



Sudan University of Science and Technology  
College of Graduate Studies

# Enhanced Three Dimension Playfair Algorithm for Image Encryption

خوارزمية البلايفير ثلاثية الأبعاد المحسنة لتشفير الصورة

A Thesis Submitted in Partial Fulfillment of the Requirement of  
Master Degree in Computer Science

By:

Fatima Abdalla Elhag Adam

Supervision:

Dr. Faisal Mohammed Abdalla Ali

December 2018

## الآية

قال الله تعالى: ( قَالَ الَّذِي عِنْدَهُ عِلْمٌ مِنَ الْكِتَابِ أَنَا آتِيكَ بِهِ قَبْلَ أَنْ يَرْتَدَّ إِلَيْكَ طَرْفُكَ  
فَلَمَّا رَأَهُ مُسْتَقْبِرًا عِنْدَهُ قَالَ هَذَا مِنْ فَضْلِ رَبِّي لِيَبْلُوَنِي أَأَشْكُرُ أَمْ أَكْفُرُ وَمَنْ شَكَرَ فَإِنَّمَا يَشْكُرُ  
لِنَفْسِهِ وَمَنْ كَفَرَ فَإِنَّ رَبِّي غَنِيٌّ كَرِيمٌ ) صدق الله العظيم.

سورة النمل الآية ٤٠

## **Acknowledgements**

Best thanks giving and completed by God Almighty, who created the best everything, which we succeeded in our Almighty what was this research to see the light without the Almighty and his generosity and kindness. And with all the meanings of thanks and sincere gratitude thanks to Dr.Faisal Mohammed Abdalla Ali, who oversaw the research, which we did not spare days in giving us information and tips and all what can benefit from it in the output of this research.

I am also pleased to thank all those who contributed to the output of this work, especially beloved girlfriends Khadiga Mohammed Adam and Mai Omar Elsadeg. We ask Allah Almighty to reward us richly rewarded and keep them stalwarts of science and knowledge.

## **Abstract**

In this research, a new modified version of Tree Dimension Playfair (3D-Playfair) cipher algorithm is introduced to encrypt image. The new method is create (16 X 4 X 4) matrix based on the key being entered by the user to become more secretive, and then manipulate image data the algorithm work on trigraph rather than using digraph, and to increasing complexity of the algorithm the key is used to generate a mask that is subsequently Exclusive OR (XORed) with the scrambled image after encrypt it.

The experimental results showed that the key space of the proposed technique makes it hard for the attacker to perform a frequency analysis based on the used pixel trigraph. Furthermore, further tests showed that a small change in the key value results in completely different cipher images. Peak Signal-to-Noise Ratio (PSNR) values and histogram comparisons were also deployed to show the robustness of the proposed cipher.

## المستخلص

في هذا البحث، يتم تقديم نسخة معدلة جديدة من خوارزمية التشفير الثلاثية الأبعاد لتشفير الصور. تتمثل الطريقة الجديدة في إنشاء مصفوفة (4 X 4 X 16) على أساس المفتاح الذي يتم إدخاله من قبل المستخدم لتصبح أكثر سرية، ومن ثم التعامل مع بيانات الصورة، تعمل الخوارزمية على التشفير الثلاثي بدلاً من استخدام التشفير الثنائي، ولزيادة تعقيد الخوارزمية يستخدم المفتاح لإنشاء قناع في وقت لاحق وإستخدامه مع بوابة الاختيار الحصري والصورة المجمعة بعد تشفيرها. أظهرت النتائج التحليلية أن المفتاح في التقنية المقترحة يجعل من الصعب على المهاجم أن يقوم بتحليل التردد على أساس إستخدام التشفير الثلاثي للصورة. علاوة على ذلك، أظهرت إختبارات أخرى أن تغييراً طفيفاً في قيمة المفتاح يؤدي إلى صور مشفرة مختلفة تماماً. كما تم نشر قيم نسبة الإشارة إلى الضوضاء ونسبة المقارنة بين المدرج التكراري لإظهار قوة التشفير المقترح.

