
4.1.3 MAPPING THE MESSAGE TO AN ELLIPTIC CURVE	38
4.1.4 ENCRYPTING THE MAPPED POINTS	39
4.2 IMPLEMENTATION OF DECRYPTION USING ECC	39
4.2.1 DECRYPTING THE MESSAGE	41
4.2.2 DECODING THE DECRYPTED MESSAGE	41
4.2.3 CONVERTING THE DECODED MESSAGE TO PLAINTEXT	41
4.3 IMPLEMENTATION OF DIGITAL SIGNATURE	42
4.3.1 SIGNING THE ENCRYPTED MESSAGE	43
4.3.2 VERIFYING THE SIGNED MESSAGE	44
4.4 INTEGRATION OF ELGAMAL SCHEME AND DIGITAL SIGNATURE	44
CHAPTER 5 SYSTEM DESIGN	46
5.1 INTRODUCTION	47
5.2 CLASS DIAGRAM	47
5.3 USE CASE DIAGRAM	48
5.4 SEQUENCE DIAGRAM	49
5.5 COLLABORATION DIAGRAM	50
CHAPTER 6 RESULTS AND DISCUSSION	51
CHAPTER 7 CONCLUSIONS AND FUTURE SCOPE	57
CHAPTER 8 REFERENCES	59
APPENDIX A	
APPENDIX B	

LIST OF FIGURES

FIGURE NO	TITLE	PAGENO
1.1	REPRESENTATION OF ELLIPTIC CURVE	4
1.2	REPRESENTATION OF ECC POINT ADDITION	5
1.3	REPRESENTATION OF ECC POINT DOUBLING	6
1.4	BLOCK DIAGRAM OF AES	7
1.5	BLOCK DIAGRAM OF DES	8
1.6	SIGNING AND VERIFICATION OF DIGITAL SIGNATURE ALGORITHM	9
1.7	DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM	11
1.8	DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM EXAMPLE USING COLORS	12
1.9	RIVEST–SHAMIR–ADLEMAN ALGORITHM	13
4.1	LIST OF MODULES INVOLVED IN THE PROPOSED MODEL	34
4.2	SYSTEM ARCHITECTURE OF PROPOSED MODE	35
4.3	ENCRYPTION ALGORITHM	36
4.4	ENCRYPTION PROCESS IN PROPOSED MODEL	37
4.5	DECRYPTION ALGORITHM	39
4.6	DECRYPTION PROCESS IN PROPOSED MODEL	41
4.7	DIGITAL SIGNATURE ALGORITHM IN PROPOSED MODEL	43
4.8	INTEGRATION OF ELGAMAL SCHEME AND DSA (PROPOSED MODEL)	45
5.1	CLASS DIAGRAM FOR PROPOSED MODEL	48
5.2	USE CASE DIAGRAM FOR PROPOSED MODEL	49
4.3	SEQUENCE DIAGRAM FOR PROPOSED MODEL	50
5.4	COLLABORATION DIAGRAM FOR PROPOSED MODEL	50
6.1	ANACONDA COMMAND PROMPT	52

6.2	ELGAMAL ENCRYPTION	52
6.3	ELGAMAL DECRYPTION	53
6.4	RSA ENCRYPTION	54
6.5	RSA DECRYPTION	54
6.6	ECC - RSA KEY SIZE COMPARISON	56
6.7	PERFORMANCE COMPARISON BETWEEN CONVENTIONAL CRYPTOSYSTEMS AND ELGAMAL	56

LIST OF TABLES

TABLE NO	TITLE	PAGENO
1.1	COMPARISION BETWEEN ENCRYPTION AND DECRYPTION	3
6.1	ECC - RSA PERFORMANCE COMPARISON	55
6.2	ECC - RSA COST COMPARISON	55
6.3	ECC - RSA KEY SIZE COMPARISON	55

LIST OF ABBREVIATIONS

ECC	: ELLIPTIC CURVE CRYPTOGRAPHY
RSA	: RIVEST, SHAMIR, ADLEMAN
AES	: ADVANCED ENCRYPTION STANDARD
DES	: DATA ENCRYPTION STANDARD
ECDH	: ELLIPTIC CURVE DIFFIE–HELLMAN
ECDSA	: ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM
EDDSA	: EDWARDS CURVE DIGITAL SIGNATURE ALGORITHM
DHKE	: DIFFIE–HELLMAN KEY EXCHANGE
MITM	: MAN IN THE MIDDLE
PKI	: PUBLIC KEY INFRASTRUCTURE
DH	: DIFFIE–HELLMAN
TLS	: TRANSPORT LAYER SECURITY
SSH	: SECURE SHELL
IPSEC	: INTERNET PROTOCOL SECURITY
DLP	: DISCRETE LOGARITHM PROBLEM
KPA	: KNOWN-PLAIN TEXT ATTACK
CPA	: CHOSEN-PLAIN TEXT ATTACK
COA	: CIPHER-TEXT ONLY ATTACK
ECDLP	: ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM
CCA	: CHOSEN-CIPHER TEXT ATTACK
IBC	: IDENTITY BASED CRYPTOSYSTEM
PKG	: PRIVATE KEY GENERATOR
CLPKC	: CERTIFICATELESS PUBLIC KEY CRYPTOGRAPHY
PAN	: PERSONAL AREA NETWORK
LAN	: LOCAL AREA NETWORK
HAN	: HOME AREA NETWORK
DFD	: DATA FLOW DIAGRAM
RMSE	: ROOT MEAN SQAURE ERROR
SDK	: SOFTWARE DEVELOPMENT KIT

Chapter – 1

INTRODUCTION

Most of the people are using mobile phones for online transactions. In this case most transactions are done through internet banking. The existing Cryptography is failed to provide security due to rendering technology hence it becomes a major concern in terms of providing security and privacy. ElGamal and Digital Signature are combined to address the existing challenges and provide effective security. ECC has a major role in improving data integrity and confidentiality. However, the data carried over a wireless network will not be highly secure as a result. We are unable to use these strategies to prevent problems with wireless networks' increased packet transmission issues, which result in data loss. To overcome the problems in the existing system, the proposed system is promoted with an authentication process involving digital signatures of the user in its implementation.

A group of points plotted on an image is provided to the user for him to use his cognitive abilities and remember the points he has selected the first time he uses the system. Coming to the process of ElGamal scheme, a private key is generated using these selected points for the user. Some mathematical operations are performed on the points to attain the digital signatures. With the help of the now obtained private key and a base point, a public key for this user is also generated. The other authenticated entry follows suit thus far. Now, in order for them to have a connection and basically validate each other, a session key needs to be generated on either end which is required to be equal. After generating and verifying the session key, both the end users are now set to communicate with each other. A message sent by either user is then encrypted using the method of elliptic curve cryptography. Thus sent message is now decrypted and digitally signed on the receiver end.

1.1 Cryptography :

Cryptography is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it. In computer science, cryptography refers to secure information and communication techniques derived from mathematical concepts and a set of rule-based calculations called algorithms, to transform messages in ways that are hard to decipher. These deterministic algorithms are used for cryptographic key generation, digital signing, verification to protect data privacy, web browsing on the internet and confidential communications such as credit card transactions and email. Cryptography is closely related to the disciplines of cryptology and cryptanalysis. It includes techniques such as microdots, merging words with images and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with scrambling (ordinary text, sometimes referred to as clear text) into ciphertext (a process called encryption), then back again (known as decryption). Individuals who practice this field are known as cryptographers.

Modern cryptography concerns itself with the following four objectives:

- **Confidentiality** : The information cannot be understood by anyone for whom it was unintended.
- **Integrity** : The information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected.
- **Non-repudiation** : The creator/sender of the information cannot deny at a later stage their intentions in the creation or transmission of the information.
- **Authentication** : The sender and receiver can confirm each other's identity and the origin/destination of the information.

Table 1.1: Comparison Between Encryption And Decryption

Parameter	Encryption	Decryption
Definition	The process of converting normal data into an unreadable format to avoid unauthorized access to sensitive data.	The process of converting the unreadable/encrypted data into its original form so that authorized users can read it.
Process	Whenever data is transferred between two separate machines, it is automatically encrypted using a secret key.	The receiver of the data automatically converts the encrypted data to its original form.
Location of Conversion	The user who is sending the encrypted data to the destination.	The user who receives the encrypted data and converts it.
Example	Sending sensitive documents to a user.	Receiving the encrypted documents from the source and decrypting it to read it.
Use of Algorithm	The encryption-decryption process uses the same algorithm with the same key.	A single algorithm is used for encryption and decryption is done with a pair of keys where each of them is used for encryption and decryption.
Primary Function	Converting decipherable messages into an incomprehensible form so that it can not be interpreted.	Converting an obscure message into a decipherable form that is understandable by humans.

1.2 Elliptic curve cryptography:

Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC allows smaller keys compared to non-EC cryptography (based on plain Galois fields) to provide equivalent security.

Elliptic curves are applicable for key agreement, digital signatures, pseudo-random generators and other tasks. Indirectly, they can be used for encryption by combining the key agreement with a symmetric encryption scheme. Elliptic curves are also used in several integer factorization algorithms based on elliptic curves that have applications in cryptography, such as Lenstra elliptic-curve factorization.

The ECC cryptography is considered a natural modern successor of the RSA cryptosystem, because ECC uses smaller keys and signatures than RSA for the same level of security and provides very fast key generation, fast key agreement and fast signatures. The private keys in the ECC are integers (in the range of the curve's field size, typically 256-bit integers). The key generation in the ECC cryptography is as simple as securely generating a random integer in certain range, so it is extremely fast. Any number within the range is valid ECC private key.

In mathematics elliptic curves are plane algebraic curves, consisting of all points $\{x, y\}$, described by the equation: $Ax^3 + Bx^2y + Cxy^2 + Dy^3 + Ex^2 + Fxy + Gy^2 + Hx + Iy + J = 0$.

ECC uses elliptic curves in a simplified form (Weierstras form), which is defined as: $y^2 = x^3 + a*x + b$.

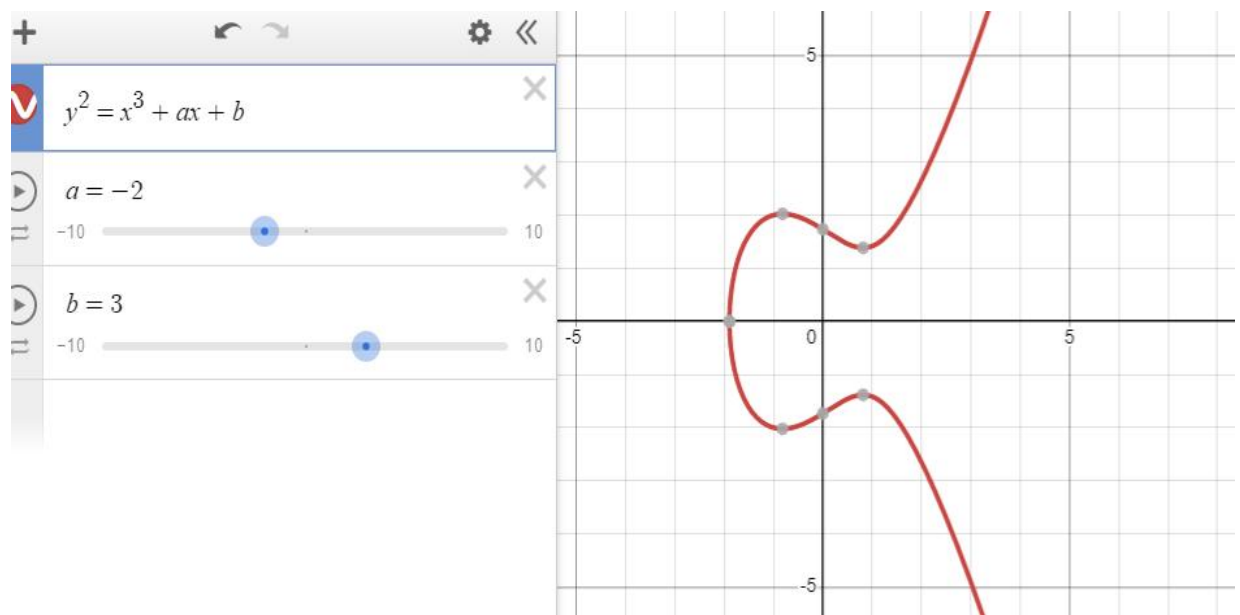


Figure 1.1 : Representation of Elliptic Curve

Alternative representations of elliptic curves include: Hessian curves, Edwards curves, Twisted curves, Twisted Hessian curves, Twisted Edwards curve, Doubling-oriented Doche-Icart–Kohel curve, Tripling-oriented Doche-Icart–Kohel curve, Jacobian curve and Montgomery curves.

Point addition: Add two points on an elliptic curve together to get a third point on the curve. let two points $A(x_1, y_1)$ and $B(x_2, y_2)$ must not have the same coordinates ($x_1 \neq x_2$ and $y_1 \neq y_2$).

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p \quad \text{and} \quad y_3 = (\lambda(x_1 - x_2) - y_1) \bmod p \quad \{ \text{where } \lambda = (y_2 - y_1) / (x_2 - x_1) \bmod p \}$$

Point doubling: Let the two points projection ($R=A+A$)

$$x_3 = (\lambda^2 - 2x_1) \bmod p \quad \text{and} \quad y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p \quad \{ \text{where } \lambda = (3x_1^2 + a) / (2y_1) \bmod p \}$$

Point multiplication: Take a point on the elliptic curve, let's call it P. Then, the repeated addition is the definition of the operation of multiplying the point P. kP equals to $P+P+P+ \dots$ k times.

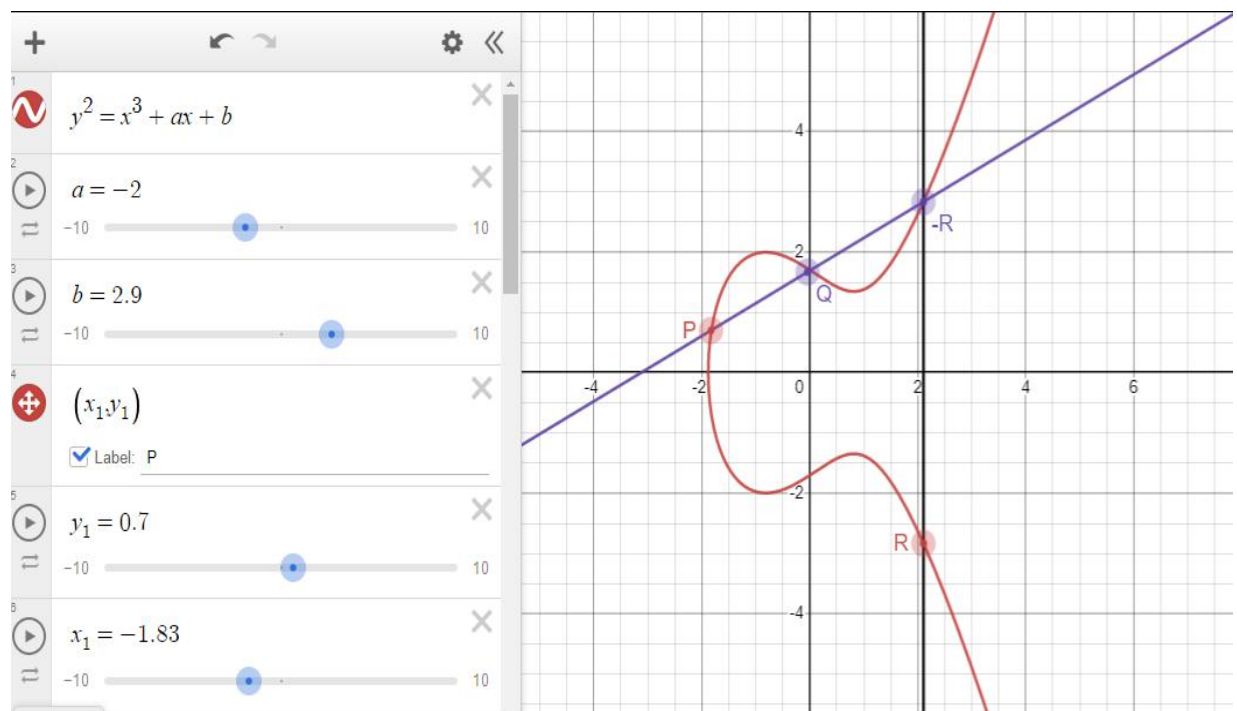


Figure 1. 2 : Representation of ECC Point Addition

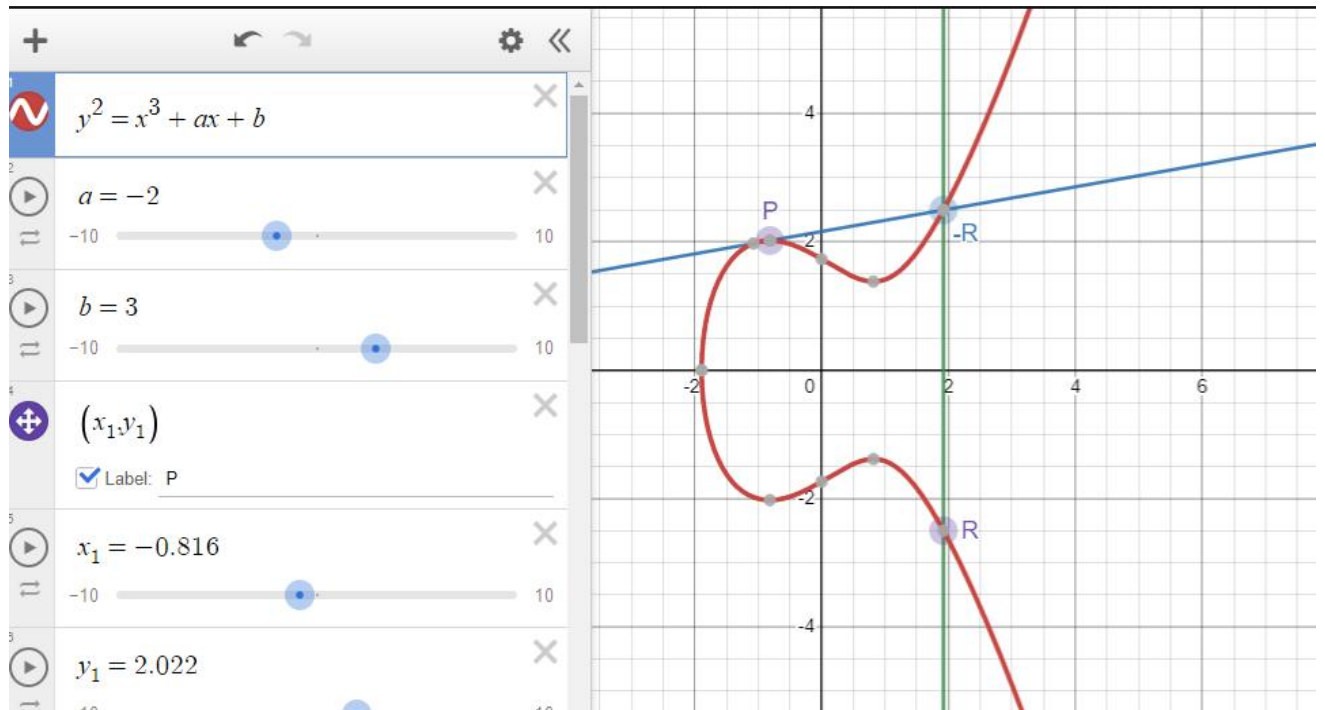


Figure 1.3 : Representation of ECC Point Doubling

1.3 AES and DES

Encryption is the process by which a readable message is converted to an unreadable form to prevent unauthorized parties from reading it. Decryption is the process of converting an encrypted message back to its original (readable) format. The original message is called the plaintext message. The encrypted message is called the ciphertext message. In cryptography two major types of encryption schemes are widely used: symmetric encryption (where a single secret key is used to encrypt and decrypt data) and asymmetric encryption (where a public key cryptosystem is used and encryption and decryption is done using a pair of public and corresponding private key).

Cryptography deals with storing and transmitting data in a secure way, such that only those, for whom it is intended, can read and process it. This may involve encrypting and decrypting data using symmetric or asymmetric encryption schemes, where one or more keys are used to transform data from plain to encrypted form and back. Symmetric encryption (like AES, Two fish and ChaCha20) uses the same key to encrypt and decrypt messages, while asymmetric encryption uses a public-key cryptosystem (like RSA or ECC) and a key-pair: public key (encryption key) and corresponding private key (decryption key). Encryption algorithms are often combined in encryption schemes (like AES-256-CTR-HMAC-SHA-256, ChaCha20-Poly1305 or ECIES-secp256k1-AES-128-GCM). Symmetric encryption schemes use the same symmetric key (or password) to encrypt data and decrypt the encrypted data back to its original form. The secret key used to cipher (encrypt) and decipher

(decrypt) data is typically of size 128, 192 or 256 bits and is sometimes referred as "encryption key" or "shared key", because both sending and receiving parties should know it. Before introducing the asymmetric key encryption schemes and algorithms, it is need to understand the concept of public key cryptography (asymmetric cryptography).

The public key cryptography uses a different key to encrypt and decrypt data (or to sign and verify messages). Keys always come as public + private key pairs. Asymmetric cryptography deals with encrypting and decrypting messages using a public/private key, signing messages, verifying signatures and securely exchanging keys. Popular public-key cryptosystems (asymmetric crypto algorithms) like RSA (Rivest–Shamir–Adleman), ECC (elliptic curve cryptography), Diffie-Hellman, ECDH, ECDSA and EdDSA, are widely used in the modern cryptography. Message encryption and signing is done by a private key. The private keys are always kept secret by their owner, just like passwords. Message decryption and signature verification is done by the public key. Public keys are by design public information (not a secret). It is mathematically infeasible to calculate the private key from its corresponding public key.

