

A Novel Approach for Robust Data Security using Homomorphic Transform

A thesis submitted to the
University of Petroleum and Energy Studies

for the award of
Doctor of Philosophy
in
Computer Science and Engineering

By
Ankit Vishnoi

June 2022

Supervisor(s)
Dr. Alok Aggarwal
Dr. Ajay Prasad
Dr. Manish Prateek



SCHOOL OF COMPUTER SCIENCE
University of Petroleum and Energy Studies
Dehradun - 248007:Uttarakhand

A Novel Approach for Robust Data Security using Homomorphic Transform

A thesis submitted to the
University of Petroleum and Energy Studies

For the Award of
Doctor of Philosophy
in
Computer Science and Engineering

By
Ankit Vishnoi
SAP ID 500033515

June 2022

Internal Supervisor
Dr. Alok Aggarwal

Professor

School of Computer Science
University of Petroleum and Energy Studies

Dr. Ajay Prasad

Professor

School of Computer Science
University of Petroleum and Energy Studies

External Supervisor
Dr. Manish Prateek

Professor

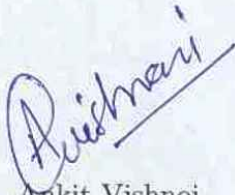
Department of Computer and Information Science
Swami Rama Himalayan University



School of Computer Science
University of Petroleum and Energy Studies
Dehradun, 248007:Uttarakhand

Declaration

I declare that the thesis entitled **A Novel Approach for Robust Data Security using Homomorphic Transform** has been prepared by me under the guidance of **Dr. Alok Aggarwal**, Professor at School of Computer Science, University of Petroleum & Energy Studies, **Dr. Ajay Prasad**, Professor at School of Computer Science, University of Petroleum & Energy Studies and **Dr. Manish Prateek**, Professor at Swami Rama Himalayan University. No part of this thesis has formed the basis for the award of any degree or fellowship previously.

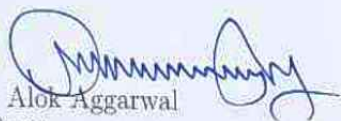


Ankit Vishnoi
School of Computer Science
University of Petroleum & Energy Studies
Bidholi via Prem Nagar, Dehradun, UK, INDIA
Date: 28/06/2022

Certificate

We certify that Ankit Vishnoi has prepared his thesis entitled **A Novel Approach for Robust Data Security using Homomorphic Transform**, for the award of PhD degree of the University of Petroleum & Energy Studies, under our guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

Internal Supervisor:



Dr. Alok Aggarwal
Professor
School of Computer Science
University of Petroleum & Energy Studies
Bidholi via Prem Nagar, Dehradun, UK, INDIA
Date: 28/06/2022

Internal Co-supervisor:



Dr. Ajay Prasad
Professor
School of Computer Science
University of Petroleum & Energy Studies
Bidholi via Prem Nagar, Dehradun, UK, INDIA
Date: 28/06/2022

Certificate

I certify that Ankit Vishnoi has prepared his thesis entitled **A Novel Approach for Robust Data Security using Homomorphic Transform**, for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

External Supervisor:



Dr. Manish Prateek
Professor

Department of Computer and Information Science

Swami Rama Himalayan University

Swami Ram Nagar, Jolly Grant, Dehradun - 248016, Uttarakhand, India

Date: 28/06/2022

ABSTRACT

Digitalization has its advantages but at the same time, it comes with various challenges, of which data security is paramount. This challenge has been dealt with devising various algorithms in the past and still has a lot of potential to develop better algorithms for data security. The study of various existing algorithms for data security shows that there are still several risk factors involved in the process. Algorithms for such as DES, AES, Blowfish, RC4, and many more are mainly defined by two processes for cryptography. One process is the key generation. The speed of the whole cryptographic process depends upon the size and/or length of the key generation process. second process is to develop encryption and decryption algorithm.

Using the homomorphic transform for the cryptographic procedure is one of the methods of safeguarding the data. This is a method that encrypts data and exploits its cyclic, dyadic, and graphical inverse features to make it homomorphic. This work attempts to develop a process, which generates a separate or independent key for each iteration of the key generation process. It aims to improve data security measures by employing the Homomorphic Transform in cryptographic processes to better aid the data protection. The proposed model was tested using random text files as input. The proposed approach has shown promising results when compared with other processes used in well-known cryptographic approaches like DES, 3DES, AES, Blowfish, and RSA. In terms of speed of encryption, the proposed approach is found to be very efficient. Testing on certain prime security parameters like avalanche effect and entropy of cipher text is also performed. Overall, the proposed model is found to be quite up to the mark.

Acknowledgements

Ph.D. journey is truly a life-changing experience and had been a confluence of multiple learning for me throughout.

I express my heartfelt gratitude to my research supervisor Dr. Alok Agarwal, he has been an ideal teacher, mentor, and thesis supervisor, offering advice and encouragement with a perfect blend of insight and humor. Under his guidance, the quality of my research work has continuously improved. He taught me how to understand a problem and exploring the best possible solutions. Without his continuous support, guidance, and constant feedback, this Ph.D., would not have been achievable.

I express my deep sense of gratitude to my co-supervisor Dr. Ajay Prasad who guided me not only like a mentor but a true friend also. I would like to acknowledge for all his help and advice with this PhD. During all the discussions, his suggestions and instructions were undoubtedly helped me a lot. I am feeling blessed and amazingly fortunate to have a mentor like him.

I gratefully acknowledge my eternal Supervisor: Dr. Manish Prateek. He has been supportive since the days I began working on the idea of the thesis; I remember he used to say something like "you're always the first one in and the last one out working on any assignment!" to encourage me to stay focused. He helped me come up with the thesis topic and guided me over almost a year of development. He gave me the moral support and the freedom I needed to move on.

My sincere thanks also goes to Dr. Durgansh Sharma, for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. He has always pushed me to think and produce things in black and white.

My deep appreciation goes to Dr. H G Sastry and Dr. Neelu Jyoti Ahuja for their much need support during this Ph.D. program. They always so helpful and provided me with their assistance throughout my research journey.

I would like to recognize the invaluable assistance of Dr. Varun Sapra, P Srikanth and Dr. Sunil Kumar for their endless support and motivation. Special thanks to Anuj Litoriya who helped me find my footing as I started this process and available whenever there was a need !

Thanks to the University of Petroleum and Energy for providing me the resources such as Library, e papers and lab resources as a means to complete this dissertation.

Would also like to thank to all the researchers whose work I have studied during this time, that have enlighten me to go ahead with the proposed work.

I am blessed with a beautiful and supportive family, I am unable to express my gratitude in words to my parents and in laws, for teaching me to appreciate and never give up approach.

My Wife, Shweta, who has been by my side throughout this Ph.D. Thanks to her for constantly listening to me rant and talk things out, for proofreading over and over (even after long days at work and during difficult times), and for the sacrifices she has made in order for me to pursue a Doctorate degree.

Nevertheless my daughter, Aaradhya, for always being there for me and for telling me that I am awesome even when I didn't feel that way. For believing in me and being my star.

Ankit Vishnoi

Table of Contents

Declaration	i
Certificate	ii
Certificate	iii
Abstract	iv
Acknowledgements	v
Table of Contents	vii
List of Abbreviations	xii
List of Figures	xiv
List of Tables	xvi
1 INTRODUCTION	1
1.1 ENCRYPTION	1
1.1.1 Encryption Algorithms	2
1.2 TYPES OF ENCRYPTION ALGORITHM	4
1.2.1 Symmetric Encryption Algorithm(SEA)	4
1.2.2 Asymmetric Encryption Algorithm	6
1.2.3 Hashing Scheme	8
1.2.4 Message Authentication Codes	8
1.2.5 Authenticated Encryption Schemes	9

1.2.6	Cryptanalysis	9
1.3	HOMOMORPHIC TRANSFORM	15
1.3.1	Introduction to Homomorphic Transform	15
1.3.2	Properties of Homomorphic transform	15
1.4	MOTIVATION OF THE PROPOSED WORK	16
1.5	APPLICATIONS OF THE PROPOSED SYSTEM	17
1.5.1	Text Encryption	17
1.5.2	Image Encryption	17
1.5.3	One Time Pads (OTP)	18
1.5.4	Bank Transactions	20
1.5.5	Military Communication	20
1.5.6	Bio-metric verification	21
1.6	CHALLENGES	21
1.7	SCOPE AND OBJECTIVE OF THE THESIS	22
1.8	CONTRIBUTIONS	23
1.8.1	Proposed Framework	23
1.9	THESIS ORGANIZATION	24
2	RELATED RESEARCH WORKS	25
2.1	INTRODUCTION	25
2.1.1	Conventional Encryption	26
2.1.2	Public Key Encryption	29
2.1.3	Contemporary Encryption	31
2.1.4	DNA Encryption	33
2.2	COMMON ENCRYPTION ALGORITHMS	34
2.2.1	DES	34
2.2.2	Triple DES	35
2.2.3	RSA	36
2.2.4	AES	37

2.2.5	Blow Fish	37
2.2.6	Twofish	38
2.3	CONCLUSION	38
3	TEXT CRYPTANALYSIS	39
3.1	INTRODUCTION	39
3.1.1	Cryptanalysis	40
3.1.2	Need for Cryptanalysis	41
3.1.3	Attacks in Cryptanalysis	42
3.1.4	Cryptanalysis Vs. Cipher Design	43
3.1.5	Types of Cryptanalysis on the text	44
3.1.6	Histogram Analysis	47
3.1.7	Key Space Analysis	47
3.2	OUTCOME OF CRYPTANALYSIS	47
3.3	CHAPTER SUMMARY	48
4	FRAMEWORK FOR DATA SECURITY USING HO-	
	MOMORPHIC TRANSFORM	49
4.1	PROPOSED FRAMEWORK	49
4.1.1	Hadnard Transform	50
4.1.2	Homomorphic Transform	50
4.1.3	Algebraic properties of the proposed Method	52
4.2	RESULTS	57
4.3	CHAPTER SUMMARY	60
5	DATA SECURITY MODEL USING SET THEORETIC	
	HOMOMORPHIC TRANSFORM	62
5.1	PROPOSED WORKING MODEL	63
5.1.1	Set-Theoretic Approach of Homomorphic Transform	63
5.1.2	Proposed Data Security Algorithm	65

5.2	CYCLIC SHIFT-INVARIANCE	65
5.3	GRAPHICAL INVARIANCE	67
5.4	DYADIC SHIFT-INVARIANCE	68
5.5	RESULTS	69
5.6	CHAPTER SUMMARY	70
6	IMPLEMENTATION OF A HOMOMORPHIC TRANS-	
	FORM TO SECURE THE TEXT DATA	72
6.1	EXPERIMENTAL SETUP AND DESIGN	73
6.2	EXPERIMENTAL OUTCOMES	73
6.2.1	Execution Time	74
6.2.2	Entropy	74
6.3	COMPARATIVE ANALYSIS	76
6.3.1	Execution Time	77
6.3.2	Avalanche Effect	79
6.3.3	Entropy	80
6.4	VERIFICATION AND VALIDATION	82
6.5	CHAPTER SUMMARY	83
7	CONCLUSION AND FUTURE DIRECTION	84
7.1	RESEARCH SUMMARY AND CONTRIBUTIONS	84
7.2	RESEARCH CONTRIBUTIONS	87
7.2.1	Framework to Enhance the Data Security Measures using Homomorphic Transform	87
7.2.2	A Hadamard Transform using DNA Amino Acid and Cryptography	87
7.2.3	A new robust cryptographic technique to improvise data security in text messages.	88
7.3	FUTURE RESEARCH DIRECTIONS	88

Bibliography	90
Annexure	107

List of Abbreviation

3DES - Triple Data Encryption Standard
ACPA - Adaptive Chosen Plain-text Attack
AE - Authenticated Encryption
AEAD - Authenticated Encryption with Associated Data
AES - Advanced Encryption Standard
CBC - Cipher Block Chaining Mode
CFB - Cipher-Feedback Mode
CoA - Cipher-text only Attack
CPA - Chosen Plain-text Attack
DES - Data Encryption Standard
DIP - Digital Image Processing
DNA - Deoxyribo Nucleic Acid
ECC - Elliptical Curve Cryptography
FISH - Fibonacci SHrinking
GPG - GNU Privacy Guard
HT - Homomorphic Transform
IAP - Internet Access Point
IDEA - International Data Encryption Algorithm
ISP - Internet Service Provider
KPA - Known Plain-text Attack
MAC - Message Authentication Code
MD5 - Message Digest Method 5
OTP - One Time Pads
PCR - Polymerase Chain Reaction
PGP - Pretty Good Privacy
PRNG - Pseudo-Random Number Generators
RC4 - Rivest Cipher 4

RSA - Ron Rivest, Adi Shamir and Leonard Adleman

SHA - Secure Hash Algorithms

SKC - Symmetric Key Cryptography

TSP - Travelling Salesman Problem

List of Figures

1.1	Standard cryptanalysis procedure [1]	2
1.2	Synoptic of cryptanalysis flow to the encryption and decryption of message sent through the untrustworthy network [2]	3
1.3	Fundamental procedure of asymmetric encryption [3]	7
2.1	Working Example of Vigenere Cipher [4]	26
2.2	Working Example of OTP [5]	26
2.3	Illustration of Stream Cipher and Block Cipher [6]	28
2.4	Working of Symmetric Cryptosystem [7]	29
2.5	Public Key Encryption [8]	31
2.6	Enciphering procedure of DES [9]	35
2.7	Triple DES Block Diagram [10]	36
2.8	RSA Encryption Process [11]	36
2.9	AES Encryption Process [12]	37
2.10	Blowfish/Twofish Procedure [13,14]	38
3.1	Classification of cryptology [15]	40
4.1	Encryption Model using Hadamard transform	50
4.2	Hadamard Transform	51
4.3	Homomorphic Transform in Signal Processing	53
4.4	Homomorphic Transform in Signal Processing (Inverse)	54
5.1	Encryption of four sets of values	64
5.2	Decryption process of four sets of values	65

5.3	Cyclic Shift-In-variance	67
5.4	Graphical In-variance	67
5.5	Input Text File	69
5.6	Processed Cipher File	69
5.7	Decrypted Text File	70
6.1	Text File Size and Execution Time	74
6.2	Input File Size and Number of Iterations	75
6.3	Histogram of Entropy values received from encryption file . .	76
6.4	Execution time of AES and the proposed scheme versus file size	78
6.5	Execution time of various schemes and proposed scheme with respect to the file size	79
6.6	Avalanche Effect in Percentage	80
6.7	Average Entropy of different cryptographic approaches Vs. proposed approach	81

List of Tables

4.1	Encrypted output of the given number sequence	54
4.2	Encrypted output of the cyclic shifted number sequence . . .	55
4.3	Encrypted output of the graphical inverse number sequence(P- 1)	56
4.4	Encrypted output of the graphical inverse number sequence (P-2)	56
4.5	Encryption output of the dyadic shifted number sequence (part 1)	56
4.6	Matrix Representation of Numeric Digits	58
4.7	Matrix Representation of Alphabets	59
4.8	Encrypted Matrix Representation of Alphabet [A] and Ci- pher Sequences	60
4.9	Error Observation of character [A] for Various Orientations .	61
6.1	Simulation Setup & Design	73
6.2	Execution time versus file size	77
6.3	Execution time of AES and proposed approach with respect to various file size	77
6.4	Execution time of available schemes and the proposed scheme versus file size	78
6.5	Comparison of Encryption Techniques with Proposed Trans- form	82

Chapter 1

INTRODUCTION

With the advancement of information technology and side-by side chip technology, a huge amount of data in the form of text, audio, video is being generated every day. While transmitting through the insecure channel of communication for various purposes the data must be ciphered in an unintelligible form [1]. Efficiency of the contemporary encryption algorithms has improved a lot during last few years due to cheaper hardware cost, especially the gate equivalents, involved in encryption process. Transformation of a information over a communication channel is made secured by hiding or substituting part of the information with numbers, special characters, etc. This process of hiding, substitution etc. of information is in general termed as cryptographic process. The cryptographic algorithm plays a vital role in maintaining security, accuracy and efficiency [1].

1.1 ENCRYPTION

Now-a-days securing digital information such as privacy data, scanned medical images, financial data, military information and digital images has become the need of the hour due to extensive use of information technology even on pocket device like smart phones. This digital information can be in

the form of a text, audio, images or video data [16]. Cautious accessing or sharing of the data on an unsecured multiparty accessing communication network is the requirement. Since the unsecured network is vulnerable to the third party access, it must be authorized with highly effective parameters [17]. Multimedia data is secured through different ways compared to text data. However both can be enciphered according to the digital value or pixel representation.

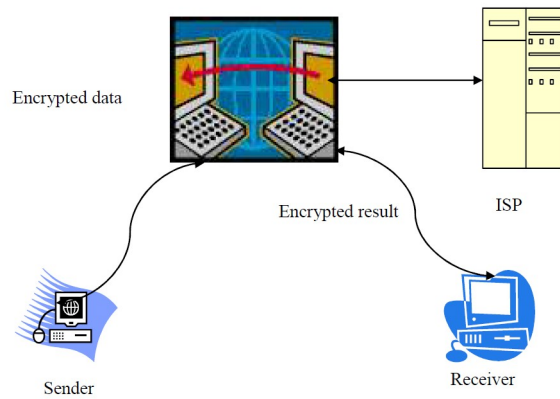


Figure 1.1: Standard cryptanalysis procedure [1]

In general, service provider gets only the encrypted cipher text. Data compression makes the communication easier and also it reduces the computational overhead due to its reduced ciphered text size [18]. Internet Service Provider (ISP) is usually an organization which gives personal as well as business access of the Internet, also called as Internet Access Providers (IAPs). All ISPs or IAPs are also connected with one another [19]. These can be accessed via the network which they provide. It is vital to use encryption process while transferring the data over the Internet from data securing purpose as shown in figure 1.1.

1.1.1 Encryption Algorithms

Contemporary encryption algorithms deals with the plaintext, encryption key, decryption key and cipher text. The encryption algorithm and its de-

tails are known to all except the keys. It is a simple flow of encryption of plaintext and the transmission of encrypted plaintext to the sender over the network. $enc(m)$ is evaluated by using the $f(m)$ function [20]. $f(m)$ function evaluates the encrypted value of m . The $enc(f(m))$ is decrypted. It should be equal to the encryption of m . This is the Homomorphic property which is used in the advanced encryption standards as depicted in figure 1.2 [2], which shows the synoptic of cryptanalysis flow to the encryption and decryption of message sent through the untrustworthy network.

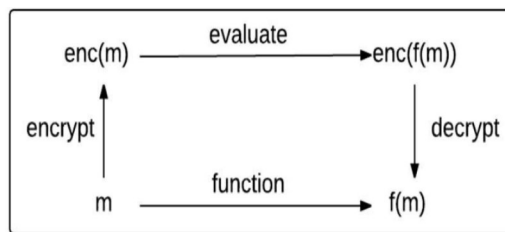


Figure 1.2: Synoptic of cryptanalysis flow to the encryption and decryption of message sent through the untrustworthy network [2]

The third party network is usually not a secure network. Due to which, system uses different types of techniques to secure the message which is sent by the sender [21]. The key size is increased for improving data security. The security of the message depends on the various parameters like the number of times of encryption, the algorithm which the system uses, the key size, the message size, encipher, decipher algorithm, etc. These parameters define the speed, efficiency, performance and security of the cryptosystem [22]. Randomness and Kirchhoff's principles are applied to make the system invulnerable to crypt analytic attacks. While processing the image, pixels are handled block by block to do the computation.

This flow of conversion can be done with the additive function and multiplicative function. Certain arbitrary functions are allowed to the $f(m)$ function. It depends on the type of cryptosystem which is used in the algorithm. Addition, multiplication and exponentiation are used in the

various Homomorphic properties. AND gate defines the multiplication, whereas XOR defines the addition operation with the inputs. In encryption, the data before encryption should be retrieved after the decryption. In image input the negotiable amount of change in the deciphered plaintext is allowed [23].

1.2 TYPES OF ENCRYPTION ALGORITHM

The strength of the encryption scheme is Untroubled by using the block size. It's strength is calculated by the key length [24]. Though any block size is acceptable, the following considerations should be examined while choosing a block size. If the size of a block is m bits, then the total number of plaintext bit combinations is 2^m . If the attacker finds the plaintext blocks that correspond to previously sent cipher-text blocks, the attacker can execute a "dictionary attack" by compiling a list of plaintext/cipher-text pairings sent with that encryption key. Because the dictionary must be larger [25], a greater block size makes the assault more difficult. The encryption process becomes less efficient to operate if the block size is quite large. Mostly, the block size is a multiple of 8, since it is simple to create and most world length of all processes is in bytes. Various encryption algorithm techniques are discussed in the subsections.

1.2.1 Symmetric Encryption Algorithm(SEA)

SEA is also called as private key encryption. In this type of algorithm, both the sender and the receiver uses the same key to do the transmission over the insecure communication channel [26]. Since the encryption uses the same key, the unknown person cannot use this encryption. It is faster compared to the asymmetric key encryption due to its simpler com-

putations. AES (Advanced Encryption Standard) [27] is one of the most popular schemes among various symmetric key cryptosystem.

While doing encryption there is a possibility of getting the same cipher text for the plaintext. The attacker may do computations over the cipher text to seek the plaintext in the transmission. The plaintext must be given with the randomness [28], so that every time plaintext gets encrypted it will give a different cipher text. The attacker can perform computation to find the secured data, but will not get it. Each time during transmission over the insecure communication channel, for the same plaintext different cipher text will be created. This is called deterministic approach with the probabilistic feature enabled system [29].

Transposition cipher is an encryption technique, where the position of the plaintext is changed or shifted. Transposition can be done either row-wise or column-wise. Double transposition is more secure compared to the single transposition [30]. The location and the number of positions are noted. In the substitution cipher case, different data is substituted to make the data invisible to the attacker. Affine cipher and Caesar cipher are the popular ciphers among substitution ciphers. Stream cipher depends on the state of the cipher hence known as state cipher. Binary additive streak cipher is a synchronous cipher. Rivest Cipher4 (RC4) [31] is a popular stream cipher. It is a Rivest cipher. Snow and Sober also very efficient and wisely used stream cipher. AES is a popular encryption scheme which has a key size of 256.

The private key encryption schemes are not very complex but faster than the public key encryption schemes. While compared to the conventional encryption schemes, the advanced contemporary schemes provide high security with the smaller key sizes [32]. The basic principle of encryption is to conceal the data which are stored, transmitted and accessed be-

tween the sender and the receiver. Both traditional and emerging advanced security methods are used by the researchers. The security techniques are used to make boundaries of access to the data. The confidentiality of the personal, professional, governmental and commercial data are preserved. Only the owner of the data can access, give access, transfer and update changes in the data [33].

Cryptanalysis is the study of analyzing the secure communication channel and tries to open the hidden items in the message. Cryptanalyst is the one who makes the secure communication as an insecure communication. Various attacks such as cipher image attacks, known image attacks, neighbor attacks, plain image attacks, known plaintext attacks are analyzed to see the performance of the system.