## Table of Contents

الآية.....	I
Acknowledgements.....	II
Abstract.....	III
المستخلص.....	IV
Table of Contents.....	V
List of Tables.....	VI
List of Figures.....	IX
List of Abbreviation.....	X
CHAPTER I	
INTRODCTION.....	1
1.1 Introduction.....	1
1.2 Research Problem.....	1
1.3 Research Significance.....	2
1.4 Proposed Solution.....	2
1.5 Research Objectives.....	2
1.6 Research Methodology.....	2
1.7 Research Scope.....	3
1.8 Research Contents.....	3
CHAPTER II	
LITERATURE REVIEW AND RELATED WORK.....	4
2.1 Introduction.....	4
2.2 Symmetric Encryption.....	5
2.2.1 Common Symmetric Key Algorithms.....	6
2.2.2 Symmetric Strengths and Weaknesses.....	8
2.3 Asymmetric Encryption.....	8

2.3.1 Common Public Key Algorithms.....	9
2.3.2 Uses for Public Key Encryption.....	9
2.3.3 Asymmetric Strengths and Weaknesses.....	10
2.4 Cryptographic Goals.....	10
2.5 Playfair Cipher.....	10
2.5.1 Key Matrix Generation.....	11
2.5.2 Encryption.....	11
2.5.3 Decryption.....	12
2.5.4 Limitations of classic Playfair.....	12
2.6 3D-Playfair Cipher.....	13
2.6.1 Key-Matrix Generation.....	13
2.6.2 Encryption.....	14
2.6.3 Decryption.....	15
2.6.4 Properties of 3D-Playfair Cipher.....	16
2.7 Related Work.....	16
CHAPTER III	
RESEARCH METHODOLOGY.....	19
3.1 Introduction.....	19
3.2 Digital Images.....	19
3.2.1 Type of Digital Images.....	19
3.3 Requirements of Images Encryption.....	20
3.4 Proposed Method.....	20
3.4.1 Key Matrix Generation Algorithm.....	20
3.4.2 Image Encryption Algorithm.....	24
3.4.3 Image Decryption Algorithm.....	26
3.5 Techniques and Tools.....	28
3.5.1 Java.....	28
3.5.2 Advantages of Java.....	28

3.5.3 Disadvantages of Java.....	29
3.5.4 MATLAB.....	29
3.5.5 Advantages of MATLAB.....	29
3.5.6 Disadvantages of MATLAB.....	29
CHAPTER IV	
IMPLEMENTATION, RESULTS AND DISCUSSIONS.....	30
4.1 Implementation.....	30
4.2 Results and Discussion.....	39
4.2.1 Key Space Analysis.....	39
4.2.2 Key Sensitivity Test.....	39
4.2.3 Visual Diffusion Test.....	41
4.2.4 Histogram.....	44
4.3 Security Aspects of Cipher.....	47
4.3.1 Brute Force Attack.....	47
4.3.2 Frequency Analysis.....	47
4.3.3 Confusion and Diffusion.....	47
CHAPTER V	
CONCLUSIONS AND RECOMMENDATIONS.....	48
5.1 Conclusion.....	48
5.2 Recommendations.....	48
References.....	49



## List of Tables

Table 2.1: Symmetric key algorithms.....	6
Table 2.2: Playfair (5 X 5) matrix, Key = null.....	11
Table 2.3: Playfair 5 X 5 matrix. Key = simple.....	11
Table 2.4: 3D-Playfair 4 X 4 X 4 matrix. Key = null.....	14
Table 2.5: 3D-Playfair (4 X 4 X 4) matrix. Key = FRIENDS4V@TJ_201.C.....	14
Table 2.6: Encryption Process of 3D-Playfair (4 X 4 X 4) matrix.....	15
Table 2.7: Decryption Process of 3D-Playfair (4 X 4 X 4) matrix.....	15
Table 3.1: Keyword= dsf:j5%2@9pLJFSWRY3H*.....	21
Table 3.2: 3D-Playfair 16 X 4 X 4 matrix. secret key = null.....	22
Table 3.3: 3D-Playfair 16 X 4 X 4 matrix, secret key = dsf:j5%2@9pLJFSWRY32H*.....	23
Table 3.4: Encryption Process of 3D-Playfair (16 X 4 X 4) matrix.....	24
Table 3.5: Decryption Process of 3D-Playfair (16 X 4 X 4) matrix.....	26
Table 4.1: PSNR, MSE values for original images, cipher images and reconstructed image.....	42
Table 4.2: PSNR, MSE values for original Images and various cipher images using different secret keys.....	43
Table 4.3: Histogram comparison between the original and cipher images use different color.....	44
Table 4.4: Original image and cipher image histograms using same key.....	45