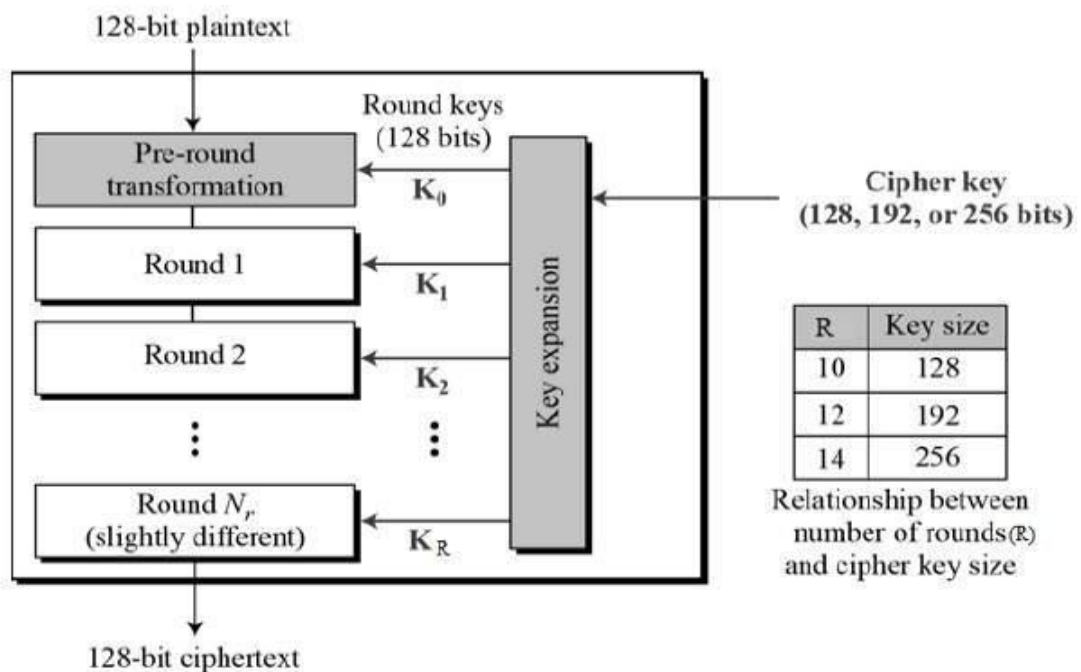


Figure 1.4 : Block Diagram Of AES

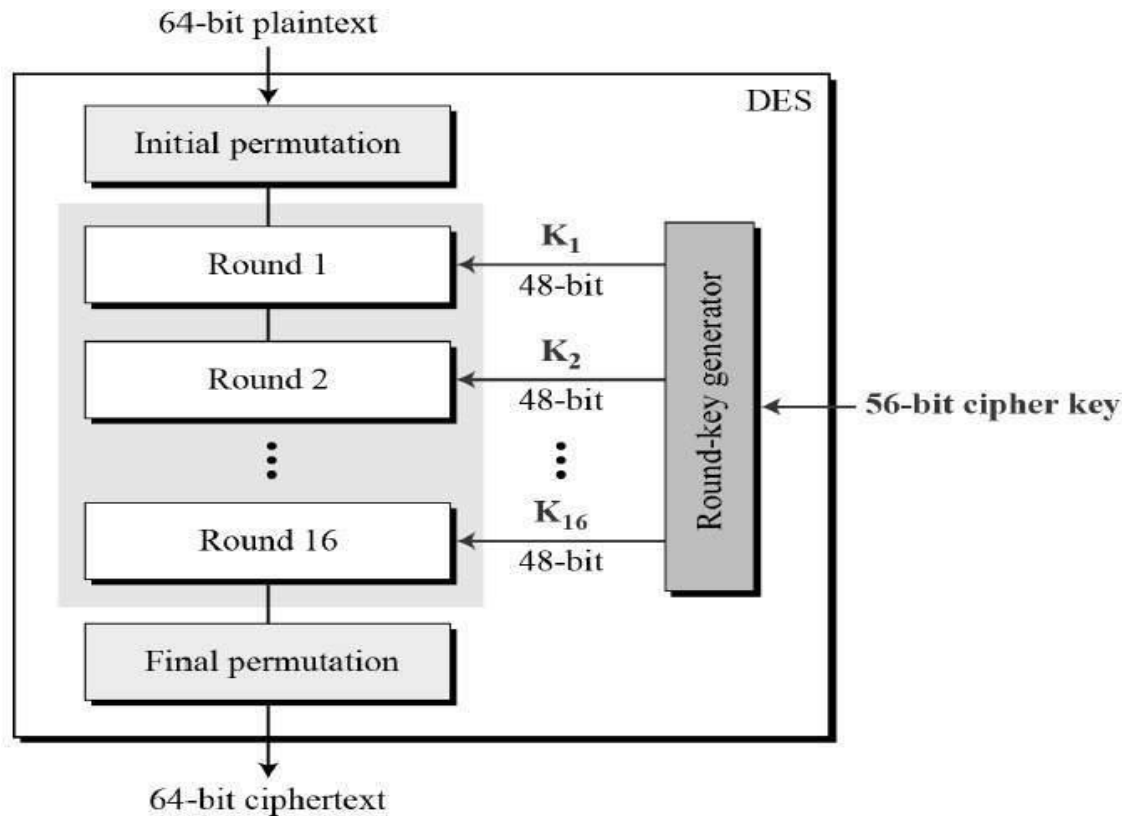


Figure 1.5 : Block Diagram Of DES

1.4 Digital Signatures:

In cryptography digital signatures provide message authentication, integrity and non-repudiation for digital documents. Digital signatures are a cryptographic tool to sign messages and verify message signatures in order to provide proof of authenticity for digital messages or electronic documents. Digital signatures work in the public-key cryptosystems and use a public / private key pairs. Message signing is performed by the private key and message verification is performed by the corresponding public key. A message signature mathematically guarantees that certain message was signed by certain (secret) private key, which corresponds to certain (non-secret) public key.

After a message is signed, the message and the signature cannot be modified and thus message authentication and integrity is provided. Anyone, who knows the public key of the message signer, can verify the signature. After signing the signature author cannot reject the act of signing (this is known as non-repudiation). Digital signatures are widely used today for signing digital contracts, for authorizing bank payments and signing transactions in the public block chain systems for transferring digital assets. Most public-key cryptosystems like RSA and ECC provide secure digital signature schemes like DSA, ECDSA and EdDSA. We shall discuss the digital signatures in greater detail later in this section.

Digital signatures objectives:

- **Message authentication** - a proof that certain known sender (secret key owner) have created and signed the message.
- **Message integrity** - a proof that the message was not altered after the signing.
- **Non-repudiation** - the signer cannot deny the signing of the document after the signature is once created.

Digital signatures are widely used today in the business and in the financial industry, e.g. for authorizing bank payments (money transfer), for exchange of signed electronic documents, for signing transactions in the public block chain systems (e.g. transfer of coins, tokens or other digital assets), for signing digital contracts and in many other scenarios. Digital signatures cannot identify who is the person, created a certain signature. This can be solved in combination with a digital certificate, which binds a public key owner with identity (person, organization, web site or other). By design digital signatures bind messages to public keys, not to digital identities.

`signMsg(msg, privKey)`

`verifyMsgSignature(msg, signature, pubKey) → valid / invalid`

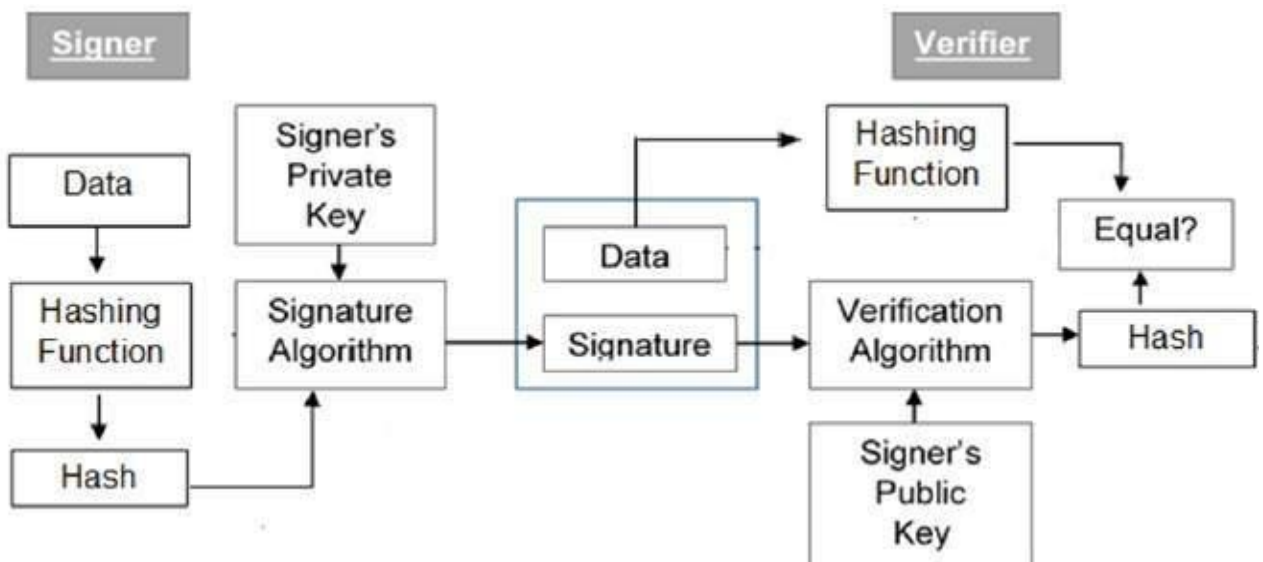


Figure 1.6 : Signing And Verification Of Digital Signature Algorithm

1.5 Key Exchange Algorithms:

In cryptography key exchange algorithms (key agreement protocols/key negotiation schemes) allow cryptographic keys to be exchanged between two parties, allowing the use of a cryptographic algorithm, in most cases symmetric encryption cipher. For example, when a laptop connects to the home WiFi router, both parties agree on a session key, used to symmetrically encrypt the network traffic between them. Most key-exchange algorithms are based on public-key cryptography and the math behind this system: discrete logarithms, elliptic curves or other. Anonymous key exchange, like Diffie–Hellman (DHKE and ECDH), does not provide authentication of the parties, and is thus vulnerable to man in the middle(MITM) attacks, but is safe from traffic interception (sniffing) attacks. Authenticated key agreement schemes authenticate the identities of parties involved in the key exchange and thus prevent man-in-the-middle attacks by use of digitally signed keys (e.g. PKI certificate), password-authenticated key agreement or other method.

1.6 Diffie-Hellman:

Diffie–Hellman Key Exchange (DHKE) is a cryptographic method to securely exchange cryptographic keys (key agreement protocol) over a public (insecure) channel in a way that overheard communication does not reveal the keys. The exchanged keys are used later for encrypted communication (e.g. using a symmetric cipher like AES). DHKE was one of the first public-key protocols, which allows two parties to exchange data securely, so that if someone sniffs the communication between the parties, the information exchanged can be revealed. The Diffie–Hellman (DH) method is an anonymous key agreement scheme: it allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel.

The DHKE method is resistant to sniffing attacks (data interception), but it is vulnerable to man-in-the-middle attacks (attacker secretly relays and possibly alters the communication between two parties). The Diffie–Hellman Key Exchange protocol can be implemented using discrete logarithms (the classical DHKE algorithm) or using elliptic-curve cryptography (the ECDH algorithm). The DHKE protocol is based on the practical difficulty of the Diffie–Hellman problem, which is a variant of the well known in the computer science DLP (discrete logarithm problem), for which no efficient algorithm still exists.

Diffie-Hellman key exchange's goal is to securely establish a channel to create and share a key for symmetric key algorithms. Generally, it's used for encryption, password-authenticated key agreement and forward security. Password-authenticated key agreements are used to prevent man-in-the-middle (MitM) attacks. Forward secrecy-based protocols protect against the compromising of keys

by generating new key pairs for each session. Diffie-Hellman key exchange is commonly found in security protocols, such as Transport Layer Security (TLS), Secure Shell (SSH) and IP Security (IPsec). For example, in IPsec, the encryption method is used for key generation and key rotation. Even though Diffie-Hellman key exchange can be used for establishing both public and private keys, the Rivest-Shamir-Adleman algorithm, or RSA algorithm, can also be used, since it's able to sign public key certificates.

The most serious limitation of Diffie-Hellman in its basic form is the lack of authentication. Communications using Diffie-Hellman by itself are vulnerable to MitM. Ideally, Diffie-Hellman should be used in conjunction with a recognized authentication method, such as digital signatures, to verify the identities of the users over the public communications medium. Diffie-Hellman key exchange is also vulnerable to logjam attacks, specifically against the TLS protocol. Logjam attacks downgrade TLS connections to 512-bit cryptography, enabling an attacker to read and modify data that's passed through the connection. Diffie-Hellman key exchange can still be secure if implemented correctly. For example, logjam attacks won't work with a 2,048-bit key.

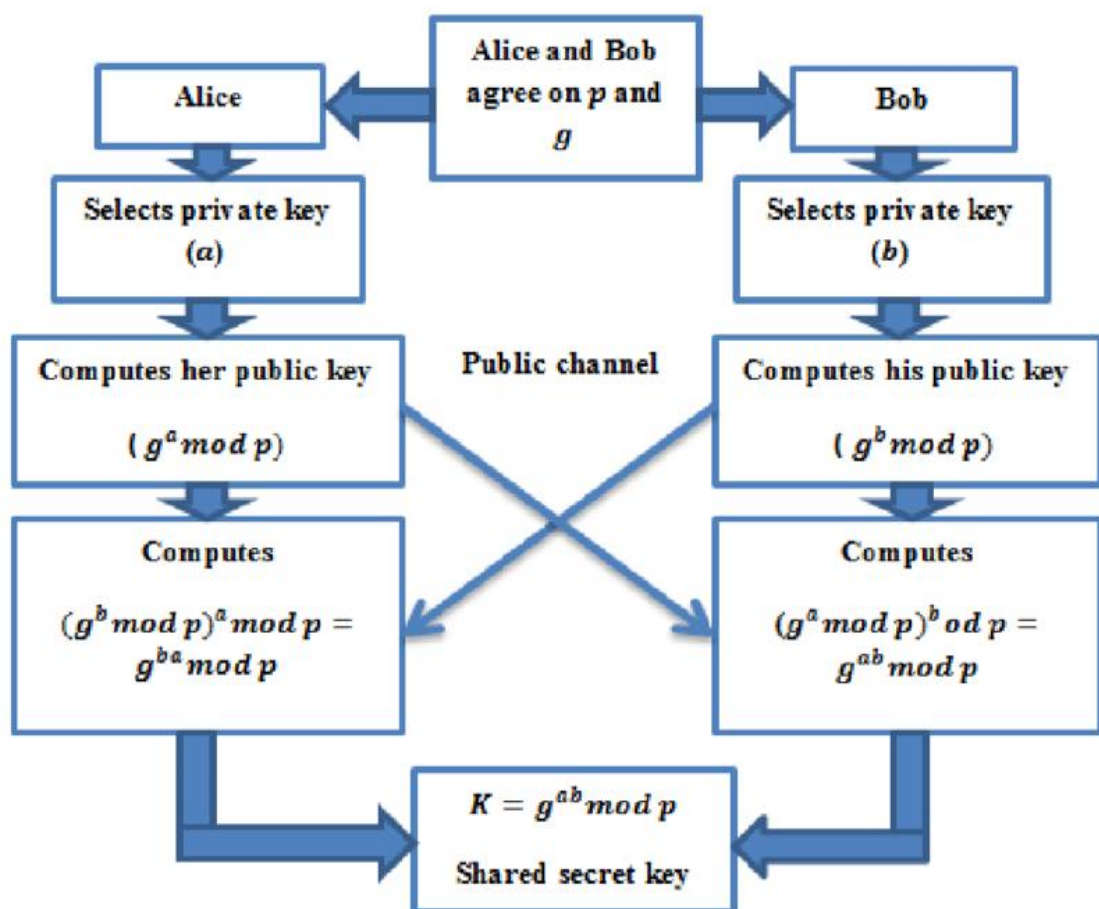


Figure 1.7 : Diffie-Hellman Key Exchange Algorithm

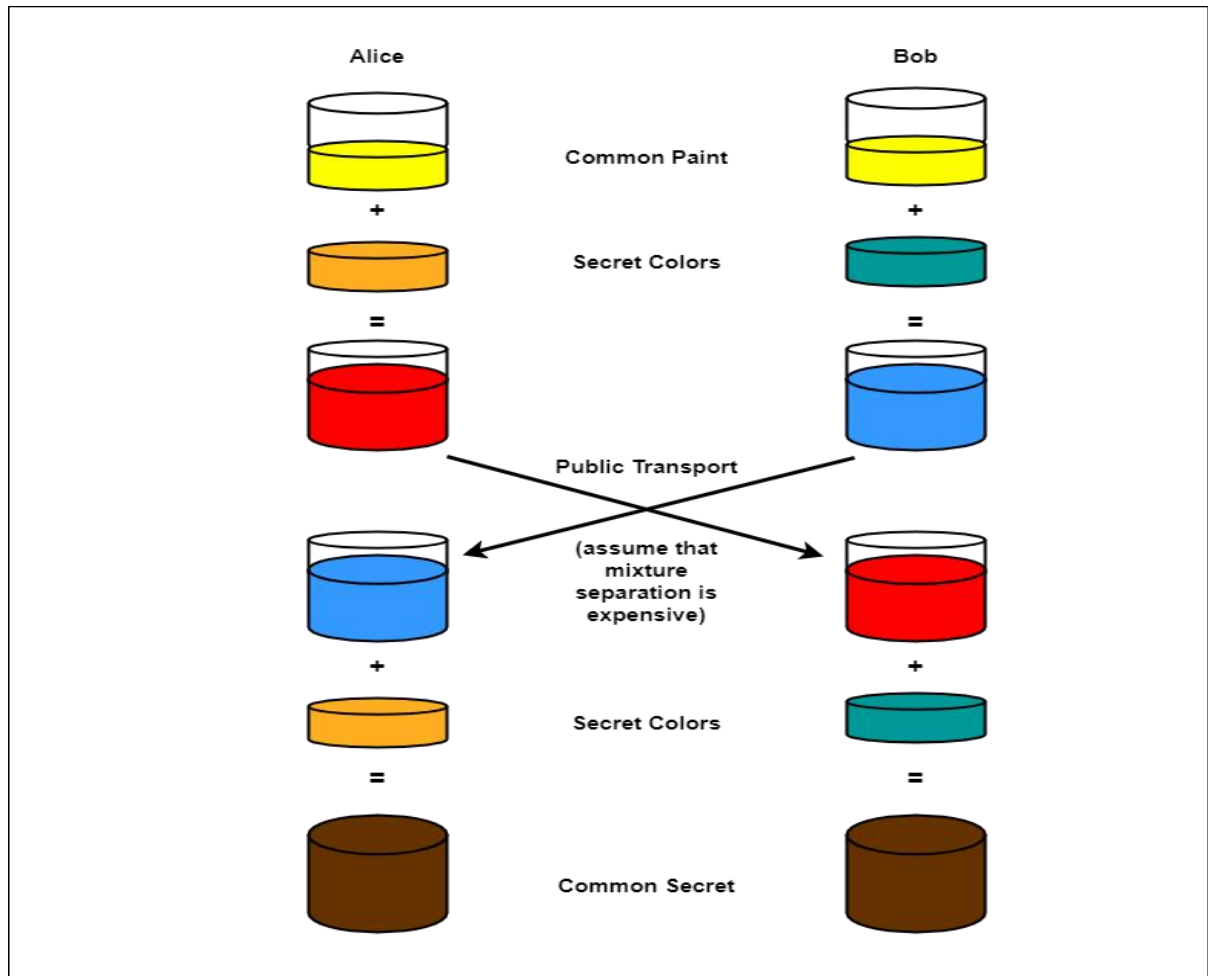


Figure 1.8 : Diffie-Hellman key exchange algorithm example using colors

1.7 Rivest–Shamir–Adleman (RSA):

The RSA cryptosystem is one of the first public-key cryptosystems, based on the math of the modular exponentiation and the computational difficulty of the RSA problem and the closely related integer factorization problem (IFP). The RSA algorithm is named after the initial letters of its authors (Rivest–Shamir–Adleman) and is widely used in the early ages of computer cryptography.

Later, when ECC cryptography evolved, the ECC slowly became dominant in the asymmetric cryptosystems, because of its higher security and shorter key lengths than RSA.

The RSA algorithm provides:

- **Key-pair generation:** generate random private key (typically of size 1024-4096 bits) and corresponding public key.
- **Encryption:** encrypt a secret message (integer in the range $[0 \dots \text{key_length}]$) using the public key and decrypt it back using the secret key.

- **Digital signatures:** sign messages (using the private key) and verify message signature (using the public key).
- **Key exchange:** securely transport a secret key, used for encrypted communication later.

RSA can work with keys of different keys of length: 1024, 2048, 3072, 4096, 8129, 16384 or even more bits. Key length of 3072-bits and above are considered secure. Longer keys provide higher security but consume more computing time, so there is a trade-off between security and speed. Very long RSA keys (e.g. 50000 bits or 65536 bits) may be too slow for practical use, e.g. key generation may take from several minutes to several hours.