Confidentiality, integrity, and authentication are one of the major security measures which are maintained in a public key encryption scheme. 3DES (Triple Data Encryption Standard) [34] algorithm is more complex than the Data Encryption Standard (DES) [35]. AES can have 128 or 256 key size. RC4-256 is an Advanced Encryption Standards-128. The speed of approaches of private key encryption is relatively faster when compared public key encryption. Key sharing between users of the communication is not easy in private key encryption. In public key, encryption keeping both the secret key and public key has its own special care to maintain.

1.2.2 Asymmetric Encryption Algorithm

Asymmetric key encryption deals with the pair of keys to perform the encryption. There is no need to share the same key as in case of symmetric key encryption [36]. It is comparatively slower compared to the asymmetric key encryption due to its computation over the plaintext. Paillier encryption algorithm and Elgamal are among the popular public key en-

encryption algorithms where Paillier is additive and Elgamal is multiplicative cryptosystem [37].

As in the outline sketch of the public key encryption, the sender sends the message M to the receiver via the insecure communication network. The encryption function encrypts the message by using the key K_1 . The fundamental procedure of asymmetric encryption is defined is shown in figure 1.3, where the receiver gets the ciphered text and it is deciphered using the key K_2 of the sender. Randomness is included to make the different cipher text to the same plaintext [3]. Cryptanalyst is an intermediate who analyses the cipher text to get the plaintext without having the key. Protection of the encryption is very high due to the randomness. It cannot be reversed easily.

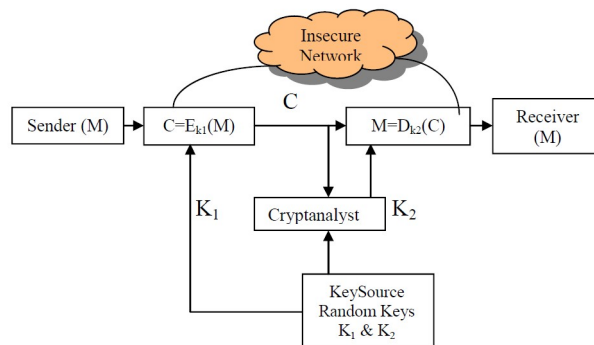


Figure 1.3: Fundamental procedure of asymmetric encryption [3]

Stream ciphers, Block ciphers, Transposition cipher and Substitution cipher are known for their variable key size in the public key encryption schemes [38]. Security of the data increases with increase in key size. In compare to public key crypto scheme, brute force attack is possible in private key encryption algorithm, because the size of the key is comparatively simpler than the public key encryption [39].

DES, AES and International Data Encryption Algorithm (IDEA) [40] are very popular and widely used private key encryption algorithm. It is a process of making assumptions by guessing values with the comparison

of available cipher text and plaintext value. Java Net beans have been used for comparing the public key encrypted arithmetic operations over big data. Runtime has been noted with that java platform for comparison. The result of the runtime proves, that the public key schemes provide more protection than the private key encryption.

1.2.3 Hashing Scheme

Hashing Function in the cryptography only encrypts the data. This is a basic tool in cryptography. It will give a message digest. With the hashing function there is no decryption. It is used to compare the data which is encrypted using the hash function. Message Digest Method 5 (MD5) [41] and Secure Hash Algorithms (SHA) [42], are hashing functions which gives a digest. Hashing schemes are deterministic in nature. So the same plaintext always gives the same output. It converts the message into message digest.

1.2.4 Message Authentication Codes

To take the portal/account access a user entered a security code, called Message Authentication Code (MAC) or Tag [43]. In general, this tag is passed to the user by user's request or through a simple message. The receiving end should decode the MAC to gain the user's account access. In financial cryptography, MAC codes has an important role. And it is essential for accessing bank, banks account, trust companies, brokerage businesses, insurance institution, investment/deposit organization. These organizations suggests the internet access to utilise MAC.

1.2.5 Authenticated Encryption Schemes

Authenticated Encryption (AE) [44] and Authenticated Encryption with Associated Data (AEAD) [45] are encryption techniques that provide data secrecy while simultaneously verifying its validity. Authenticated encryption, in addition to message integrity and secrecy, can protect against a specific cipher-text attack. In these attacks, an attacker sends precisely prepared cipher-texts to a “decryption oracle” and analyses the decrypted output to gain an edge over a cryptosystem (e.g., information about the secret decryption key). Authenticated encryption systems can identify and refuse to decode badly created ciphertext. As a result, the attacker is precluded from requesting the decryption of any cipher-text unless the cipher-text was successfully created using the encryption algorithm, implying that the plaintext is already known. Authenticated encryption is properly implemented, renders the decryption oracle ineffective by preventing an attacker from getting relevant information that they do still not possess.

1.2.6 Cryptanalysis

Cryptanalysis is the process of breaking up the cryptosystems by identifying and exploiting its vulnerability. The symmetric and asymmetric encryption, both are based on the central dogma that the algorithm or working principle can be made publicly available, but the keys must be kept surreptitious from the attackers to avoid its breakdown. To withstand the possible attacks on a cryptogram, the cryptographer must design the cipher using the following goals:

- **Confidentiality:** This is the main requirement of any cryptographic algorithm as this ensures that the message contents remained private or confidential and are not divulged intentionally or unintentionally to any unauthorized user [46].

- **Integrity:** It simply means that the message contents have not been altered during its transmission from the sender to the receiver. It also assures that message is protected from any unauthorized access. Usually, the one-way hash functions are employed for achieving integrity [47].
- **Availability:** It ensures that the information is available to the authenticated and authorized users anytime whenever required. Timely and reliable information access is also the leading goal of an information system.
- **Authentication:** It is the process of validating the individuals involved in the communication. There can be many ways to authenticate users like the username and the corresponding password [48].
- **Non-Repudiation:** It guarantees that the communicating parties cannot deny that the communicate ever happened, if it actually happened. Through it, over the network, a sender and a receiver cannot deny to send and receive messages [49].

Besides the above-mentioned safety goalmouths, two other factors need to be considered: key generation and keyspace. The key generation is usually implemented through a different algorithm and the cryptographer must ensure that this algorithm is not producing recurrent keys as this will dwindle the cryptogram. Keyspace refers to the possible number of keys generated, and larger keyspace offers a greater number of keys making the brute-force cryptanalysis even harder. However, there are other possible elucidations already known to the cryptanalyst through which they can exploit the pitfalls in the encryption key generation. Also, there are some attacks that can help the cryptanalyst to recover the secret key without trying all possible combinations.

The security attacks can be many and varied which usually depend on

the various factors like the attacker does with the information retrieved. One categorization can be passive attacks, where the attacker just reads the secret documents but does not alter them in any way. Examples of such attacks are traffic analysis, eavesdropping, or releasing the message contents in the public domain. Another tagging can be active attacks, where the attacker may choose to alter or destroy the information. Illustrations of active attacks can be message modifications or denial-of-service attacks where the server is loaded with heaps of illicit requests that may cause the server to break down. Some other major attacks are listed below [50]:

- **Brute-Force Attack:** It refers to the most primitive weapon in the arsenal of a hacker, where he tries every possible combination to break the key. Obviously, it is more time-consuming and totally dependent on the key length.
- **Replay Attack:** It is a variety of passive attacks in which the large amount of content is repeatedly sent to the receiver to cause irritation or harm. Although now-a-days, it can be avoided by using expiration period and timestamp.
- **Man-in-the-Middle Attack:** This is the mischievous act where a person basically sits in the middle of communication and can masquerade to one user as the other legit user.
- **Frequency Analysis:** As the name signifies, the attacker takes benefit of the highly frequent characters or words that are usually present in the message. For Example, vowels like a, e, i, o, u tend to appear more often than other consonants. Such behaviors and patterns are identified to ripple the communication.
- **Ciphertext only Attack (CoA):** This attack refers to the situation where the attacker has access to both ciphertext and the underlying

cryptographic algorithm. Based on these two vital materials, the hacker tries to deduce the plaintext or cryptographic key.

- **Known Plaintext Attack (KPA):** It is a more advanced version of the ciphertext-only attack, where the foe expert has access to cipher algorithm, ciphertext, and also a part of plaintext.
- **Chosen Plaintext Attack (CPA):** It is an advanced form of known plaintext attack, where the attacker may have access to multiple plaintext and their corresponding ciphertext. It may also mean that hackers may access the cryptogram with a rooted cipher key.
- **Adaptive Chosen Plaintext Attack (ACPA):** Progressive version of chosen-plaintext attack where an attacker can adapt plaintext based on previous crypto results.

The cryptographic algorithms must be able to withstand all the above-mentioned attacks. Cryptanalyst may deploy eccentric tactics like statistical analysis like birthday attack or may take advantage of the human language used in the cryptosystem or use the knowledge of programming language data types, exploit peculiar patterns, or even exploit the viable characteristics of media like entropy in the image encryption etc., to compromise the security of the cryptosystems. Cryptanalysis is divided into two categories.

1.2.6.1 Linear Cryptanalysis

It is a mathematical approach that combines algebraic, linear cryptanalysis, statistical, and numerical techniques [51]. It works by looking for affine approximations to the cipher's action. An algorithm (or cryptographic methodology) and a cryptographic key are the two major keys of cryptography. A complex mathematical function, which is used to apply the cryptographic protection such as encrypting the data, constitutes the algo-

rithm. The cryptographic key is utilized to decrypting the encrypted data, which entails reversing or verifying the process; it is a factor used by the process. A secure key makes cryptographic process secure and reliable. It is an important factor, since the efficacy of the protocols is related to the keys and its shielding. There are varied existing algorithms for encryption such as DES, AES, Blowfish, RC4 and many more [52]. As discussed above there are two processes for cryptography. One of the main processes is key generation. The size of the key generation defines the time complexity of encryption techniques. Algorithm speed depends upon the key generated for cryptographic process. A wide variety of methods is present for key generation but the basic problems arise during key generation. The key should be secured enough for data security.

For the current study, cryptography is an encoding algorithm, a piece of software, which encrypts the information. This piece of information is useless, until it is decrypted. On arrival of encrypted information at the receiver's end, it is decoded by the computer via same software. This message transforms into an information then. No change occurs in remaining ciphertext's decryption if one character altered in input ciphertext but after addition/deletion a character in input ciphertext, the message organization (synchronization) is broken and remainder decryption aborted. In asynchronous (self-synchronizing) cipher, from the key, a keystream prepared based on the key. This keystream previously defined the ciphertext characters. The present state depends only on the fixed number of ciphertext characters, so this ciphertext stream can superior while addition/deletion of characters. After processing the alteration process, the overall ciphertext again synchronized.

1.2.6.2 Differential Cryptanalysis

It is the study of how changes in information input affect the outcome difference. It refers to a set of techniques for tracing differences through a network of transformations, identifying non-random behaviour in the cipher, and exploiting such qualities for recovering the secret key in the case of a block cipher. The attack is based on the fact that particular input/output difference patterns only occur for specific input values. In most cases, the non-linear components were present in S-boxes/ look-up tables and if the non-linear components were solid components, the attack was applied on it [53]. The change in intended output proposes alternative key (between known/two chosen text inputs).

In today's digital era, the most talked about phrase is data Security, also referred as information security. Generally, security can be summed up as "the state of acquiring protection to attain a danger free domain/status". The level of security is variable, which can be customized from domain to domain. It indicates to defensive advanced protection measures, which are connected to deter unauthorized access to websites, PCs and databases. The degradation of the information can be shielded by information security, which is core part of Information Technology for associations of each sort and size. The information security contributes to shield the reliability and authenticity of information source, which could be in the form of storage, processing, or transmission. Information security is attained through the implementation of policy, education, learning and technology. The process initiates with encoding the information, which is done by the computer, this is in turn fired towards its destination.

1.3 HOMOMORPHIC TRANSFORM

Homomorphic Transform (HT) [54] is a tool which is featured as high-speed spectral domain tool. The Digital Image Processing (DIP) [55] operation are utilized for object recognition. The research carried out so far reflects the algebraic properties of homomorphic transform, it is the best process could be employed in cryptography to make data secure.

1.3.1 Introduction to Homomorphic Transform

Homomorphic transform is a fast technique that is similar to Fast Fourier and Hadamard transform but takes a dissimilar approach. The Homomorphic Transform converts a numerical sequence $a(r)$ of length $R = 2^o$ where $o \geq 0$ to the cipher sequence $A(o)$. It can encrypt every sequence of numbers via isomorphism. Homomorphism is defined as a transform that follows the cyclic, dyadic, and graphical inverse features of contemporary algebra. The HT incorporates these several algebraic sequences $a(r)$ into the same $A(o)$ but with distinct key $E(r)$. In other words, for a given set of values, a new set of cyclic, dyadic, and graphical inverse processes are encoded into the same sequence of the cipher-text using various authentication keys, demonstrating why it is required as a transform [55].

1.3.2 Properties of Homomorphic transform

HT system is a type of public key encryption scheme which allows algebraic operations on the cipher texts. It is an important advanced technique for enabling computation and analytical processing on encrypted data given to insecure third communication channel. This is multiplicative encryption system. In this scheme, the multiplication of ciphered text can give the encrypted plaintext. Without knowing the original data, the operations are done on the encrypted text. Expansion of the cipher text is longer than

the plaintext. The ratio is high for ciphertext compared to the plaintext. Comparatively minimum expansion gives minimum noise to the computation. Diffie-Hellman, Paillier, Elgamal, Goldwasser Micali and DGK encryption schemes [56] are very popular and widely used encryption standards in public key encryption schemes. An efficient binary operations on the real numbers used for Homomorphic property is utilized in all types of data (text, audio, video and image) files. This has computational overhead due to the high security feature.

1.4 MOTIVATION OF THE PROPOSED WORK

Most of the available algorithms either use the same key or a pair of keys for the cryptographic process. The process of key generation is also some permutation and combination, use of real numbers, and a few key generation algorithms. These algorithms generate the same sets of keys for the entire data, which in turn makes the entire process compromised if the key is exposed to the hackers. This is the motivation for this proposed work to introduce an approach to generate different keys with each set of data which avoids unwanted access to the encrypted file. The proposed approach is compared to AES, the well-known approach to cryptography. Results show that for various file sizes ranging from 2.5 MB to 23.3 MB the proposed approach gives a minimum of 54% improvement compared to AES. There is no overhead on the message and key transfer using the proposed technique. The execution time of the proposed algorithm is better than available algorithms like DES, 3DES, AES, Blowfish, and RSA, below 1 MB data size. However, it gives significant execution time for 1 MB file size or more.

1.5 APPLICATIONS OF THE PROPOSED SYSTEM

Several potentials application areas are identified where a variation of different types of Homomorphic cryptosystem can be applied. The main purpose of the proposed cryptosystem is used in Industrial, governmental, financial, insurance and healthcare. In this research the applications of the Optimized Homomorphic transform is in the fields of message encryption, one time pads, bank transactions, military communication, image encryption, bio-metric verification and private information retrieval. The following subsection further explores the application domain of the proposed work.

1.5.1 Text Encryption

The process of encryption and decryption of text is now-a-days extended to image, audio and video also. The improvement in the web application development and the percentage of people accessing global internet has increased. So the secret way of multimedia data encryption using different geometrical shape and properties has increased over communication media.

The conventional encryption algorithms have some problems like data size, relationship among pixels before and after changing. The traditional cryptography techniques cannot be directly applied to Image, audio & video because of pixel neighbourhood relationships, pixel redundancy. The symmetric key encryption algorithms such as DES, Triple DES, AES etc., can be applied directly on text but not on images.

1.5.2 Image Encryption

To improve security to the data in today's communication systems, there is a large demand in various cryptographic techniques. The security ex-

perts search for new ways of ensuring data security. One such technique is image based encryption which provides a strong secret key and different permutations and combinations in the encryption process. Encryption of image data is different from the encryption of text data as images have high resolution and the pixel redundancy is also more. Based on the type and size of the image, the whole volume of data changes.

Image encryption methods are the better techniques to encrypt multiple data inputs such as text, image and audio data. When information is exchanged via a network, information is concealed through the encryption process. With some inherent qualities, such as a higher link between pixels and a mass information capacity, image security differs the from text security. Because of this, image encryption using traditional encryption algorithms like the IDEA, DES, and AES is not very sufficient. For the security of images, a number of approaches, including steganography, watermarking, visual cryptography etc., have been used since very long. Now, a number of fresh approaches using diverse techniques and calculations have been put forth. Chaotic systems, optical transforms, DNA cryptography, and compressive sensing are a few of these [57].

1.5.3 One Time Pads (OTP)

One-Time Pads (OTPs) [58] refer to the most secure cryptographic systems at least theoretically and are considered almost impossible to crack. The most important characteristic of an OTP is the secure key generation that is random in nature. It comes under the flag of stream cipher where plaintext is picked byte-wise and a unique key s applied on it. After the encoding process, this very key is destroyed and not used again. In the earlier times, these keys were written on paper, hence named One-time pad.

The central dogma of OTP was first described by Frank Miller in 1882

[59], but the credit of modern-day OTP goes to Gilbert Vernam Joseph Mauborgne in 1919 who suggested the idea of always using a random key for encryption. Some mandatory requirements of finding the perfect one-time pads are: truly random key, single time usage only of the key, only two copies of OTP must exist, utmost secrecy of the key, and finally, the key length must be greater than or equal to the length of plaintext that needs to be enciphered.

Historians believe that OTP ciphers were extensively used throughout both world wars and in the cold war era. The Russian OTPs were extremely popular and contained numbers only. Their implementation was done using OTP booklets which were a stack of thin pages with a series of five number sequences; and after usage, each page was removed from the booklet. Spy radio sets like R-353 were extensively used to share the message using short-wave radio bands. The trend of using OTPs continued to the modern day, but the biggest challenge is to generate long and truly random keys for encryption. To realize this, Pseudo-Random Number Generators (PRNGs) can be used, but the predicament here is that these numbers are pseudo-random but not truly random; hence they are not considered a good contender to produce OTPs.

The solution here is the use of real-world physical phenomena like cosmic emissions or even DNA. This very inkling of high arbitrariness and unpredictability has made DNA a popular choice to generate random long keys for OTP ciphers. There are primarily four biological processes for it: Polymerase Chain Reaction (PCR) [60], electrophoresis gel screening, DNA preparation, and sequencing. Apart from the genomic processes, DNA-based OTPs can also be generated using rotating, multiplexing, and concatenation of two or more partial chromosomal sequences generated from a single downloaded DNA reference.

The usual cryptographic operation applied is the XOR binary operation because of two reasons: first, the output after the operation does not increase or decrease the bits and second, the operation produces similar results applied again; hence it is a faster and efficient way to obtain ciphertext from plaintext. Nevertheless, the key feature of DNA-based OTP is that enormous DNA sequences make it possible to spawn random keys, and the key length must be equal to (or greater than) the plaintext that needs to be enciphered. However, the key length increases or decreases according to the media type; for example, the sound file requires a longer sequence than the image file which in turn needs lengthier chromosomal series than the text file and all of them could be satisfied by downloading just 1 DNA sequence of 8 bytes from the genetic databases. The largest DNA reference is desirable for the video files that use nearly 48 bytes of DNA sequence for encoding an approximately 330 MB video.

1.5.4 Bank Transactions

To grant the user access, the receiving system must recognize the message authentication codes attached to the message. For accessing a bank account, MAC are very essential codes. These codes can be used by brokerage firms, banking organizations, deposit/investment institutes, insurance company and trust companies that provides Internet access. They're an important part of financial cryptography.

1.5.5 Military Communication

For a long time, cryptography has been a crucial aspect of combat. It's a method for the military to safely transfer communications without their enemy monitoring it. Even if the opponent retrieves the communication, it must decode it before it can be used.

1.5.6 Bio-metric verification

Bio-metric verification is a category of emerging technologies that effectively link a cryptographic key or create a digital key from the bio-metric. There is no bio-metric template or image available. In the bio-metric verification template, no digital key (bio-metric) can be recovered from the bio-metric verification template that has been saved. This process is also known as a “bio-metric identification encoded key” or “helper information”. Bio-metrics differs fundamentally from other systems that use normal encryption to encode bio-metric images or templates, or that retain a secret key and release it upon valid bio-metric identification. The digital secret may be reproduced using bio-metrics as long as the correct bio-metric sample is offered during the validation. The result of bio-metric identification is either a digitized key or a failure notification.

1.6 CHALLENGES

Major challenges in the existing systems are summarized below:

- **Computational overhead:** It should be less as much as it can. It is one of the main performances metric which affect the output of the crypto systems.
- **Data security of the system:** The security of the saved data is very essential to keep it safe. Either it may be a user data or the database server data, it need to be protected with the latest methods and techniques of the system.
- **Large storage space:** Since the size of the data and the encrypted data is high and it should be stored separately hence a huge amount of storage space is required. Use of hand held devices like smart phones, are increasing constantly even by a common man for various

kind of data like text, image, audio or video, due to the latest advancements in the hardware technology. This further completes the storage requirements.

- **Predictability by the hacker:** It is one of the main domains in the cryptography which desires the performance of the system. Various attacks are applicable and can be done on the available data to increase the security.
- **High precision of keys:** The output of the encryption is not a simpler data. Due to the use of big size key the output will also have so many bits. It should be protected to make the system as more protected.

1.7 SCOPE AND OBJECTIVE OF THE THESIS

The scope and objective of the thesis is to devise a novel framework to overcome the issues in the cryptographic processes using Homomorphic Transform. The sub-objectives of the thesis as follows,

1. To design a novel key generation technique using Homomorphic Transform.
2. To create an end-to-end framework for data encryption and decryption.
3. To implement the framework and evaluate its robustness by comparing it with respect to the existing techniques.

The data preserving techniques claim to be secure enough to prevent any breach in its identity. During this research, the analysis of various existing algorithms for data security concluded the risk factors involved

in the process of data security. The analysis of identified algorithms for respective risk factors in data security will enable this research work to propose “A Novel Approach for Robust Data Security using Homomorphic Transformation”.

1.8 CONTRIBUTIONS

The thesis contribution towards the framework design and development of key generation process along with Encryption process. The thesis work is elaborated further with Framework Design, Algebraic Properties like, Cyclic, Dyadic and Graphical In-variance, Experimental setup, Experiment Results, Validation and Verification, and Comparative Analysis.

1.8.1 Proposed Framework

In this research work, an approach to generate different keys with each set of data which avoids unwanted access to the encrypted file is proposed. The proposed approach is compared to AES, the well-known approach to cryptography.

The proposed methodology is designed on adding and subtracting with a private key and a predetermined conversion. The secrets are all is kept safe by the secret key (private key). Without the correct key and transform, deciphering the encrypted message is extremely difficult. According to the obtained results, the output of the homomorphic transform is the similar for inverse, dyadic, and cyclic variance data sequences. To elude cipher attacks, the thing that is different is that the best approach will be distinct for each set of data sequences for each cryptosystem. Because of the private key, recovering the authentic communication systems is incredibly hard. Brute force attacks on secret key become more challenging as the key's size

increases. The proposed method employs the Homomorphic Transition. Secrecy is maintained at two levels by the secret key and the Homomorphic Transitions. Even when the mechanism is known, decrypting the final output of encrypted messages has proven to be more challenging. As or when required, the number of iterations can be increased.

1.9 THESIS ORGANIZATION

The overview of the presented thesis is discussed as follows:

Chapter one discussed the introduction of encryption algorithms, types of cryptanalysis, and various types of Homomorphic Transform, challenges, objective and scope of this work.