## List of Figures

Figure 2.1: The encryption and decryption processes of cipher.....	4
Figure 2.2: Symmetric encryption process.....	5
Figure 2.3: Asymmetric Encryption.....	8
Figure 3.1: The Key-Matrix generation process.....	21
Figure 3.2: Show the process of encryption algorithm.....	25
Figure 3.3: Show the process of decryption algorithm.....	27
Figure 4.1: System home screen.....	30
Figure 4.2: Key matrix, secret key = maxk(123*.....	31
Figure 4.3: Graphical user interface for encryption process.....	32
Figure 4.4: Encryption process whiles it done.....	33
Figure 4.5: Image after encryption process (encrypted image).....	34
Figure 4.6: Graphical user interface for decryption process.....	35
Figure 4.7: Decryption process whiles it done.....	36
Figure 4.8: Image after decryption process (reconstructed image).....	37
Figure 4.9: The result of applying 3D-Playfair encryption for images sample.....	38
Figure 4.10: The original Peppers and Ciphered images using different keys.....	40
Figure 4.11: The ciphered Peppers and reconstructed images using different keys	40

## List of Abbreviation

3D-Playfair	Tree Dimension Playfair	III
XOR	Exclusive OR	III
AES	Advanced Encryption Standard	1
RC4	Rivest Cipher 4	1
3DES	Triple Data Encryption Algorithm	1
DES	Data Encryption Algorithm	6
IDEA	International Data Encryption Algorithm	6
MARS	multivariate adaptive regression splines	6
MD5	message-digest	7
SHA 1	Secure Hash Algorithm 1	7
HMAC	Hash based Message Authentication Code	7
DSS	Digital Signature Standards	9
FIPS	Federal Information Processing Standard	9
DSA	Digital Signature Algorithm	9

# CHAPTER I

## INTRODCTION

### 1.1 Introduction

As the general public became more aware of cryptographic uses, the personal and social need for privacy is increased. Nowadays, cryptography referred almost exclusively to encryption, which is the process of converting ordinary information (unencrypted message) into unintelligible gibberish (encrypted message).

There are several ways of classifying cryptographic algorithms, one of these ways is to be categorized based on the number of keys that are employed for encryption and decryption, based on this criterion there are three types of encryption systems: Symmetric key, Public key and Hash functions [1]. Examples of popular and well-respected cryptographic algorithms include Advanced Encryption Standard (AES) [2], Rivest Cipher 4 (RC4) and Triple Data Encryption Algorithm (3DES) [1].

Among the classic techniques, Playfair cipher was the best know multiple-letter encryption cipher, which treats digraphs in plain text as a single unit and translates it into cipher text, this cipher was invented by the British scientist Sir Charles Wheatstone in 1854.

One of the extends of traditional Playfair cipher in addition to encrypt images in safer manner, is method created matrix of (16 X 16) based on the key being entered by the user to become more secretive, to make it compatible with image data content [8].

The classic 3D-Playfair cipher is enhance of classic Playfair, which encrypts a trigraph of plain text into its corresponding cipher text trigraph, using (4 X 4 X 4) matrix to store 26 alphabets, 10 numerals and 28 special symbols [3]. The proposed method, is enhanced 3D-Playfair for encrypt images.

### 1.2 Research Problem

The drawback in classic Playfair cipher is that a digraph and its reverse will encrypt in the same fashion, and the classic 3D-Playfair cipher is not compatible with image data content (can't encrypt image) [5].

### **1.3 Research Significance**

Nowadays, information security is becoming more important in data storage and transmission to protect image contents:

1. Enhances the security by increasing complexity.
2. Images are wider used in different-different processes, therefore the security of image data from unauthorized uses is important.
3. Image encryption plays an important role in the field of information hiding.
4. Image encryption method makes information unreadable. Therefore, no hacker or eavesdropper, including sewer administrators and others, have access to original message or any other type of transmitted information through public networks such as internet.

### **1.4 Proposed Solution**

This research proposed 3D-Playfair Cipher (16 X 4 X 4) version, which works on trigraph rather than using digraph which eliminates the fact that a digraph and its reverse will encrypt in a similar fashion, by using (16 X 4 X 4) matrix it take into consideration the algorithm is compatible with image data content for encryption.