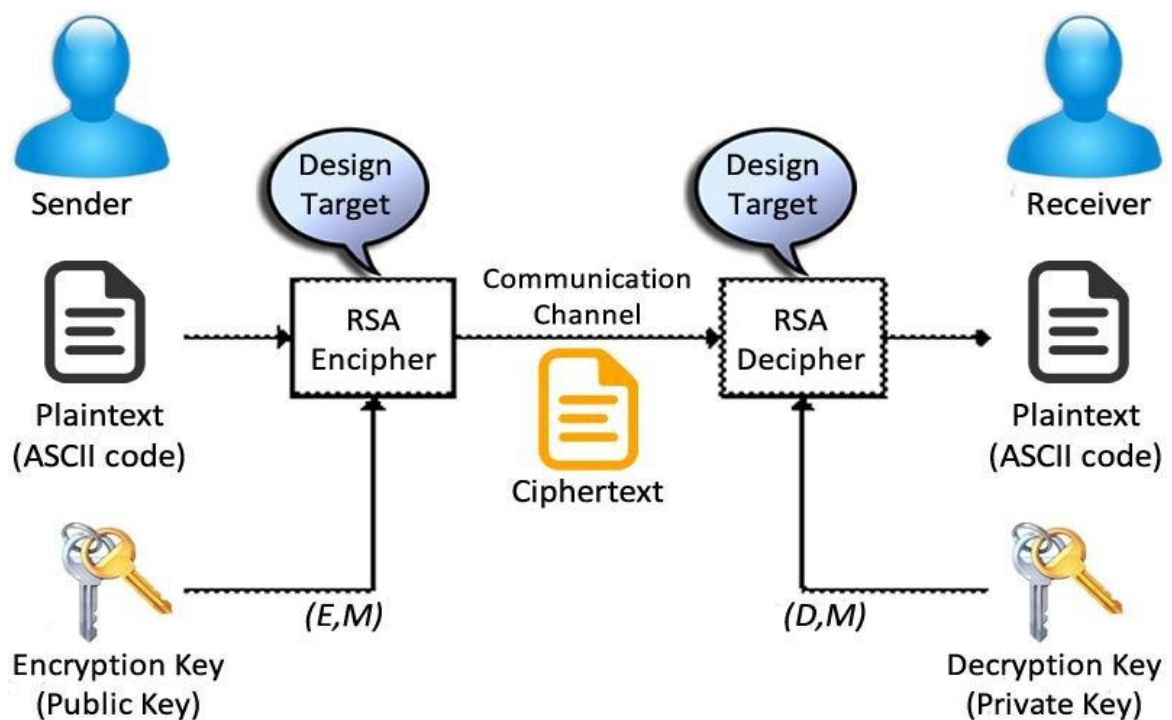


Figure 1.9 : Rivest–Shamir–Adleman Algorithm

1.8 Flask Application

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. To create flask application it is need to initialize the Flask object and then root each URL followed by it's corresponding function to that URL. Every routed function must return the displayable content on the web page.

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug

WSGI toolkit and Jinja2 template engine. Both are Pocco projects. It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases.

Jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages. Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have form a validation support. Instead, Flask supports the extensions to add such functionality to the application. Flask is pretty impressive. Because it has built-in development server and fast debugger, integrated support for unit testing, Restful request dispatching, Jinja2 templating, support for secure cookies, WSGI 1.0 compliant and Unicode based etc.

@app.route: App Routing means mapping the URLs to a specific function that will handle the logic for that URL. Modern web frameworks use more meaningful URLs to help users remember the URLs and make navigation simpler. Example: In our application, the URL ("/") is associated with the root URL.

This system consists of following URLs :

1. Root ('/')
2. encrypted data ('/encrypt')
3. input for RSA ('/rsa')
4. Keys and encrypted data ('/rsaa')

Root (/) : This will be the base URL that is opened directly when we open the web server (localhost). On this page Index.html template is rendered. This consists of basic interface with keys used and the text-area to enter the text. This page will have submit button which is an hyperlink for another page. If we submit data through that form then the POST request will be sent.

/encrypt : This function is responsible for calling encrypt_ECC and decrypt_ECC and to print the encrypted data. At the end this function will render the index.html with the back button. On this page ECC encryption.html template is rendered.

/rsa: On this page RSA.html template is rendered. This consists of basic interface with text-area to enter the text. This page will have submit button which is an hyperlink for another page.

/rsaa: This URL call the function that is responsible for RSA encryption and finally render the results.html page.

Running application : The Flask application is started by calling the run() function. The method should be restarted manually for any change in the code. To overcome this, the debug support is enabled so as to track any error.

1.9 WEB TECHNOLOGIES

1.9.1 HTML

HTML stands for Hyper Text Markup Language. HTML is the universal markup language for the Web. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. HTML was created by Tim Berners-Lee in 1991.

1.9.2 CSS

CSS Stands for Cascading Style Sheets. It is used to apply styles to web pages. Cascading Style Sheets are fondly referred to as CSS. It is used to make web pages presentable. The reason for using this is to simplify the process of making web pages presentable. It allows you to apply styles on web pages. It enables you to do this independently of the HTML that makes up each web page. Styling is an essential property for any website. It increases the standards and overall look of the website that makes it easier for the user to interact with it. So that is why CSS makes a huge important role in web page creation.

1.9.3 Java Script

JavaScript is the world's most popular lightweight, interpreted compiled programming language. It is also known as scripting language for web pages. It can be used for Client-side as well as Server-side developments. JavaScript can be added to your HTML file in two ways. One way is Internal JavaScript i.e adding JavaScript code directly to our HTML file the <script> tag. Another way is External JavaScript File i.e Create a file with .js extension and paste the JavaScript code inside it. After creating the file, add this file in <script src="file_name.js"> tag inside <head> tag of the HTML file.

Chapter – 2

LITERATURE SURVEY

Following research papers are studied in detail to understand the proposed recommendation technique and experimental result for predicting the output.

Paper 1: H. N. Almajed and A. S. Almogren, "SE-Enc: A Secure and Efficient Encoding Scheme Using Elliptic Curve Cryptography," in IEEE Access, vol. 7, pp. 175865-175878, 2019, doi: 10.1109/ACCESS.2019.2957943.

In this paper, author(s) discusses about a new cryptography technique using elliptic curve cryptography.

Many applications use asymmetric cryptography to secure communications between two parties. One of the main issues with asymmetric cryptography is the need for vast amounts of computation and storage. While this may be true, elliptic curve cryptography (ECC) is an approach to asymmetric cryptography used widely in low computation devices due to its effectiveness in generating small keys with a strong encryption mechanism. The ECC decreases power consumption and increases device performance, thereby making it suitable for a wide range of devices, ranging from sensors to the Internet of things (IoT) devices.

This work objective is to propose a trusted and proofed scheme that offers authenticated encryption (AE) for both encoding and mapping a message to the curve. In addition, this paper provides analytical results related to the security requirements of the proposed scheme against several encryption techniques. Additionally, a comparison is undertaken between the SE-Enc and other state-of-the-art encryption schemes to evaluate the performance of each scheme.

Cryptography is vulnerable to many well-known attacks that threaten the encryption process of these schemes. Examples include the known-plain-text attack (KPA), chosen-plain-text attack (CPA), cipher-text-only attack (COA), and chosen-cipher-text attack (CCA). The first attack, KPA, can occur when an adversary has the ability to obtain the plain-text and its corresponding cipher-text, as a consequence of which the adversary attempts to obtain the secret key. The second attack, CPA, can occur when an adversary has the ability to choose random plain-texts transmitted for encryption, and then to obtain the corresponding cipher-texts. Therefore, the adversary in the CPA attempts to reduce the security of the scheme. In the third attack, COA, the adversary is assumed to have access only to a set of cipher-texts, meaning that they can extract the plain-text or the secret key. Finally, the CCA is characterized by an adversary's attempt to acquire information from plain-texts by obtaining the decryption of selected cipher-texts. Therefore, the adversary attempts to obtain the secret key used to decrypt the message.

The main focus and contribution of this work is to offer an AE scheme using ECC as following: Describe the security flaws in ECC encoding and mapping phases. These flaws fall under several encryption attacks such as COA and CPA. Secure message encoding phase by applying Block Cipher Modes of Operation that resistant to COA, KPA, CPA, Replay Attack and Malleability Attack. Provide a comprehensive study of the padding step that significantly affect the performance of the scheme in the encoding phase. Provide security analysis for the proposed scheme that satisfying the security requirements in the second item.

The Elliptic Curve Cryptography (ECC) is modern family of public-key cryptosystems, which is based on the algebraic structures of the elliptic curves over finite fields and on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP). The ECC cryptography is considered a natural modern successor of the RSA cryptosystem, because ECC uses smaller keys and signatures than RSA for the same level of security and provides very fast key generation, fast key agreement and fast signatures.

ECC digital signature algorithms like ECDSA (for classical curves) and EdDSA (for twisted Edwards curves).

In point multiplication a point P on the elliptic curve is multiplied with a scalar k using elliptic curve equation to obtain another point Q on the same elliptic curve, i.e $kP=Q$.

public key(point)=private key (integer)*generator point (fixed constant, a base point on the EC)

Point multiplication is achieved by two basic elliptic curve operations:

- **Point addition**, adding two points J and K to obtain another point L , i.e ($L = J + K$).
- **Point doubling**, adding a point J to itself to obtain another point L , i.e ($L = 2J$).

ECC consists of several phases to exchange keys and secure communications. These phases are used separately and/or together, and they can be listed as follows: Generating parameters such as defining the elliptic curve, and calculating P_r and P_u ; numerical encoding of the message (for encryption); hashing the message (for signing); mapping the encoded message to an elliptic curve.

Paper 2: Islam, SK Hafizul, and G. P. Biswas. "Provably secure and pairing-free certificate-less digital signature scheme using elliptic curve cryptography." *International Journal of Computer Mathematics* 90.11 (2013): 2244-2258.

The digital signature is the most fundamental tools used in public key cryptography (PKC)/public key infrastructure (PKI) in order to accomplish the message integrity, authenticity and non-repudiation, when the messages are exchanged over any public channel. The notion of PKC was first proposed by Diffie and Hellman in 1976, based on which several digital signature schemes have been proposed . Out of all, the RSA signature is one of them and its security mainly depends on the two prime numbers used and if these numbers are not sufficiently large like 1024-bit or more, the RSA signature is easily breakable. On the other hand, the ElGamal signature is unforgeable based on the decisional Diffie–Hellman assumption in large multiplicative group.

The digital signature schemes based on PKI need complex public key certificate management process in order to authenticate the public key, which decreases the applicability in real environments. Note that the generation, management, delivery, revocation, etc. of certificate need to bear high computing cost and the huge storage space. To defeat these troubles, Shamir introduces the idea of identity-based cryptosystem (IBC) in 1984, which eliminates the use of public key certificate as needed in PKI-based cryptosystems. In this scheme, a user's public key is computed from the publicly known identity of the user such as email identity, passport number, social security number, etc. and a trusted third party, called private key generator (PKG) is responsible to generate the corresponding private key of the user by binding user's identity and PKG's private key. However, the IBC schemes have an inherent problem known as private key escrow problem since PKG have the knowledge about the private keys of all users in the system and thus, the user secrecy can be compromised if PKG is not trusted.

In order to manage these problems, the certificate-less public key cryptography (CL-PKC), which captures the advantages of both the cryptosystems PKI and IBC, was proposed by Al-Riyami and Paterson. In this setting, like PKI the user chooses a secret key completely unknown to PKG and similar to IBC, and the PKG computes the identity-based long-term private key of the user, and these two secrets are combined together to form the full private key of the user. Thus, the needs of public key certificate in PKI and the private key escrow problem exists in IBC are abolished in the CL-PKC cryptosystems. In CL-PKC, the identity-based public key is easily computable from the identity of the user and the PKI-based public key is easily accessible from the directory of the corresponding user. In the following, we briefly address some of the efficient certificateless digital signature (CL-DS) schemes and their shortcomings.

The ECC was proposed by Miller and Koblitz, and its security is based on the difficulty of solving the ECDLP. Any cryptosystem based on ECC provides high security with small key size, for example, a 160-bit ECC is considered to be as secured as 1024-bit RSA key. Let F_q be the field of integers modulo a large prime number q . A non-singular elliptic curve $E_q(a, b)$ over F_q is defined by the following equation: $y^2 \bmod q = (x^3 + ax + b) \bmod q$ (1), where $a, b, x, y \in F_q$ and $4a^3 + 27b^2 \bmod q \neq 0$. A point $P(x, y)$ is an elliptic curve point if it satisfies Equation (1), and the point $Q(x, -y)$ is called the negative of P , i.e. $Q = -P$. Let $P(x_1, y_1)$ and $Q(x_2, y_2)$ ($P \neq Q$) be two points in Equation (1), the line l (tangent line to Equation (1) if $P = Q$) joining the points P and Q intersects the curve (1) at $-R(x_3, -y_3)$ and the reflection of $-R$ with respect to x -axis is the point $R(x_3, y_3)$, i.e. $P + Q = R$. The points $E_q(a, b)$ together with a point O (called point at infinity) form an additive cyclic group G_q , that is, $G_q = \{(x, y): a, b, x, y \in F_q \text{ and } (x, y) \in E_q(a, b)\} \cup \{O\}$ of prime order q . The scalar point multiplication on the group G_q can be computed as follows: $kP = P + P + \dots + P$ (k times). A point P has order n if n is the smallest positive integer such that $nP = O$.

Definition 1: A function $\epsilon(k)$ is said to be negligible if, for every $c > 0$, there exists k_0 such that $\epsilon(k) \leq 1/k^c$ for every $k \geq k_0$.

Definition 2: Elliptic curve discrete logarithm problem (ECDLP): Given a random instance $(P, Q) \in G_q$, find a number $a \in \mathbb{R} Z_q^*$ such that $Q = aP$. The probability that a polynomial time bounded algorithm B can solve the ECDLP problem is defined as advanced ECDLP $B, G_q(k) = \Pr[B(P, Q) = a : a \in \mathbb{R} Z_q^*, Q = aP]$.

Definition 3: Elliptic curve discrete logarithm assumption: For every probabilistic polynomial time-bounded algorithm B , advanced ECDLP $B, G_q(k)$ is negligible, i.e. advanced ECDLP $B, G_q(k) \leq \epsilon$ for some negligible function ϵ .

- **Setup:** The PKG runs this algorithm, which takes the security parameter $k \in \mathbb{Z}^+$ as inputs and outputs the system's parameter and a master secret key, msk , of PKG. Note that the system's parameter is known to all the users whereas the msk is kept secret by PKG.
- **Set-Secret-Value:** This algorithm is run by every user in the system separately to generate a secret value for oneself. It takes the system's parameter and an identity ID_i of the user i as inputs and then returns the secret key x_i as output to ID_i .

- **Partial-Private-Key-Extract:** The PKG runs this algorithm to generate the partial private key of the users. The inputs of this algorithm are the system's parameter, master key msk and an identity ID_i of the user i , and then it outputs the partial private key D_i of ID_i .
- **Set-Private-Key:** In order to generate the full private key, every user in the system executes this algorithm that needs the system's parameter, partial private key D_i and the secret value x_i of ID_i as inputs and it outputs the full private key sk_i for ID_i .
- **Set-Public-Key:** This algorithm is also run by every user independently in order to compute one's full public key intended for him. It takes the system's parameter and the secret value x_i of ID_i as inputs and then returns the full public key pk_i to ID_i as output.
- **CL-DS-Sign:** The signer ID_i runs this algorithm to obtain a signature on a given message. This algorithm takes system's parameter, full private key sk_i of ID_i and a message $m_i \in \{0, 1\}^*$ as inputs and then outputs a signature δ_i for the same message m_i .
- **CL-DS-Verify:** This algorithm is executed by the verifier in order to verify the message–signature pair generated by the signer. This algorithm takes system's parameter, full public pk_i of ID_i and a message–signature pair (m_i, δ_i) as inputs. It outputs true if the signature δ_i is valid, otherwise outputs false.

The rigorous security analysis of the proposed CL-DS scheme in the random oracle model is presented in this subsection. According to, two types of adversary A_I and A_{II} are present in any certificateless cryptosystem, the former adversary is nothing but a dishonest user who has the ability to replace the public key of any user with the value of his choice and the later adversary is a malicious PKG who can access the master key, but cannot replace the public key of any user. The existential unforgeability of the proposed CL-DS scheme is based on the intractability of ECDLP problem in the elliptic curve group against the adversaries under the adaptive chosen message and identity attacks. For proving the strong security of the proposed scheme, the following theorems are presented.

Theorem 1 The proposed pairing-free CL-DS scheme is existential unforgeable under the adaptive chosen message and identity attacks against the adversary A_I in the random oracle model provided the ECDLP problem is intractable by any polynomial time-bounded algorithm in the elliptic curve group.

In recent years, the CL-PKC scheme has received much attention of many researchers as it eliminates the certificate management problems occurring in traditional PKI and the private key escrow problem of IBC schemes. In this paper, we proposed an efficient CL-DS scheme using ECC for the message integrity, non-repudiation and authentication and it is proven to be existentially

unforgeable in the random oracle model under the adaptive chosen message and identity attacks against the different adversaries with different attack powers. However, it may be noted that the security of the proposed scheme is based on the intractability of ECDLP problem. The proposed scheme is easily implementable as no bilinear pairing and MTP hash function have been used. Security and computation cost comparisons of our signature scheme with other existing schemes prove to be secured and efficient. Due to the low computation cost and strong security features, the proposed scheme is applicable in the areas where the communication bandwidth, computation cost and storage space are limit.

Paper 3: Y. Luo, X. Ouyang, J. Liu and L. Cao, "An Image Encryption Method Based on Elliptic Curve Elgamal Encryption and Chaotic Systems," in IEEE Access, vol. 7, pp. 38507-38522, 2019, doi: 10.1109/ACCESS.2019.2906052.

Due to the potential security problem about key management and distribution for the symmetric image encryption schemes, a novel asymmetric image encryption method is proposed in this work, which is based on the elliptic curve ElGamal (EC-ElGamal) cryptography and chaotic theory. Specifically, the SHA-512 hash is firstly adopted to generate the initial values of chaotic system, and a crossover permutation in terms of chaotic index sequence is used to scramble the plain-image. Furthermore, the generated scrambled image is embedded into the elliptic curve for the encrypted by elliptic curve ElGamal which can not only improve the security but also can help solve the key management problems. Finally, the diffusion combined chaos game with DNA sequence is executed to get the cipher image. Experimental analysis and performance comparisons demonstrate that the proposed method has high security, good efficiency, and strong robustness against chosen-plaintext attack which make it have potential applications for the image secure communications.

In order to solve this problem, the asymmetric encryption requires that the encryption key should be different from the decryption key, and the decryption key cannot be calculated from the encryption key. The asymmetric encryption achieves the secure communication among multiple users, and distributing key on the unsecure channel can also be avoided. Two different types of keys namely the public key and the private key are required in asymmetric encryption. The public key can be made public, which is available for everyone while the private key can only be kept by receiver. The theory of asymmetric encryption. The key pair is firstly kept by receiver, and the public key is sent to transmitter. Then, transmitter encrypt the information with public key, where the public key can be sent in any way because the cipher information can be only decrypted successfully with the private key. Finally, the cipher information is sent to the receiver and decrypted with the private key. In the

asymmetric encryption, the private key is only kept by receiver, which can greatly facilitate key management and distribution.

Elliptic Curve Cryptography (ECC), as an significant asymmetric encryption technology. The theory of ECC is based on elliptic curve mathematics, which is the difficulty to calculate the discrete logarithm of the ellipse curve on the Abelian group by using rational points. The ECC has relative high security, short keys, and it is faster than the classical counterparts such as Ron Rives Adi Shamir Leonard Adleman (RSA) and Digital Signature Algorithm (DSA). Therefore, ECC has bring extensive attention in the fields of authentication, digital signature, secure communication and signal processing etc. Many methods of image encryption based on the characteristics of ECC are proposed. For example, a colour image encryption method is proposed in [10], which utilizes the combination of discrete chaotic map (2D-TFCDM) and Menezes-Vanstone ECC (MVECC). In this method, the keys and parameters are generated by the MVECC, and fractional 2D-TFCDM is used to scramble and diffuse the image.

The experiment results demonstrate that this method can resist various attacks. In [11], a method based on DNA encoding and elliptic curve Diffie-Hellman encryption (ECDHE) is introduced, in which the plain image is transformed into DNA matrix by DNA encoding. DNA addition operation is performed on each component, and the cipher image is obtained by ECDHE. Results show that this method has large key space and it can defend common attacks effectively.

Based on the ECC and the public key cryptosystem, Elliptic Curve ElGamal (EC-ElGamal) cryptosystem is introduced in [12], which is widely applied in the field of image encryption. In approach of [13], a method of colour image encryption based on chaotic systems and EC-ElGamal is presented. Firstly, the plain image is compressed for the purpose of grayscale expansion, and then the compressed image is encrypted by the improved four-dimensional cat map. Finally, the EC-ElGamal encryption algorithm is used for global expansion to obtain the cipher image. This method performs well in statistical analysis and differential attacks, which shows better security compared with other algorithms. Similarly, EC-ElGamal based method of image encryption is proposed in [14], where a new additive homomorphism in the EC-ElGamal cryptosystem is employed. Medical image encryption method using improved ElGamal encryption technique is presented in [15] in which the operation of embedding plaintext pixels into elliptic curve is discarded. This method can encrypt multiple pixels at the same time, and the results show that it has fast encryption speed.

As an excellent candidate for the key generation of cryptosystem, chaotic systems have characteristics of ergodicity, sensitivity to initial conditions, and long-term unpredictability, which are

applied in different fields. A plenty of chaos-based image encryption methods have been proposed, which include the intertwining Logistic map, the Chebyshev map, the Arnold cat map the Tent map, the Lorenz system, the spatiotemporal chaotic system, the hyper-chaotic system, the memristive chaotic system etc. Chaos-based design patterns are developed to make the encryption process more complicated, which have better encryption performance and capability in resisting different attacks. Furthermore, in, the concept of chaos game is proposed in the field of fractal. This is a fast-statistical method for analyzing the internal structure of DNA sequences. It uses a simple Iterative Function System (IFS) to generate fractal graphics in a two-dimensional plane. This method has been widely applied in many research areas such as biotechnology and signal processing.