Chapter two outlines the survey of the existing methodologies related to the work presented in this thesis.

Chapter three discusses the various cryptanalysis techniques and comparison among various models.

Chapter four describes the general analysis and performance of various types of Homomorphic Transform in various applications.

Chapter five discusses the implementation of set theoretic Homomorphic Transform, and discusses the proposed algorithm by various encryption modules involved in it.

Chapter six discusses the comparative analysis, validation and verification for a 32-bit input, 64-bit input, 128-bit input, and 256-bit input data and discusses the secure text data transmission using Homomorphic Transform, are considered for the proposed experimentation.

Chapter seven concludes the thesis with the summary and outcome of the research work. Future scope of this research work that arises from the investigations carried out is also suggested.

Chapter 2

RELATED RESEARCH WORKS

2.1 INTRODUCTION

The classic era of cryptography is considered the age of kings, politicians, and devout personnel which laid the earlier stones of encryption. There are primarily two categories of classic ciphers: Substitution and Transposition, where the first classification substitutes another character, and the second classification alters the character positions to add bamboozlement. Substitution cryptographs are further divided into mono-alphabetic and poly-alphabetic codes depending upon the number of characters available for replacement. Obviously, the poly-alphabetic codes bring more choice and more robustness to the algorithm [4].

People employed pigeons for important message communication prior to the adoption of cryptography, but the usage of encryption began about 50 BC, as Roman emperor Julius Caesar utilized Caesar's Cipher for military communication. This was a simple substitution cipher, where each English letter was replaced by the 3rd letter to its right. e.g., A replaced with D and B with E, and so on. It was implemented on two co-centric disks or plates

with a letter inscribed on them. It was so effective that the rival generals and kings believed that the message was written in some foreign language that they could not understand. This novel concept took the crypto world by storm and myriad ciphers (both transposition and substitution based) were developed after it as per Li et al. [5].

A noteworthy reference here is the Rail-fence Cipher, which writes the data column-wise in multiple rows but reads the data row-wise to create the ciphertext. To add more bafflement, the number of rows can be increased [61]; however, rail-fence cipher of 2 or 3 rows was more popular. Another notable mention is the Vigenere Cipher that was considered to be unbreakable for many years but failed eventually due to short key length and repetition of characters in the plaintext. But this failure also gave birth to the dogma of OTPs which according to Claude Shannon can provide the ultimate security. Working of Vigenere and OTP cipher are displayed in Figure 2.1 and Figure 2.2 resp.

$$\begin{array}{r}
 \text{samplemessage} \\
 + \text{keykeykeykey} \\
 \hline
 = \text{cekzpcwiqceo}
 \end{array}$$

Figure 2.1: Working Example of Vigenere Cipher [4]

$$\begin{array}{r}
 \text{samplemessage} \\
 + \text{hqnyjiefsehpb} \\
 \hline
 = \text{zqznumqjkw hvf}
 \end{array}$$

Figure 2.2: Working Example of OTP [5]

2.1.1 Conventional Encryption

Zhao and Iwaihara [61] explained the classic age of cryptography was based on the natural languages used by humans and hence all the classic ciphers

were limited by the characters in the respective human language. This was a serious issue, and the repercussions were devastating. Furthermore, the usage of mechanical devices developed by the Germans especially Enigma hit the final blow and there was a necessity for better and cutting-edge cryptosystems that practice complex mathematical functions. The dominant dogma of conventional cryptography which surfaced in the latter half of the 20th century is that the secrecy lies with the key, given the cryptographic algorithm can be disclosed publicly.

Predominantly, the conventional encryption scheme can be classified into stream and block ciphers according to the way plaintext is processed. These encryption algorithms apply static transformations on a larger set of plaintext called blocks and hence the name derived. Block ciphers are comparatively faster because they operate on a big chunk of plaintext, but they may suffer from the predicament of producing the same ciphertext if the plaintext is repeated. Hence, advanced block cipher modes like Cipher Block Chaining Mode (CBC) and Cipher-Feedback Mode (CFB). were implemented by Ghrare et al. [6], to strengthen the security of information.

The other alternative - Stream Cipher encodes and decodes using a time variable alteration on either a single character or small group of characters. Obviously, this approach takes more time to complete but does not suffer from the problem of producing the same ciphertext for the same input data. The ciphers (stream) are further alienated into synchronous and asynchronous ciphers depending upon whether they consider previous output as feedback or not while producing output. Both methodologies have been quite popular and copious ciphers are proposed for both block and stream cryptograms. Some popular examples of block cipher are DES, 3DES, AES, RC5, and some renowned stream ciphers are RC4, Fibonacci SHrinking (FISH) discussed by Agbedemrab et al. in [62], A5/1, A5/2,

etc. Figure 2.3 shows how the working of stream ciphers differ from that of block ciphers.

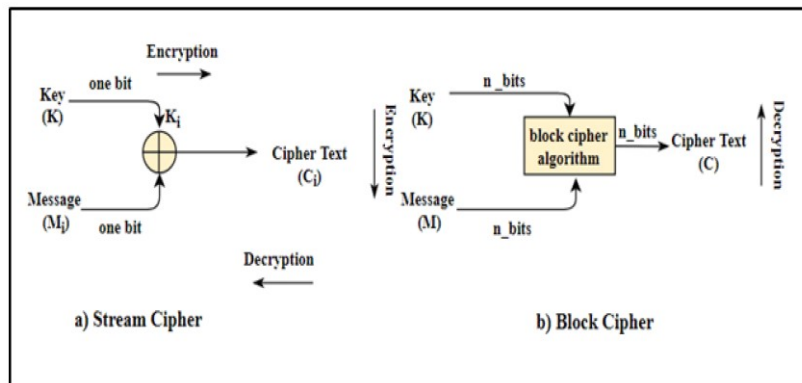


Figure 2.3: Illustration of Stream Cipher and Block Cipher [6]

Conventional cryptography is also called Symmetric Key Cryptography (SKC) presented by Liang et al. in [7] due to the fact that it applies the similar key for the encoding and decoding process (which are exactly opposite to each other) and this key is kept secret between the communicating parties to ensure confidentiality. It simply states that the secret key must be shared prior to the communication and not through the insecure channel as it will compromise the whole cryptosystem. Nevertheless, the messages can be relayed over the insecure channel and the encryption algorithm can also be made public. Figure 2.4 demonstrates the working of a general symmetric cryptogram.

A crucial feature of conventional ciphers is that they exhibit avalanche effect meaning a minor modification in the plaintext or secret key can cause immense changes in the produced ciphertext. Conventional cryptography offers numerous benefits like faster communication, robust conjoint authentication between communicating parties using passwords and above all, no adversary can decode the input text message without using a secret key, also highlighted by Ali in [63]. The downsides of this cryptosystem are mainly two: key distribution and key management. The first problem refers to

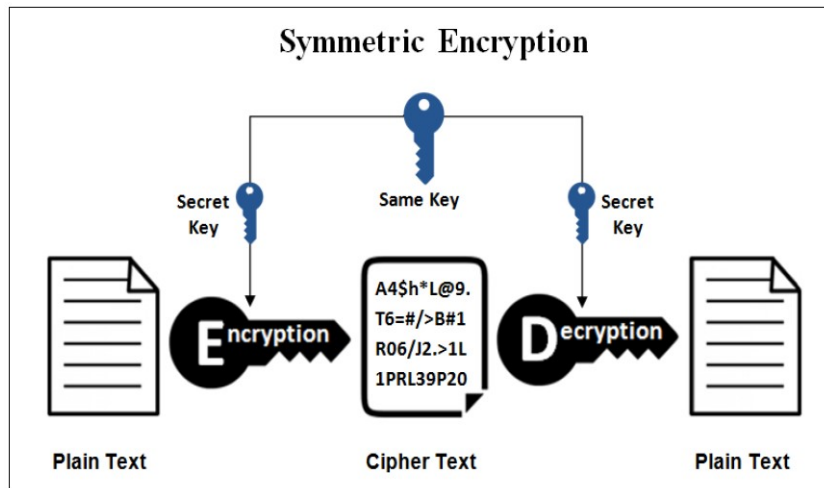


Figure 2.4: Working of Symmetric Cryptosystem [7]

the mode of sharing the shared key between users because unveiling the shared key will reveal every message sent. Hence, the key must be shared before communicate in person or through the trusted passage. The second issue is also prominent but ignored quite often; it refers to the problem of having one secret key between a pair of users. So, if a user wants to converse with another person, there is a requirement for an additional key for it. This problem increases exponentially with the growing number of users and communication between them. To handle all these quandaries, Public Key Cryptosystems were proposed, and these are discussed in the next section.

2.1.2 Public Key Encryption

Srikant et al. [8] discussed the core idea of Public Key Encryption (PKE) was proposed by Whitfield Diffie and he along with Martin Hellman published a research article titled "New Directions in Cryptography" in 1976 to eliminate the biggest issue in symmetric encryption, i.e., key exchange between communicating parties. They proposed a novel approach to exchange the secret key using complex modular arithmetic operations. The PKC algorithms believe in the idea of using two separate keys for encipher-

ing and deciphering the text, hence named asymmetric encryption. As the name advocates, the private key always keep secretly and the public key is disclosed to everyone by keeping it in a public catalogue or other accessible files available electronically.

This approach has manifold benefits: firstly, the user needs to generate and manage only a pair of keys, which also manages the key management problem. Secondly, all users have access to the public keys of any individual, hence the message can be transmitted securely to him by enciphering the message through the user's public key. Thirdly, the private key of the user need not be shared with anyone and can be used to either decrypt a received message or digitally sign an electronic document for non-repudiation purposes. Additionally, if at any point the user feels his private key is compromised, he can generate a new pair of keys and replace his old public key with the newly generated one. Thus, the bottom line is that PKC can be used for cryptography, digital signature as well as key exchange between trusted parties. An illustration of Public Key Cryptosystem is shown in Figure 2.5.

There has been some misconception that arrived after the development of PKC like asymmetric encryption is more secure than its symmetric counterpart and it will make secret-key cryptography obsolete. However, public-key encryption is not considered as a replacement for secret-key encryption, but both can be used in conjunction to increase the speed and security. Some of the notable asymmetric algorithms are RSA and El Gamal Encryption (named after its creator Tahir El Gamal), based on the computational complication of factoring hefty prime numbers and discrete logarithms respectively highlighted in [64] by Sadeghikhorami and Safavi. The only major downside of the asymmetric ciphers is their slower encryption and decryption speed than their symmetric equivalents. The

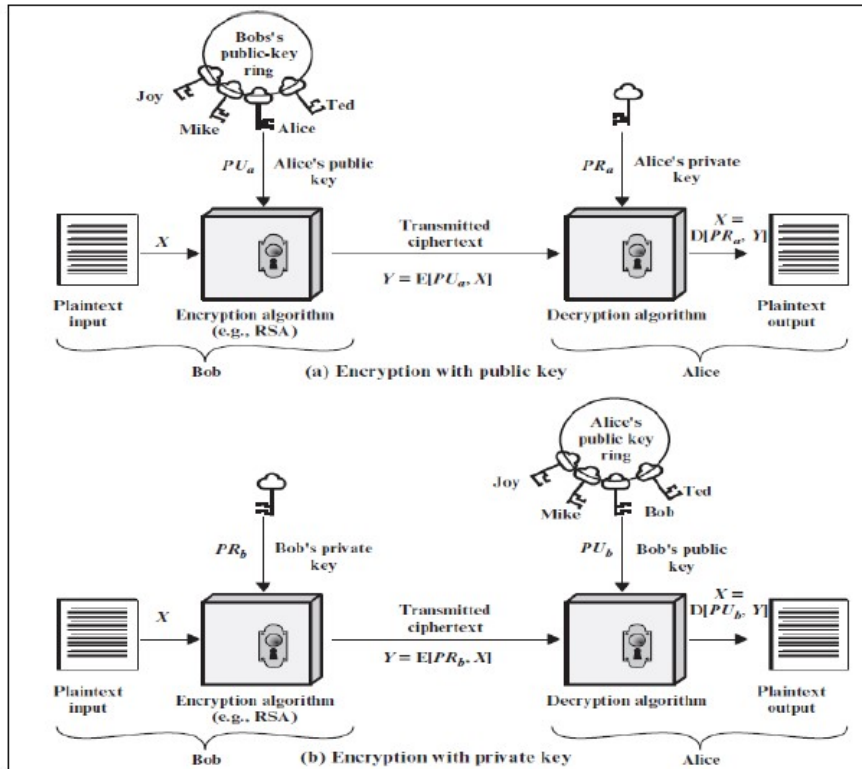


Figure 2.5: Public Key Encryption [8]

contributions of public-key cryptography are not limited to secure key exchange but are extended to key concepts like authentication using one-way hash functions, digital watermarking to safeguard the intellectual property rights, and the usage of digital signature to deliver the non-repudiation to prove the validity of the associated person.

2.1.3 Contemporary Encryption

Sobbahi [65] and Agbedem nab et al. [66] elaborate the evolution of public key cryptography paved the way for new research areas in the cryptology domain like Elliptical Curve Cryptography (ECC) and Quantum Cryptography. Elliptic curves are the prime or binary curves that are equally distributed over the x-axis and indigenous way to generate the public-private key pair using point addition and point doubling methods. The biggest plus is that the keys produced have a smaller length than asymmetric ciphers

but are equally strong. Due to this immense advantage, ECC is highly recommended for mobile applications as they need less computing power and lower depletion of battery pointed by Waleed et al. in [67]. Apart from this, Odeh et al. [68] in the elliptic curve ciphers provide faster encryption and decryption speed than asymmetric cryptograms like RSA. ECC uses addition operation instead of modular multiplication operation in RSA and to counter modular exponentiation, ECC uses multiple addition process. Cryptanalysis of elliptic cryptography shows comparable results with RSA i.e., time to break ECC and RSA is nearly the same, but if the short key length of ECC is taken into consideration, it makes ECC more powerful than RSA. One disadvantage of using elliptic curve cryptography can be the real number calculation, as in this case, ECC behaves slowly. The experiments have shown that ECC performs better at key exchange than at encoding information; nevertheless, ECC can provide faster and robust alternatives to RSA and even Diffie-Hellman Key Exchange.

Quantum cryptography is another variant of security algorithms used widely across the globe whether it is the encoded video calls or online elections in Switzerland presented by Prakash et al. in [69]. Quantum cryptography hails from quantum computing which uses qubits or photons instead of binary digits (bits) as their atomic unit of communication. The polarity of qubits is calculated according to their vertical or horizontal polarity. Also, quantum cryptology offers security at the physical layer rather than the higher layers using natural laws for encryption. The core idea behind quantum cryptography is the key distribution and its strength against the man-in-the-middle attack. According to Heisenberg's uncertainty norm, the qubits shift to a solo state intercepted in between; hence the recipient will come to know that the message has been reformed and should not be trusted by Kaur and Sodhi in [70]. Furthermore, it faces some serious

challenges due to higher error rates that need to be diminished and communication over longer distances. Currently, this technique is quite costly and challenging. However, it shows to be worthwhile to encrypt optical fibre network data used.

2.1.4 DNA Encryption

Kumar and Murti [71] explained that the DNA encryption is the latest branch of cryptology that promises ultimate information security by the conglomeration of encryption and biology. Primarily it takes advantage of the biological methodologies to safeguard the data with the help of laboratory experiments. Later on, the digital way of doing encryption also came into the play which smoothed things and also reduced the costs considerably. The core idea of DNA cryptography came from DNA computing's popularity to solve confound problems with ease. In 1994, Leonard Adleman (one of the co-inventors of RSA cipher) demonstrated how the Hamilton Path or Travelling Salesman Problem (TSP) explained in [72] by Sumartono et al., can be resolved using the biological approach. This pioneering research caused a whirlwind among the peers and aspiring research scholars to undertake DNA computing. They all understood that DNA computing can be a one-stop solution to unravel the NP-hard and NP-complete problems using the existing biological operations like Polymerase Chain Reaction (PCR) discussed in [73] Fan et al., DNA slicing and replication, etc. DNA computing to do the DES cryptanalysis to simplify things and the information is represented by DNA strands using multiple A, C, G & T letters in DNA computing instead of using 0s and 1s in electronic computing.

Later on, the DNA computing dogma was applied to the mystic world of encryption and the result was DNA cryptography. DNA principles can

be harnessed in the cryptography. The DNA can be altered in multiple ways to make it suitable to modify the information, hide bulks of information in the biological DNA, or use various biological processes to generate long and random DNA sequences that can be used as one-time pads or as homo-phonic substitution cipher. Apparently, there are myriad ways through which the traditional ciphers can be amalgamated with the DNA ciphers to produce a newer and stronger version. The fusion can produce unexpectedly amazing results in the areas of symmetric encryption, asymmetric encoding, and one-time pads. Steganography and Cryptography may also make use of DNA to store the partial or full results as it serves as an excellent storage material. In most cases, the plaintext is first converted to binary form, and then the DNA encryption is applied.

2.2 COMMON ENCRYPTION ALGORITHMS

Wang and Liu in [74] explained that the cryptographic algorithms shall be used to process data in a secured manner. A thorough security analysis is needed prior to the approval of these algorithms and they remain to be inspected to define that the algorithms provide necessary security. Cryptographic algorithms in general may be more intensified by the use of larger keys.

2.2.1 DES

The conventional encryption algorithms have some problems like data size, relationship among pixels before and after changing. The traditional cryptography techniques cannot be directly applied to image, audio & video because of pixel neighbourhood relationships and pixel redundancy. The Symmetric key encryption algorithms such as DES, Triple DES, AES, etc.

discussed by Silva in [9], can be applied directly on text but not on images, as images have high correlation among pixels and high redundancy and it is more time consuming to encrypt the image. The drawback with conventional encryption algorithms like DES, Triple DES, RSA are lack of addressing the features like bulk data size, correlation among pixels etc. Figure 2.6 shows the general structure of enciphering procedure of DES.

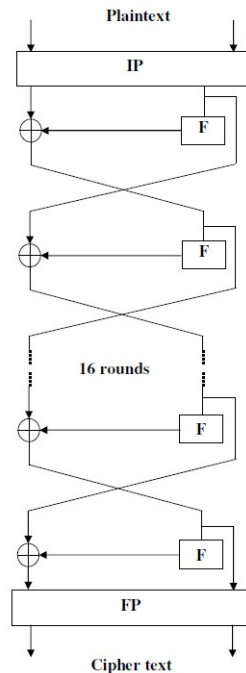


Figure 2.6: Enciphering procedure of DES [9]

2.2.2 Triple DES

Geethavani et al. [10] explained that though the Standard DES algorithm is the most popular security algorithm, it has been replaced by Triple DES, as it is easily hacked by the intruders. Triple DES was most widely used symmetric algorithm and also was suggested as a standard in the industry. Triple DES uses three distinct keys each with a key length of 56 bits as Figure 2.7 depicted.

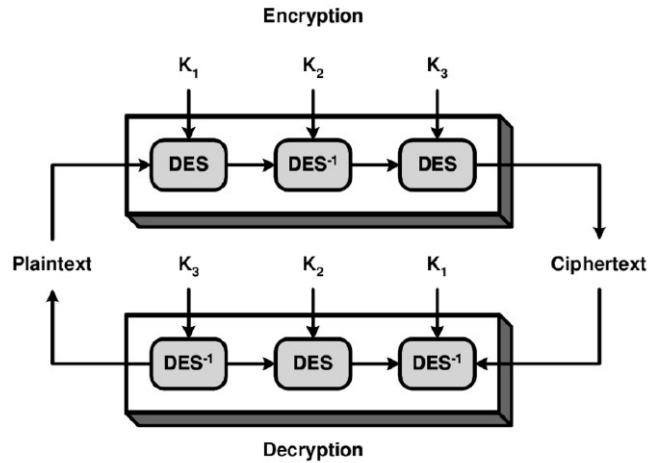


Figure 2.7: Triple DES Block Diagram [10]

2.2.3 RSA

RSA is a public-key encryption technology that is also a well-known source of information encryption. Because it employs a key pair, it is classified as an asymmetric algorithm. It is also one of the approaches used in the Gnu Privacy Guard (GPG) and Pretty Good Privacy (PGP) programs by Nagar and Alshamma in [11], as shown in figure 2.8.

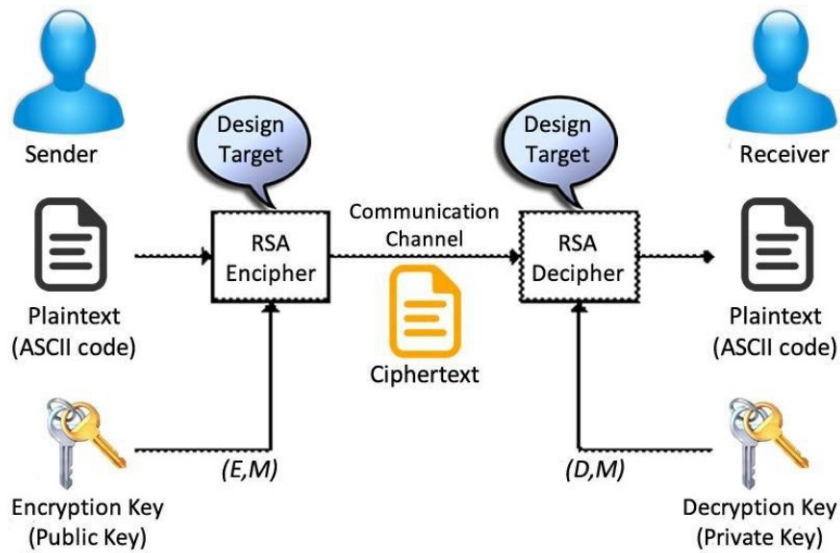


Figure 2.8: RSA Encryption Process [11]

2.2.4 AES

Cao and Fu [12] discussed that the AES algorithm is widely accepted as the industry benchmark by the US government and other agencies. For heavily loaded encryption, AES employs keys of 192 and 256 bits as shown in figure 2.9. Apart from brute force technique, which strives to decode input messages by applying all conceivable combinations in the encryption (128, 192, or 256-bit), it is widely regarded as impervious to all attacks. Nonetheless, security experts anticipate that AES will be recognized as the defacto method for securing information in the corporate sector.