### **1.5 Research Objectives**

1. The 3D-Playfair encryption process can be applied on three pixels color components in the original colored image data.
2. Using modified 3D-Playfair which makes the algorithm complex and have high rate of confusion and diffusion hard for applying brute force attack on.

### **1.6 Research Methodology**

The 3D-Playfair cipher has mainly three algorithms: Key Matrix Generation, Encryption and Decryption. Analyzing colored image data to three color components: Red, Green and Blue, their values range is between 0 and 255. So, using (16 X 4 X 4) matrix to fill the values between 0 and 255, then using 3D-Playfair cipher for encrypt and decrypt image. Use Java for implementation code, and MATLAB for analysis.

## **1.7 Research Scope**

Implementation enhancement 3D-Playfair cipher algorithm (16 X4 X 4) as improve for combine algorithms 3D-Playfair (4 X 4 X 4) and Playfair (16 X 16) to encrypted and decrypted colored images (data RGB image).

## **1.8 Research Contents**

This research contains the following chapters:

**CHAPTER I:** The first chapter contains a general introduction to research, research problem, significance of research, proposed solution, research objectives, methodology and the scope of research.

**CHAPTER II:** The second chapter contains literature review and related work.

**CHAPTER III:** The third chapter contains the algorithm used to know the diseases under study and explain how they work, and techniques that were used in the research and advantages of each technique contain.

**CHAPTER IV:** Chapter four contains system implementation, result and discussion.

**CHAPTER V:** Chapter five contains conclusion and recommendations.

## CHAPTER II

### LITERATURE REVIEW AND RELATED WORK

#### 2.1 Introduction

The word cryptography has come from a Greek word, which means secret writing, in the present day it refers to the tools and techniques used to make messages secure for communication between the participants and make messages immune to attacks by hackers, for private communication through public network, cryptography plays a very crucial role.

Encryption is a modern form of cryptography that allows a user to hide information from others, and is one of the most important methods for providing data security, especially for end-to-end protection of data transmitted across networks, which plain text or any other type of data is converted from a readable form to an encoded version that can only be decoded by another entity if they have access to a decryption key.

Encryption process use encryption algorithm and encryption key to translating plain text data into cipher text that is unreadable by other, decryption process use decryption algorithm and decryption key to converting cipher text back to plain text as shown in Figure 2.1.

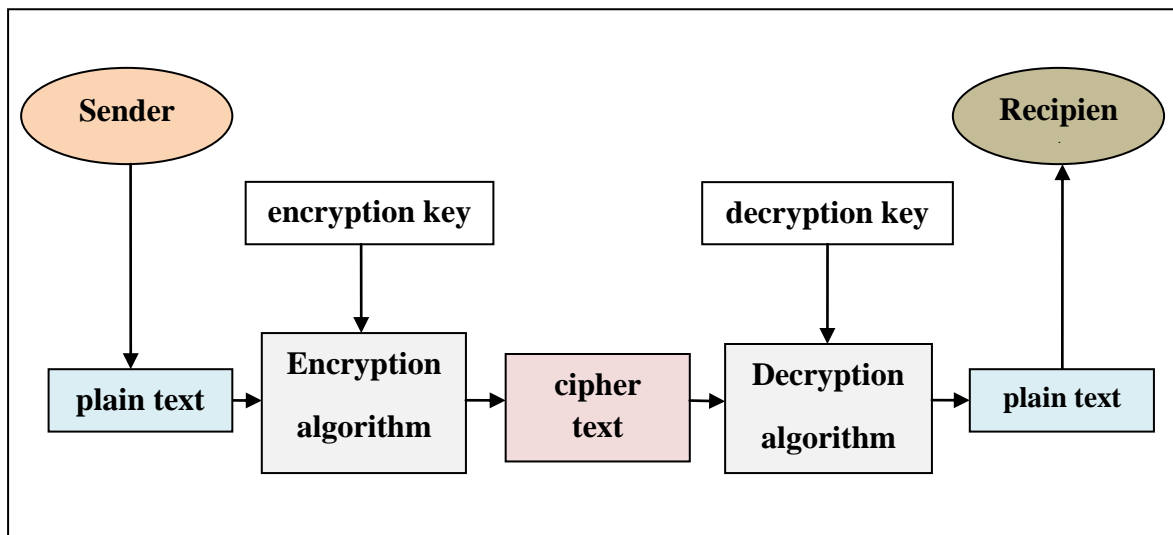


Figure 2.1: The encryption and decryption processes of cipher.