There are two main processes in traditional image encryption, which are permutation and diffusion. In the permutation phase, correlation of adjacent pixels are reduced and the information entropy is increased. However, the tonal distribution of scrambled image is same as plain image, which is vulnerable to statistical attacks. In order to improve the security of the proposed method, the EC-ElGamal encryption is employed before diffusion operation. That is, there is no connection between permutation and diffusion, which makes separate attacks become more difficult, and only one round of encryption can achieve good performance. Suppose the size of the plain image I is $M \times N$. The flowchart of the proposed images encryption method. There are three main processes in the proposed encryption method, which include the crossover permutation, EC-ElGamal encryption and diffusion. Specially, EC-ElGamal is asymmetric encryption, so that the encryption keys and the decryption keys are different.

In this study, a novel image encryption method based on elliptic curve ElGamal and chaotic theory is proposed. Specifically, the SHA-512 hash is used to generate the initial values of the LTM, TSM and chaos game which reduces the strong correlations between adjacent pixels in plain image as well as resists the known-plaintext attack and chosen plaintext attack. Then, the proposed scrambled method is used to permute the plain-image, which is then embedded into elliptic curve to be further encrypted by EC-ElGamal cryptosystem. Moreover, the diffusion based on chaos game and DNA code is executed to get the final cipher, which can improve the randomness of the pixel distribution in advance. The comprehensive performance analysis demonstrates that the proposed method has high security and good efficiency. In the future work, we will focus on the optimization of time consumption, which aims to better satisfy the requirement of real-time communications.

Paper 4: Abitha, K. S., Anjalipandey, A., & Kaliyamurthie, D. K. P. (2015). "Secured data transmission using elliptic curve cryptography". IJIRCCE, 3(3), 1-7.

Secured data transmission using elliptic curve cryptography can be defined as transmission of data. This paper proposes an survey about Secured data transmission using elliptic curve cryptography. Efficiency, and reliability will be increased for each transmission of data, While enclosing the proposed method by using the ECC algorithm which allow itself to encrypt and decrypt the data that is to be transferred and performs the active classification, we are concluding that the Secured data transmission using elliptic curve cryptography provide a efficiency higher than DSDV when compared with AODV. In a network environment, authorized users may access data and information stored on written for the client process, which initiates the communication, and for the server process, which waits for the communication to be initiated. Both endpoints of the communication flow are implemented as network sockets; hence network programming is basically socket programming. Networks are often classified by their physical or organizational extent or their purpose. Usage, trust level, and access rights differ between these types of networks. Some of these networks are: personal area network (PAN), local area network (LAN), home area network (HAN), storage area network (SAN), campus area network (CAN), backbone network, Metropolitan Area Network (MAN), Wide Area Network (WAN), enterprise private network, virtual private network (VPN), Virtual Network and finally Inter-network.

Secured data transmission using elliptic curve cryptography is based on the encryption and decryption, they are most widely used in video conferencing, confidentiality of data other social medias and they are the efficient one that deal with the confidentiality of information and are most widely used to analyze, find the methods for security of data. Generally, the cryptography techniques are classified as three categories: symmetric ciphers, asymmetric ciphers and key exchanges. Symmetric ciphers is based on the size of the key and the same keys are used to encrypt and decrypt data. Assymetric ciphers consist of two different keys where one is the public key and private key. The security is based upon the module and the exponent used. The next and most widely faced problem in Secured data transmission using elliptic curve cryptography is the eavesdropping and forging of data that was overcome by encryption and decryption. It also faces a tedious flaws during the and familiar with many related works, some of their problems are still continuing in the market, the problem may be estimated as the rating of items, and one of the important and main issue is the low performance that too in real time applications, other related issues may be the limited content analysis, data insecurity etc.

It provides a reliable and robust security environment for the operation of smart grid to emphasize economic, environmental, and social benefits using efficient security algorithm. It proposes the smart meters which are distributed in nodes of the SG. It achieves authentication and establishes the shared session key with Diffie-Hellman exchange protocol. Then, with the help of shared session key between smart meters and hash-based authentication code technique, the following messages can be authenticated in a lightweight way. It proposes a distributed data separation technique that occurs in smart meters that cover the entire routing environment. Homomorphic encryption is used for the security of data. It discusses key security technologies for a smart grid system, including public key infrastructures and trusted computing. It proposes an efficient and scalable key management protocol for secure uni-cast, multicast, and broadcast communications in a smart grid network. The proposed protocol is based on a binary tree approach, and supports all these three types of secure communications by using only one binary tree. The analysis and discussion show that the proposed protocol is versatile, and hence suitable for secure smart grid communications. A novel key management scheme which combines symmetric key technique and elliptic curve public key technique.

The symmetric key scheme is based on the Needham-Schroeder authentication protocol. We show that the known threats including the man-in-the-middle attack and the replay attack can be effectively eliminated under the proposed scheme. It is based on the hybrid recommendation system from the perspective of the types, architectures, and applications, algorithm to overcome the encryption and decryption using mesh topology. It proposes an idea to transmit only when a significant power consumption change occurs using link budget or signal processing algorithms. CAT, AMI, proposes efficient and privacy-preserving aggregation scheme, named EPPA, for smart grid communications using three algorithms like key generation, encryption and decryption. It proposes frequency agility-based interference avoidance algorithm, ZigBee protocol to detect interference and adaptively switch nodes to “safe” channel to avoid WLAN interference with small accuracy and small energy consumption.

In this Existing concepts Cryptography plays a significant role in improving the integrity and confidentiality of the data in SG. Many existing standard encryption algorithms and authentication schemes are adopted in SG. In that cryptography will not give full security to data transmission in wireless network. Wireless network gets more problem during packet transmission loss data as well as dropping data, so we can't prevent. In this paper having same problem. Symmetric cryptographic such as DES (Data Encryption Standard), Triple DES, AES (Advanced Encryption Standard) are widely employed in SG to efficiently defend against possible threats. This kind of algorithms compares with others very low security.

This paper propose a new way, that increases security considerations of the network using AODV algorithm for transfer of data and to increment the efficiency of AODV algorithm using ECC(Elliptic Curve Cryptography). Efficiency, and reliability will be increased for each transmission of data, While enclosing the proposed method by using the ECC algorithm which allow itself to encrypt and decrypt the data that is to be transferred and performs the active classification, we are concluding that the Secured data transmission using elliptic curve cryptography provide a efficiency higher than DSDV when compared with AODV. In this method we Elliptic curve cryptography (ECC) algorithm which allow itself to encrypt and decrypt and it performs Key generation is an important part where we have to generate both public key and private key. The sender will be encrypting the message with receiver's public key and the receiver will decrypt its private key., the proposed method will increase the efficiency of AODV protocol and it is useful that ECC is efficient in terms of the data file size and encrypted files. It will be useful to the military intelligence to transfer data by encrypting and decrypting data where only the source and the destination can view the information.

Paper 5: O. R. Merad Boudia, S. M. Senouci and M. Feham, "Elliptic Curve-Based Secure Multidimensional Aggregation for Smart Grid Communications," in IEEE Sensors Journal, vol. 17, no. 23, pp. 7750-7757, 1 Dec.1, 2017, doi: 10.1109/JSEN.2017.2720458.

In smart grid, data aggregation is considered as an essential paradigm in assessing information about current energy usage. To achieve the privacy-preserving goal, several homomorphic-based solutions have been proposed. However, these solutions either consider one-dimensional information or use costly pairing computation in order to ensure source authentication. In fact, smart grid data are likely to be multidimensional (e.g., time, purpose, and so on) for more accurate control. In addition, the aggregation node in smart grid needs to verify data that come from several smart meters in a residential area; hence, the verification must be cost-efficient. In this paper, we propose a scheme that considers multidimensional aggregation with privacy preserving and an efficient verification of smart grid data. The proposal is based on elliptic curve cryptography, along with homomorphic encryption and without pairings. The performance analysis shows the efficiency of the scheme for smart grid communications in comparison with existing schemes. For instance, we show that, when an aggregator node is responsible of 600 smart meters, it spends approximately 14 s to verify the data in pairing-based schemes, while only 0.3 s is needed for verification within the proposed scheme.

A conceptual smart grid model from NIST is presented. The SG consists of seven domains: Transmission, Distribution, Operations, Generation, Markets, Customer, and Service Provider. One of the most important components in SG is the Smart Meter (SM) in the customer domain, which is responsible for recording the electricity use and sending the information to the utility company. In

order to improve the efficiency and reliability, the SG considers two-way communication between the utility company and its customers, which improves sustainability and security as well. For instance, customers can see, using SM, their electricity use on a real-time basis. As a result, they can reduce their use during peak periods when it is more expensive, which in one hand saves money and on the other hand decreases the pressure on the central power grid. In smart grid, data aggregation is an essential paradigm to efficiently manage the energy supply. In fact, for billing, the Control Center (CC) at the operations domain needs individual data that are collected over long intervals. However, the CC requires data that are collected much more frequently for monitoring and control. So, the huge amount of real-time data, in this case, can be aggregated in the network before being processed and used. Consequently, by reducing the traffic, data aggregation can improve the network's efficiency.

We employ MRES, a Multi-Recipient Encryption Scheme to secure multidimensional data. The multi-dimensional data allow CC to use other information than electricity use for more accurate control. The encryption scheme considered is ECEG (Elliptic Curve El-Gamal) for its efficiency in terms of computation and communication. In addition, a reference technique is introduced to overcome the ECEG issues, we adopt ECDSA (Elliptic Curve Digital Signature Algorithm) with batch verification to allow intermediate nodes to efficiently verify data integrity and authenticate the senders, the analysis and performance evaluation show that compared with existing secure aggregation schemes for SG communications, the scheme significantly reduces the computation cost and communication overhead.

ECC becomes an attractive area of research in the last twenty years. The major benefits of using ECC are the highest strength-per-bit provided and the smallest key size. The security of ECC is based directly on the intractability of ECDLP (the Elliptic Curve Discrete Logarithm Problem). ECC is very useful for wireless communications and low power devices. In fact, compared with traditional cryptosystem like RSA or $\text{mod } p$ systems, ECC provides the same level of security with reduced key size. For example, an elliptic curve over a 160-bit field currently provides the same level of security as a 1024-bit RSA modulus or Diffie-Hellman prime. The difference becomes even more dramatic as the desired security level increases. In this paper, we use ECEG. More specifically, we use the elliptic curve analog of the Multi-Recipient El-Gamal Encryption Scheme (MRES) in which the sender re-uses the random coin to encrypt different plain texts under different public keys. MRES approximately halves the computational cost (number of exponentiation) for encryption as compared to the naive method.

The encryption in this scheme is an ECEG encryption. The security of ECEG is based on ECDLP, which makes the encryption secure if ECDLP is intractable. In this work, we consider a

security level of 160 bits, the same provided by RSA with 1024 bits key. Also, the ECEG is IND-CPA secure, so the encryption is secure against any form of ciphertext analysis. The reference technique used in the proposal is only considered in order to speed-up encryption and decryption operations. Roughly speaking, this technique does not negate the security of ECEG. In fact, recall that the encrypted data is the difference $d_i - f_i$, so even if the attacker has f_i , he cannot deduce the difference $d_i - f_i$ nor d_i . The privacy-preserving is guaranteed for all types of data because for El-Gamal, it is safe (w.r.t IND-CPA) to encrypt data under different public keys with the same randomness. Furthermore, AGG_j performs aggregation directly on cipher texts, so even if an adversary intrudes the AGG_j 's database, it cannot deduce the individual user's data. Also, CC performs decryption on aggregated cipher texts, and retrieves aggregated plain texts, so even if an adversary intrudes the CC's database, it still cannot deduce the individual user's data. Therefore, the Privacy-Preserving is ensured in our scheme.

Chapter – 3

REQUIREMENTS SPECIFICATION

3.1 Software Requirements

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view.

3.1.1 Operating System

Windows is a graphical operating system developed by Microsoft. It allows users to view and store files, run the software, play games, watch videos, and provides a way to connect to the internet. It was released for both home computing and professional works.

Mac OS is the computer operating system (OS) for Apple desktops and laptops. It is a proprietary graphical OS that powers every Mac. OSes interact with a computer's hardware, allocating the resources necessary to complete tasks given to it, for example, running an application. OSes allocate resources including memory, processing power and file storage.

Linux is an operating system. In fact, one of the most popular platforms on the planet, Android, is powered by the Linux operating system. An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn't function.

3.1.2 SDK

SDK stands for software development kit or devkit for short. It's a set of software tools and programs used by developers to create applications for specific platforms. SDK tools will include a range of things, including libraries, documentation, code samples, processes, and guides those developers can use and integrate into their own apps. SDKs are designed to be used for specific platforms or programming languages.

SDK used in this project is : Flask.

3.2 Hardware Requirements :

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating

system or application. The following sub-sections discuss the various aspects of hardware requirements. The Hardware Interfaces Required are:

1. Ram: Minimum 8GB or higher
2. GPU: 4GB dedicated
3. SSD: 128GB

Processor : Intel i5 10th Gen or Ryzen 5 with Octa core.

3.3 Non-Functional Requirements :

Non-Functional Requirements are the constraints or the requirements imposed on the system. They specify the quality attribute of the software. Non- Functional Requirements deal with issues like scalability, maintainability, performance, portability, security, reliability, and many more. Non-Functional Requirements address vital issues of quality for software system. It includes below things: Capacity, Availability and Performance etc.

3.4 Python Libraries To be installed:

- tinyec
- pycryptodome
- crypto_commons
- rsa
- cryptography

Chapter – 4

METHODOLOGY

The proposed Main Project is divided into the following four modules keeping every module consisting of main functionality in the whole project.

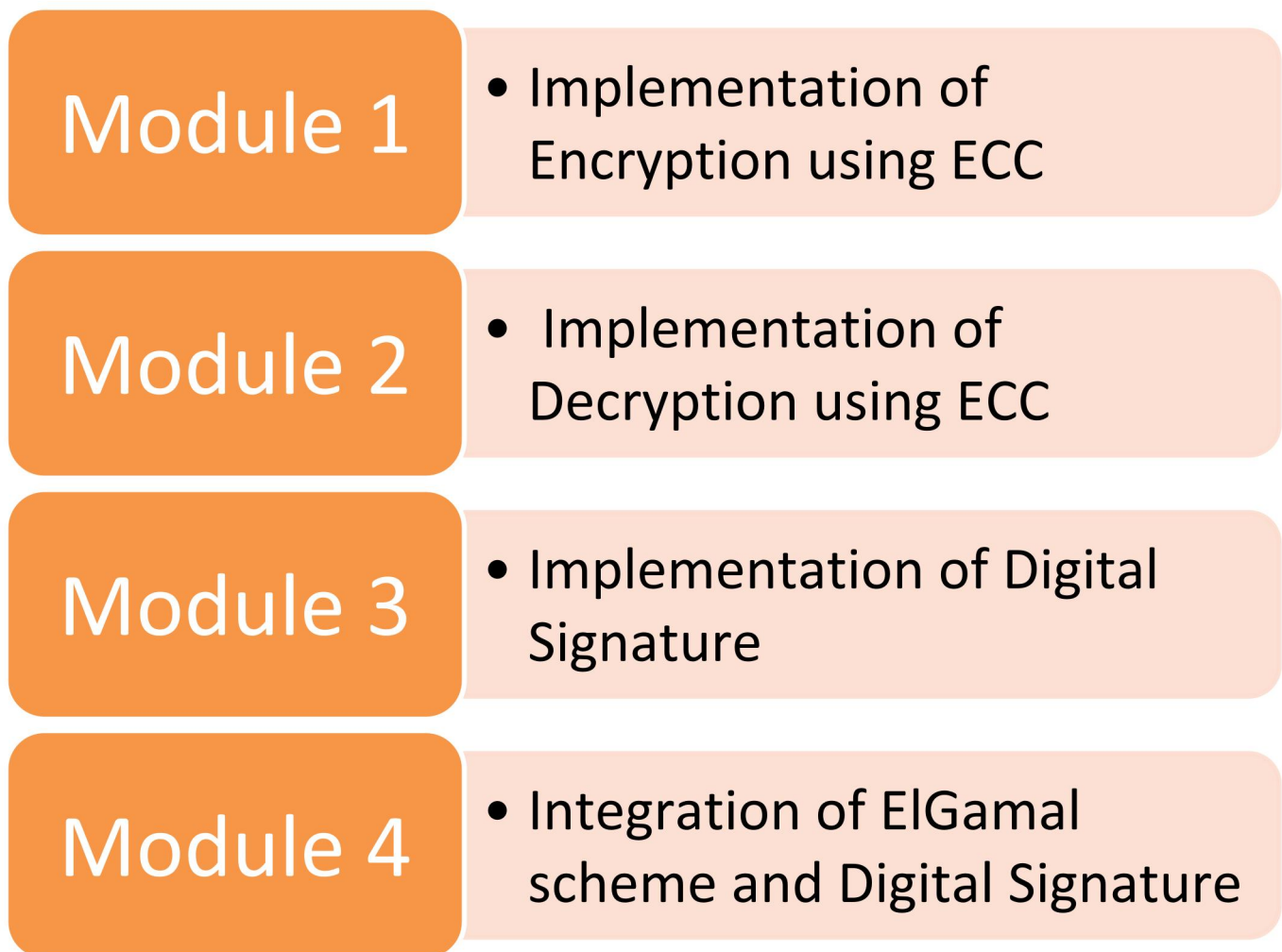


Figure 4.1 : List Of Modules Involved In The Proposed Model

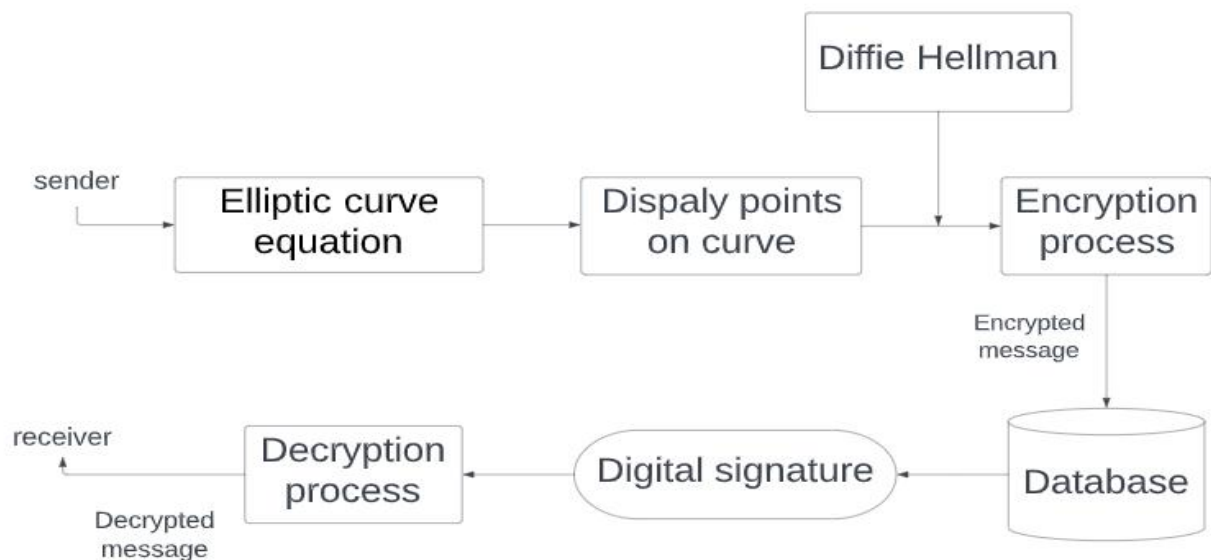


Figure 4.2 : System Architecture Of Proposed Model

4.1 Implementation of Encryption using ECC

Encryption is the method by which information is converted into secret code that hides the information's true meaning. The science of encrypting and decrypting information is called cryptography. In computing, unencrypted data is also known as plaintext, and encrypted data is called ciphertext. The formulas used to encode and decode messages are called encryption algorithms, or ciphers. Encryption plays an important role in securing many different types of information technology (IT) assets. It provides the following:

- **Confidentiality** - the protection of information in the system so that an unauthorized person cannot access it.
- **Authentication** - verifies the origin of a message.
- **Integrity** - proves the contents of a message have not been changed since it was sent.
- **Non-repudiation** - prevents senders from denying they sent the encrypted message.

Encryption is commonly used to protect data in transit and data at rest. Every time someone uses an ATM or buys something online with a smartphone, encryption is used to protect the

information being relayed. Businesses are increasingly relying on encryption to protect applications and sensitive information from reputation damage when there is a data breach. There are three major components to any encryption system: the data, the encryption engine and the key management. In laptop encryption, all three components are running or stored in the same place: on the laptop. In application architectures, however, the three components usually run or are stored in separate places to reduce the chance that compromise of any single component could result in compromise of the entire system.



Figure 4.3 : Encryption Algorithm

At the beginning of the encryption process, the sender must decide what cipher will best disguise the meaning of the message and what variable to use as a key to make the encoded message unique. The most widely used types of ciphers fall into two categories: symmetric and asymmetric. Symmetric ciphers, also referred to as secret key encryption, use a single key. The key is sometimes referred to as a shared secret because the sender or computing system doing the encryption must share the secret key with all entities authorized to decrypt the message. Symmetric key encryption is usually much faster than asymmetric encryption. The most widely used symmetric key cipher is the Advanced Encryption Standard, which was designed to protect government-classified information.

Asymmetric ciphers, also known as public key encryption, use two different -- but logically linked -- keys. This type of cryptography often uses prime numbers to create keys since it is computationally difficult to factor large prime numbers and reverse-engineer the encryption. The Rivest-Shamir-Adleman (RSA) encryption algorithm is currently the most widely used public key algorithm. With RSA, the public or the private key can be used to encrypt a message; whichever key is not used for encryption becomes the decryption key. Today, many cryptographic processes use a symmetric algorithm to encrypt data and an asymmetric algorithm to securely exchange the secret key.

Generally, ElGamal encryption consists of two components - Key generator and the encryption algorithm. In ECC, Encryption is an asymmetric key encryption algorithm which is based on

the Diffie–Hellman key exchange. Encryption is the method by which information is converted into secret code that hides the information's true meaning.