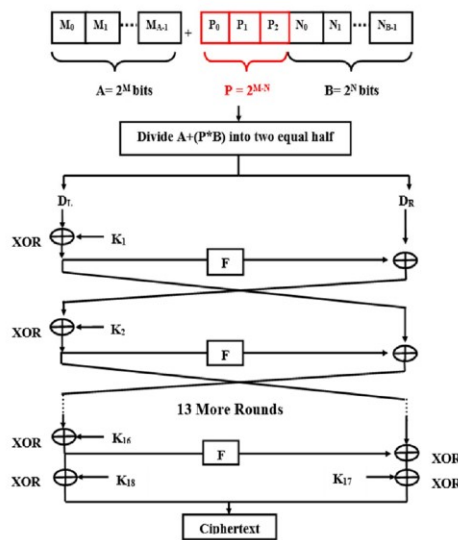


Figure 2.9: AES Encryption Process [12]

2.2.5 Blow Fish

Dongjiang et al. in [13] explained the Blowfish approach which is intended to substitute DES. This algorithm divides the messages into 64 bit blocks and each message is encrypted independently. This algorithm is known for its efficiency and rapidity. Blowfish found in different software industries, extending from e-commerce networks for banking transactions to password control systems for password protection. It is, without a doubt, one of the

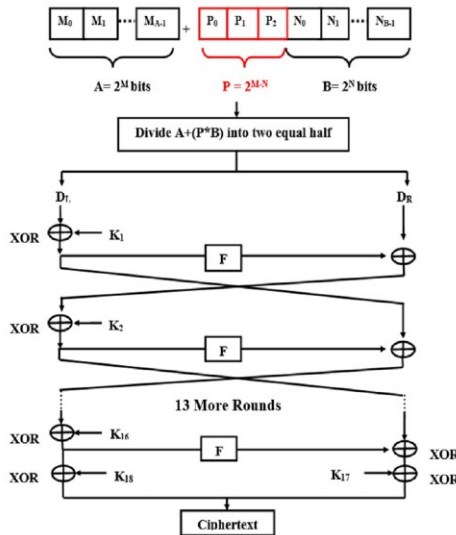


Figure 2.10: Blowfish/Twofish Procedure [13, 14]

more adaptable crypto algorithms accessible.

2.2.6 Twofish

Blowfish's replacement is Twofish by Si in [14]. This application's keys can be up to 256 bits long, and because it is a block cipher approach, a single private key is required for encryption as well as for decryption. Twofish is regarded as the quickest key (in this area) and it is suitable for the usage in software/hardware contexts. It is available for free to everyone who wishes to use it. It is commonly used in cryptographic algorithms such as PhotoEncrypt, GPG, and TrueCrypt. The procedure of Blowfish/Twofish is shown in Figure 2.10

2.3 CONCLUSION

This chapter presented the different encryption algorithms and their workflow with clear descriptions. In the next chapter, we consider the cryptanalysis and their types on text encryption.

Chapter 3

TEXT CRYPTANALYSIS

3.1 INTRODUCTION

With the remarkable improvement in the usage of the Internet over the past decade, the need for secure data storage and communication has significantly increased [75]. Vital information is continuously accessed and shared over the internet, thus demanding the need of cryptographic algorithms to fight security threats. While cryptography is an art of enciphering the information for ensuring security, cryptanalysis is an art of deciphering the ciphertext without the information of the secret key. Cryptanalysis is generally used to analyze the strength of the cryptographic algorithms or to find the secret information i.e., the key [76]. cryptographic systems are the encryption techniques used for encrypting the data. The "breaking the code" or cryptanalysis are the approaches that can be applied to decrypt the input text without intervention of any adopted encryption technique [77]. The cryptanalysts are the cryptanalysis practitioners. The process of enciphering and deciphering is possible with the help of Cryptographic Key and the mathematical logic behind these operations is acknowledged as the cryptographic algorithm or simply Cipher [78]. The whole process of encoding and decoding comes under the heading Cryp-

tosystem, which involves cryptography and cryptanalysis [79]. While the former deals with data encoding and decoding; the latter means the art of breaking the cryptographic algorithms. Cryptanalysis is the study of analyzing the secure communication channel and tries to open the hidden items in the message. Cryptanalyst is the one who is making the secure communication as an insecure communication.

3.1.1 Cryptanalysis

Cryptology is classified into two parts, as in Figure 3.1.

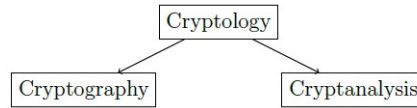


Figure 3.1: Classification of cryptology [15]

The first one, cryptography, deals with encoding the original message (or plaintext) to the hidden message (or ciphertext) called encryption and decoding the hidden message to the original message called decryption [80]. First, the plaintext is encrypted using public or private (secret) key, and then the ciphertext is decrypted by using a secret key. The ciphertext is transferred through a public channel. The latter, cryptanalysis, deals with analyzing the weakness and strength of the encryption algorithm without knowing the secret key. The challenge is to find the secret key with or without using the encryption (or decryption) oracle [81]. The leakage of any information about the plaintext or key is also associated with cryptanalysis. The ciphertext transferred in the public channel should look like a random sequence (called a pseudo-random sequence) of a secure cipher [82].

One way of attacking the encryption algorithm is by means of cryptanalysis, i.e., if an intruder get a secret message or data then he tries to break the code. Thus an intruder put the maximum effort to find the plain

text or original message from the secret message and this way of recovering plaintext from cipher text without knowing keyword is called Cryptanalysis [83,84].

As has been drawn in the universal definition of the term 'cryptanalysis' the exploration involves the defeating of techniques which ensure the secrecy or authenticity of information sent through a channel of communication. To elaborate, there are three levels of security architecture in information communication: classifying security attacks, designing the security mechanism and the obtainable security services [85]. The security attacks can be either active or passive; the security mechanism may incorporate encryption algorithms and the security services rope in realms such as authentication and access control. Cryptanalysis is the system which aims at cracking the security mechanism in the encryption process by directing its strike on the retrieval of the key rather than the repossession of the plain text of a single cipher text [86].

3.1.2 Need for Cryptanalysis

Cryptanalysis is defined as an art of science of interpreting the available cipher text. Cryptanalysis, also referred to as code-breaking, has the opposite objective of cryptography [87]. It is the study of information and analysing the hidden aspects of the system. It deciphers different types of codes. It is used to circumvent cryptography protective measures and obtain access to the relevant encryption methods even when the encryption algorithm is unknown [88]. The term cryptology comprises the two persons namely Cryptographer and Cryptanalyst. The aim of cryptographer is to develop strong cryptosystems for sending secret information over networks and the aim of cryptanalyst is to find out weakness of the cryptosystems by the way of breaking the cipher in less time [89]. Cryptanalytic attacks

are based on the algorithm's nature, essential aspects of simple text, and a sampling pair of simple text and block cipher. One of the main principles of cryptography is that the only secret which should need to be shared between two communicating parties is the secret key [90,91]. The goal of a cryptographic mechanism is to retrieve the key instead of the original message. As a result, cryptanalysis is essential to well before the encryption technique is deployed in order to determine its strengths and shortcomings. Linear and Differential cryptanalysis are thus the two most powerful and promising methodologies [92–95].

3.1.3 Attacks in Cryptanalysis

Cryptanalysis schemes differ depending upon the cryptanalyst's ability or nature of gaining access to the plaintext, ciphertext or other aspects of the cryptosystem. The most universal categories of attacks are catalogued as under [96–101],

- **Chosen plaintext cryptanalysis:** Attacking an encryption system with an unknown embedded key [102, 103].
- **Partial or truncated differential:** Exploiting the truncated differentials i.e. the differentials where only part of the difference can be predicted [104, 105].
- **Higher order differentials:** Expanding the differential characteristics to higher degrees to break ciphers susceptible to higher order attacks [106].
- **Miss in the middle attacks, impossible differentials:** Prediction of a differential that some particular differences should never occur [107].
- **Boomerang rectangle attacks:** Breaking constructions with highly

probable differential patterns propagating halfway through the cipher from the top as well as the bottom [108].

- **Related cipher attack:** Exposing the theoretical weaknesses in a cipher's key scheduling algorithm by comparing the keys in two different encryptions [109].
- **Related key attack:** Using relationships between keys which are different but related and in such attacks, cipher is fixed and the keys are fixed [110].
- **Weak key attack:** Attacking the weak keys which are defined as the keys which are very vulnerable to block cipher cryptanalysis [111].
- **Cube attack:** In this attack, a set of cube variables are chosen from the 4-bits such that the polynomial generated by adding the output function for all possible cube values (called the superpoly) is a simple polynomial (e.g., linear or quadratic) [112]. In the preprocessing phase of the cube attack, the linear (or quadratic) expressions of the superpolies of corresponding cubes are collected using the linearity test (or quadratic test) [113]. In the online phase, for a fixed key, the oracle is queried by setting the values of non-cube 4-variables as zero, and the values of the expressions are collected. Finally, a system of equations is solved, and the corresponding key bits are recovered. Also, another variant of cube attack, cube tester, deals with the properties of the superpoly to implement the distinguishing attack [114]. In conditional cube tester or conditional cube attack, conditions on some state bits are imposed [115].

3.1.4 Cryptanalysis Vs. Cipher Design

Cryptanalysis is an experimentation of the design of the cipher. In designing a cipher, the creator is pitted against the challenger in terms of

computational powers [116]. The establishment of the lower bounds of the computational security based on the complexity of attacks is definitely impractical [117]. At the same time, it can be stated with a certain measure of confidence that the upper limit on the complexity of breaking the cipher may possibly be established based on the nature of the cryptanalytic attack on the cipher [118]. As is the case, it is the foremost principal in any widely held design strategy that the cipher should be so modelled securely as to hold out against all notorious potent attacks [119].

Hence, it is the universal axiom that cryptanalytic techniques and the design of a cipher go in tandem. This does not pledge that the design approach does not absolutely offer warranty against the yet to be identified cryptanalytic attacks in future [120]. This is the best possible step that could be contemplated at the existing point in time. This stand is necessitated by the different perspective adopted by the modern researchers that cryptanalysis alone goes a long way in designing a secure cipher with inbuilt safety measures identified through the estimation of possible attempts that could be adopted by the hackers [121]. Cryptanalysis, has thus, become associated with positive credentials wiping out its bad connotation.

3.1.5 Types of Cryptanalysis on the text

Cryptanalysis approaches are used to break the encrypted message without intervention of any previously adopted key values. Following subsections are represented various attacks which discussed the similar behaviour.

3.1.5.1 Brute Force Attack

The cryptanalyst attempts every key on a chunk of the encrypted message until an understandable conversion into unencrypted is obtained [122]. To be successful, 50% of all potential keys must be attempted. This means

that the attacker will look for any and all possible inputs in required to persuade encrypted text to simple text [123].

An encryption system is inherently secured if the ciphertext created the scheme does not include information to uniquely determine the matching plaintext, regardless of the amount of ciphertext generated [124]. That is, no matter how much time an opponent has, decrypting the ciphertext is difficult since the necessary data is lacking. There is no cryptographic algorithm that is inherently strong, with the exception of a process developed as the OTP. The attacker has access to the cipher text only. Such attacks prove formidable as the ciphers are easily vulnerable to attacks.

3.1.5.2 Known Plain Attack (KPA)

The KPA [125] is a cryptanalysis attack paradigm in which the attacker obtains copies of both the text and its ciphertext and is free to utilize them to expose additional confidential messages about the key. It is much more effective than the plaintext approach, but less practical than the cipher text-only approach.

A cryptanalyst having access of both plaintext and ciphertext of several messages but its working behavior is to infer the key applied for message encryption or to design an approach for decryption for previously encrypted messages. [126].

For : $CT_1 = ET_k ey(X_1), CT_2 = ET_k ey(X_2) \dots CT_i = ET_k ey(X_i)$

Infer: Either *key*; or

an approach to deduce X_{i+1} , from $CT_{i+l} = ET_k ey(X_{i+l})$.

The attacker is aware of a slice of the corresponding plain text besides the cipher text. This could be assumed somewhat realistic since the attacker may perhaps guess the plain text from the context of communication.

3.1.5.3 Chosen Plain text only Attack

A chosen-plaintext attack is one in which the cryptanalyst may create his or her own text, send this into the cipher, then evaluate the resultant ciphertext. A cryptanalyst needed for chosen-plaintext attack to be able to feed information of his choosing into the encrypted mechanism, as well as examine the result from the machine [127].

The attacker possesses the skill to encrypt the plain text of his choice. This might be a probable situation with the attacker possessing the encryption system alone with the implanted key unidentified by him.

3.1.5.4 Jigsaw Puzzle Attack

This is the most practical case where no information regarding the plaintext and key is known. The cryptanalyst has only the cipher text and the encryption method employed and has no information regarding the contents or the key employed. Finding the cryptographic key is the most difficult with this method, but most of the time this is the only method that can be used. This technique is analogous to the earlier one but the difference is in the choice of the cipher text rather than the plain text.

3.1.5.5 Neighbor Attack

The cryptanalyst having several ciphertext messages, encrypted by applying same type of encryption technique. The job of cryptanalyst is to find out the plaintext messages as much as possible.

For given: $CT_1 = ET_k ey(X_1), CT_2 = ET_k ey(X_2) \dots CT_i = ET_k ey(X_i)$

Infer: Either $X_1, X_2, \dots X_i; key$; or

a technique to deduce X_{i+1} , from $CT_{i+l} = ET_k ey(X_{i+l})$.

In this kind of attack, it is assumed that the attacker is adept in adapting the plain text with his knowledge and experience with reference to

earlier encryptions. Such attacks are theoretically feasible although practically implausible.

3.1.6 Histogram Analysis

Generally, the attacks applies large amount of information and having capability to alter the ciphertext, without having a key value. Suppose the attackers having the complete details of the applied technique, he can get the plain text or the original image with only few attempts. In practice, it is factual and can be easily arrange for known chosen text to be processed (send text file that will be transmitted or sometimes showing automated response systems etc.). The informational distribution of the pixel values in an image can be seen in the histogram. The histogram of the cipher image needs to be consistent enough to withstand statistical attacks; otherwise, attackers could use the histogram of the encrypted image to extract meaningful information about the plain image.

3.1.7 Key Space Analysis

This analysis has two dimensions. Firstly the attacker uses one of the above kinds of attacks. Secondly, he is knowledgeable in the correlation between the keys in two separate encryption. These attacks lay bare the weaknesses in the key scheduling algorithm in the cipher.

3.2 OUTCOME OF CRYPTANALYSIS

It is perchance that the results or consequences of cryptanalysis could have variance in effectiveness. As an instance, the classification of attacks on block ciphers could be a point of contemplation. The categorization is developed on the discovery of the amount and quality of secret information.

- **Total break:** The hacker might figure out the security code.
- **Global deduction:** Although the actual key is not divulged, the hacker might find an essentially similar technique for encrypting and decrypting.
- **Instance (Local) deduction:** The hacker may find previously unknown plain messages (ciphertext).
- **Information deduction:** The hacker might obtain previously unknown acoustic data regarding plain messages or encrypted texts.
- **Distinguishing algorithm:** The hacker could identify the cipher apart from a randomized permutation.

Many, although not all, cyberattacks are becoming massively harder to complete as rounds are added to cryptanalysis, hence it is possible for the full cryptosystem to be strong which means the converse is true - the reduced round variants are weak. It is also a general statement that the partial breaks which are closer to breaking the original cryptosystem leads to the inevitable sequence of achieving a full break.

3.3 CHAPTER SUMMARY

In this chapter, the crptoanalysis and their text encryption algorithms were discussed. In the next chapter, we will present the analysis of homomorphic encryption techniques and simulated the results with the proposed method and other techniques considered.

Chapter 4

FRAMEWORK FOR DATA SECURITY USING HOMOMORPHIC TRANSFORM

The previous chapter discussed cryptanalysis with its needs and various attacks in it. The difference between Cryptanalysis Vs. Cipher Design was also highlighted. The text cryptanalysis with the need, attack, and its text encryption algorithms were discussed. Various types of text cryptanalysis, histogram, and key space analysis along with outcomes of text cryptanalysis were discussed in this chapter.

4.1 PROPOSED FRAMEWORK

The proposed framework is designed on the basis of homomorphic transformation. This section is classified into three subsection that are transformation techniques, algebraic theory behind proposed work, and proposed data security algorithm.

4.1.1 Hadmard Transform

It is the transform used to generate $2m$ genuine numbers S_k from $2m$ genuine numbers s_n . It can be characterized by applying twofold portrayal. We try to create a model (Figure 4.1) for Encryption process which involves the use of Hadamard transform along with the DNA Encryption for key generation process. Hadamard transform use an input sequence of $2m$ and covert them into a transformed sequence of the same length as the length of input sequence. Figure 4.2 shows the conversion of a input sequence $s(m) = (1, 1, 0, 1, 1, 0, 0, 1)$ into the transformed sequence $S(K) = (4, 0, 0, 0, -2, -2, 2, 2)$.

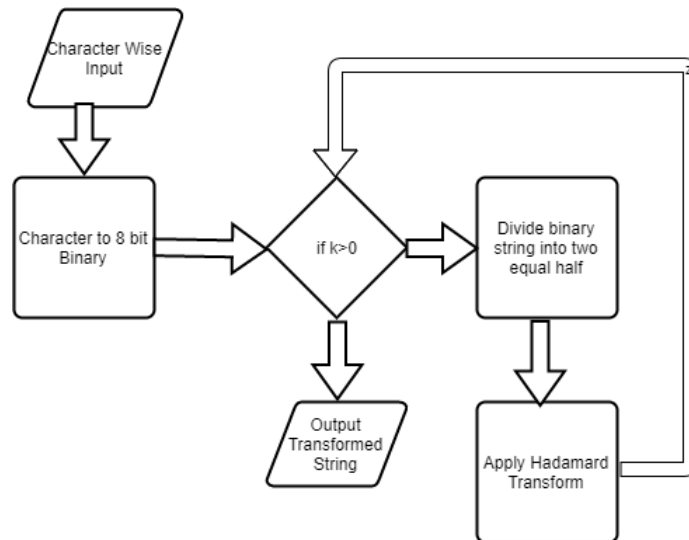


Figure 4.1: Encryption Model using Hadamard transform

This output sequence is further processed with DNA cryptography to make it a hybrid process. We tried to apply amino acid sequence by making a dictionary of alphabets, but it could not turn out as a feasible solution for the encryption process, so we finally discarded it.

4.1.2 Homomorphic Transform

It encrypts $A(t)$ which a sequence of numbers having length $S = 2^o$ where $o > 0$, to the sequence of ciphertext $a(o)$. In this process, it can encrypt any

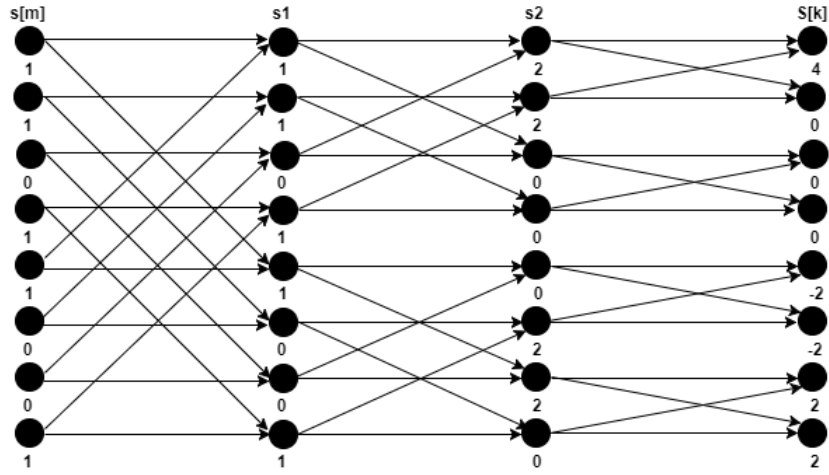


Figure 4.2: Hadamard Transform

sequence of numbers and also uses isomorphism. By applying three inverse properties (graphical, dyadic, and cyclic), makes this approach as homomorphic technique. The proposed homomorphic approach implemented these three algebraic number sequences $A(t)$ into the $a(o)$, however with dissimilar key $ET(x)$. In another side, for a set having dissimilar set of graphical, dyadic, and cyclic inverse sequences set gets encrypted into the similar cipher text sequence, but with a dissimilar cryptographic key value demonstrated the reason.

Let's having a sequence of t bits, $A(t)$, where t is in power of 2. If not considered, require to add a zero bits (temporary) after that it can easily partitioned into two identical half for further processing. Each half having similar $\frac{R}{2}$ points and also adequate the following equations:

$$p(n) = a(m) + a(m + (\frac{R}{2})); 0 \leq n \leq (\frac{R}{2}); 0 \leq m \leq (\frac{R}{2}) \quad (4.1)$$

$$q(n) = | a(m) - a(m - (\frac{R}{2})) |; 0 \leq n \leq (\frac{R}{2}); 0 \leq m \leq (\frac{R}{2}) \quad (4.2)$$

Until further division is unable to processed, each $\frac{R}{2}$ segment is partitioned

into $\frac{R}{4}$.

$$p1(o) = p(n) + p(n + (\frac{R}{4})); 0 \leq o \leq (\frac{R}{4}); 0 \leq n \leq (\frac{R}{4}) \quad (4.3)$$

$$p2(o) = | p(n) - p(n - (\frac{R}{4})) |; 0 \leq o \leq (\frac{R}{4}); 0 \leq n \leq (\frac{R}{4}) \quad (4.4)$$

$$q1(o) = q(n) + q(n + (\frac{R}{4})); 0 \leq o \leq (\frac{R}{4}); 0 \leq n \leq (\frac{R}{4}) \quad (4.5)$$

$$q2(o) = | q(m) - q(n - (\frac{R}{4})) |; 0 \leq o \leq (\frac{R}{4}); 0 \leq n \leq (\frac{R}{4}) \quad (4.6)$$

The above process will continuously followed till further partitioned can be done. It can further divide and the total number of states can calculated as $Log_2(R)$.

The Figure 4.3 depicts the implementation of the forward homomorphic transform in the form of a signal. Firstly, the input values are eight which can further partitioned into two halves. For the 1st iteration, the 1st value of each half is added, and if the resultant sign is positive then key value is 0. For the 2nd half, 1st values of each half is subtracted, and if the resultant sign is negative, then key value is 1. In the 2nd iteration, the division process will be followed similarly till the further division not possible. .

By using the same approach, as shown in figure 4.4, the further processes i.e. inverse function implemented. The results are extracted by applying subtraction and addition of mod values and these results are dependent on the key values (corresponding values). e.g., if the value is 0, then 1st apply subtraction operation then addition and if the value is 1, then apply 1st addition operation then subtraction.

4.1.3 Algebraic properties of the proposed Method

In this work, we formulate and describe various algebraic properties of the proposed method. The proposed method follows some algebraic properties,

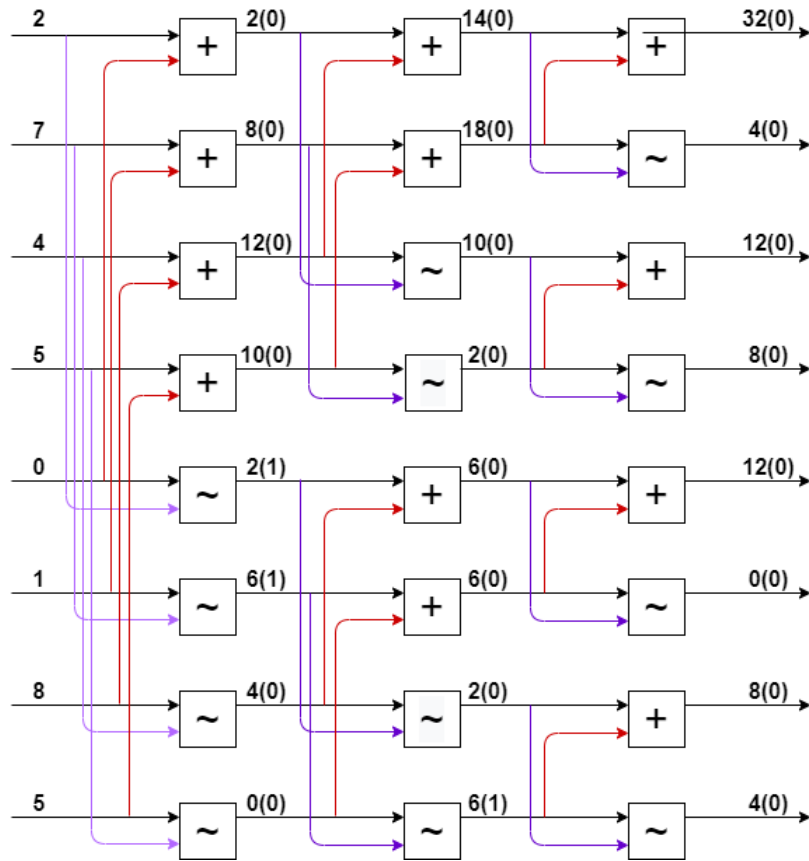


Figure 4.3: Homomorphic Transform in Signal Processing

which presents how it generate separate key for each set of input values, while output for each values will always be the same. Next section explains the algebraic properties:

Cyclic Shift In-variance

The input sequence $a(r) = 2, 7, 4, 5, 0, 1, 8, 5$ is considered to design the following cipher text may produce sequences of 7 cyclic shifted: $ac1(r) = 5, 2, 7, 4, 5, 0, 1, 8$

$$ac2(r) = 8, 5, 2, 7, 4, 5, 0, 1$$

$$ac3(r) = 1, 8, 5, 2, 7, 4, 5, 0$$

$$ac4(r) = 0, 1, 8, 5, 2, 7, 4, 5$$

$$ac5(r) = 5, 0, 1, 8, 5, 2, 7, 4$$

$$ac6(r) = 4, 5, 0, 1, 8, 5, 2, 7$$

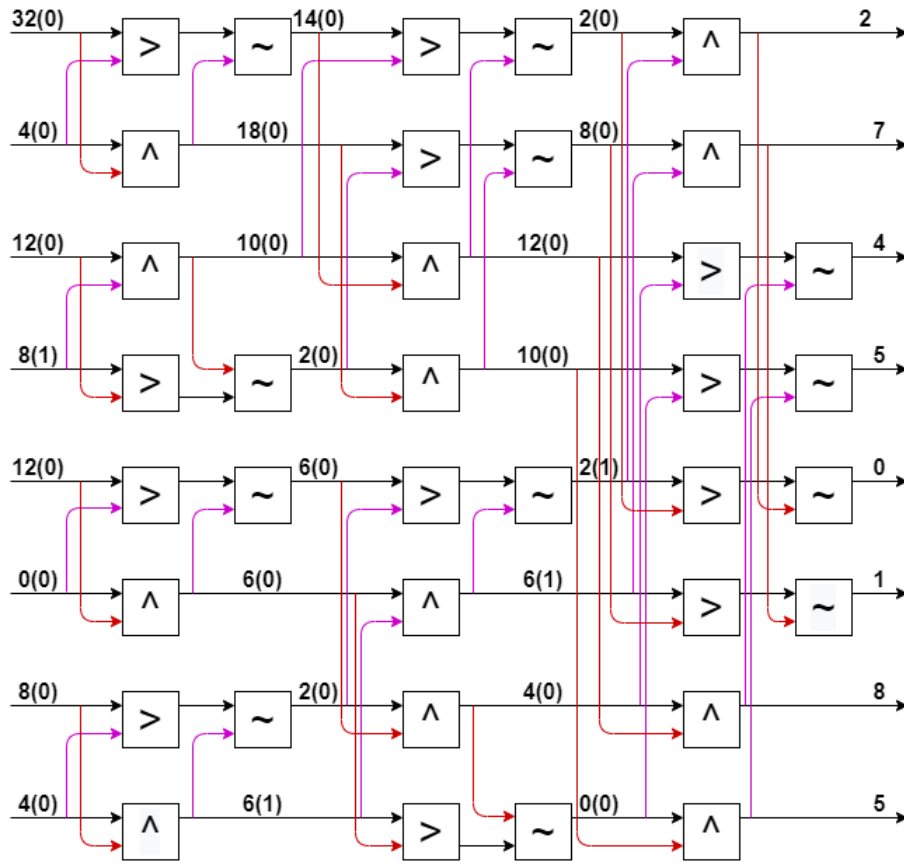


Figure 4.4: Homomorphic Transform in Signal Processing (Inverse)

$$ac7(r) = 7, 4, 5, 0, 1, 8, 5, 2$$

Interestingly, after applying the encryption process, the newly produced 7 sequences of cyclic shifted into cipher text, the calculated output will be $A(o) = 32, 4, 12, 8, 12, 0, 8, 4$, which can be same as in given input sequence, but an omission of $E(y)$ (encryption key).

Table 4.1: Encrypted output of the given number sequence

$x(n)$	$a(n)$	$E(1)$	$b(n)$	$E(2)$	$X(n)$	$E(3)$
2	2	0	14	0	32	0
7	8	0	18	0	4	0
4	12	0	10	0	12	0
5	10	0	2	0	8	1
0	2	1	6	0	12	0
1	6	1	6	0	0	0
8	4	0	2	0	8	0
5	0	0	6	1	4	0

Table 4.2: Encrypted output of the cyclic shifted number sequence

x(n)	a(n)	E(1)	b(n)	E(2)	X(n)	E(3)
7	8	0	18	0	32	0
4	12	0	14	0	4	1
5	10	0	2	0	12	0
0	2	0	10	1	8	0
1	6	1	6	0	12	0
8	4	0	6	0	0	0
5	0	0	6	1	8	0
2	2	0	2	1	4	1

Graphical In-variance

As in inverse function, similar approach can be applied to the graphical inverse function. The input sequence in graphical inverse is $a(r) = 2, 7, 4, 5, 0, 1, 8, 5$. For graphical inverse sequence, the extracted cyclic shift version is also true and a normal user can easily calculate the cipher text of $A(o) = 32, 4, 12, 8, 12, 0, 8, 4$ as the input sequence, however with the distinct key $E(y)$ (encrypted). Table 4.3 and 4.4 shows the graphical inverse number sequence. $a_{c1}^{-1}(r) = 2, 5, 8, 1, 0, 5, 4, 7$

$$a_{c2}^{-1}(r) = 7, 2, 5, 8, 1, 0, 5, 4$$

$$a_{c3}^{-1}(r) = 4, 7, 2, 5, 8, 1, 0, 5$$

$$a_{c4}^{-1}(r) = 5, 4, 7, 2, 5, 8, 1, 0$$

$$a_{c5}^{-1}(r) = 0, 5, 4, 7, 2, 5, 8, 1$$

$$a_{c6}^{-1}(r) = 1, 0, 5, 4, 7, 2, 5, 8$$

$$a_{c7}^{-1}(r) = 8, 1, 0, 5, 4, 7, 2, 5$$

Dyadic Shift In-variance

This shifting defines the chances formation of in-grouping of two dyadic that simply means a group of two sequences. Assume that the similar sequence $a(r) = 12, 17, 14, 15, 10, 11, 18, 15$. In this two ways can be seen to extracted the dyadic shift from the input sequence. Following paragraphs

Table 4.3: Encrypted output of the graphical inverse number sequence(P-1)

x(n)	a(n)	E(1)	b(n)	E(2)	X(n)	E(3)
5	10	0	18	0	32	0
8	12	0	14	0	4	1
1	8	0	2	1	12	0
0	2	0	10	1	8	0
5	0	0	6	0	12	0
4	4	1	6	0	0	0
7	6	0	6	0	8	0
2	2	0	2	1	4	1

Table 4.4: Encrypted output of the graphical inverse number sequence (P-2)

x(n)	a(n)	E(1)	b(n)	E(2)	X(n)	E(3)
8	12	0	14	0	32	0
1	8	0	18	0	4	0
0	2	0	10	1	12	0
5	10	0	2	0	8	1
4	4	1	6	0	12	0
7	6	0	6	0	0	0
2	2	0	2	1	8	0
5	0	0	6	1	4	0

presented it one by one. The input sequence partitioned into 2 same halves with block size:

$$S_d^2[a(r)] = 7, 2, 5, 4, 1, 0, 5, 8.$$

$$S_d^4[S_d^8[a(r)]] = 4, 5, 2, 7, 8, 5, 0, 1$$

$$S_d^8[S_d^4[S_d^8[a(r)]]] = 5, 8, 1, 0, 5, 4, 7, 2$$

And, if someone applies encryption of dyadic sequences and takes the same ciphertext $A(o) = 32, 4, 12, 8, 12, 0, 8, 4$ but with distinct key $E(y)$.

Table 4.5: Encryption output of the dyadic shifted number sequence (part 1)

x(n)	a(n)	E(1)	b(n)	E(2)	X(n)	E(3)
0	2	0	14	0	32	0
1	8	0	18	0	4	0
8	12	0	10	0	12	0
5	10	0	2	0	8	1
2	2	0	6	0	12	0
7	6	0	6	0	0	0
4	4	1	2	0	8	0
5	0	0	6	1	4	0

Alternate way to get the dyadic shift of input sequence $a(r) = 2, 7, 4, 5, 0, 1, 8, 5$ is to get,

$$S_d^8[a(r)] = 0, 1, 8, 5, 2, 7, 4, 5.$$

$$S_d^4[S_d^2[a(r)]] = 8, 5, 0, 1, 4, 5, 2, 7.$$

$$S_d^2[S_d^4[S_d^2[a(r)]]] = 5, 8, 1, 0, 5, 4, 7, 2.$$

Remarkably, $S_d^2[S_d^4[S_d^8[a(r)]]] = S_d^8[S_d^4[S_d^2[a(r)]]]$ and it become true for all input sequences. A cryptanalyst can easily calculate the cyclic shift and graphical inverse of these input values. The calculated ciphertext will always be $A(o) = 32\ 4, 12, 8, 12, 0, 8, 4$ which is similar to the input sequence, however with the key $E(y)$ (encrypted).

4.2 RESULTS

After applying the encryption and decryption in the plain text file and vice-versa, a few test cases were designed to check the performance of the proposed approach. All designed cases include digits and alphabets and depicted a pattern of 8×8 matrix, as mentioned in Table 4.6 and Table 4.6. After successful conversion of a text file and vice-versa, it has been observed that there is no error encountered during the transformation process.

Table 4.6: Matrix Representation of Numeric Digits

00000000	00000000	00000000	00000000
00011000	00001000	00111100	01111100
00100100	00001000	01000010	00000010
00100100	00001000	00000010	00011100
00100100	00001000	00111100	00000010
00100100	00001000	01000000	00000010
00011000	00011100	01111110	01111100
00000000	00000000	00000000	00000000
0	1	2	3
00000000	00000000	00000000	00000000
00000100	01111110	00111110	01111110
00001100	01000000	01000000	00000010
00010100	01111100	01000000	00000100
00100100	00000010	01111100	00001000
01111110	00000010	01000010	00010000
00000100	01111100	00111100	00100000
00000000	00000000	00000000	00000000
4	5	6	7
00000000	00000000		
00111100	00111100		
01000010	01000010		
00111100	00111100		
01000010	00000010		
01000010	00000010		
00111100	00111100		
00000000	00000000		
8	9		

To analyze the performance of the proposed work, the letter [A] can be represented in matrix form and its encrypted text matrix is demonstrated in Table 4.8.

Table 4.7: Matrix Representation of Alphabets

00000000	00000000	00000000	00000000
00011000	00111100	00011100	00111100
00100100	00100010	00100000	00100010
00100100	00111100	00100000	00100010
00111100	00100010	00100000	00100010
00100100	00100010	00100000	00100010
00100100	00111100	00011100	00111100
00000000	00000000	00000000	00000000
A	B	C	D
00000000	00000000	00000000	00000000
00111110	00111110	00111100	01000010
00100000	00100000	01000010	01000010
00111100	00111100	01000000	01111110
00100000	00100000	01001110	01000010
00100000	00100000	01000010	01000010
00111110	00100000	00111110	00000000
00000000	00000000	00000000	00000000
E	F	G	H
00000000	00000000	00000000	00000000
00011100	00001100	00100010	00100000
00001000	00000100	00100100	00100000
00001000	00000100	00101000	00100000
00001000	01000100	00110000	00100000
00001000	01000100	00101000	00100000
00011100	00111000	00100100	00111110
00000000	00000000	00100010	00000000
I	J	K	L
00000000	00000000	00000000	00000000
01000010	01000010	00111100	00111100
01100110	01100010	01000010	01000010
01011010	01010010	01000010	01111100
01000010	01001010	01000010	01000000
01000010	01000110	01000010	01000000
01000010	01000010	00111100	01000000
00000000	00000000	00000000	00000000
M	N	O	P
00000000	00000000	00000000	00000000
00111100	01111000	00111100	01111100
01000010	01000100	01000010	00010000
01000010	01000100	01000000	00010000
01000010	01111000	00111100	00010000
00111100	01000100	00000010	00010000
00001000	01000010	01000010	00010000
00000100	00000000	00111100	00000000
Q	R	S	T
00000000	01000001	10000010	0 0000000
01000010	01000001	10010010	1 0000010
01000010	01000001	10101010	01000100
01000010	00100010	11000110	00101000
01000010	00010100	10000010	00010000
01000010	00001000	00000000	00010000
00111100	0 0000000	00000000	00010000
00000000	00000000	01000010	00000000
00000000	1 0000010	00100100	
U	V	W	X
00000000	00000000		
01111110	01111110		
00000100	00000100		
00001000	00001000		
00010000	00010000		
00100000	00100000		
01111110	01111110		
00000000	00000000		
Y	Z		

Table 4.8: Encrypted Matrix Representation of Alphabet [A] and Cipher Sequences

00000000	1400060014000600
00011000	0200020002000200
00100100	0600020006000200
00100100	0200020002000200
00111100	1000020010000200
00100100	0200020002000200
00100100	0600020006000200
00000000	0200020002000200
[A]	Cipher(A)

As mentioned in the above table, the character [A] represented an encrypted form in an 8×8 matrix representation form. It can also be observed that character [A] can be denoted by six different forms of matrix representation. After analysis of six encrypted and decrypted values (RT and IRT) of character [A], few errors have been over-served, as mentioned in Table 4.9.

4.3 CHAPTER SUMMARY

The present chapter presented two transformation techniques that are Hadamard and homomorphic transformation. The proposed work applied homomorphic transform only due to its quick responsive nature. The present chapter also discussed the algebraic properties of homomorphic transformation which is further classified into cyclic shift-invariance, graphical invariance, and dyadic shift-invariance. The result section showed the evaluation of the present work by designing few test cases. The present chapter represented the 8×8 matrix representation for digits (0-9) and alphabets (A-Z) along with error observations of [A] character for various orientations. The next chapter discusses the set-theoretic approach for the proposed transformation with their results by applying encryption and decryption on plain text.

Table 4.9: Error Observation of character [A] for Various Orientations

Input	Encryption	Error	Remarks
00000000	14 00 06 00 14 00 06 00	00000000	Orientation: Normal
00011000	02 00 02 00 02 00 02 00	00000000	Scanning: Row-Column
00100100	06 00 02 00 06 00 02 00	00000000	Error in Encryption=NIL
00100100	02 00 02 00 02 00 02 00	00000000	Efficiency=100%
00111100	10 00 02 00 10 00 02 00	00000000	
00100100	02 00 02 00 02 00 02 00	00000000	
00100100	06 00 02 00 06 00 02 00	00000000	
00000000	02 00 02 00 02 00 02 00	00000000	
00000000	14 00 06 00 14 00 06 00	00000000	Orientation: Inverted
00100100	02 00 02 00 02 00 02 00	00000000	Scanning: Row-Column
00100100	06 00 02 00 06 00 02 00	00000000	Error in Encryption=NIL
00111100	02 00 02 00 02 00 02 00	00000000	Efficiency=100%
00100100	10 00 02 00 10 00 02 00	00000000	
00100100	02 00 02 00 02 00 02 00	00000000	
00011000	06 00 02 00 06 00 02 00	00000000	
00000000	02 00 02 00 02 00 02 00	00000000	
00001100	14 00 06 00 14 00 06 00	00000000	Orientation: Translated
00010010	02 00 02 00 02 00 02 00	00000000	Scanning: Row-Column
00010010	06 00 02 00 06 00 02 00	00000000	Error in Encryption=NIL
00011110	02 00 02 00 02 00 02 00	00000000	Efficiency=100%
00010010	10 00 02 00 10 00 02 00	00000000	
00010010	02 00 02 00 02 00 02 00	00000000	
00000000	06 00 02 00 06 00 02 00	00000000	
00000000	02 00 02 00 02 00 02 00	00000000	
00000000	14 02 02 02 16 02 06 02	02428202	Orientation: 90° Rotated
00000000	00 00 00 00 00 00 00 00	20202020	Scanning: Row-Column
01111100	10 02 04 02 02 02 02 02	42424202	Error in Encryption=NIL
00010010	00 00 00 00 00 00 00 00	20202020	Efficiency=71%
00010010	14 02 02 02 06 02 06 02	42024242	
01111100	00 00 00 00 00 00 00 00	20202020	
00000000	10 02 06 02 02 02 02 02	42424202	
00000000	00 00 00 00 00 00 00 00	20202020	
00000000	14 02 02 02 16 02 06 02	02428202	Orientation: 270° Rotated
00000000	00 00 00 00 00 00 00 00	20202020	Scanning: Row-Column
01111100	10 02 04 02 02 02 02 02	42424202	Error in Encryption=NIL
01001000	00 00 00 00 00 00 00 00	20202020	Efficiency=71%
01001000	14 02 02 02 06 02 06 02	42024242	
01111100	00 00 00 00 00 00 00 00	20202020	
00000000	10 02 06 02 02 02 02 02	42424202	
00000000	00 00 00 00 00 00 00 00	20202020	

Chapter 5

DATA SECURITY MODEL USING SET THEORETIC HOMOMORPHIC TRANSFORM

The approaches concerned with data security refer to the practices for the protection of data/information from unauthenticated access, theft, or corruption. Various existing techniques are successful to achieve a certain level while maintaining data security. The previous chapter presented the pros & cons of Hadamard transformation and also discussed the proposed framework for data security using homomorphic transformation along with its algebraic properties. The present chapter discusses the working model of the proposed framework with the set-theoretic approach for homomorphic transformation. The cyclic, graphical, and dyadic shift-in-variance properties of the proposed approach are also highlighted.

5.1 PROPOSED WORKING MODEL

The proposed working model is basically designed on the working behavior of homomorphic transformation. This section is classified into two subsections, i.e. the set-theoretic approach of homomorphic transformation and the proposed data security algorithm based on homomorphic transformation.

5.1.1 Set-Theoretic Approach of Homomorphic Transform

An encryption and decryption approach is proposed using a homomorphic transform. The process is divided into three parts; encryption, key generation, and decryption. The encryption process takes a set values as input $A(r)$ defining the length R . R is always defined in the power of two. For example, the sequence (1,2,3), (2,3,4), (1,4,5), (2,4,6) which has 4 different sets $A(1)$, $A(2)$, $A(3)$ and $A(4)$ is shown as follows:

$$A(1) = (1,2,3)$$

$$A(2) = (2,3,4)$$

$$A(3) = (1,4,5)$$

$$A(4) = (2,4,6)$$

The above input $A(1)$, $A(2)$, $A(3)$, and $A(4)$ is processed as per the proposed algorithm and is explained in figure 5.1.

As shown in figure 5.1, the length R is four, which is in the power of two. Any such sequence is divided into two halves, each having $\frac{R}{2}$ values so that:

$$P(n) = A(m) \cup A(m + (\frac{R}{2})); \quad (5.1)$$

$$Q(n) = A(m) \Delta A(m - (\frac{R}{2})); \quad (5.2)$$

where $0 \leq n \leq (\frac{R}{2})$ and $0 \leq m \leq (\frac{R}{2})$. Each $(\frac{R}{2})$ set sequence is further

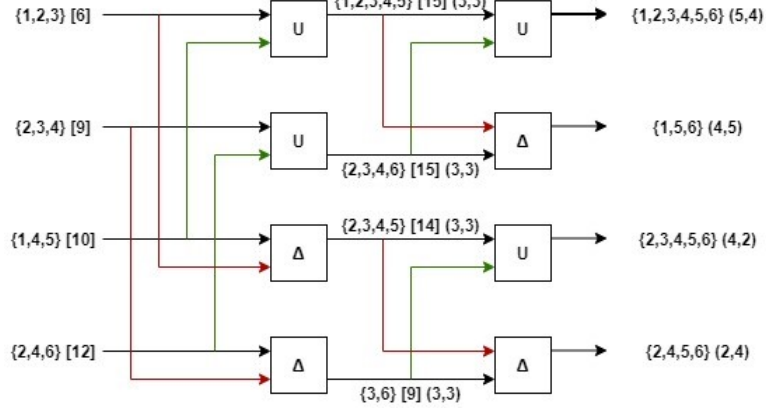


Figure 5.1: Encryption of four sets of values

divided into two halves, each having $(\frac{R}{4})$ set of values, and satisfy the following conditions:

$$P1(k) = A(n) \cup A(n + (\frac{R}{4})); \quad (5.3)$$

$$P2(k) = A(n) \Delta A(n - (\frac{R}{4})); \quad (5.4)$$

$$Q1(k) = A(n) \cup A(n + (\frac{R}{4})); \quad (5.5)$$

$$Q2(k) = A(n) \Delta A(n - (\frac{R}{4})); \quad (5.6)$$

where $0 \leq k \leq (\frac{R}{2})$ and $0 \leq n \leq (\frac{R}{2})$. The process is repeated until no further division is possible. It is discovered that there are $\log_2 R$ stages. The results of the encryption process are given to the decryption process, together with the key produced. During the encryption, some weight is assigned to the values of each set along with the cardinality of each set of values. Union operation is executed on the first two sets of halves where the cardinality of the first set is to be written first and the cardinality of the second set later. Figure 5.2 shows the decryption process of four sets of values, which shows the round-wise conversion of encrypted data into normal data.

However, during the symmetric difference operation, the cardinality of

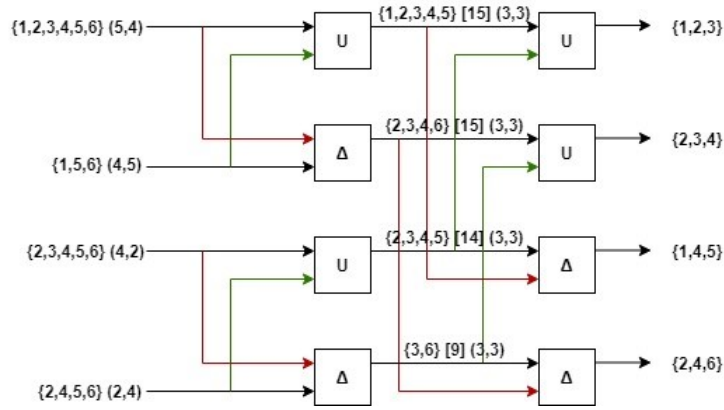


Figure 5.2: Decryption process of four sets of values

the second set is considered first and the cardinality of the first set later. Thus, the encryption process will provide the three sets of values; the decrypted values, set of weight, and set of cardinality. The combination of weight and cardinality is the key for the encryption process, which is unique for each set of values.