High-level view of this encryption consists of four phases: generating system parameters; encoding the message, mapping to the elliptic curve; encrypting the mapped points. The main focus and contribution of this module is to offer an asymmetric encryption using ECC with secure message encoding, mapping, and encryption. In addition to these phases, it is noteworthy that the first phase is a major step, and many proposed schemes neglect to consider the importance of having a shared key between the two parties to encrypt the message. This phase was included in the SE-Enc scheme because it can reasonably be viewed as a major phase for any system that needs to facilitate AE.

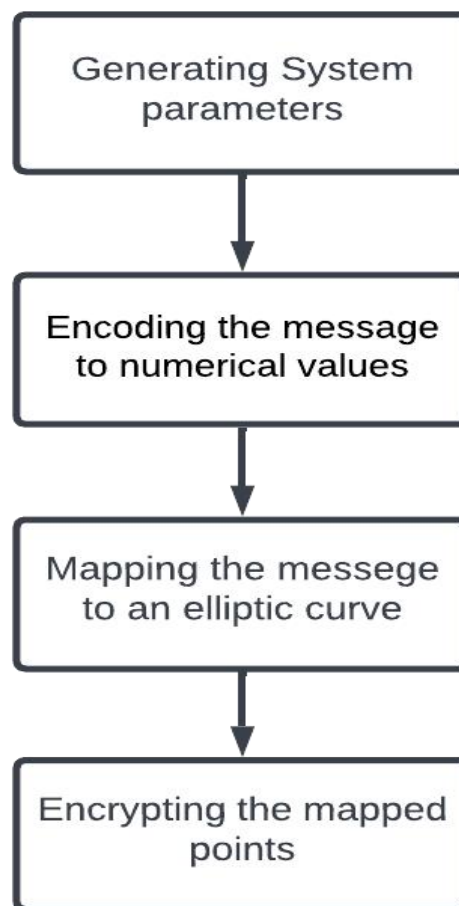


Figure 4.4 : Encryption Process In Proposed Model

4.1.1 Generating system parameters:

The main advantage of this phase is the generation of the shared secret key between the two parties. This key is used to encrypt the mapped points on the elliptic curve. It is necessary for the

sender to create a shared session key k_{sh} in order to encrypt mapped points on the elliptic curve. Thus, using their private key d_s , as well as the recipient's public key PU_r , the sender can generate k_{sh} . Benefiting from ECDLP, both the sender and the recipient can agree on a shared key. The private keys in the ECC are integers in the range of the curve's field size, typically 256-bit integers. The key generation in the ECC cryptography is as simple as securely generating a random integer in certain range, so it is extremely fast. Any number within the range is valid ECC private key. Following is the system generation notations used in this session.

4.1.2 Encoding the message to numerical values:

In this phase, encoding the message is done to overcome the security flaws. Each message is divided into several blocks B , where each block B contains N characters. The following equation is used to calculate N : $N \leq (p-8)/8$. The number of blocks B required is obtained by dividing the total number of characters in M by N , i.e, $B=M/N$. The reason for dividing the message to this length blocks from the need to encrypt each mapped point to the same length as p . In addition, one character is removed from each block to pad it with the 3 bits of zeros that are necessary for the mapping phase. For each message, after obtaining the blocks for the message M , each set of characters in each block is converted to its binary value. Afterward, the first block of binary values are XORed with the initial vector IV . Accordingly, each following blocks are XORed with previous XORed block. Finally, each XORed block is padded with 3bits for mapping in the next phase.

4.1.3 Mapping the message to an elliptic curve:

Mapping a message to an elliptic curve means that (x_i, y_i) satisfies the elliptic curve $y^2 \equiv x^3 + a*x + b \mod p$. Therefore, it is necessary to find the y_i value which corresponds to x_i for each point. For each block from the encoded message, the block containing a binary value is converted into a decimal value. Following that, using the generated EC equation we map this value to EC by find the correspond y_i . ECC crypto algorithms can use different underlying elliptic curves. Different curves provide different level of security (cryptographic strength), different performance (speed) and different key length, and also may involve different algorithms. ECC curves, adopted in the popular cryptographic libraries and security standards, have name (named curves, e.g. secp256k1 or Curve25519), field size (which defines the key length, e.g. 256-bit), security strength (usually the field size/2 or less), performance (operations/sec) and many other parameters. ECC keys have length, which directly depends on the underlying curve.

In most applications (like OpenSSL, Open SSH and Bitcoin) the default key length for the ECC private keys is 256 bits, but depending on the curve many different ECC key sizes are possible: 192-bit

(curve secp192r1), 233-bit (curve sect233k1), 224-bit (curve secp224k1), 256-bit (curves secp256k1 and Curve25519), 283-bit (curve sect283k1), 384-bit (curves p384 and secp384r1), 409-bit (curve sect409r1), 414-bit (curve Curve41417), 448-bit (curve Curve448-Goldilocks), 511-bit (curve M-511), 521-bit (curve P-521), 571-bit (curve sect571k1) and many others.

4.1.4 Encrypting the mapped points:

Many schemes overlook the encryption phase and assume that the mapping phase is sufficient to secure the message. However, this view is not correct, since mapping points to an elliptic curve means that the points are eligible to be multiplied with the private key to gain the ECDLP hardness. There are several ways to secure these points. In the SE-Enc scheme, these points are encrypted by adding each point to k_{sh} . Consequently, it is cryptographically hard to retrieve the mapped points without the shared key. Elliptic-curve cryptography (ECC) provides several groups of algorithms, based on the math of the elliptic curves over finite fields. ECC digital signature algorithms like ECDSA (for classical curves) and EdDSA (for twisted Edwards curves). ECC key agreement algorithms like ECDH, X25519 and FHEMQV. All these algorithms use a curve behind (like secp256k1, curve25519 or p521) for the calculations and rely on the difficulty of the ECDLP (elliptic curve discrete logarithm problem). All these algorithms use public / private key pairs, where the private key is an integer and the public key is a point on the elliptic curve (EC point). Let's get into details about the elliptic curves over finite fields.

4.2 Implementation of Decryption using ECC

Decryption is a technique that is used in cyber security that makes it challenging for hackers to intercept and read unauthorized information. While encryption is in place to protect the data, recipients must have the access to the appropriate decoding or decryption tool to access the original details. Let's explore what decryption entails. Decryption is the transformation of data that has been encrypted and rendered unreadable back to its unencrypted form. The garbled data is extracted by the system and converted and transformed into texts and images that are easily understandable by the reader as well as the system. Simply put, decryption is essentially the reverse of encryption, which requires coding data to make it unreadable, but the matching decryption keys can make it readable.



Figure 4.5 : Decryption Algorithm

The recipients must have the right decryption or decoding tools to access the original details. Decryption is performed using the best decryption software, unique keys, codes, or passwords. The original file can be in the form of text files, images, e-mail messages, user data, and directories. The original format is called plaintext while the unreadable format is referred to as ciphertext. Parties use an encryption scheme called an algorithm and keys for encryption and decryption of messages in a private conversation. The decryption algorithm is also known as a cipher. One of the primary reasons for having an encryption-decryption system in place is privacy. Information over the World Wide Web is subject to scrutiny and access from unauthorized users. Therefore, the data is encrypted to prevent data theft.

Here are some significant reasons why decryption is used:

- It helps secure sensitive information like login credentials like usernames and passwords.
- Provides confidentiality to private data.
- It helps ensure that the record or file remains unchanged.
- It avoids plagiarism and protects IP.
- It is beneficial for network communications like the internet where a hacker can gain access to unencrypted data.
- It lets one protect their data safely without the fear of someone else accessing it.

The person who is responsible for data decryption receives a prompt or window for a password to be entered to gain access to the encrypted information. Decryption is the method by which cipher text is converted into plain text. It is the reverse process of encryption. The message which is encrypted in module 1 is decrypted in this module. The following phases are the reverse of the previous phases.

High-level view of this decryption consists of three phases: decrypting the message; decoding the decrypted message; and converting the decoded message into plaintext.

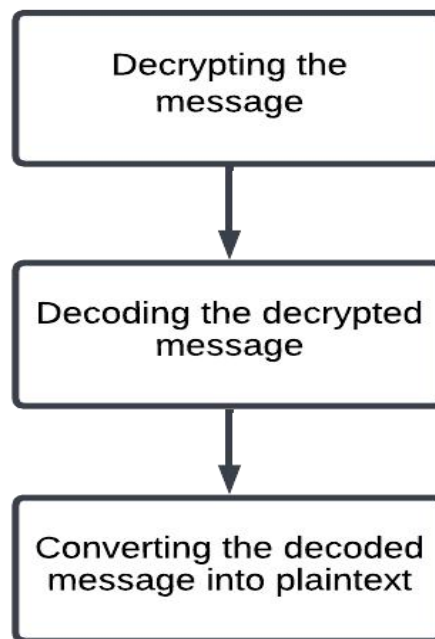


Figure 4.6 : Decryption Process In Proposed Model

4.2.1 Decrypting the message:

Similar to the encryption phase, the mapped points are obtained from the encrypted points using the shared key k_{sh} . In this phase the encrypted data which are the mapping points are decrypted, but this is not the original data, because sender encodes the data before mapping to the elliptic curve. This encrypted data is gone through the next phases to derive the original data.

4.2.2 Decoding the decrypted message:

The output of the previous phase is not the original data but it is a set of mapped points. These points consist of two pairs, represented by x_i and y_i . In encryption phase, For each block from the encoded message is mapped to the elliptic curve by using the pair (x_i, y_i) which satisfies the elliptic curve $y^2 \equiv x^3 + a*x + b$ and by using the prime factor p . The y_i value is only used in mapping the points to the elliptic curve. Therefore, the decoding phase is simply concerned with x_i , which is used to represent the binary values that are employed in the converting to plaintext phase.

4.2.3 Converting the decoded message to plaintext:

The final phase is concerned with converting the binary values into their corresponding characters. These characters represent the plaintext message M . In this phase the decoded values are converted to plaintext. The total process done on sender side is to send the data without effecting by

the security flaws, where as on the receiver side is to get the original message. Decoded message in previous phases is in the form of blocks containing binary values, so it is need to convert the decoded message to plain text. This is the the reverse process to the process in encoding the message.

4.3 Implementation of Digital Signature

A digital signature is an electronic, encrypted, stamp of authentication on digital information such as email messages, macros, or electronic documents. A signature confirms that the information originated from the signer and has not been altered. A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document. It's the digital equivalent of a handwritten signature or stamped seal, but it offers far more inherent security.

A digital signature is intended to solve the problem of tampering and impersonation in digital communications. Digital signatures can provide evidence of origin, identity and status of electronic documents, transactions or digital messages. Signers can also use them to acknowledge informed consent. Signature algorithm is used for authenticating a device or a message sent by the device. For example consider two devices A and B.

To authenticate a message sent by A, the device A signs the message using its private key. The device A sends the message and the signature to the device B. This signature can be verified only by using the public key of device A. Since the device B knows A's public key, it can verify whether the message is indeed send by A or not. ECDSA is a variant of the Digital Signature Algorithm (DSA) that operates on elliptic curve groups. For sending a signed message from A to B, both have to agree up on Elliptic Curve domain parameters. Generally, Elliptic Curve Digital Signature Algorithm (ECDSA) consists of two components - key generation, signature generation and signature verification. The private key is generated as a random integer in the range $[0...n-1]$. The public key is a point on the elliptic curve, calculated by the EC point multiplication.

Objective of digital signature in the proposed model.

- **Authenticity** - The signer is confirmed as the signer.
- **Integrity** - The content has not been changed or tampered with since it was digitally signed.
- **Non-repudiation** - Proves to all parties the origin of the signed content. Repudiation refers to the act of a signer denying any association with the signed content.curve.
- **Notarization** - Signatures in Microsoft Word, Microsoft Excel, or Microsoft PowerPoint files, which are time stamped by a secure time-stamp server, under certain circumstances, have the validity of a notarization.

High-level view of this decryption consists of three phases: signing the encrypted message; verifying the received message. AE schemes assure confidentiality and integrity of the transmitted message between two parties. In the this scheme, confidentiality is maintained throughout the previously mentioned phases. To maintain integrity, the sender signs the encrypted points using ECDSA. Signing and verifying is done after the four encryption phases done. Most public-key cryptosystems like RSA and ECC provide secure digital signature schemes (signature algorithms). Examples of well known digital signature schemes are: DSA, ECDSA, EdDSA, RSA signatures, Elgamal signatures and schnorr signatures. The above mentioned signature schemes are based on the difficulty of the DLP (discrete logarithm problem) and ECDLP (elliptic-curve discrete logarithm problem) and are quantum-breakable (powerful enough quantum computers may calculate the signing key from the message signature).

Benefits of digital signatures : personal identification numbers (PINs), passwords and codes, asymmetric cryptography, checksum, cyclic redundancy check, certificate authority (CA) validation, trust service provider (TSP) validation, time stamping, globally accepted and legally compliant, time savings, cost savings, positive environmental impact, traceability. Uses for digital signatures : government, healthcare, manufacturing, financial services, cryptocurrencies. Digital signature tools and vendors : Adobe Sign, DocuSign standards-based services, GlobalSign, SignEasy, SignNow, Vasco.

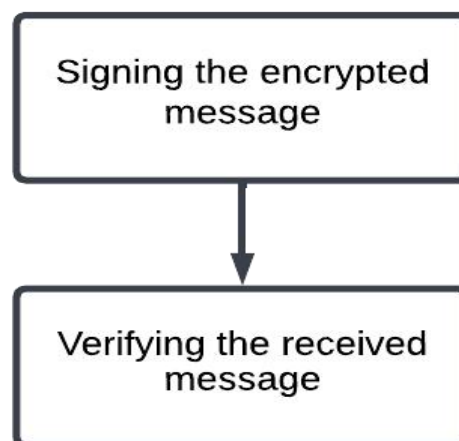


Figure 4.7 : Digital Signature Algorithm In Proposed Model

4.3.1 Signing the encrypted message:

The ECDSA signing algorithm (RFC 6979 takes as input a message + a private key and produces as output a signature, which consists of pair of integers $\{r, s\}$. The ECDSA signing algorithm

is based on the ElGamal signature approach. The signing process encodes a random point R (represented by its x-coordinate only) through elliptic-curve transformations using the private key $privKey$ and the message hash h into a number s , which is the proof that the message signer knows the private key $privKey$. The signature $\{r, s\}$ cannot reveal the private key due to the difficulty of the ECDLP problem.

ECDSA signatures are 2 times longer than the signer's private key for the curve used during the signing process. For example, for 256-bit elliptic curves (like secp256k1) the ECDSA signature is 512 bits (64 bytes) and for 521-bit curves (like secp521r1) the signature is 1042 bits.

4.3.2 Verifying the signed message:

The algorithm to verify a ECDSA signature takes as input as the signed message + the signature $\{r, s\}$ produced from the signing algorithm + the public key, corresponding to the signer's private key. The output is boolean value: valid or invalid signature. At signature verification, the message for verification is hashed (either alone or together with the public key) and some computations are performed between the message hash, the digital signature and the public key, and finally a comparison decides whether the signature is valid or not. The signature verification decodes the proof number s from the signature back to its original point R , using the public key $pubKey$ and the message hash h and compares the x-coordinate of the recovered R with the r value from the signature.

4.4 Integration of ElGamal scheme and Digital Signature

This is the final module where above three modules are integrated. In this module ElGamal scheme and Digital Signature are integrated to improve the security of the existing systems. The ECDSA (Elliptic Curve Digital Signature Algorithm) is a cryptographically secure digital signature scheme, based on the elliptic-curve cryptography (ECC). ECDSA relies on the math of the cyclic groups of elliptic curves over finite fields and on the difficulty of the ECDLP problem (elliptic-curve discrete logarithm problem).

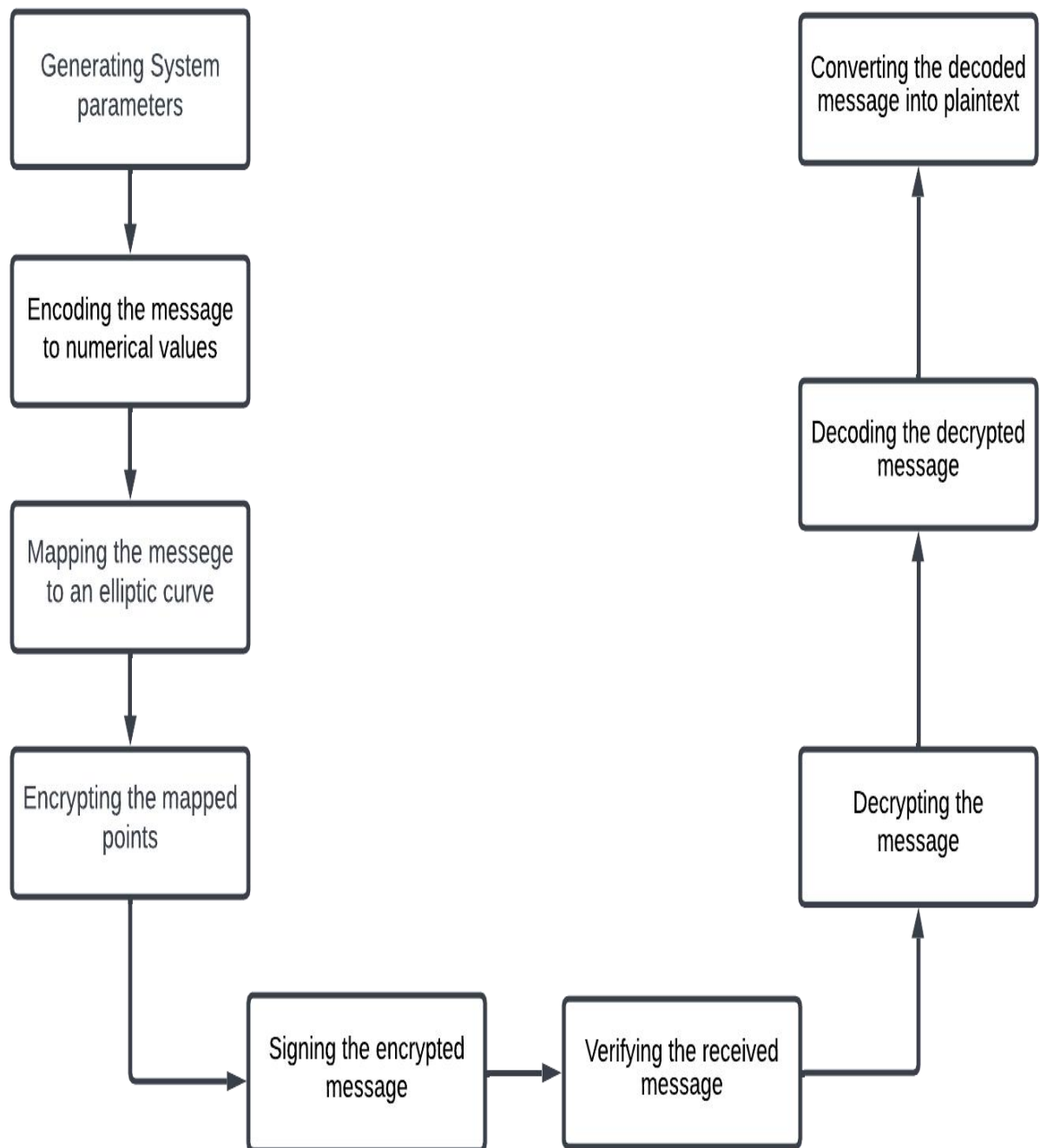


Figure 4.8 : Integration of ElGamal scheme And DSA (Proposed Model)

Chapter – 5

SYSTEM DESIGN

5.1 Introduction

Once the requirements document for the software to be developed is available, the software design phase begins. While the requirement specification activity deals entirely with the problem domain, design is the first phase of transforming the problem into a solution. In the design phase, the customer and business requirements and technical considerations all come together to formulate a product or a system.

The design process comprises a set of principles, concepts and practices, which allow a software engineer to model the system or product that is to be built. This model, known as design model, is assessed for quality and reviewed before a code is generated and tests are conducted. The design model provides details about software data structures, architecture, interfaces and components which are required to implement the system. This chapter discusses the design elements that are required to develop a software design model. It also discusses the design patterns and various software design notations used to represent a software design.