For the decryption, the cardinality defines the number of elements in each set. To get the original values the symmetric difference of both the sets is to be calculated, write the element of each set considering the weight of each set and distribute the remaining elements accordingly.

5.1.2 Proposed Data Security Algorithm

5.2 CYCLIC SHIFT-INVARIANCE

Let's assume a set of following sequence, separated by comma

$$X(n) = \{1,2\}\{3,4,6\}\{4,5\}\{1,5\}\{1,4,5\}\{3,4,5\}\{2,6\}\{1,4,6\}$$

The following seven cycle shift variations can be produced using this set sequence:

It is clear that $X(n)$ itself is $X_{c2}(n)$, the cycle shift variant. It is simple to confirm that each of these eight sequences has the same $Y(k)$ and that $Y(K)$ sequence is mentioned below $Y(K) = \{1,2,3,4,5,6\}, \{2,3\}, \{1,3,6\}, \{3,6\},$

Algorithm 1 Crypt

```
1: Input: A list
2: Output: Generating the tuples of the union and symmetric difference
3: Goal: Performing the union and symmetric difference operations
4: procedure Crypt(list)
5: if len(list) = 1 then
6:     return list
7: else
8:      $k = \text{len}(\text{list})$ 
9: end if
10: temp=list
11: half=  $\frac{k}{2}$ 
12: un=[]
13: sy=[]
14: for  $i \leftarrow 0$  to half do
15:     un.append(union(list[i],list[half + i]))
16: end for
17: for  $i \leftarrow 0$  to half do
18:     sy.append(diff(temp[i],temp[half + i]))
19: end for
20: return un,sy
21: end procedure
```

Algorithm 2 Union

```
1: Input: Tuple1 and Tuple2
2: Output: Union of the two tuples
3: Goal:Performing the combined operation of two tuples
4: procedure union (tuple1, tuple2)
5: res = set(tuple1 + tuple2)
6: return res
7: end procedure
```

Algorithm 3 Difference

```
1: Input: Tuple1 and Tuple2
2: Output: Symmetric difference between tuple1 and tuple2
3: Goal: Computing the symmetric difference
4: procedure diff(tuple1, tuple2)
5: res_1=set (tuple1).symmetric_difference (set(tuple 2))
6: return res_1
7: end procedure
```

$\{2,4,5,6\}, \{2\}, \{4,6\}, \{4,6\}$, as depicted in figure 5.3.

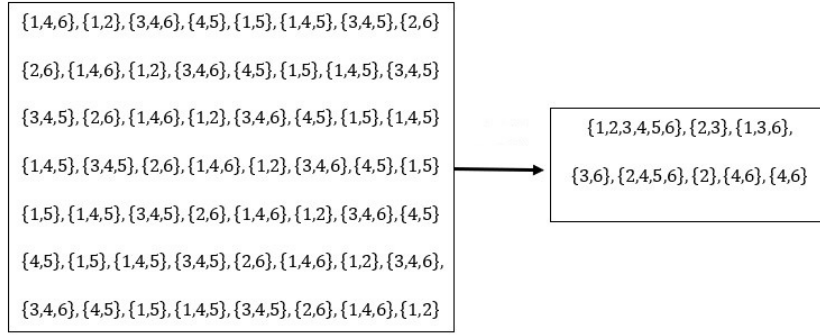


Figure 5.3: Cyclic Shift-In-variance

5.3 GRAPHICAL INVARIANCE

As similar in cyclic shift-invariance, in the graphical invariance, the following sets are separated by comma,

$X(n) = \{1,2\}\{3,4,6\}\{4,5\}\{1,5\}\{1,4,5\}\{3,4,5\}\{2,6\}\{1,4,6\}$ has eight additional graphical variants of this predetermined order, including,

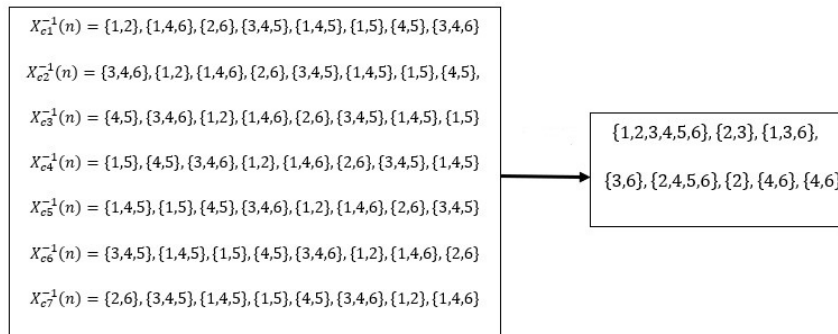


Figure 5.4: Graphical In-variance

It is clear that $X_{c7}^{-1}(n)$ itself has a pictorial representation. The identical $X^{-1}(n)$ in each of these eight sequences and the fact that the $Y(k)$ sequence is listed below the $Y(k)$ sequence may be simply proven.

$\{1,2,3,4,5,6\}, \{2,3\}, \{1,3,6\}, \{3,6\}, \{2,4,5,6\}, \{2\}, \{4,6\}, \{4,6\}$ as depicted in figure 5.4.

5.4 DYADIC SHIFT-INVARIANCE

Dyadic shift is the process of transposing two blocks of elements in a sequence, and the name "dyadic" refers to a group of two. e.g., lets us take $X(n) = \{1,2\},\{1,2\},\{3,4,6\},\{4,5\},\{1,5\},\{1,4,5\},\{3,4,5\},\{2,6\},\{1,4,6\}$ and transpose its 1st half with the 2nd half. The resulting sequence $Td^2[X(n)] = \{1,4,5\},\{3,4,5\},\{2,6\},\{1,4,6\},\{1,2\},\{3,4,6\},\{4,5\},\{1,5\}$ is $X(n)$ with a 2-block shift. The two-block dyadic shift operator is denoted by the letter Td^2 . In a similar way, we may get

$$Td^4[Td^2] = \{1,2\},\{3,4,6\},\{4,5\},\{1,5\},\{1,4,5\},\{3,4,5\},\{2,6\},\{1,4,6\} \text{ and}$$

$$Td^8[Td^4[Td^2[X(n)]]] = \{1,4,6\},\{2,6\},\{3,4,5\},\{1,4,5\},\{1,5\},\{4,5\},\{3,4,6\},\{1,2\}.$$

It is simple to confirm that the same $Y(k)$ exists in all of these dyadic shifted sequences, i.e.,

$$Y(K) = \{1,2,3,4,5,6\},\{2,3\},\{1,3,6\},\{3,6\},\{2,4,5,6\},\{2\},\{4,6\},\{4,6\}$$

Another method of dyadic input sequence shifting exists $X(n)$ to $Td^2[Td^4[Td^8[X(n)]]]$.

Let's take $X(n)$ as below:

$$X(n) = \{1,2\},\{3,4,6\},\{4,5\},\{1,5\},\{1,4,5\},\{3,4,5\},\{2,6\},\{1,4,6\}$$

The following dyadic shift can be obtained from $X(n)$

$$Td^8[X(n)] = \{3,4,6\},\{1,2\},\{1,5\},\{4,5\},\{3,4,5\},\{1,4,5\},\{1,4,6\},\{2,6\}$$

$$Td^4[Td^8[X(n)]] = \{4,5\},\{1,5\},\{3,4,6\},\{1,2\},\{1,4,6\},\{2,6\},\{3,4,5\},\{1,4,5\} \text{ and}$$

$$Td^2[Td^4[Td^8[X(n)]]] = \{1,4,6\},\{2,6\},\{3,4,5\},\{1,4,5\},\{4,5\},\{1,5\},\{3,4,6\},\{1,2\}$$

Note that, $Td^8[Td^4[Td^2[X(n)]]] = Td^2[Td^4[Td^8[X(n)]]]$

It is clear from the examples above that, in addition to $Td^4[Td^2[X(n)]]$ and $Td^8[X(n)]$,

The cyclic permutation class of $X(n)$ encompasses all other dyadically permuted sequences. Essentially, this means that the cyclic permutation class of $X^{-1}(n)$ has 8 independent sequences of non-repeating values, while the class of $X(n)$ also has eight such sequences. Dyadic permutation $X^{-1}(n)$ class consists of two separate, non-repeating sequences. In con-

clusion, it was possible to see that these 18 sequences had the similar $Y(k)$ value.

5.5 RESULTS

This section can be discussed into two parts, first part will explain the conversion of normal text file to cipher file & vice-versa and the second part will highlight some test cases which is based on numeric digits and alphabets (uppercase only). Figure 5.5 was considered as an input text file, where input alphabets and numbers are in a human-readable form. The file 'text001.txt' contains more than 50 alphabet and numbers. After applying the proposed approach, the plain text is converted into cipher text and it is represented in figure 5.6 with the file name 'enc001.txt'.

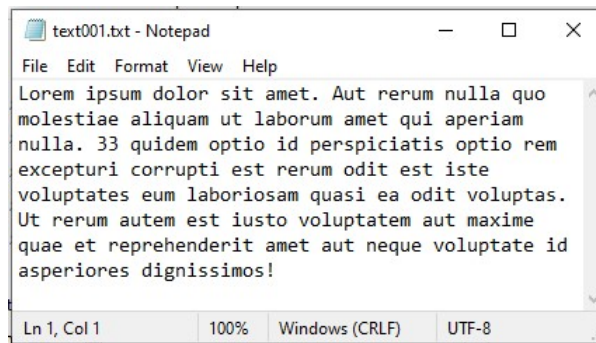


Figure 5.5: Input Text File

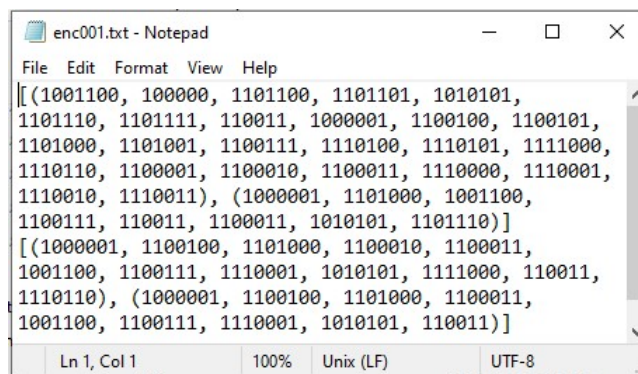


Figure 5.6: Processed Cipher File

After applying the proposed data security approach, the plane text file

was converted into cipher text and in the next step, a user should have to decrypt this encoded file to normal text. Figure 5.7 represents the final decrypted file as an output file.

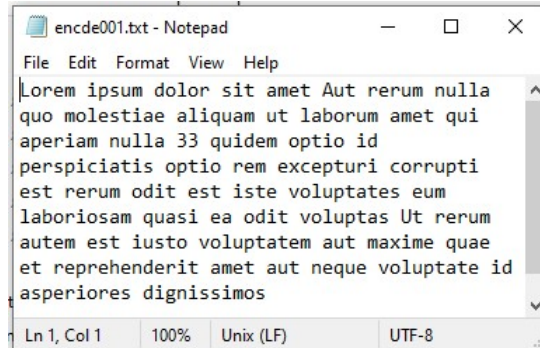


Figure 5.7: Decrypted Text File

5.6 CHAPTER SUMMARY

The present chapter represented the proposed approach which is based on addition (then subtraction) and subtraction (then addition) by applying a private (secret) key value and a chosen homomorphic transformation. The private key retained all the encrypted secrets within it. It can be concluded that without using the right private key and appropriate transformation, it is hard to decipher the input ciphertext. The results shown in present chapter also demonstrated that the output extracted from three variances of homomorphic transformation (inverse, dyadic, and cyclic) are giving similar sequences. To prevent the cipher attacks, for each data sequence set, different natured encryption methods can be applied. One can easily conclude, the brute force attacks are unable to find the original text once the key size grows bigger. The private key along with the homomorphic transformation presented as a proposed approach on two different levels. If required, the number of iterations can be enlarged. So, in worst case scenario, the algorithm is known, after that it is very hard to break the

resultant encrypted text.

Chapter 6

IMPLEMENTATION OF A HOMOMORPHIC TRANSFORM TO SECURE THE TEXT DATA

Various existing cryptography techniques have been used to enhance data security for various applications. These techniques are based on data enciphering and deciphering. The previous chapter presented a novel framework for data security using homomorphic transformation. Using addition and subtraction in a secret key and a homomorphic transformation were selected to enhance the security at two levels. After applying the proposed approach, even if the algorithm is known, it is a challenging task to extract the encrypted text. The number of rounds in the proposed approach can also be increased if required. The present chapter discusses the outputs of the proposed approach and discussion on these outputs. The organization of the present chapter is as follows: section 6.1 demonstrate the experimental setup & design, section 6.2 shows the results & discussion, and section 6.3 discusses the comparison of the proposed work with existing

contemporary data encryption techniques.

6.1 EXPERIMENTAL SETUP AND DESIGN

This section describes the experimental setup simulation for the proposed work. The Python Spyder was installed on a personal computer having the configuration of a 3.2 GHz i5 processor to implement the proposed work. The selected sample size for testing the proposed work was 50 different nature text files. By applying the proposed approach, these files are successfully converted into cipher files and vice-versa. Three main parameters are selected to complete the encryption process and to test the efficiency of the proposed approach. These parameters are the number of rounds, execution time, and size of the file. Table 6.1 shows the simulation setup for the proposed data security algorithm.

Table 6.1: Simulation Setup & Design

Hardware	CPU	Intel® Core™ i3-6006U CPU @ 2.0 GHz x 2.0 GHz
	RAM	8 GB
	External storage	500 GB
Software	Internet	Gigabit Ethernet
	Operating system	Windows 10 pro
	Software	Python Spyder

6.2 EXPERIMENTAL OUTCOMES

The performance efficiency of the proposed approach can be calculated by the number of rounds, block size, and key size. To improve further, Rajan Transform has been applied and it provided flexibility during the use of key size and block size. These results show that if the input size is 2^n bits then the key size is $2n \times n$, where n is the number of rounds to produce 2^n bits data. Similarly, Rajan transformation [128] takes 9 rounds to produce ciphertext with the key size of 4096 bits for the input value of 512 bits.

The calculated time complexity of the proposed algorithms is an order of n means $O(n)$.

6.2.1 Execution Time

The graphical representation in figure 6.1 demonstrated the relationship between the input size of the text file and corresponding to its execution time. The execution time of the input text file is directly proportional to the size of the file which depicts the increase in size and shows the increase in execution time. There is a linear relationship between these two entities. Similarly, figure 6.2 depicts the relationship between the input file size and the number of rounds (iteration) taken.

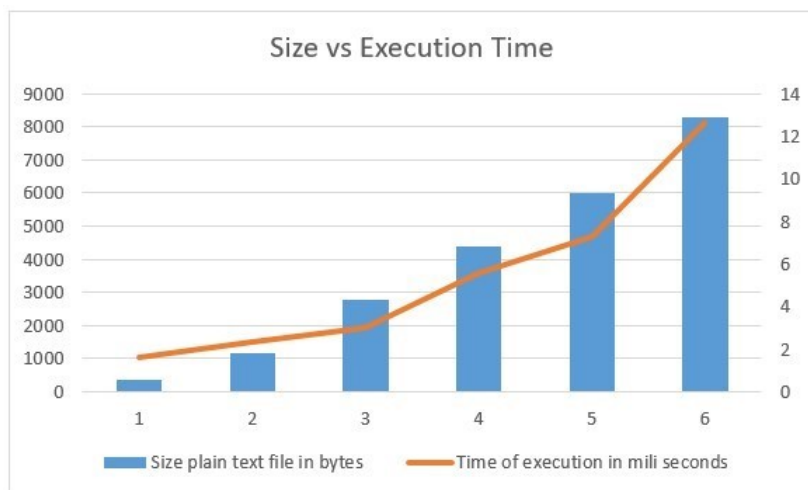


Figure 6.1: Text File Size and Execution Time

6.2.2 Entropy

Cryptography is one of the many areas where entropy is helpful. A useful way to distinguish between strong encryption and poor or nonexistent encryption is to measure the randomness in a system. If the cipher texts that an encryption algorithm generates can be distinguished from a random binary string, is one way to tell that, the algorithm is effective. Maxi-

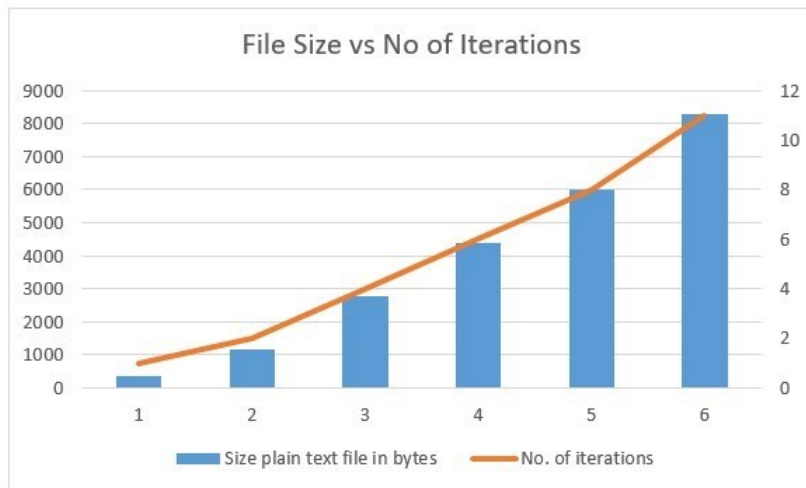


Figure 6.2: Input File Size and Number of Iterations

imum entropy, or the absence of information, is found in a wholly random binary string. This makes the ciphertext completely opaque to the associated plaintext, which is desired in an encryption scheme. To distinguish between ciphertext produced by a strong encryption technique and the usage of possibly weak and broken encryption, the entropy of data can be calculated.

There are numerous methods for calculating entropy. Shannon entropy, was developed by Claude Shannon, the inventor of information theory, is the sort of entropy that is most frequently employed in cryptography. Based on the observed likelihood that an event will occur, Shannon entropy can be estimated. This pertains to cryptography and the quantity of 0 and 1 in the ciphertext. The entropy decreases and more knowledge about the associated plaintext may be gleaned the more peculiar the ciphertext is. It is feasible to tell if data is encrypted using a reliable encryption method and if a specific ciphertext was produced using a flawed encryption technique to find high-entropy data.

The experiment was performed on 50 different plain text files, and the entropy was calculated for each encrypted file. figure 6.3 shows the histogram of the entropy values received from encrypted files.

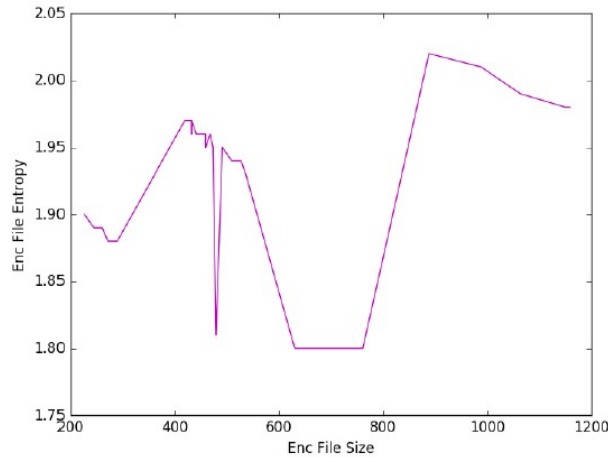


Figure 6.3: Histogram of Entropy values received from encryption file

The appearance of an encrypted file as histogram is shown in Figure 6.3. Entropy of encrypted files is close to 2. They wouldn't be properly encrypted if they are not more than 1. Patterns result in lower entropy, explains the bad quality of encryption, where as good quality of encryption requires higher entropy.

However, the entropy calculated with the parameter assigned for the contemporary encryption algorithms, which are isomorphism in nature, where the mapping from input to output is one to one. Whereas the proposed methodology is homomorphic in nature, where one input is mapped to many outputs. Each time it cipher the plain text, it will receive the different value of cipher text.

6.3 COMPARATIVE ANALYSIS

This section is divided into three parts that are execution time, avalanche effect, and entropy. The detailed discussed on these three categories are mentioned as follows,

6.3.1 Execution Time

The proposed approach is compared with the AES encryption scheme. The experiment was performed with some sample size of the text file to calculate the execution time with respect to the size. Table 6.2 shows the sample file size and their respective execution time.

Table 6.2: Execution time versus file size

S. No.	File Size (MB)	Execution Time (Sec)
1	2.5	23.12
2	4.3	44.6
3	5.6	52.3
4	7.3	71.34
5	8.4	84
6	17	172.5
7	19.4	186.28
8	20.3	210
9	23.3	244.6

Results obtained from the proposed algorithm are compared with AES, which is shown in figure 6.3.

Table 6.3: Execution time of AES and proposed approach with respect to various file size

Algorithm File Size (MB)	2.5	4.3	5.6	7.3	8.4	17.0	19.4	20.3	23.3
AES (Sec)	40.5	71.07	90.63	118.17	137.3	271.1	313.8	328.4	377.8
Proposed Technique (Sec)	23.12	44.6	52.3	71.34	84.0	172.5	186.28	210	244.6
Improvement (%)	0.75	0.59	0.73	0.65	0.63	0.57	0.68	0.65	0.54

The range of execution of the AES algorithm is between 40.5 seconds to 377.8 seconds for various file sizes ranging from 2.5MB to 23.3MB. The range of execution time of the proposed algorithm is in between 23.12 seconds to 244.6 seconds for similar file sizes. Figure 6.4 shows the execution time of AES and the proposed scheme versus file size.

The proposed method is also compared to the encryption schemes AES, DES, 3DES, Blowfish, and RSA. The experiment was carried out with a small sample size of the text file to determine the execution time concerning the file size. Table 6.4 presents the size of the sample files and their execution times.

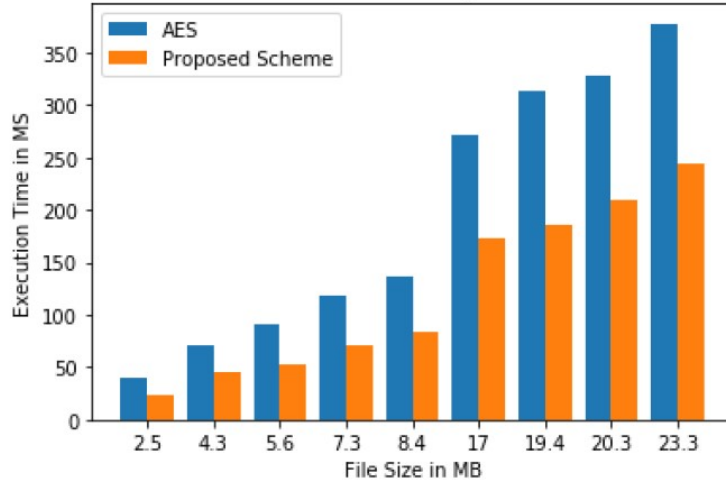


Figure 6.4: Execution time of AES and the proposed scheme versus file size

Table 6.4: Execution time of available schemes and the proposed scheme versus file size

Algorithm File Size	10 KB	50 KB	1MB
DES	400 ms	500 ms	750 ms
3 DES	400 ms	400 ms	750 ms
AES	600 ms	650 ms	650 ms
Blowfish	100 ms	350 ms	450 ms
RSA	600 ms	800 ms	1400 ms
Proposed	1400 ms	41 ms	93 ms

The range of execution of the DES and 3DES algorithms is between 400 ms to 750 ms. The range of execution of the AES algorithm is between 600 ms to 650 ms. The range of execution of the blowfish algorithm is between 100 ms to 450 ms. The range of execution of the RSA algorithm is between 600 ms to 1400 ms. The range of execution time of the proposed algorithm is between 41 ms to 1400 ms. Figure 6.5 shows the execution time of various other algorithms and the proposed scheme with respect to the file size.