5.2 Class diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

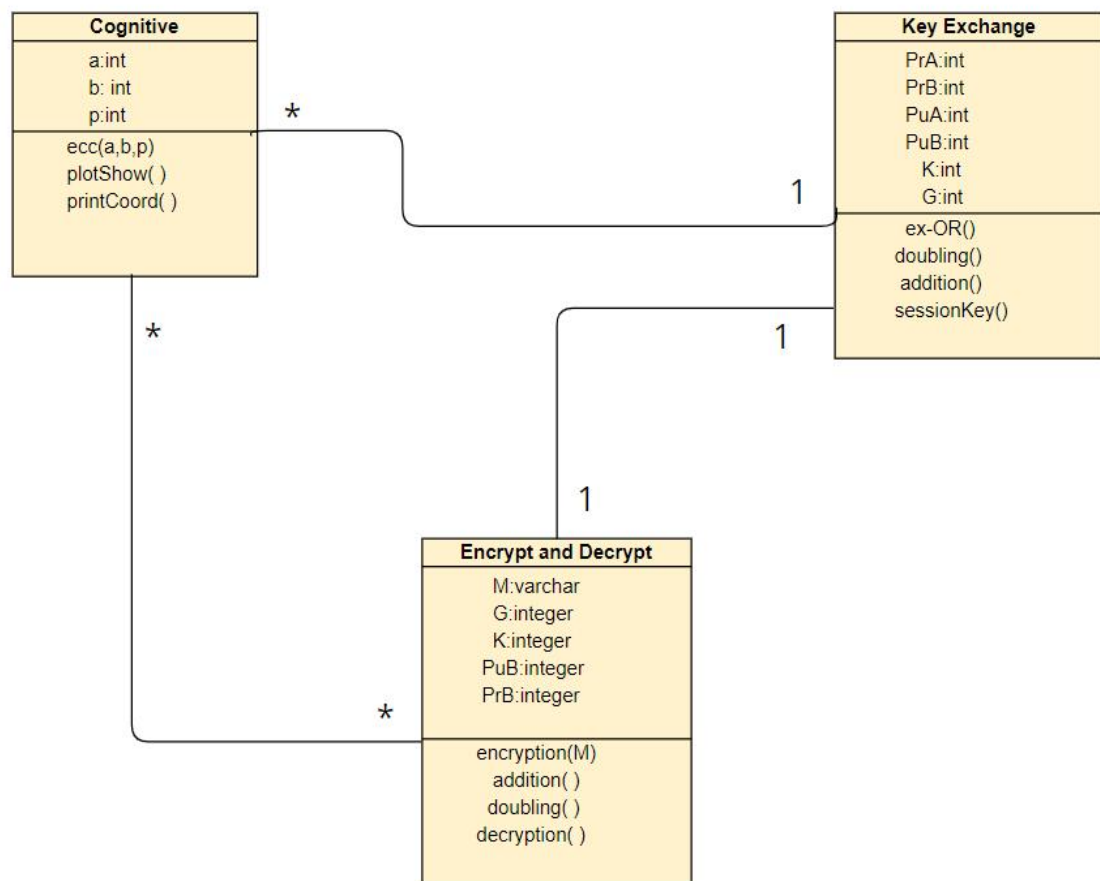


Figure 5.1 : Class Diagram For Proposed Model

5.3 Use case diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system. The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

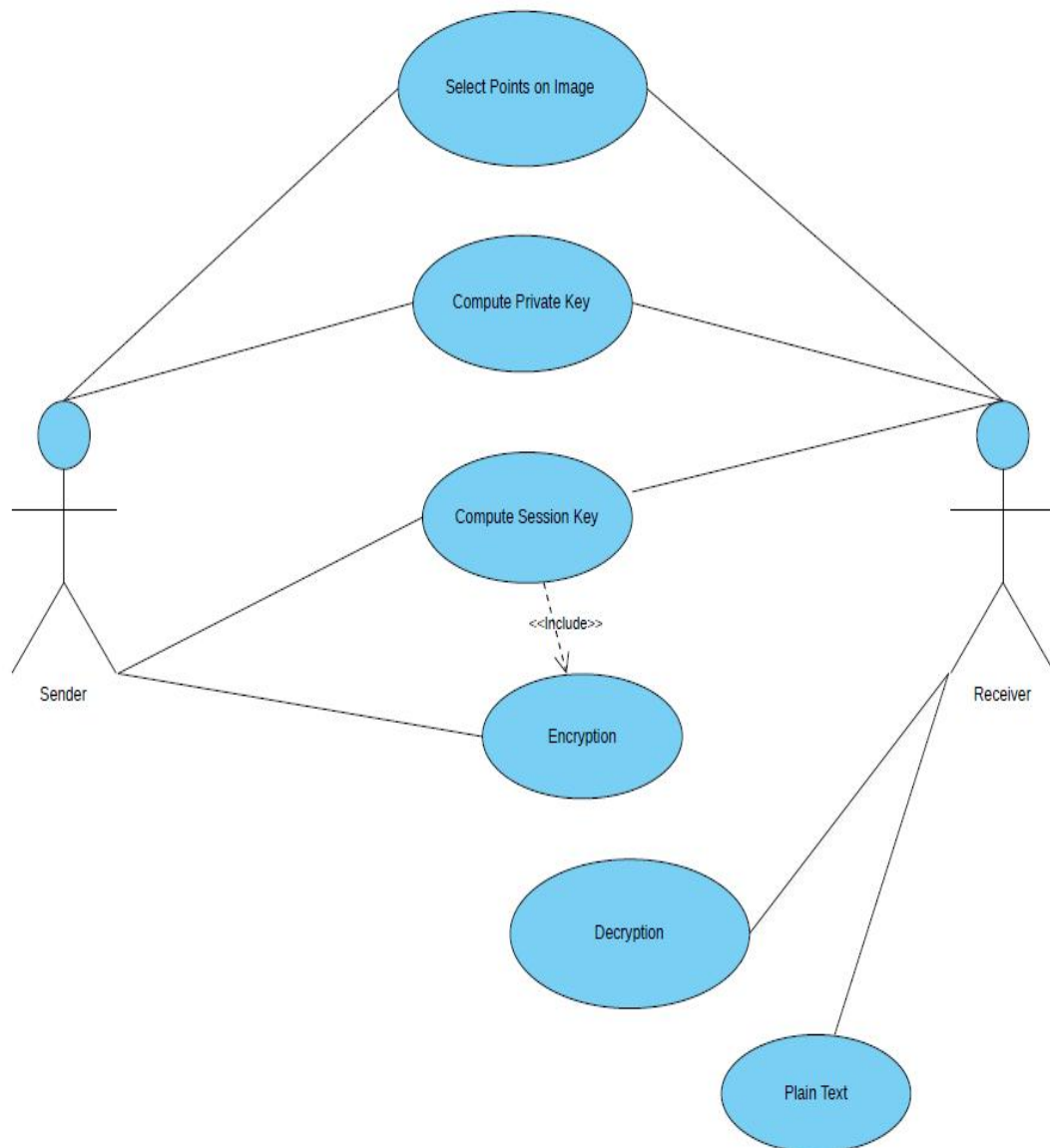


Figure 5.2 : Use Case Diagram For Proposed Model

5.4 Sequence diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

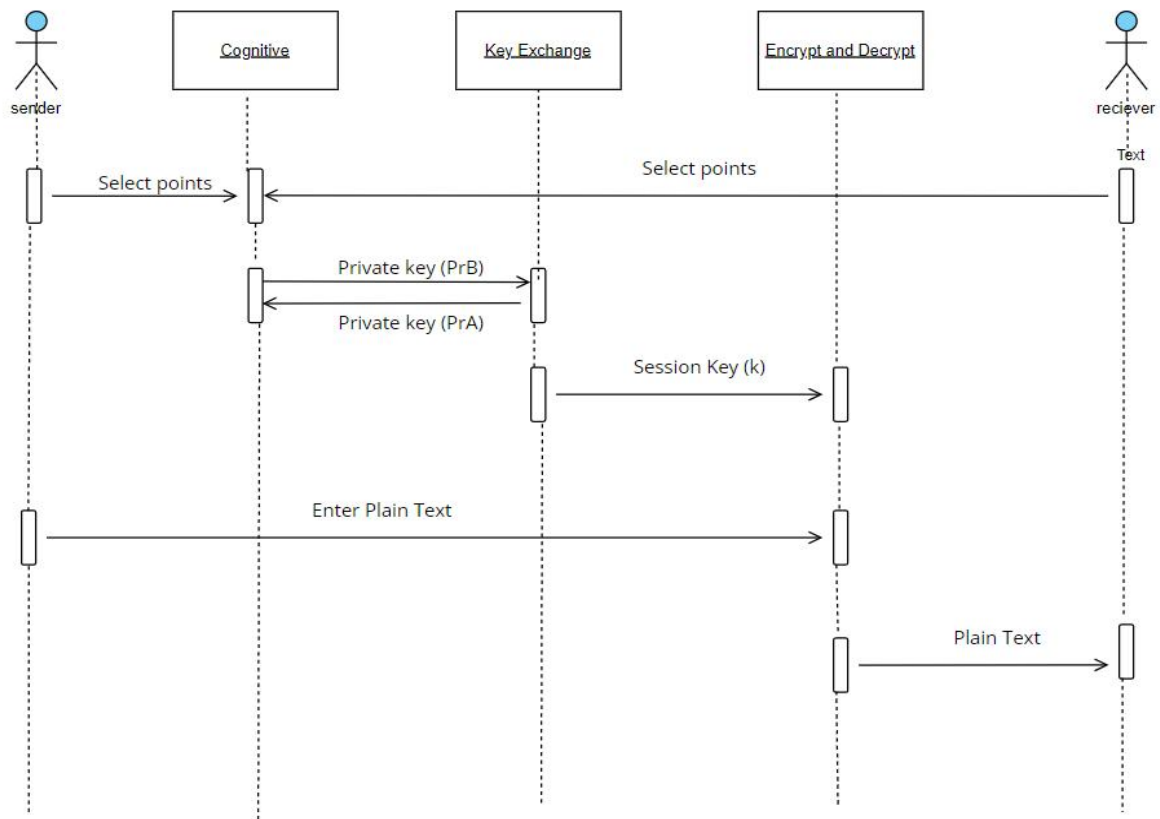


Figure 5.3 : Sequence Diagram For Proposed Model

5.5 Collaboration diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

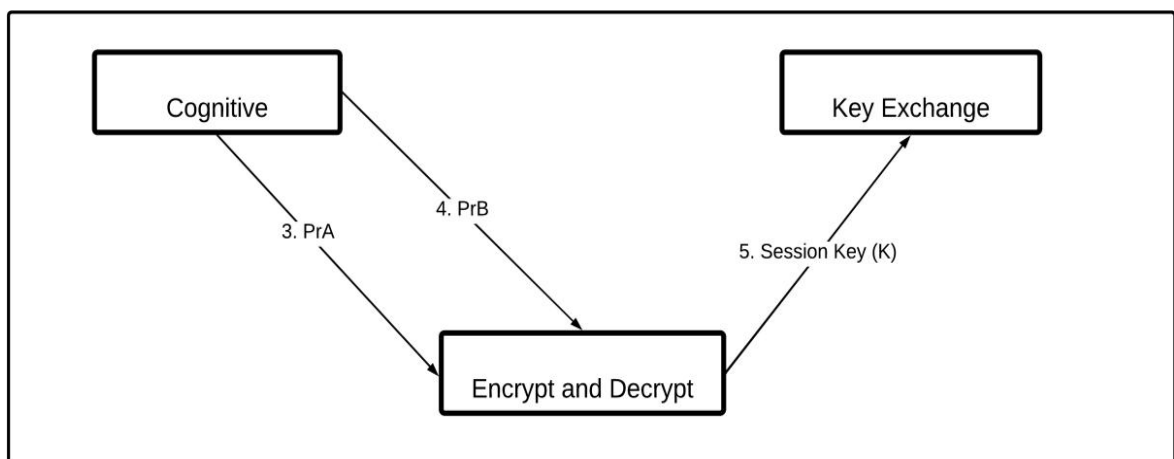
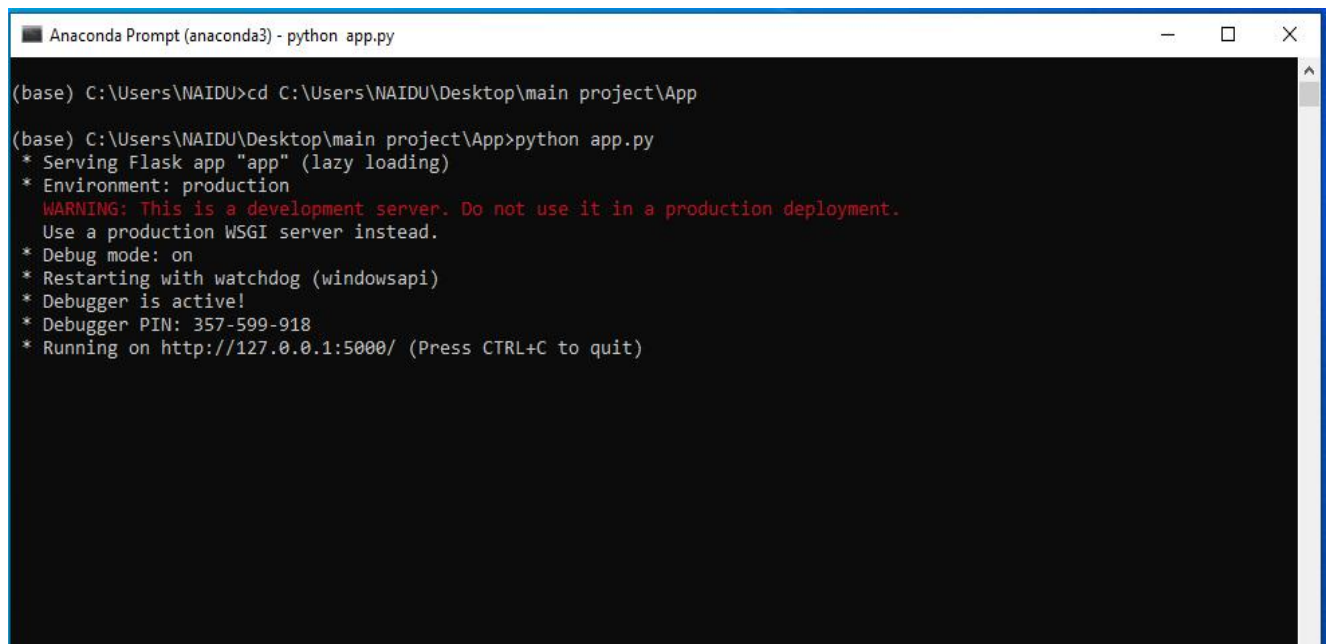


Figure 5.4 : Collaboration Diagram For Proposed Model

Chapter – 6

RESULTS & DISCUSSION



```
Anaconda Prompt (anaconda3) - python app.py

(base) C:\Users\NAIDU>cd C:\Users\NAIDU\Desktop\main project\App

(base) C:\Users\NAIDU\Desktop\main project\App>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 357-599-918
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 6.1 : Anaconda Command Prompt

Elgamal

private key : 0x241a21e9560732b9aa39e0f3bbf464424512704a85eff0d2ef617c6019bd81e4

public key : 0x35c26049e552ab7fc6e546a42a7d874c623c541b64b94a7fecbf80d82fe0f930

Receiver pubKey :
0x85493b0071f1ed45339a742d44aac4e28e1cdb580ea9698af5eac5b9342b50290

encryption key: : 0x846813a2e4415af463d1548637cc571f48f3bb82dd3d131da7e7add0e4e1f4ee1

decryption key: : 0x846813a2e4415af463d1548637cc571f48f3bb82dd3d131da7e7add0e4e1f4ee1

Message:

Submit

Figure 6.2 : Elgamal Encryption

Elgamal

ciphertext :
b'd1dc6ccc51655bd773d77a23da4f1a5b42b6dd431db1d566abb625a2c3d3abdaa293dab09aaf0ccfe1497da10e6f'

nonce(n) :
b'aa25e1e8a7590c692ba1d771d514e4e7'

authTag(a) :
b'35bd2ed923a3eddb35c9874c0eba2635'

Sender PubKey :
0x82dd3ae8df14041ca27bda33b0c3fe19e7ebdea6a16e228fcea16f5c5a9789e0

[BACK](#)

Signature is valid

Decrypted message:
Text to be encrypted by ECC public key and decrypted by its corresponding ECC private key

Figure 6.3 : Elgamal Decryption

RSA

Enter a prime number (17, 19, 23, etc):

Enter another prime number (Not one you entered above):

Enter a message to encrypt with your public key:

Figure 6.4 : RSA Encryption

RSA Ecnryption

Generating your public / private key-pairs now . . .

Your public key is: (37, 437)

Your private key is: (289, 437)

Your Encrypted message is: 1754092644276114

Decrypting message with private key(289, 437)

Decrypting message with private key: cipher

Figure 6.5 : RSA Decryption

Table 6.1 : ECC - RSA Performance Comparison

Key size	RSA		ELGAMAL		ECC ELGAMAL	
	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption
80	0.231	0.483	0.145	0.203	0.011	0.003
112	1.58	3.384	0.991	1.542	0.015	0.004
128	4.78	7.626	2.947	5.159	0.017	0.006

Table 6.2 : ECC - RSA Cost Comparison

S.No	ECC key size	RSA key size	Ratio of cost
1	160	1024	1:3
2	224	2048	1:6
3	256	3072	1:10
4	384	7680	1:32
5	512	15360	1:64

Table 6.3 : ECC - RSA Key Size Comparison

S.No	ECC Key Size	RSA Key Size	Key Size Ratio
1	112	512	1:5
2	163	1024	1:6
3	192	1536	1:8
4	224	2048	1:9
5	256	3072	1:12
6	384	7680	1:20
7	512	15360	1:30

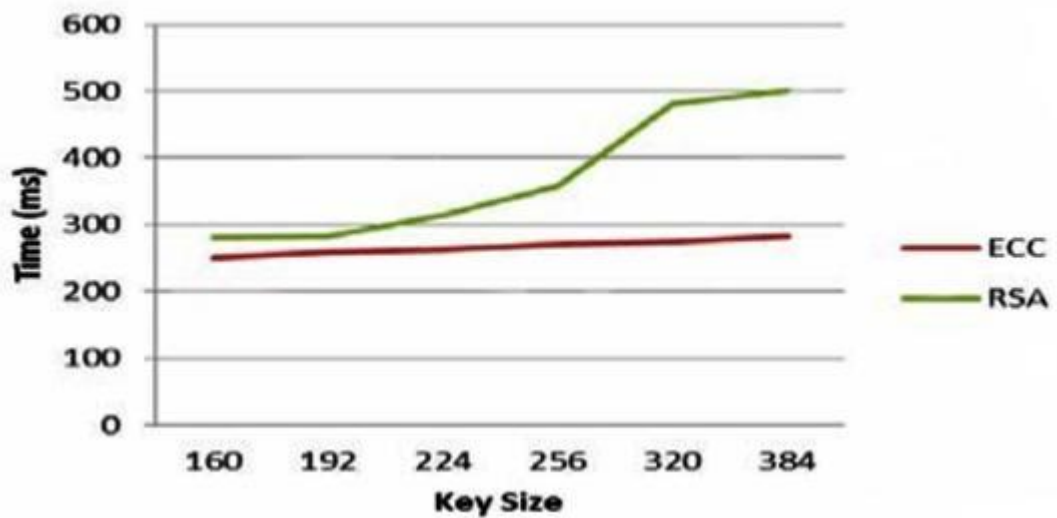


Figure 6.6 : ECC - RSA Key Size Comparison

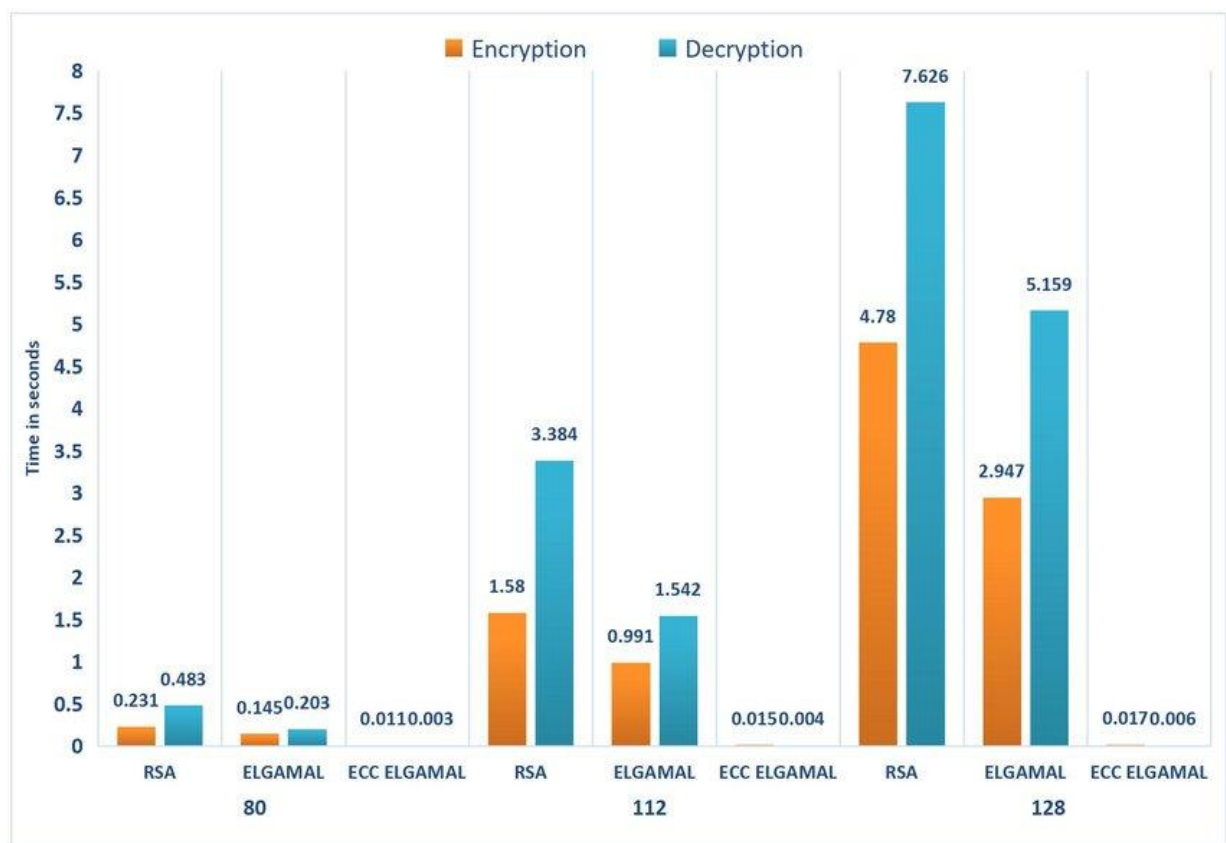


Figure 6.7 : Performance Comparison Between Conventional Cryptosystems And ElGamal

Chapter – 7
CONCLUSION & FUTURESCOPE

This paper proposes a new ECC scheme by combining digital signatures that provide high security than existing scheme. ECC provides same level security as RSA by using small keys only. This is done to make a system that is less vulnerable to different types of attacks like KPA, CPA, CCA, COA. This project ensures that the message is authenticated with the senders authenticity and verifies at the receiver's side. The message which is also send by the sender is safely reach to the receiver without any modifications. The proposed scheme need to evaluated on different key sizes for further improvement. This needs much research on various Elliptic Curve Discrete Logarithmic Problems. A study on different curves is required to improve the efficiency.

One of the major concerns of high-end data communications is security. These concerns have been resolved through various cryptographic methods throughout the years. The highly sensitive data is encrypted when sent, and decrypted on the receiver's end making use of the suitable cryptographic method among the existing. One of these methods is the highly reputed Elliptic Curve Cryptography. This method is known for being well utilized in systems that require security up a notch when compared to others. Although Elliptic Curve Cryptography in itself is effective enough to fulfill this need, it lacks in providing authentication of anyone trying to use its privilege. Therefore providing authentication makes it a full package of security being provided to those who need it. Hence, the concept of cognitive feature comes into picture to help with the authentication process. Any two authenticated users can send sensitive information through a platform/interface at once. This system can be further improved or worked on in the future by making multiple users be able to communicate at the same time on a platform.

Chapter – 8

REFERENCES

- [1] H. N. Almajed and A. S. Almogren, "SE-Enc: A Secure and Efficient Encoding Scheme Using Elliptic Curve Cryptography," in IEEE Access, vol. 7, pp. 175865-175878, 2019, doi: 10.1109/ACCESS.2019.2957943.
- [2] Islam, SK Hafizul, and G. P. Biswas."Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography." International Journal of Computer Mathematics 90.11 (2013): 2244-2258.
- [3] Y. Luo, X. Ouyang, J. Liu and L. Cao, "An Image Encryption Method Based on Elliptic Curve Elgamal Encryption and Chaotic Systems," in IEEE Access, vol. 7, pp. 38507-38522, 2019, doi: 10.1109/ACCESS.2019.2906052.
- [4] Abitha, K. S., Anjalipandey, A., & Kaliyamurthie, D. K. P. (2015). "Secured data transmission using elliptic curve cryptography". IJIRCCE, 3(3), 1-7.
- [5] M. A. Mehrabi, C. Doche and A. Jolfaei, "Elliptic Curve Cryptography Point Multiplication Core for Hardware Security Module," in IEEE Transactions on Computers, vol. 69, no. 11, pp. 1707-1718, 1 Nov. 2020, doi: 10.1109/TC.2020.3013266.
- [6] C. Qi, "A Zero-Knowledge Proof of Digital Signature Scheme Based on the Elliptic Curve Cryptosystem," 2009 Third International Symposium on Intelligent Information Technology Application, 2009, pp. 612-615, doi: 10.1109/IITA.2009.505.
- [7] Rajeswari, Ms PG, and K. Thilagavathi. "An Efficient Authentication Protocol Based on Elliptic Curve Cryptography for Mobile Networks." IJCSNS 9, no. 2 (2009): 176.
- [8] Shankar, Tarun Narayan, and G. Sahoo."Cryptography with elliptic curves." International Journal of computer science and applications 2.1 (2009): 38-42.
- [9] Qiuxia Zhang, Zhan Li and Chao Song, "The Improvement of digital signature algorithm based on elliptic curve cryptography," 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011, pp. 1689-1691, doi: 10.1109/AIMSEC.2011.6010590.
- [10] S. Sciancalepore, G. Piro, G. Boggia and G. Bianchi, "Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption," in IEEE Embedded Systems Letters, vol. 9, no. 1, pp. 1-4, March 2017, doi: 10.1109/LES.2016.2630729.

APPENDIX – A

```

//app.py
from flask import *
from tinyec import registry
import secrets
from Crypto.Cipher import AES
import hashlib, secrets, binascii
import crypto_commons as commons
import random

app = Flask(__name__)

#encoding the values to hexadecimal
def compress_point(point):
    return hex(point.x) + hex(point.y % 2)[2:]

curve = registry.get_curve('brainpoolP256r1')

def ecc_calc_encryption_keys(pubKey):
    ciphertextPrivKey = secrets.randbelow(curve.field.n)
    ciphertextPubKey = ciphertextPrivKey * curve.g
    sharedECCKey = pubKey * ciphertextPrivKey
    return (sharedECCKey, ciphertextPubKey)

def ecc_calc_decryption_key(privKey, ciphertextPubKey):
    sharedECCKey = ciphertextPubKey * privKey
    return sharedECCKey

def key_generation():
    privKey = secrets.randbelow(curve.field.n) # alicePrivKey
    temp=privKey
    pubKey = privKey * curve.g # alicePubKey
    temp2=pubKey
    print("private key:", hex(privKey))
    print("public key:", compress_point(pubKey))

```



```

print(" ")
(encryptKey, ciphertextPubKey) = ecc_calc_encryption_keys(pubKey)
print("ciphertext pubKey:", compress_point(ciphertextPubKey))
print("encryption key:", compress_point(encryptKey))

decryptKey = ecc_calc_decryption_key(privKey, ciphertextPubKey)
print("decryption key:", compress_point(decryptKey))

homedata = {
    "private key" : hex(privKey),
    "public key" : compress_point(pubKey),
    "Receiver pubKey": compress_point(ciphertextPubKey),
    "encryption key:": compress_point(encryptKey),
    "decryption key:": compress_point(decryptKey)
}
return homedata

def encrypt_ECC(msg, pubKey):
    ciphertextPrivKey = secrets.randbelow(curve.field.n)
    sharedECCKey = ciphertextPrivKey * pubKey
    secretKey = ecc_point_to_256_bit_key(sharedECCKey)
    ciphertext, nonce, authTag = encrypt_AES_GCM(msg, secretKey)
    ciphertextPubKey = ciphertextPrivKey * curve.g
    return (ciphertext, nonce, authTag, ciphertextPubKey)

def decrypt_ECC(encryptedMsg, privKey):
    (ciphertext, nonce, authTag, ciphertextPubKey) = encryptedMsg
    sharedECCKey = privKey * ciphertextPubKey
    secretKey = ecc_point_to_256_bit_key(sharedECCKey)
    plaintext = decrypt_AES_GCM(ciphertext, nonce, authTag, secretKey)
    return plaintext

def encrypt_AES_GCM(msg, secretKey):
    aesCipher = AES.new(secretKey, AES.MODE_GCM)
    ciphertext, authTag = aesCipher.encrypt_and_digest(msg)
    return (ciphertext, aesCipher.nonce, authTag)

```

```
def decrypt_AES_GCM(ciphertext, nonce, authTag, secretKey):
    aesCipher = AES.new(secretKey, AES.MODE_GCM, nonce)
    plaintext = aesCipher.decrypt_and_verify(ciphertext, authTag)
    return plaintext
```

```
def ecc_point_to_256_bit_key(point):
    sha = hashlib.sha256(int.to_bytes(point.x, 32, 'big'))
    sha.update(int.to_bytes(point.y, 32, 'big'))
    return sha.digest()
```

```
def encryptMessage(msg):
#   msg = b'Text to be encrypted by ECC public key and decrypted by its corresponding ECC private
key'
    msg = bytes(msg,'UTF-8')
    print("original msg:", msg)
    print(" ")
    privKey = secrets.randbelow(curve.field.n)
    pubKey = privKey * curve.g
    encryptedMsg = encrypt_ECC(msg, pubKey)
    #decoding the message
    encryptedMsgObj = {
        'ciphertext': binascii.hexlify(encryptedMsg[0]),
        'nonce(n)': binascii.hexlify(encryptedMsg[1]),
        'authTag(a)': binascii.hexlify(encryptedMsg[2]),
        'Sender PubKey': hex(encryptedMsg[3].x) + hex(encryptedMsg[3].y % 2)[2:]
    }
    return (encryptedMsgObj)
```

```
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a
```

```
def multiplicative_inverse(e, phi):
```

```
    d = 0
```

```
    x1 = 0
```

```
    x2 = 1
```

```
    y1 = 1
```

```
    temp_phi = phi
```

```
    while e > 0:
```

```
        temp1 = temp_phi//e
```

```
        temp2 = temp_phi - temp1 * e
```

```
        temp_phi = e
```

```
        e = temp2
```

```
        x = x2 - temp1 * x1
```

```
        y = d - temp1 * y1
```

```
        x2 = x1
```

```
        x1 = x
```

```
        d = y1
```

```
        y1 = y
```

```
    if temp_phi == 1:
```

```
        return d + phi
```

```
def is_prime(num):
```

```
    if num == 2:
```

```
        return True
```

```
    if num < 2 or num % 2 == 0:
```

```
        return False
```

```
    for n in range(3, int(num**0.5)+2, 2):
```

```
        if num % n == 0:
```

```
            return False
```

```
    return True
```

```

def generate_key_pair(p, q):
    if not (is_prime(p) and is_prime(q)):
        raise ValueError('Both numbers must be prime.')
    elif p == q:
        raise ValueError('p and q cannot be equal')
    n = p * q
    # Phi is the totient of n
    phi = (p-1) * (q-1)

    # Choose an integer e such that e and phi(n) are coprime
    e = random.randrange(1, phi)

    # Use Euclid's Algorithm to verify that e and phi(n) are coprime
    g = gcd(e, phi)
    while g != 1:
        e = random.randrange(1, phi)
        g = gcd(e, phi)

    # Use Extended Euclid's Algorithm to generate the private key
    d = multiplicative_inverse(e, phi)

    # Return public and private key_pair
    # Public key is (e, n) and private key is (d, n)
    return ((e, n), (d, n))

def encrypt(pk, plaintext):
    # Unpack the key into its components
    key, n = pk
    # Convert each letter in the plaintext to numbers based on the character using a^b mod m
    cipher = [pow(ord(char), key, n) for char in plaintext]
    # Return the array of bytes
    return cipher

```

```

def decrypt(pk, ciphertext):
    # Unpack the key into its components
    key, n = pk
    # Generate the plaintext based on the ciphertext and key using  $a^b \bmod m$ 
    aux = [str(pow(char, key, n)) for char in ciphertext]
    # Return the array of bytes as a string
    plain = [chr(int(char2)) for char2 in aux]
    return ''.join(plain)

```

```

def modInverse(A, M):
    for X in range(1, M):
        if (((A % M) * (X % M)) % M == 1):
            return X
    return -1

```

```

def testPrimeness(number):
    for i in range(2, number):
        if number % i == 0:
            return False
            break
    return True

```

```

base = 3
p = 1279 #prime
#p = a * q + 1

```

```

for i in range(10, p):
    if (p-1) % i == 0 and testPrimeness(i):
        q = i
        break

```

```

a = int((p-1)/q)
g = pow(base, a, p)
x = 15 # private key
y = pow(g, x, p)
print("signing")
k = 10 #random key
h = 123
r = pow(g, k, p) % q
s = modInverse(k, q) * (h + x*r) % q
print("verification")
h = 123
w = modInverse(s, q)
u1 = h * w % q
u2 = r * w % q
v = ((pow(g, u1, p) * pow(y, u2, p)) % p) % q
if v == r:
    print(" Both are samesignature is valid")
else:
    print("invalid signature is detected")

```

```

@app.route("/")

```

```

def home():

```

```

    homedata=key_generation();

```

```

    #global homedata

```

```

    return render_template("index.html",data = homedata);

```

```

@app.route('/encrypt',methods = ["POST"])

```

```

def encrypt1():

```

```

    msg = request.form['input_message']

```

```

    encryptmsg = encryptMessage(msg)

```

```

    return render_template("ECC encryption.html",data = encryptmsg,message=msg )

```

```
@app.route("/back",methods=["POST"])
```

```
def back():
```

```
    return redirect(url_for('home'))
```

```
@app.route("/rsa")
```

```
def rsa():
```

```
    return render_template("RSA.html")
```

```
@app.route("/rsaa", methods=["POST"])
```

```
def rsaa():
```

```
    p = int(request.form['prime_number1'])
```

```
    q = int(request.form['prime_number2'])
```

```
    r = request.form['message1']
```

```
    public, private = generate_key_pair(p, q)
```

```
    encrypted_msg = encrypt(public, r)
```

```
    encrypted_msg="".join(map(lambda x: str(x), encrypted_msg))
```

```
    return render_template("results.html",a=public,b=private,c=encrypted_msg,ram=r)
```