Figure 6.5 shows that the execution time of the proposed is algorithm is better than available algorithms below 1 MB data size. However, beyond 1 MB data size the proposed algorithm gives a longer execution time to other algorithms.

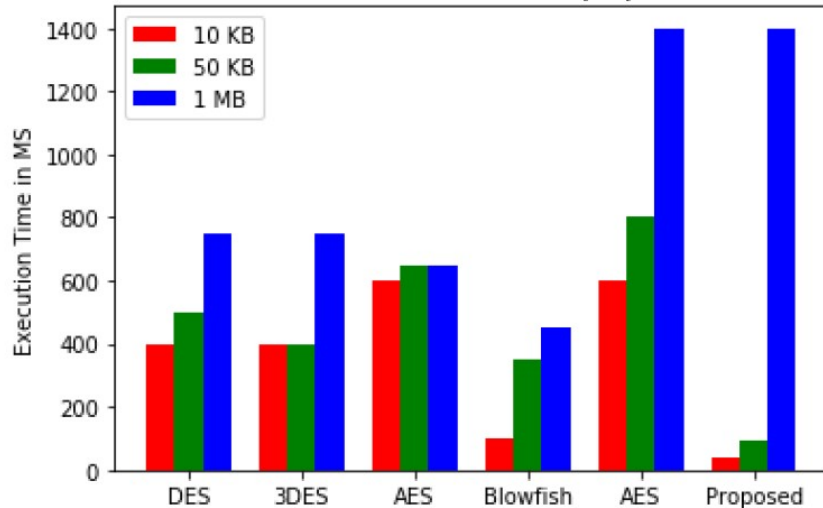


Figure 6.5: Execution time of various schemes and proposed scheme with respect to the file size

This implementation focus on lowering the length of the encrypted message in this proposed technique, which increases the complexity of the decryption done by the attacker. The message generates the key, which is then sent along with the encrypted message. The ciphertext is then reduced to the original message, and the key is of the same length as the ciphertext. Finally, the encrypted text and key take up the same amount of space as the original message.

6.3.2 Avalanche Effect

Diffusion is a cryptographic property that measures an algorithm's security. The output changes drastically when an input is even slightly changed. Avalanche effect is another name for this. Using hamming distance, we have measured the avalanche impact. In information theory, the hamming distance is used to quantify dissimilarity. As it is simple to do programmatically, we determine the hamming distance as the total of bit by bit XOR while taking into account ASCII values. It is preferred to have a high diffusion rate, or high avalanche effect. The avalanche effect shows how

well a cryptographic algorithm works.

Avalanche effect = (hamming distance \tilde{A} · file size)

Figure 6.6 shows the comparison of avalanche effect of different cryptography approaches versus proposed encryption scheme.

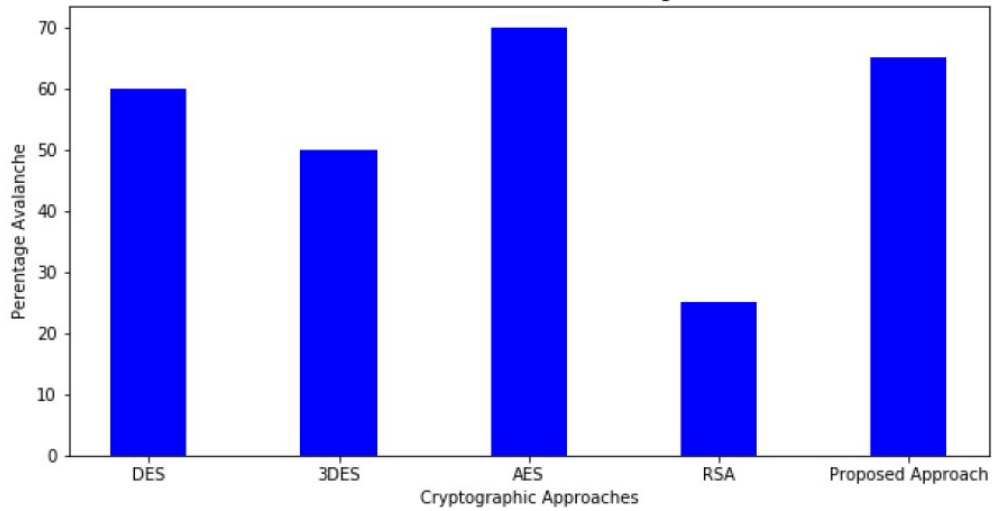


Figure 6.6: Avalanche Effect in Percentage

From Figure 6.6, it is understood that the proposed approach shows significant percentage of avalanche effect compares to DES, 3DES, AES and RSA.

6.3.3 Entropy

Entropy shows the degree of randomness. The value of average entropy should always be greater than one, otherwise it is always considered the weak encryption. The average entropy value comparison is shown in figure 6.7.

Figure 6.7 shows that the average entropy value of proposed algorithm is less than DES, 3DES, AES and RSA algorithm; however it is greater than two. Hence, proposed approach shows the reasonable entropy value to make it an effective cryptographic approach.

The entropy calculated with the parameter assigned for the contem-

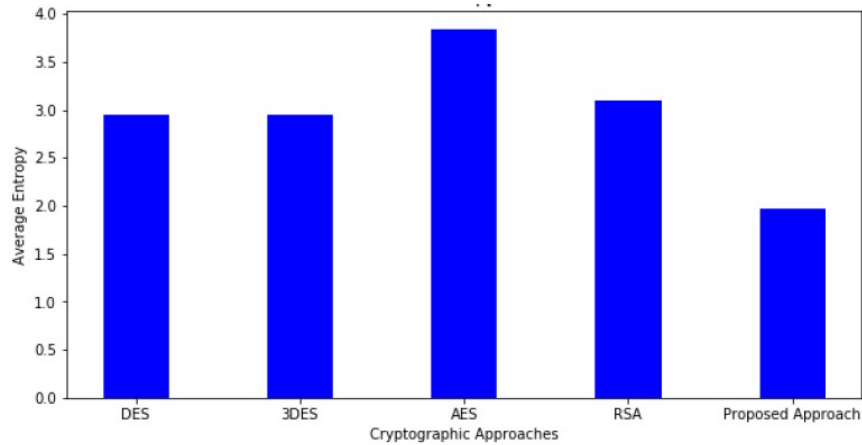


Figure 6.7: Average Entropy of different cryptographic approaches Vs. proposed approach

porary encryption algorithms, which are isomorphism in nature, where the mapping from input to output is one to one. Whereas the proposed methodology is homomorphic in nature, where one input is mapped to many outputs. Each time it cipher the plain text, it will receive the different value of cipher text. Thus the entropy of proposed approach which has average value close to two is for isomorphic process, not for homomorphic process. As this work uses homomorphic transform for cryptography process, which is unique in nature, has no method of calculating the entropy. However, the entropy shows the degree of uniqueness in cipher text, the work ensure that the value of each output is different from the earlier output, i.e., two cipher text of the same plain text file are always different.

Table 6.5 shows the comparison between encryption algorithms with proposed enhanced Rajan transformation. This table shows the comparative analysis of contemporary approaches like AES, DES, 3DES, Blowfish, and RSA with the proposed algorithm. The input file size is small to determine the exact execution time taking consideration into input file size. A significant improvement was shown by the proposed technique for input three parameters (file size, execution time, and no of rounds)

Table 6.5: Comparison of Encryption Techniques with Proposed Transform

Algorithm	Key Size	Block	Round	Flexible	Features
DES [8]	64 bits	64 bits	16	No	Not Strong Enough
3DES [9]	112 or 168	64 bits	48	Yes	Adequate Security
AES [10]	128, 192, 256 bits	128 bits	10, 12, 14	Yes	Replacement for DES, Excellent Security
Blowfish [11]	32-448	64 bits	16	Yes	Excellent Security
RC4 [12]	Variable	40-2048	256	Yes	Fast Cipher in SSL
Serpent [13]	128- 256	128 bits	32	Yes	Good Security
IDEA [14]	128 bits	64 bits	8.5	No	Not Strong Enough
RSA [15]	1,024 to 4096	128 bits	1	No	Excellent Security, low speed
Diffie Hellman[16]	1024 to 4096 bits	512	-	Yes	Many attacks
MD5 [17]	Series of MD	512	4	Hash function	Not Strong Enough
Rajan Transformation	4096 bit	512 bit	9	Yes	Excellent

6.4 VERIFICATION AND VALIDATION

For the verification and validation of the proposed approach, the cipher text file is given to Delhi based IT organization working in the area of cyber security, named Codec Network. The objective was to break the cryptographic processes by applying various attacks like bruit force attack, on cipher text and find out the plain text. The cipher text file was given in December 2020, and they reverted in January 2021, with the following findings.

According to the report received from the Codec Networks, the cipher-text by itself reveals nothing other than the fact that it produces a 7-bit word and that it comprises two lines of cipher-text, implying that the method was run twice or that there were two blocks of data submitted to the algorithm. They tried many test vectors to decipher this cipher-text and obtain the plain-text, but we were unable. Based on the tried test vectors, we can conclude that implementing a bare brute-force attack on the cipher-text is a very difficult operation, making this cipher-text extremely difficult.

6.5 CHAPTER SUMMARY

The present chapter discussed the experimental setup and design for the proposed approach. Initially, the process of converting a normal text file to a cipher file & vice-versa was successfully executed. From the experimental section, it was observed that the proposed technique confirmed the expected outcomes even in the case of inverted, translated, and normal matrix forms of characters used. The comparative analysis of the proposed approach with respect to the contemporary techques like AES, DES, 3DES, Blowfish, and RSA also highlighted. The comparison tables show that the proposed approach is more robust when compare to the existing techniques.

Chapter 7

CONCLUSION AND FUTURE DIRECTION

7.1 RESEARCH SUMMARY AND CON- TRIBUTIONS

With 3-dimensional improvement in various information technological techniques, data transfer became an essential task. During network communication, due to various purposes, the data is transmitted through different insecure channels. This insecurity is one of prime challenge in data cryptography. The term cryptography defines the codes to secure the data/information on or during the communication processes by reading/altering through an unauthenticated body. The various existing deterministic algorithms are used for private communications, email to secure various types of data, digital signature, cryptographic key generation, on-line browsing, credit card transactions, and data privacy verification. Data security is a foremost problem that arises during an unauthorized access to the datasets and data streams. To address the problem of data security, various encryption algorithms are already available but due to a lack in

their key generation technique, they stay in a vulnerable situation, so a robust data security algorithm is highly required.

The present thesis focuses to create an end-to-end framework for data encryption and decryption. The thesis works not only on the data security part for network communication but provides a novel key generation technique using the proposed approach. The main objective of the present work is to devise a novel framework to overcome the issues of key generation, for robust data security using Homomorphic Transformation secure data transmission during a lower level.

The proposed work is presented in seven chapter segments, where chapter one discussed the introduction of the various modern encryption algorithms such as symmetric encryption algorithm, asymmetric encryption algorithm, hashing scheme, message authentication codes, authenticated encryption schemes, and homomorphic transform, etc. This chapter also discussed the different applications of homomorphic cryptosystems such as text encryption, image encryption, one time pads, bank transactions, military communication, and bio-metric verification. These points are the base points of the present research work. To resolve these research issues, crisp and clear research objectives are also defined. Various research challenges along with thesis organization are also discussed in this chapter. In the last segment of chapter one, the proposed research methodology is formulated to significantly achieve the designed objectives.

The second chapter covered a thorough explanation of the related work on encryption schemes on the conventional encryption model, public key encryption schemes, contemporary encryption techniques, and DNA encryption for data security approaches. After a detailed discussion of the above-said encryption techniques, common existing encryption algorithms are explained. Out of various encryption algorithms, DES, triple-DES,

RSA, AES, BlowFish, and Twofish approaches are mainly focussed for data encryption and decryption. This literature review chapter summarizes the earlier researchers findings related to various approaches for data encryption and decryption

Chapter three narrows down the research on text cryptanalysis with the need for cryptanalysis for data encryption, and different types of attacks in the cryptanalysis. A brief discussion on cryptanalysis versus cipher design is also highlighted in this chapter. Types of cryptanalysis on the text cryptanalysis, histogram, and key space analysis are explored. The chapter also presented the outcomes of cryptanalysis and finally summarizes the text of cryptanalysis

Chapter four presented the proposed approach for data encryption to devise a novel framework to overcome the issue of key generation for robust data security using homomorphic transform. This chapter also shows, how algebraic properties of homomorphic transform helps to achieve desired output.

Chapter five discussed the set theoretic approach of homomorphic transform. This chapter discussed the results ordained after implementation of the proposed approach. This chapter discussed how algebraic properties were used to make the process homomorphic. It is shown that the proposed approach is able to provide the promising results. The proposed method was able to successfully covert plain text to cipher text and vice versa.

Chapter six discussed the experimental results and discussion of the proposed work with the help of various graphically and pictorial representations. The complete explanation of research contributions for the proposed algorithms is also discussed. The summary of the chapter briefly highlighted the results and experimentation work. At the end of the thesis

report, the present work is concluded with the significant future research directions in chapter seven.

7.2 RESEARCH CONTRIBUTIONS

Major contribution of the work proposed in this thesis are listed below:

7.2.1 Framework to Enhance the Data Security Measures using Homomorphic Transform

An enhanced data security measures using Homomorphic Transform, i.e. an enhanced method for securing data during network communication is presented. Without applying any decryption technique in its original form, the encrypted version of the dataset is available for data processing step. Performance of the presented method is shown to highlight it in terms of the algorithm's efficiency, block size, no of round, and key size required to execute the encryption process. It is observed that if the input data size of 512 bits is processed then it will get 512-bit of ciphertext and produce 1 key of 4608-bit in 9 rounds.

7.2.2 A Hadamard Transform using DNA Amino Acid and Cryptography

A approach is proposed coined as Hadamard Transform using DNA amino acid and cryptography for improvising data security using data transformation and encapsulation approaches.

7.2.3 A new robust cryptographic technique to improve data security in text messages.

A new robust cryptographic technique is presented to improve the data security in text messages from different types of unwanted attacks such as brute force. It is shown that from various input size files, ranging from 2.5 to 23.3 MB file size, the proposed techniques show 54% (at least) improvement compared to the traditional AES approach. The proposed approach is compared with DES, 3DES, AES, Blowfish, & RSA. It is observed that the proposed approach is better in terms of execution time parameters for data size less than 1 MB. But in the case of data size is larger than 1 MB, the proposed approach shows a larger execution time when compared with DES, 3DES, AES, Blowfish, & RSA. The proposed approach shows that there is no burden on the key and message transfer.

7.3 FUTURE RESEARCH DIRECTIONS

Data security is a prime concern when transferring the data from one end to another in any computing device over a network. The study shows that various data security approaches have vulnerability in data security, those can be addressed and the presented approach shows a significant improvement. Following future improvements in the said area, can be explored as a future direction work:

- Currently, there is a large number of application areas, which need a high-level of message security, so the present research work can be utilized for the majority of critical applications that demand a high level of message security.
- The future areas of the presented research work can be data collection & data measurement issues, cross-cultural InfoSec research, improv-

ing information security compliance, unmasking the mystery of the hacker world and insider deviant behavior, etc.

- The extendable work of the presented work can be implemented for text and images multimedia files as well.
- The contemporary cryptographic approaches are isomorphic in nature, where as use of homomorphic transform open new doors to identify methods to break such algorithms.
- Use of homomorphic transform in cryptography not only encrypt the data, but also compress the cipher text. Hence the proposed approach can also be used as encryption algorithm.

The future enhancement in existing data security techniques continuously opens new research domains along with an open call for various ongoing research applications. The present research work can be move-forward with the above-highlighted future directions. The implementation of future directions may be an appreciable step in the area of data security.

Bibliography

- [1] A. Asif and S. Hannan, “A review on classical and modern encryption techniques,” *International Journal of Engineering Trends and Technology*, vol. 12, no. 4, pp. 199–203, 2014.
- [2] D. R. Lide, *A century of excellence in measurements, standards, and technology*. CRC Press, 2018.
- [3] M. Yamuna and A. Elakkiya, “Amino acids in data encryption,” *Journal of Analytical & Pharmaceutical Research*, vol. 2, no. 5, pp. 29–31, 2016.
- [4] R. Shokri and V. Shmatikov, “Privacy-preserving deep learning,” in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321, 2015.
- [5] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, “Multi-key privacy-preserving deep learning in cloud computing,” *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.
- [6] S. E. Ghrare, H. A. Barghi, and N. R. Madi, “New text encryption method based on hidden encrypted symmetric key,” 2018.
- [7] X. Liang, S. Xiang, L. Yang, and J. Li, “Robust and reversible image watermarking in homomorphic encrypted domain,” *Signal Processing: Image Communication*, vol. 99, p. 116462, 2021.

- [8] P. Srikanth, A. Mehta, N. Yadav, S. Singh, and S. Singhal, “Encryption and decryption using genetic algorithm operations and pseudo-random number,” *IJCSN-International Journal of Computer Science and Network*, vol. 6, no. 3, pp. 455–459, 2017.
- [9] J. C. L. da Silva, “Factoring semiprimes and possible implications for rsa,” in *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, pp. 000182–000183, IEEE, 2010.
- [10] B. Geethavani, E. Prasad, and R. Roopa, “A new approach for secure data transfer in audio signals using dwt,” in *2013 15th International Conference on Advanced Computing Technologies (ICACT)*, pp. 1–6, IEEE, 2013.
- [11] S. A. Nagar and S. Alshamma, “High speed implementation of rsa algorithm with modified keys exchange,” in *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pp. 639–642, IEEE, 2012.
- [12] Y.-y. Cao and C. Fu, “An efficient implementation of rsa digital signature algorithm,” in *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, vol. 2, pp. 100–103, IEEE, 2008.
- [13] L. Dongjiang, W. Yandan, and C. Hong, “The research on key generation in rsa public-key cryptosystem,” in *2012 Fourth international conference on computational and information sciences*, pp. 578–580, IEEE, 2012.
- [14] H. Si, Y. Cai, and Z. Cheng, “An improved rsa signature algorithm based on complex numeric operation function,” in *2010 International*

Conference on Challenges in Environmental Science and Computer Engineering, vol. 2, pp. 397–400, IEEE, 2010.

- [15] N. B. Silva, D. F. Pigatto, P. S. Martins, and K. R. Branco, “Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer,” *Journal of Network and Computer Applications*, vol. 60, pp. 130–143, 2016.
- [16] A. VISHNOI, D. SHARMA, and M. PRATEEK, “Improvising data security measures using rajan transform,” *Journal of Engineering Science and Technology*, vol. 16, no. 6, pp. 4920–4932, 2021.
- [17] A. Gupta and N. K. Walia, “Cryptography algorithms: a review,” 2014.
- [18] S. Koko, A. Babiker, *et al.*, “Comparison of various encryption algorithms and techniques for improving secured data communication,” *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 17, no. 1, pp. 62–69, 2015.
- [19] R. Sharma and S. Bollavarapu, “Data security using compression and cryptography techniques,” *International Journal of Computer Applications*, vol. 117, no. 14, 2015.
- [20] I. Basharat, F. Azam, and A. W. Muzaffar, “Database security and encryption: A survey study,” *International Journal of Computer Applications*, vol. 47, no. 12, 2012.
- [21] G. Singh, “A study of encryption algorithms (rsa, des, 3des and aes) for information security,” *International Journal of Computer Applications*, vol. 67, no. 19, 2013.

- [22] N. Mathur and R. Bansode, “Aes based text encryption using 12 rounds with dynamic key selection,” *Procedia Computer Science*, vol. 79, pp. 1036–1043, 2016.
- [23] B. S. Ross and V. Josephraj, “Performance enhancement of blowfish encryption using rk blowfish,” *International Journal of Applied Engineering Research*, vol. 12, no. 20, pp. 9236–9244, 2017.
- [24] S. Sriadhi, R. Rahim, and A. S. Ahmar, “Rc4 algorithm visualization for cryptography education,” in *Journal of Physics: Conference Series*, vol. 1028, p. 012057, IOP Publishing, 2018.
- [25] G. T. Cayabyab, A. M. Sison, and A. A. Hernandez, “Giskop: A modified key scheduling operation of international data encryption algorithm using serpent key scheduling,” in *Proceedings of the 2nd International Conference on Computing and Big Data*, pp. 53–57, 2019.
- [26] D. Abdullah, R. Rahim, A. P. U. Siahaan, A. F. Ulva, Z. Fitri, M. Malahayati, and H. Harun, “Super-encryption cryptography with idea and wake algorithm,” in *Journal of Physics: Conference Series*, vol. 1019, p. 012039, IOP Publishing, 2018.
- [27] S. Nisha and M. Farik, “Rsa public key cryptography algorithm—a review,” *International journal of scientific & technology research*, vol. 6, no. 7, pp. 187–191, 2017.
- [28] S. Boni, J. Bhatt, and S. Bhat, “Improving the diffie-hellman key exchange algorithm by proposing the multiplicative key exchange algorithm,” *International Journal of Computer Applications*, vol. 130, no. 15, 2015.

- [29] H. W. Dhany, F. Izhari, H. Fahmi, M. Tulus, and M. Sutarman, "Encryption and decryption using password based encryption, md5, and des," in *International conference on public policy, social computing and development*, vol. 2018, 2017.
- [30] S. Kumari and J. Chawle, "Comparative analysis on different parameters of encryption algorithms for information security," *International Journal of Innovation & Advancement in Computer Science*, vol. 2, pp. 123–129, 2015.
- [31] O. M. A. Al-Hazaimeh, "A new approach for complex encrypting and decrypting data," *International Journal of Computer Networks & Communications*, vol. 5, no. 2, p. 95, 2013.
- [32] Y. Aono, T. Hayashi, L. Wang, S. Moriai, *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.
- [33] P. Mahajan and A. Sachdeva, "A study of encryption algorithms aes, des and rsa for security," *Global Journal of Computer Science and Technology*, 2013.
- [34] K. Acharya, M. Sajwan, and S. Bhargava, "Analysis of cryptographic algorithms for network security," *International Journal of Computer Applications Technology and Research*, vol. 3, no. 2, pp. 130–135, 2013.
- [35] C.-L. Wu and C.-H. Hu, "Computational complexity theoretical analyses on cryptographic algorithms for computer security application," in *2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications*, pp. 307–311, IEEE, 2012.

- [36] B. K. Mandal, D. Bhattacharyya, and S. K. Bandyopadhyay, "Designing and performance analysis of a proposed symmetric cryptography algorithm," in *2013 International Conference on Communication Systems and Network Technologies*, pp. 453–461, IEEE, 2013.
- [37] S. Halder and T. Newe, "Enabling secure time-series data sharing via homomorphic encryption in cloud-assisted iiot," *Future Generation Computer Systems*, vol. 133, pp. 351–363, 2022.
- [38] V. Kadykov, A. Levina, and A. Voznesensky, "Homomorphic encryption within lattice-based encryption system," *Procedia Computer Science*, vol. 186, pp. 309–315, 2021.
- [39] M. Sabry, M. Hashem, and T. Nazmy, "Three reversible data encoding algorithms based on dna and amino acids' structure," *International Journal of Computer Applications*, vol. 54, no. 8, 2012.
- [40] S. Namdev and V. Gupta, "A dna and amino-acids based implementation of four-square cipher," *Int. Journal of Engineering Research and Applications*, vol. 6, pp. 90–96, 2016.
- [41] A. Atito, A. Khalifa, and S. Rida, "Dna-based data encryption and hiding using playfair and insertion techniques," *Journal of Communications and Computer Engineering*, vol. 2, no. 3, p. 44, 2012.
- [42] P. Singh, M. Masud, M. S. Hossain, and A. Kaur, "Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid," *Computers & Electrical Engineering*, vol. 93, p. 107209, 2021.
- [43] L. Brenna, I. S. Singh, H. D. Johansen, and D. Johansen, "Tfhe-rs: A library for safe and secure remote computing using fully homomor-

- phic encryption and trusted execution environments,” *Array*, vol. 13, p. 100118, 2022.
- [44] C. Fu, G.-y. Zhao, M. Gao, and H.-f. Ma, “A chaotic symmetric image cipher using a pixel-swapping based permutation,” in *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*, pp. 1–6, IEEE, 2013.
- [45] A. Ghosh, “Comparison of encryption algorithms: Aes, blowfish and twofish for security of wireless networks,” *International Research Journal of Engineering Technology*, vol. 7, pp. 4656–4658, 2020.
- [46] D. Venu, A. Mayuri, S. Neelakandan, G. Murthy, N. Arulkumar, and N. Shelke, “An efficient low complexity compression based optimal homomorphic encryption for secure fiber optic communication,” *Optik*, vol. 252, p. 168545, 2022.
- [47] S. Meftah, B. H. M. Tan, K. M. M. Aung, L. Yuxiao, L. Jie, and B. Veeravalli, “Towards high performance homomorphic encryption for inference tasks on cpu: An mpi approach,” *Future Generation Computer Systems*, vol. 134, pp. 13–21, 2022.
- [48] S. D. Samarin, D. Fiore, D. Venturi, and M. Amini, *A compiler for multi-key homomorphic signatures for Turing machines*, vol. 889. Elsevier, 2021.
- [49] W. Stallings, “Cryptography and network security principles and practice seventh edition global edition british library cataloguing-in-publication data,” 2017.
- [50] A. Aloufi, P. Hu, H. Liu, S. S. Chow, and K.-K. R. Choo, “Universal location referencing and homomorphic evaluation of geospatial query,” *Computers & Security*, vol. 102, p. 102137, 2021.

- [51] R. A. Rueppel, *Analysis and design of stream ciphers*. Springer Science & Business Media, 2012.
- [52] W. Ren, X. Tong, J. Du, N. Wang, S. C. Li, G. Min, Z. Zhao, and A. K. Bashir, *Privacy-preserving using homomorphic encryption in Mobile IoT systems*, vol. 165. Elsevier, 2021.
- [53] H. E. D. Kang, D. Kim, S. Kim, D. D. Kim, J. H. Cheon, and B. W. Anthony, “Homomorphic encryption as a secure phm outsourcing solution for small and medium manufacturing enterprise,” *Journal of Manufacturing Systems*, vol. 61, pp. 856–865, 2021.
- [54] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts,” in *2016 IEEE symposium on security and privacy (SP)*, pp. 839–858, IEEE, 2016.
- [55] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, “Blockchain,” *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 2017.
- [56] R. Henry, A. Herzberg, and A. Kate, “Blockchain access privacy: Challenges and directions,” *IEEE Security & Privacy*, vol. 16, no. 4, pp. 38–45, 2018.
- [57] S. Dahiya, “Multilevel data encryption using hadamard transform based image steganography,” 2016.
- [58] G. Sosa-Gómez, O. Rojas, and O. Páez-Osuna, “Using hadamard transform for cryptanalysis of pseudo-random generators in stream ciphers,” *EAI Endorsed Transactions on Energy Web*, vol. 7, no. 27, 2020.

- [59] M. N. A. Wahid, A. Ali, B. Esparham, and M. Marwan, “A comparison of cryptographic algorithms: Des, 3des, aes, rsa and blowfish for guessing attacks prevention,” *Journal Computer Science Applications and Information Technology*, vol. 3, no. 2, pp. 1–7, 2018.
- [60] Y. Rajput, D. Naik, and C. Mane, “An improved cryptographic technique to encrypt text using double encryption,” *International journal of Computer Applications*, vol. 86, no. 6, 2014.
- [61] R. Zhao and M. Iwaihara, “Lightweight efficient multi-keyword ranked search over encrypted cloud data using dual word embeddings,” *arXiv preprint arXiv:1708.09719*, 2017.
- [62] P. A.-N. Agbedemrab, E. Y. Baagyere, and M. I. Daabo, “A novel text encryption and decryption scheme using the genetic algorithm and residual numbers,” in *Proceedings of 4th International Conference on the*, vol. 12, pp. 20–31, 2019.
- [63] A. Ali, “Randomly encryption using genetic algorithm,” *Int. J. Appl. Innov. Eng. Manage.*, vol. 2, no. 8, pp. 242–246, 2013.
- [64] L. Sadeghikhorami and A. A. Safavi, “Secure distributed kalman filter using partially homomorphic encryption,” *Journal of the Franklin Institute*, vol. 358, no. 5, pp. 2801–2825, 2021.
- [65] R. Al Sobhahi and J. Tekli, “Low-light homomorphic filtering network for integrating image enhancement and classification,” *Signal Processing: Image Communication*, vol. 100, p. 116527, 2022.
- [66] P. A.-N. Agbedemrab, E. Y. Baagyere, and M. I. Daabo, “A novel text encryption and decryption scheme using the genetic algorithm and residual numbers,” in *Proceedings of 4th International Conference on the*, vol. 12, pp. 20–31, 2019.

- [67] K. Waleed, M. Mohammed, and S. Norrozila, “An enhanced encryption algorithm for database protection based on dynamic key and reverse string,” *Journal of Engineering and Applied Sciences*, vol. 12, no. 5, pp. 1186–1191, 2017.
- [68] A. Odeh, S. R. Masadeh, and A. Azzazi, “A performance evaluation of common encryption techniques with secure watermark system (sws),” *International Journal of Network Security & Its Applications (IJNSA)*, vol. 7, no. 3, pp. 31–38, 2015.
- [69] G. Prakash, M. Prateek, and I. Singh, “Data encryption and decryption algorithms using key rotations for data security in cloud system,” in *2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014)*, pp. 624–629, IEEE, 2014.
- [70] N. Kaur and S. Sodhi, “Data encryption standard algorithm (des) for secure data transmission,” in *International Conference on Advances in Emerging Technology (ICAET)*, pp. 31–37, 2016.
- [71] B. R. Kumar and P. Murti, “Data encryption and decryption process using bit shifting and stuffing(bss) methodology,” *International Journal on Computer Science and Engineering*, vol. 3, no. 7, pp. 2818–2827, 2011.
- [72] I. Sumartono, A. P. U. Siahaan, and N. Mayasari, “An overview of the rc4 algorithm,” *IOSR J. Comput. Eng.*, vol. 18, no. 6, pp. 67–73, 2016.
- [73] Y. Fan, J. Bai, X. Lei, W. Lin, Q. Hu, G. Wu, J. Guo, and G. Tan, “Ppmck: Privacy-preserving multi-party computing for k-means clustering,” vol. 154, pp. 54–63, Elsevier, 2021.

- [74] S. Wang and G. Liu, "File encryption and decryption system based on rsa algorithm," in *2011 International Conference on Computational and Information Sciences*, pp. 797–800, IEEE, 2011.
- [75] T. Wenxue, W. Xiping, X. Jinju, and P. Meisen, "A mechanism of quantitating the security strength of rsa key," in *2010 Third International Symposium on Electronic Commerce and Security*, pp. 357–361, IEEE, 2010.
- [76] R. S. Dhakar, A. K. Gupta, and P. Sharma, "Modified rsa encryption algorithm (mrea)," in *2012 second international conference on advanced computing & communication technologies*, pp. 426–429, IEEE, 2012.
- [77] A. Nadeem and M. Y. Javed, "A performance comparison of data encryption algorithms," in *2005 international Conference on information and communication technologies*, pp. 84–89, IEEE, 2005.
- [78] D. S. Abd Elminaam, H. M. Abdual-Kader, and M. M. Hadhoud, "Evaluating the performance of symmetric encryption algorithms.," *Int. J. Netw. Secur.*, vol. 10, no. 3, pp. 216–222, 2010.
- [79] A. K. Mandal, C. Parakash, and A. Tiwari, "Performance evaluation of cryptographic algorithms: Des and aes," in *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science*, pp. 1–5, IEEE, 2012.
- [80] F. Maqsood, M. Ahmed, M. Mumtaz, and M. Ali, "Cryptography: a comparative analysis for modern techniques," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 442–448, 2017.

- [81] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, and E. Roback, “Report on the development of the advanced encryption standard (aes),” *Journal of Research of the National Institute of Standards and Technology*, vol. 106, no. 3, p. 511, 2001.
- [82] C. E. Shannon, “Communication theory of secrecy systems,” *The Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.
- [83] A. Kahate, *Cryptography and network security*. Tata McGraw-Hill Education, 2013.
- [84] M. Nofer, P. Gomer, O. Hinz, and D. Schiereck, “Blockchain,” *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 2017.
- [85] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press, 2020.
- [86] Y. Rajput, D. Naik, and C. Mane, “An improved cryptographic technique to encrypt text using double encryption,” *International journal of Computer Applications*, vol. 86, no. 6, 2014.
- [87] C. W. Wu and N. F. Rul’kov, “Studying chaos via 1-d maps-a tutorial,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 10, pp. 707–721, 1993.
- [88] D. Younes and P. Steffan, “Universal approaches for overflow and sign detection in residue number system based on $\{2n-1, 2n, 2n+1\}$,” in *Proceedings of the Eighth International Conference on Systems (ICONS)*, vol. 1, pp. 77–84, 2013.
- [89] C. Narasimham and J. Pradhan, “Evaluation of performance characteristics of cryptosystem using text files,” *Journal of Theoretical & Applied Information Technology*, vol. 4, no. 1, 2008.

- [90] N. Singhal and J. Raina, "Comparative analysis of aes and rc4 algorithms for better utilization," *International Journal of Computer Trends and Technology*, vol. 2, no. 6, pp. 177–181, 2011.
- [91] P. C. Mandal, "Evaluation of performance of the symmetric key algorithms: Des, 3des, aes and blowfish," *Journal of Global Research in Computer Science*, vol. 3, no. 8, pp. 67–70, 2012.
- [92] G. Singh, A. K. Singla, and K. Sandha, "Throughput analysis of various encryption algorithms," *IJCST*, vol. 2, no. 3, 2011.
- [93] S. S. Mehrotra and M. Rajan, "Comparative analysis of encryption algorithm for data communication," *International Journal of Computer Science and Technology*, vol. 2, no. 2, pp. 292–294, 2011.
- [94] G. Singh, A. K. Singla, and K. Sandha, "Superiority of blowfish algorithm in wireless networks," *International Journal of Computer Applications*, vol. 44, no. 11, pp. 23–26, 2012.
- [95] A. Ramesh and A. Suruliandi, "Performance analysis of encryption algorithms for information security," in *2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, pp. 840–844, IEEE, 2013.
- [96] M. Rani and S. Kumar, "Analysis on different parameters of encryption algorithms for information security," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 8, pp. 104–108, 2015.
- [97] D. Sharma, M. Prateek, and T. Chattopadhyay, "Dct and simulink based realtime robust image watermarking," *International Journal of Image Processing (IJIP)*, vol. 8, no. 4, pp. 214–219, 2014.

- [98] D. Sharma, M. Prateek, and T. Chattopadhyay, “Realtime energy efficient digital image watermarking on mobile devices using android,” *International Journal of Image Processing (IJIP)*, vol. 9, no. 2, p. 61, 2015.
- [99] K. Deepak and D. Pawan, “Performance comparison of symmetric data encryption techniques. issn: 2278–1323,” *Int. J. Adv. Res. Comput. Eng. Tech*, vol. 1, no. 4, 2012.
- [100] E. N. Mandalapu and E. Rajan, “Rajan transform and its uses in pattern recognition,” *Informatica*, vol. 33, no. 2, 2009.
- [101] P. Sirohi and A. Agarwal, “Cloud computing data storage security framework relating to data integrity, privacy and trust,” in *2015 1st international conference on next generation computing technologies (NGCT)*, pp. 115–118, IEEE, 2015.
- [102] C. Regueiro, I. Seco, S. de Diego, O. Lage, and L. Etxebarria, “Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption,” *Information Processing & Management*, vol. 58, no. 6, p. 102745, 2021.
- [103] L. Cui, G. Xie, Y. Qu, L. Gao, and Y. Yang, “Security and privacy in smart cities: Challenges and opportunities,” *IEEE access*, vol. 6, pp. 46134–46145, 2018.
- [104] B. K. Dewangan, A. Agarwal, A. Pasricha, *et al.*, “Credential and security issues of cloud service models,” in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 888–892, IEEE, 2016.
- [105] M. Lakhera, M. Rauthan, and A. Agarwal, “An efficient cryptographic algorithm for securing biometric template using aes and

- scrambling the pixels of row and column,” in *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, pp. 228–231, IEEE, 2016.
- [106] G. Prakash, D. M. Prateek, and D. I. Singh, “Data security algorithms for cloud storage system using cryptographic method,” *International Journal of Scientific & Engineering Research*, vol. 5, no. 3, pp. 54–61, 2014.
- [107] G. Prakash, M. Prateek, and I. Singh, “Efficient data security method to control data in cloud storage system using cryptographic techniques,” in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, pp. 1–6, IEEE, 2014.
- [108] A. v. Oppenheim, R. Schafer, and T. Stockham, “Nonlinear filtering of multiplied and convolved signals,” *IEEE transactions on audio and electroacoustics*, vol. 16, no. 3, pp. 437–466, 1968.
- [109] T. Huang, “Stability of two-dimensional recursive filters,” *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 2, pp. 158–163, 1972.
- [110] P. Whittle, “On stationary processes in the plane,” *Biometrika*, pp. 434–449, 1954.
- [111] M. Ekstrom and J. Woods, “Two-dimensional spectral factorization with applications in recursive digital filtering,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 2, pp. 115–128, 1976.
- [112] D. Goodman and M. Ekstrom, “Multidimensional spectral factorization and unilateral autoregressive models,” *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 258–262, 1980.

- [113] M. Ekstrom, R. Twogood, and J. Woods, “Two-dimensional recursive filter design—a spectral factorization approach,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 1, pp. 16–26, 1980.
- [114] J. Woods, J.-H. Lee, and I. Paul, “Two-dimensional iir filter design with magnitude and phase error criteria,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 31, no. 4, pp. 886–894, 1983.
- [115] J.-H. Le and Y.-M. Chen, “A new method for the design of two-dimensional recursive digital filters,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 589–598, 1988.
- [116] D. G. Childers, D. P. Skinner, and R. C. Kemerait, “The cepstrum: A guide to processing,” *Proceedings of the IEEE*, vol. 65, no. 10, pp. 1428–1443, 1977.
- [117] H. W. Swan, “Phase averaging of image ensembles by using cepstral gradients,” *JOSA*, vol. 73, no. 11, pp. 1488–1492, 1983.
- [118] D. Goodman, “Some properties of the multidimensional complex cepstrum and their relationship to the stability of multidimensional systems,” *Circuits, Systems and Signal Processing*, vol. 6, no. 1, pp. 3–30, 1987.
- [119] K. Davidson and M. Vidyasagar, “Causal invertibility and stability of asymmetric half-plane digital filters,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 1, pp. 195–201, 1983.
- [120] I. J. Maddox, *Elements of functional analysis*. CUP Archive, 1988.

- [121] P. Henrici, *Applied and computational complex analysis, Volume 3: Discrete Fourier analysis, Cauchy integrals, construction of conformal maps, univalent functions*, vol. 41. John Wiley & Sons, 1993.
- [122] R. Delanghe, “On some properties of the hilbert transform in euclidean space,” *Bulletin of the Belgian Mathematical Society-Simon Stevin*, vol. 11, no. 2, pp. 163–180, 2004.
- [123] D. G. Luenberger, *Introduction to linear and nonlinear programming*, vol. 28. Addison-wesley Reading, MA, 1973.
- [124] D. Goodman, “M-dimensional linear systems: determination of impulse response support, computing cepstra, and recursibility issues,” tech. rep., California Univ., Livermore (USA). Lawrence Livermore Lab., 1979.
- [125] B. P. Bogert, “The quefrency alalysis of time series for echoes; cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking,” *Time series analysis*, pp. 209–243, 1963.
- [126] N. K. Bose, *Applied multidimensional systems theory*. Springer, 1982.
- [127] D. E. Dudgeon and R. M. Mersereau, *Multidimensional digital signal processing*. Prentice-Hall, 1984.
- [128] E. Bertino, “Data security—challenges and research opportunities,” in *Workshop on Secure Data Management*, pp. 9–13, Springer, 2013.

Annexure

List of Publications

Journals

- [1]. A. Vishnoi, D. Sharma, and M. Prateek, "Data security measures using hybrid encryption technique:", *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 323-326, 2019. Doi: 10.35940/ijitee.J1058.08810S19. [Index: SCOPUS].
<https://www.ijitee.org/wp-content/uploads/papers/v8i10S/J105808810S19.pdf>
- [2]. A. Vishnoi, D. Sharma, and M. Prateek, "Improvising Data Security Measures using Rajan Transform", *Journal of Engineering Science and Technology (JESTEC)*, vol. 16, no 6, pp. 4920-4936, 2021. [Index: SCOPUS] https://jestec.taylors.edu.my/Vol%2016%20Issue%206%20December%20%202021/16_6_39.pdf
- [3]. A. Vishnoi, A. Agarwal, A. Prasad, and M. Prateek, "An Approach to Secure Text Transmission for Lower Bandwidth Channel Using Homomorphic Transform," *Turkish Journal of Electrical Engineering Computer Sciences*, 2022. [Index: SCI] {Under Review}
- [4]. A. Vishnoi, A. Agarwal, A. Prasad, and M. Prateek, "Securing Medical data using homomorphic transform," *Int. Journal SSE of Inder-science* [Index: SCOPUS]. {Under Review}
- [5]. A. Vishnoi, A. Agarwal, A. Prasad, and M. Prateek, "An Improved approach to secure Financial Transactions using Homomorphic Transform," *J Arch. Egyptol*, vol. 19, no. 2, 2022. [Index: SCOPUS] {Under Review}

Conferences

- [1]. A. Vishnoi, A. Agarwal, A. Prasad, and M. Prateek, "An Encryption Method Involving Homomorphic Transform", *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), 2021*, pp. 359-363.
doi: 10.1109/CENTCON52345.2021.9688040. [Index: SCOPUS] <https://ieeexplore.ieee.org/abstract/document/9688040>
- [2]. A. Vishnoi, A. Agarwal, A. Prasad and M. Prateek, "Use of Homomorphic Transform in Encryption," *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), 2021*, pp. 1-6.
doi: 10.1109/CSITSS54238.2021.9683207. [Index: SCOPUS] <https://ieeexplore.ieee.org/document/9683207>
- [3]. A. Vishnoi, A. Aggarwal, A. Prasad, M. Prateek, "Text encryption for lower bandwidth channels: Design and implementation:", *3rd IEEE Int. Conf. on Intelligent Computing, Instrumentation and Control Technologies, (ICICT 2022)*, Kerala, August 11-12, 2022. (Accepted for presentation and publication)
- [4]. A. Vishnoi, A. Aggarwal, A. Prasad, M. Prateek, "Image Encryption Using Homomorphic Transform", *3rd IEEE Int. Conf. on Intelligent Computing, Instrumentation and Control Technologies, (ICICT 2022)*, Kerala, August 11-12, 2022. (Accepted for presentation and publication)
- [5]. A. Vishnoi, A. Aggarwal, A. Prasad, M. Prateek, "A Cryptosystem analysis for text messages using Homomorphic Transform", *3rd IEEE Int. Conf. on Intelligent Computing, Instrumentation and Control Technologies, (ICICT 2022)*, Kerala, August 11-12, 2022. (Ac-

cepted for presentation and publication)

- [6]. A. Vishnoi, A. Aggarwal, A. Prasad, M. Prateek, “The improved encryption technique using homomorphic transform”, *3rd IEEE Int. Conf. on Intelligent Computing, Instrumentation and Control Technologies, (ICICT 2022)*, Kerala, August 11-12, 2022. (Accepted for presentation and publication)
- [7]. A. Vishnoi, A. Agarwal, A. Prasad, and M. Prateek, “An improved cryptographic technique using homomorphic transform”, *Int. Conf. on Adv. in Interdisciplinary Research (ICAIR 2021)*, Dr. BR Ambedkar University Agra, Oct. 29-31, 2021.
- [8]. A. Vishnoi, A. Agarwal, A. Prasad, and M. Prateek, “Analysis of Text Encryption techniques based on homomorphic transform”, *Int. Conf. on Adv. in Interdisciplinary Research (ICAIR 2021)*, Dr. BR Ambedkar University Agra, Oct. 29-31, 2021.
- [9]. A. Vishnoi, A. Agarwal, A. Prasad, and M. Prateek, “Design and implementation of text encryption for lower bandwidth channels using homomorphic transform”, *Int. Conf. on Adv. in Interdisciplinary Research (ICAIR 2021)*, Dr. BR Ambedkar University Agra, Oct. 29-31, 2021.
- [10]. A. Vishnoi, A. Agarwal, A. Prasad, and M. Prateek, “A cryptosystem analysis for text messages using homomorphic transform”, *Int. Conf. on Adv. in Interdisciplinary Research (ICAIR 2021)*, Dr. BR Ambedkar University Agra, Oct. 29-31, 2021.

Patents

- [1]. “Secure Multimodal Biometric System and Method Thereof” by A. Vishnoi, D. Sharma, and M. Prateek, (Patent Application no. 20201

ORIGINALITY REPORT

10%

SIMILARITY INDEX

8%

INTERNET SOURCES

5%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1	jestec.taylors.edu.my Internet Source	1%
2	Ankit Vishnoi, Alok Aggarwal, Ajay Prasad, Manish Prateek. "An Encryption Method Involving Homomorphic Transform", 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), 2021 Publication	1%
3	globaljournals.org Internet Source	1%
4	Submitted to Indian Institute of Technology Tirupati Student Paper	<1%
5	Submitted to University of Petroleum and Energy Studies Student Paper	<1%
6	Submitted to Indian Institute of Technology, Madras Student Paper	<1%