```
if __name__ == "__main__":
```

```
    app.run(debug= True)
```

APPENDIX – B

Digital Signature and ElGamal Scheme Integration for Secure Data Transmission in Digital Transactions: Survey Paper

Y Surya Prakash

*Department of Information Technology
GMR Institute of Technology
Srikakulam, Andhra Pradesh, India
suryaparakash.y@gmrit.edu.in*

P Harisankar Narayan

*Department of Information Technology
GMR Institute of Technology
Srikakulam, Andhra Pradesh, India
19341A1284@gmrit.edu.in*

R Ramakrishna

*Department of Information Technology
GMR Institute of Technology
Srikakulam, Andhra Pradesh, India
19341A1295@gmrit.edu.in*

G Sai Sandeep

*Department of Information Technology
GMR Institute of Technology
Srikakulam, Andhra Pradesh, India
19341A1298@gmrit.edu.in*

V Srikara Sai Ramesh

*Department of Information Technology
GMR Institute of Technology
Srikakulam, Andhra Pradesh, India
19341A12C4@gmrit.edu.in*

I Balaraju

*Department of Information Technology
GMR Institute of Technology
Srikakulam, Andhra Pradesh, India
20345A1207@gmrit.edu.in*

Abstract-- Security can be considerably a major concern when it comes to high-end data transmissions like satellite parameter communication, Defence security codes, communication etc. Sensitive information of any type can lead to various attacks from intruders who have malicious thoughts. These attacks can be significantly harmful to the loss of user data (sensitive or non-sensitive data). This problem is look thoughtfully for a long time and resolved by integrating Digital Signature and ElGamal scheme(Elliptic Curve Cryptography). This Integration provides a Cognitive Feature for Authorization. The encryption is done using the standard ElGamal scheme with a well-built reputation for itself in secure data transmission. It provides a secure communication channel between the two ends by authenticating the sender with Digital Signature. One of the thrust areas of this project is digital transactions.

Keywords--Cryptography, Elliptic Curve Cryptography, attacks, cognitive feature, Digital Signature, ElGamal.

I. INTRODUCTION

Most of the people are using mobile phones for online transactions. In this case most transactions are done through internet banking. The existing Cryptography is failed to provide security due to rendering technology hence it becomes a major concern in terms of providing security and privacy. ElGamal and Digital Signature are combined to address the existing challenges and provide effective security. ECC has a major role in improving data integrity and confidentiality. However, the data carried over a wireless network will not be highly secure as a result. We are unable to use these strategies to prevent problems with wireless networks' increased packet transmission issues, which result in data loss. To overcome the problems in the existing system, the proposed system is promoted with an authentication process involving digital signatures of the user in its implementation. A group of points plotted on an image is provided to the user for him to use his cognitive abilities and remember the points he has selected the first time he uses the system. Coming to the process of ElGamal scheme, a private key is

generated using these selected points for the user. Some mathematical operations are performed on the points to attain the digital signatures. With the help of the now obtained private key and a base point, a public key for this user is also generated. The other authenticated entry follows suit thus far. Now, in order for them to have a connection and basically validate each other, a session key needs to be generated on either end which is required to be equal. After generating and verifying the session key, both the end users are now set to communicate with each other. A message sent by either user is then encrypted using the method of elliptic curve cryptography. Thus sent message is now decrypted and digitally signed on the receiver end.

II. LITERATURE SURVEY

In the modern business world, when companies offer their working environments to clients, customers, and colleague online, digital security has a significant part in creating trust. The typical technique used to protect Web content is SSL and HTTP/HTTPS. SSL does not, however, satisfy all security needs. Even though non-repudiation is a crucial security need for business applications, it does build a solid private connection and validate peer identities. Due to SSL's status as a transport layer protocol, data is protected while being transmitted between the Web server and client browser. End to end protection is not entirely possible with SSL. Non-repudiation, authenticity, and integrity can all be provided through digital signatures. At this time, a digital signature architecture that offers client and server-side elements for signature validation that are independent of the browser is needed in order to facilitate the integration of signatures into web applications. SSL and digital signatures together enhance the inherent capabilities of each technology. A digital signature that makes use of a public key infrastructure is generally considered as being equivalent to a handwritten signature in terms of legitimacy and legality. Although the procedure is straightforward, using it for web apps makes it more difficult. Two key causes account for this. First one is, the HTML tool doesn't contains signature support in it. This can not be overcome by

using third-party signature tools for this purpose to sign the content because This restricts the web apps to specific operating systems and browsers. The second one is that there are multiple signature forms for different business situations, including the XML Digital Signature and cryptographic message syntax. Designing a centralised and shape-driven framework which controls the behaviour of both client and server modules is necessary to achieve a variety of business essentials. In that the following modules: Configuration Repository, Signature Generator, Signature Validator, Certificate Validator, Signature Storage and Archival, Offline Signature Verification Utilities, Certificate and CRL Update Utilities, Certificate & CRL Repository, Secure Private Key Storage, Security Assurance. [1]

There are numerous varieties of high-security cryptographic techniques. However, their only flaw is a huge size key that demands a lot of processing power. ECC is a substitute for that offers great security with a size reduced key. This essay is structured around the mathematics of the elliptic curve. Compared to other traditional cryptographic systems like RSA, elliptic curve cryptography (ECC) has a key length advantage. It offers higher security with a very short key length, making it a great option when the device's processing power is limited. Elliptic curve and it's mathematics : Different mathematical operations can be performed on elliptic curves over a set of finite points such as like point doubling, point addition, and point multiplication.

Point_addition : let two points $A(x_1, y_1)$ and $B(x_2, y_2)$ must not have the same coordinates ($x_1 \neq y_1$ and $x_2 \neq y_2$).

$$X_3 = (\lambda^2 - x_1 - x_2) \bmod p \quad \text{and} \quad y_3 = (\lambda(x_1 - x_2) - y_1) \bmod p \quad \text{where} \quad \lambda = (y_2 - y_1) / (x_2 - x_1) \bmod p$$

Point_doubling: Let the two points projection ($R = A + A$)

$$X_3 = (\lambda^2 - 2x_1) \bmod p \quad \text{and} \quad y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p \quad \text{where} \quad \lambda = (3x_1^2 + a) / (2y_1) \bmod p$$

Point_multiplication : Take a point on the elliptic curve, let's call it P. Therefore, repeated addition is the definition of the operation of multiplying the point P. kP equals to $P + P + P + \dots k$ times. The brute force method is a technique for locating the private key that involves all values with generator and comparing it to the user's public key to see whether they match. There are more approaches that use the Pollard-Ro technique, but have a different iteration function, such as random walks and $(r+h)$ mixed walks. The time complexity of the Baby-Step-Giant-Step method and the Pollard-Rho algorithm is $O(\sqrt{n})$. [2]

In multiple proxy multiple signature systems, an original group of signers may have the arrangement of all signers in the initial group and proxy group, approve another group of proxy signers. The work suggests elliptic curve cryptography based brand-new multi-proxy multi-signature. The insider attack, a potent attack on multiple signature methods, cannot succeed against this new scheme. A proxy signature can allow a person to authorize another person to sign documents on his behalf during absences or to grant authority to subordinates for responsibilities they have been given. The multiple proxy multiple signature is a unique kind of proxy signatures in which original signers may allow a different group of proxy signers with the consent of each signer from the original group as well as the proxy group. The Elliptic

Curve Cryptography is a high sophisticated cryptography than the Rivest Shamir Adleman cryptosystem since it uses less memory and resources than RSA and has a tiny key length 160 bits, compared to RSA key length 1024 bits. Any public key cryptosystem that creates a one-way function must use a challenging mathematical problem. The abelian set of points plotted on the elliptic curve and logarithm issues are the mathematical foundation of ECC. Another effective tool for determining prime factors and verifying prime numbers is the elliptic curve (EC). The difficulties of dealing a discrete logarithm issue on an elliptic curve serves as the foundation for the security of the multiple proxy multiple signature system. The equation represents an elliptic curve, a type of the curve used in mathematics: $y^2 = x^3 + ax + b$. These benefits apply to the proposed multiple proxy multiple signature scheme. The size of a multiple proxy multiple signature is also free of the number of members of the proxy, just as the proxy certificate's size is unrelated to the total number of initial signers. The group of initial signer and the proxy group are well protected under our new method, too. It is more resistant to the formidable insider attack, which targets multiple signature methods. Lastly, the fact that checking numerous proxy signatures only costs half as much as validating signatures created by original signers demonstrates the efficiency of our techniques. That is, the cost of verifying multiple proxy signatures is unrelated to the number of proxy signers. [3]

With the help of the longest possible length random sequence generation, this work suggests a new EC Cryptography (ECC) method and a data mapping method for elliptic curves over finite fields. This study also suggests a brand-new algorithm for ECC scalar multiplication during its implementation. The suggested system is evaluated on several bits of prime field length, and the practical findings demonstrate very high strength against cryptanalytic attack such as random walk and higher production in terms of computation time compared with current approaches. When compared with other Public Key Cryptography methods, ECC is popular due to its short key length, cheap processing, and good security. Small processor device producers, including those who make smart cards, wireless gadgets, raspberry computers, smart phones, pagers, and tablets, are drawn to these features. The main applications of ECC are key exchanging, digital signatures, and authentication. However, it can be utilised in any security application with constrained computational resources and integrated circuit area. This study examines the fundamentals of ECC and suggests a brand-new ECC scheme as well as a new scalar multiplication algorithm. Let E be the elliptic curve and $A(x_1, y_1)$, $B(x_2, y_2)$ are the two points on E. **Point Addition:** Start by drawing a line across A and B; this line eventually meets the elliptic curve three times. The sum of the points A and B is R, which is the projection of this intersection point along the x-axis. **Point Doubling:** Make a tangent line to the elliptic curve at point A, which touches the elliptic curve. The projection of this point on the x-axis is then R. **Random Number Generation :** Linear Feedback Shift Registers (LFSR) may produce random numbers with ease from polynomials of the maximum length. For example, for the polynomial $f(x) = 3x^5 + 2x + 7$, the highest power in the polynomial is same as the length of the shift register i.e, $n = 5$. As a result, the length of the sequence generated by the polynomial is defined by $2^n - 1$. For given polynomial, $2^5 - 1 = 32 - 1 = 31$. **Scalar Multiplication - Proposed Algorithm :** The method used to implement the Scalar or Point Multiplication determines how effective an ECC implementation is. The majority of the algorithms now in use concentrate on minimizing the hamming weight of input value by changing it to sign binary integers or to binary integers. The suggested algorithm also strives to minimize the hamming weight by selecting the

most appropriate sign binary or binary multiplication method. This work implements the recommended system while also introducing a novel, less expensive scalar multiplication algorithm that benefits from both binary and signed binary presentation. [4]

This paper contains examples to make an idea on implementation and applications of digital signature in network security. Problems achieved in digital signature and solutions for those problems are discussed in this paper. This paper also explain the role and the importance of digital signature. The ISO7498-2 specification states that a digital signature is In order to prevent data from being falsified, some data connected to data cells or data cells that have undergone cryptographic transformation enable the receiver of the data cells to confirm the integrity of the data. The security of a digital signature depends on the degree of security of the cryptosystem being used for its generation and processing. By doing this, higher levels of security than a handwritten signature are achieved. There are many different types of digital signatures, including standard digital signatures, arbitrated digital signatures, threshold signatures, blind signatures, irrefutable signatures, and group signatures. The one-way Hash function encryption algorithm creates a 128 bit cryptographic text for a message's digital summary. It is a one-way function, making decryption for it impossible. Digital summaries are a solution to the problem of information integrity. On the one-way Hash function, digital summaries are based. The speed of the PC affects how quickly information is transferred in online programmes. High costs and ineffectiveness are problems that hamper today's digital signature technology. The choice of algorithm and the administration of private keys affect the security of digital signatures. The AES asymmetric encryption algorithm outperforms the symmetric AES algorithm. RSA, DSA, and ECC serve as examples for AES (elliptic curve algorithm). ECC algorithm is faster and safer than RSA algorithm since it is based on elliptic curve logarithm problems. Although RSA is also the most widely used asymmetric encryption technique, we employ it as a digital signature algorithm for a number of reasons. We employ a 1024-digit private key for increased security. [5]

Nowadays, asymmetric cryptography is used mostly in e-commerce application development to ensure the parties' authenticity. On the other hand, an increase in the popularity of mobile appliances has sparked a move toward mobile e-commerce applications. PDAs, cell phones, and pagers are the results of this quest for smaller, faster, and less expensive platforms. This work places a focus on the existing RSA-based asymmetric cryptographic authentication techniques that are currently in use, but are unsuitable owing to limitations in key sizes, memory capacity, processing power, and cryptographic support. The performance gained is good when compared to RSA thanks to this protocol's complete reliance on elliptic curve asymmetric encryption, according to the results. Sensitive data transmission has, however, run into certain issues because the internet is an open and vulnerable network. The use of encryption and secure authentication protocols, which ensure the confidentiality, authenticity, and integrity of communications, is the solution. Existing and widely utilized in current online shopping applications are such protocols, such as SSL and SET. On RSA public key cryptography, the majority of them are built. A protocol based on elliptic curve cryptography is created, maintaining a high level of security. A secure channel in between two the principals over an unsecured network, such as the internet, must be able to be established by the authentication protocol. It is simple to hack a router or spy on a line in order to listen in on and change all communications. The Mutual authentication between

the parties must be ensured via the protocol, as well as the secrecy and integrity of any data transmitted through it, in order to prevent this. Even with a key length of 160 bits compared to RSA's 1024 bits, the implementation of ECC asymmetric-key on mobile devices performs well and offers more security. There are 5 phases to this: In phase 1, by providing its ID or serial number to the server, the mobile device initiates the protocol. In phase 2, the mobile ID is stored on the server for authentication purposes, and the elliptic curve is used to generate the mobile's private key and public key. The mobile device receives these mobile keys as well as the server's public key. The keys are sent via the diffie-hellman key exchange technique. In phase 3, With the help of the server's public key and the mobile's private key, the mobile generates a challenge and transmits it together with its ID to the server. The server uses the mobile's public key and private key to encrypt the message, decrypts it, and then checks to see if the ID it decrypted matches the ID it received in step 1. It serves to verify the client's identity. In phase 4, the server sends the mobile's challenge from the previous step along with a randomly generated session key. All three are encrypted using the server's private key and the mobile device's public key. With the help of the server's private key and public key, the mobile device decrypts this message and validates the challenge. If it is same as the one that was sent in step 3, the mobile device can be confident that it is speaking to the correct server. Elliptic curve cryptography technology is used for the encryption and decryption processes, which are described in steps 3 and 4. In phase 5, All information is encrypted using a session key across a secure channel that has been established. A different key is set up for each communication to protect against replay attacks. [6]

Elliptic curves garnered a lot of attention and quickly developed after being introduced to cryptography. Elliptic curves are the best option in comparison to the arithmetic of finite fields, and there is no sub exponential time attack for the discrete logarithm problem of elliptic curves. Elliptic curves are frequently used in a variety of cryptographic procedures, such as digit signatures, date encryption, and key exchange protocols. The most common operation in cryptosystems based on elliptic curves is point multiplication on elliptic curves, which is nothing more than computing $[n]P$ for a point P on elliptic curves. The fundamental operation in the implementation of point multiplication is the addition of a point's double to another point, i.e., computing $2P+Q$ for two points P and Q on an elliptic curve.[7]

In a proxy signing arrangement, the proxy signer has the authority to sign on the other signer's behalf. Currently, a variety of proxy signature schemes with various properties have been proposed; however, it is still very difficult to put these schemes into practise due to a variety of security risks and flaws, such as the proxy signer's lack of confidentiality protection, their susceptibility to coalition attacks and generalised signature forgery, and the efficiency of their hardware and software implementation. A novel proxy signature based on ECC is proposed to address security risks and flaws in existing systems. The plan explains how a probabilistic encryption mechanism can effectively secure the privacy of proxy signers. The usage of the upgraded one-way hash function based on ECDLP allows the algorithms to greatly benefit from the advantages of ECC, including its high efficiency and small key length. The suggested technique not only fully exploits the ECDLP's complexity to solve and the proxy scheme's security, but it also employs the one-way trapdoor function to increase security. [8]

Information security is now playing a critical role in communication and data storage due to the internet's explosive growth. Cryptography

can be used to secure data being passed from one party to another through an unsecured channel (like the Internet). Suli Wu and colleagues suggested a method for matrix scrambling based on a two-way circular queue. This leads to the evolution of a new elliptic curve-based matrix scrambling encryption system. The decision to operate on rows or columns depends on a random elliptic curve point. Choosing the procedures for scrambling in the first and second phases at random places on an elliptic curve increases the complexity of decryption. This eliminates regularity in the cypher text that results from the transformation of the plaintext matrix. We would like to highlight the effective usage of the elliptic curves and the integer value "a" in this, which produces a significant binary sequence and is utilised to scramble the matrix in both directions. There is a lot of room to experiment with the choice of the "aP" point in the end. One can evaluate the application of this technique in low memory appliances such as smart cards and mobile gadgets depending on the memory demand. This algorithm can also be used for text, image, and multimedia encryption, among other things. ECC is a field with a lot of potential for further investigation. [9]

Public key cryptosystems are the main applications for elliptical curve cryptography (ECC). The fundamental benefit is that, compared to conventional asymmetric cryptosystems like RSA, key sizes are substantially smaller at a given security level. Smaller keys need less memory, use less energy, and require less cryptographic processing. Security is one of the main issues with embedded devices, in addition to performance. Although mathematically secure, cryptosystems like ECC are not regarded as secure when used in actual transactions. Attackers can keep an eye on these exchanges in order to launch fault assaults. To defend the Montgomery Scalar Multiplication method from fault attacks, countermeasures have been developed. An effective defence against fault attacks for the Montgomery Scalar Multiplication method based on a duplication scheme and the scrambling approach. The strategy is straightforward and simple to implement on hardware. Additionally, we run simulations of injection-based errors and showed that the error coverage is nearly 99.9%. The scrambling technique and duplicating scheme are the foundation of the countermeasure. The method is straightforward and straightforward to implement with hardware. The results of the fault injection affected reveal that, after examining/analyzing the application of the suggested countermeasure against fault assaults, the technique can detect 99.99% of the randomly injected errors. [10]

The third and most fundamental cryptographic primitive after encryption and digital signature is key agreement. These protocols enable two or more parties to communicate securely later on by allowing them to transmit information across a controlled, insecure network while also deciding on a common session key. Building secure, complicated, higher-level protocols requires the use of secure key agreement protocols as a fundamental building block. The Diffie-Hellman protocol, presented by Diffie and Hellman in 1976, is the first contemporary key agreement protocol. Its security is predicated on how challenging discrete logarithm problems are to solve. It is vulnerable to a man-in-the-middle attack as the first usable key agreement protocol without authentication. The security of the protocols for known key agreement is based on two fundamental mathematical issues: Determining the structure and order of a finite Abelian group, Computing the discrete logarithm in a cyclic group with computable order. Shor's algorithm for a quantum computer allows for

polynomial-time resolution of both issues. As a result, the majority of present public-key cryptosystems will become vulnerable once a quantum register is large enough. [11]

Throughout this paper, we learned about the fundamentals of ECC applications in online shopping applications. In an online shopping transaction, there are card issuers, card owners, businesses, banks, payment gateways, and certificates. The transaction id is converted into unreadable text famously known as ciphertext by using ECC to give safety to the users who are maintaining their accounts, and the transaction data is also modified to unreadable text with ECC to save from giving a misleading account. We use the ECC signature algorithm to certify the business and card owners. As a result, an online shopping platform based on the EC Curve guaranteed information validity, confidentiality, integrity, and non-repudiation. Online shopping is now a means of online exchange, so transaction security is becoming increasingly important. Businesses and consumers are both focusing on the development of a secure online shopping platform. While the Secure Electronic Transaction (SET) rule is a common e-commerce safety rule, it is used to give data is not available for unknown and invalid login users. The data safety is attained by the plaintext is converted into ciphertext by using the Signature rule. The SET protocol is designed to settle bank card transactions between as well as merchants and banks. Encryption, a digital signature, and an envelope are all included. Encryption and signatures are typically used throughout all phases. In the SET protocol, which involves a number of communications, the data is typically encrypted using a symmetrical encryption algorithm, and the digital signature and envelope are encrypted using a public encryption algorithm before being transmitted using a public encryption algorithm. [12]

We obtained a description of the Sign Change assault and assessed its likelihood based on this information. The cryptography under the elliptic curves are prone to two fault attacks have been described. This technique involves extremity on the elliptic loop for modification of sign. Colleagues discussed the usage of a key expressed in NAF form. The use of a NAF representation does not raise the likelihood of a sign flip attack because the assault is on the point's memory rather than a computer instruction. In an ECC implementation, the ECC point wouldn't be represented by a sign bit. Additionally, the negative of a point is utilized in the multiple areas that are used to develop an ECC application. This appears to be a highly selective alteration of the key stream because it is exceedingly improbable that a sign bit would be used for an extended integer. Here, we outline two conceivable fault attacks against an elliptic curve cryptography smart card implementation. The integrated circuit card may give a signature in high-end applications like banking and identity software. The card issuer, not the cardholder, often holds the secret signature key that is stored on the smart card. The card's owner may try to gain malicious access to this confidential data. To find the secret key, numerous adversarial attacks have been created. Once the enemy is aware of the signing key, they can create an authentication and authorization the essential path to ensure the entry for the reading the files of the transaction. The fault attack is a side channel attack that is active. The fault attack occurs when a fault is introduced into the integrated circuit card while the device is running the programmed. The antagonist states complete data chambers, with inclusion of the result, in the process of

to get better data with the reference key. The type of attacks are not the same as passive side-channel attacks. [13]

We obtained a description of the Sign Change assault and assessed its likelihood based on this information. Majorly Two different types of accountability attacks on elliptic curve cryptography have been described. This technique involves exchanging the signs of the elliptic curve point. Bloomer and colleagues described the utilization of these keys which are expressed in the form of NFA. By Using NFA representation for the Keys helps in reducing the possibility of occurrence of the sign flip attack will be on the point's mother rather than the computer instruction. In an ECC implementation, the ECC point wouldn't be represented by a sign bit. Additionally, the negative of a point is utilized in the multiple areas that are used to develop an ECC application. Implementing the ECC seems to be a highly preferable alteration of the key stream because it is exceedingly improbable that a sign bit would be used for an extended integer. Here,

we outline two conceivable fault attacks against an elliptic curve cryptography smart card implementation. The integrated circuit card may be used to give a signature in high applications like banks and identity software. The card issuer, not the cardholder, often holds the hidden signature key that is kept in the smart card. The card's owner may try to gain malicious access to this confidential data. To find the secret key, numerous adversarial attacks have been created. Once the enemy is aware of the signing key, the user can create a signature and detour the necessary procedure. if the banking application is compromised it results in huge financial loss. A fault attack is a side channel attack which is active. A Fault Attack occurs when a fault is introduced into a smart card when the tool is running the program. The attacker then examines all the channels with information, which includes the output also, in the attempt to recover information about the key. This type of attacks are not the same as passive side-channel attacks. [14].

Table 1 : Comparative study

	Author	Approach	Advantages	Disadvantages
[1]	H. K. B. Ponnappalli and A. Saxena	New Architecture Digital Signature	Due to this, Web applications may handle digital signatures in a useful, adaptable, and extendable way.	The HTML specification excludes functionality for signatures. Web applications are only compatible with specific browsers and operating systems when signed with third-party signature technologies.
[2]	L. D. Singh and T. Debbarma	elliptic equations utilising a tiny key and it's naive algorithm	lower key size and good security. Secure against Pollard's rho attacks	complexity in solving the discrete logarithm problem
[3]	D. M. Tuan and N. A. Viet	ECC with multiple proxy signature	The insider attack, a potent attack on multi-signature techniques, cannot succeed against this scheme.	complexity in hash function and solving the discrete logarithm problem
[4]	Fatema Akhter	ECC using random sequence approach	increased resistance to cryptographic attacks like random walk and faster performance in terms of computing time when compared to conventional methods.	Any faults occurs during Scalar Multiplication leads to system fail
[6]	S. P. Ganesan	An Asymmetric authentication protocol using ECC and J2ME wireless toolkit 2.5.1	Enhances computational speed, memory size, key lengths, and cryptographic support	Wrong implementation leads compromise
[7]	Y. Qi, M. Xu and C. Tang	an approach for computing $2p+q$ on elliptic curves	can enhance point multiplication computation.	The discrete logarithm problem of elliptic curves does not lend itself to a sub-exponential time attack.
[8]	X. Sun and M. Xia	A Better Proxy Signature System Using ECC	to fix the flaws and security risks in the current systems.	ability to sign in the proxy signature scheme on the original signer's behalf
[9]	F. Amounas and E. H. El Kinani	Elliptic Curve Cryptography Using the Matrix Scrambling Technique	avoids uniformity in the cypher text that results from the transformation of the plaintext matrix, increasing the	utilises sophisticated mathematical calculations to muddle the message

			complexity of decryption.	
[10]	M. Bedoui, B. Bouallegue, B. Hamdi and M. Machhout	Montgomery Algorithm for Scalar Multiplication with Elliptic Curve Fault Detection	An effective defence against fault attacks for the Montgomery ECSM algorithm	mathematically sound, but not always regarded as safe when put into use.
[11]	H. Weiwei and H. Debiao	Isogenies Between Elliptic Curves in the Authenticated Key Agreement Protocol	utilises the isogeny star. Additionally, it was demonstrated that the protocol satisfies the security requirements under the presumption that the isogeny search issue between elliptic curves is secure.	The problem appears to be difficult to solve using a quantum computer because of its exponential temporal complexity.
[12]	Xia Lin	For decryption and encryption of web traffic, ECC relies on pairs of public and private keys.	excellent security with quick, short keys	The standard curves, in particular, are complicated and difficult to implement securely.
[13]	Qiuxia Zhang, Zhan Li and Chao Song	The algebraic structure of elliptic curves over finite fields is the foundation of the public-key cryptographic technique known as elliptic curve cryptography.	DSA is a digital signature algorithm that utilises keys derived from elliptic curve cryptography (ECC)	The hard problem underlying elliptic curve cryptography is the elliptic curve discrete logarithm. Mathematicians have been looking for an algorithm that can solve this issue more effectively than the naïve method for over three decades.
[14]	J. Ling and B. King	Using ECC as an RSA substitute is a reliable cryptographic strategy.	stronger than RSA for current key sizes	The discrete elliptic curve alogarithm is the challenging issue at the heart of elliptic curve encryption.

III. CONCLUSION AND FUTURE SCOPE:

This paper proposes a new ECC scheme by combining digital signatures that provide high security than existing scheme. ECC provides same level security as RSA by using small keys only. This is done to make a system that is less vulnerable to different types of attacks like KPA, CPA, CCA, COA. This project ensures that the message is authenticated with the senders authenticity and verifies at the receiver's side. The message which is also send by the sender is safely reach to the receiver without any modifications. The proposed scheme need to evaluated on different key sizes for further improvement. This needs much research on various Elliptic Curve Discrete Logarithmic Problems. A study on different curves is required to improve the efficiency.

REFERENCES :

[1] H. K. B. Ponnappalli and A. Saxena, "A Digital Signature Architecture for Web Apps," in IT Professional, vol. 15, no. 2, pp. 42-49, March-April 2013, doi: 10.1109/MITP.2012.23.

[2] L. Dolendro Singh and T. Debbarma, "A new approach to Elliptic Curve Cryptography," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, 2014, pp. 78-82, doi: 10.1109/ICACCCT.2014.7019213.

[3] D. M. Tuan and N. A. Viet, "A new multi-proxy multi-signature scheme based on elliptic curve cryptography," 2017 4th NAFOSTED Conference on Information and Computer Science, 2017, pp. 105-109, doi: 10.1109/NAFOSTED.2017.8108047.

[4] F. Akhter, "A novel Elliptic Curve Cryptography scheme using random sequence," 2015 International Conference on Computer and Information Engineering (ICCIE), 2015, pp. 46-49, doi: 10.1109/CCIE.2015.7399314.

[5] Junling Zhang, "A study on application of digital signature technology," 2010 International Conference on Networking and Digital Society, 2010, pp. 498-501, doi: 10.1109/ICNDS.2010.5479249.

[6] S. Prasanna Ganesan, "An asymmetric authentication protocol for mobile devices using elliptic curve cryptography," 2010 2nd International Conference on Advanced Computer Control, 2010, pp. 107-109, doi: 10.1109/ICACC.2010.5486928.

[7] Y. Qi, M. Xu and C. Tang, "An improved algorithm for computing $2P + Q$ on elliptic curves," 2011 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, 2011, pp. 122-125, doi: 10.1109/CYBER.2011.6011776.

[8] X. Sun and M. Xia, "An Improved Proxy Signature Scheme Based on Elliptic Curve Cryptography," 2009 International Conference on Computer and Communications Security, 2009, pp. 88-91, doi: 10.1109/ICCCS.2009.36.

[9] F. Amounas and E. H. El Kinani, "An elliptic curve cryptography based on matrix scrambling method," 2012 National Days of Network Security and Systems, 2012, pp. 31-35, doi: 10.1109/JNS2.2012.6249236.

[10] M. Bedoui, B. Bouallegue, B. Hamdi and M. Machhout, "An Efficient Fault Detection Method for Elliptic Curve Scalar Multiplication

Montgomery Algorithm," 2019 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), 2019, pp. 1-5, doi: 10.1109/DTSS.2019.8914743.

[11] H. Weiwei and H. Debiao, "An Authenticated Key Agreement Protocol Using Isogenies Between Elliptic Curves," 2010 Second International Workshop on Education Technology and Computer Science, 2010, pp. 366-369, doi: 10.1109/ETCS.2010.22.

[12] Xia Lin, "The application of Elliptic Curve Cryptography in Electronic Commerce," 2012 IEEE Symposium on Electrical & Electronics Engineering (EEESYM), 2012, pp. 547-549, doi: 10.1109/EEESym.2012.6258715.

[13] Qiuxia Zhang, Zhan Li and Chao Song, "The Improvement of digital signature algorithm based on elliptic curve cryptography," 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011, pp. 1689-1691, doi: 10.1109/AIMSEC.2011.6010590.

[14] J. Ling and B. King, "Smart card fault attacks on elliptic curve cryptography," 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), 2013, pp. 1255-1258, doi: 10.1109/MWSCAS.2013.6674882.

Letter of Acceptance

TO


**Y Surya Prakash, P Harisankar Narayan, R Ramakrishna, G Sai Sandeep, V Srikara
Sai Ramesh, I Balaraju
GMR Institute of Technology**

Herewith, the conference committee of the International Conference on Augmented Intelligence and Sustainable Systems ICAISS 2022 is pleased to inform you that the peer reviewed research paper **“Paper ID: ICAISS235”** entitled **“Digital Signature and ElGamal Scheme Integration for Secure Data Transmission in Digital Transactions: Survey Paper”** has been accepted for oral presentation as well as it will be recommended in ICAISS Conference Proceedings.

ICAISS will be held on 24-26, November 2022, in CARE College of Engineering, Trichy, India. ICAISS encourages only the active participation of highly qualified delegates to bring you various innovative research ideas.

We congratulate you on being successfully selected for the presentation of your research work in our esteemed conference.

Yours' Sincerely



Dr. A. Pasumpon Pandian
Conference Chair

Proceedings by

19IT701 Project Work**0 0 16 8**

At the end of the project work the students will be able to

1. Identify a contemporary engineering application to serve the society at large
2. Use engineering concepts and computational tools to get the desired solution
3. Justify the assembled/fabricated/developed products intended.
4. Organize documents and present the project report articulating the applications of the concepts and ideas coherently
5. Demonstrate ethical and professional attributes during the project implementation.
6. Execute the project in a collaborative environment.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2				3	2						3	3
CO2	3	3			3								3	3
CO3	3	3	3	2							2		3	3
CO4										3		2	3	3
CO5								3					3	3
CO6									3				3	3