Two of the most widely used encryption methods are Private key (symmetric) encryption and Public key (asymmetric) encryption, the two are similar in the sense that they both allow a user to encrypt data to hide it from others, and then decrypt it in order to access the original plaintext, they differ in how they handle the steps between encryption and decryption [3].

## 2.2 Symmetric Encryption

This system uses only private keys, which can be anything from a numerical symbol to a string of random letters, these private keys are used to encode a message, so that only the sender and the recipient of the message who know what the secret key is can unlock it and decrypt it, the system works pretty much like two best friends using a decoder ring to send secret messages to each other. The symmetric system's only downside is the potentially unsafe private key transmission via the Internet, where other people can crack it and decode the message [3].

In symmetric key cryptography same key is shared, the same key is used in both encryption and decryption as shown in Figure 2.2, symmetric keys are also called secret keys because this type of encryption relies on each user to keep the key secret and properly protected. If this key got into an intruder's hand, that intruder would have the ability to decrypt any intercepted message encrypted with this key.

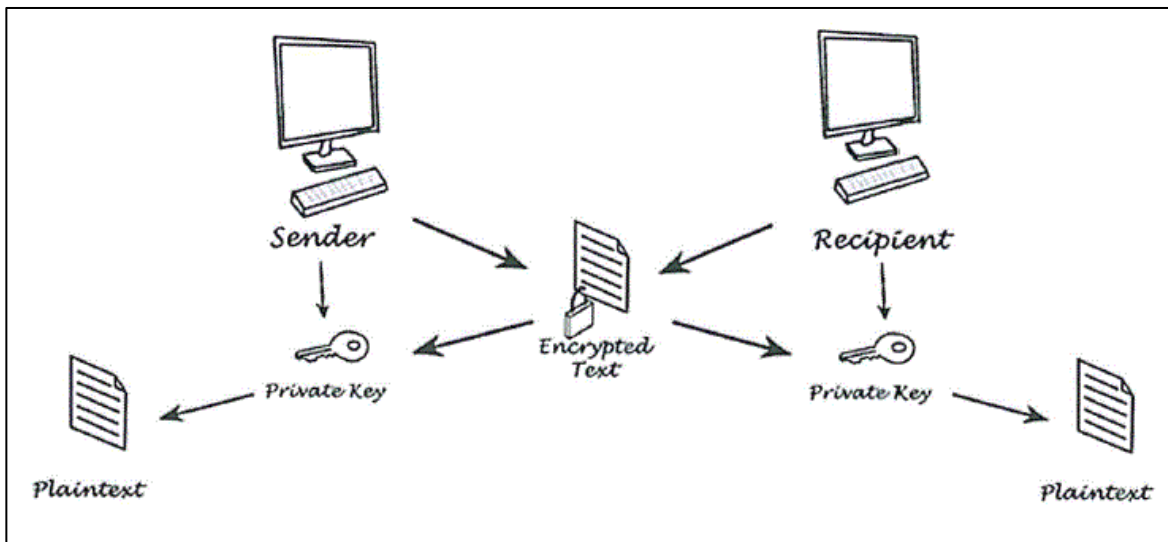


Figure 2.2: Symmetric encryption process



The algorithm used to decrypt is just the inverse of the algorithm used for encryption, for example, if addition and division is used for encryption, multiplication and subtraction are to be used for decryption.

### 2.2.1 Common Symmetric Key Algorithms

There are many symmetric key algorithms use today, shown in Table 2.1 [13]:

Table 2.1: Symmetric key algorithms.

Algorithm	Description	Key Length	Rating
Blowfish	Block cipher developed by Schneier	1-448 bits	Λ
DES	DES adopted as a U.S. government standard in 1977	56 bits	§
IDEA	Block cipher developed by Massey and Xuejia	128 bits	Λ
MARS	AES finalist developed by IBM	128-256 bits	∅
RC2	Block cipher developed by Rivest	1-2048 bits	Ω
RC4	Stream cipher developed by Rivest	1-2048 bits	Λ, §
RC5	Block cipher developed by Rivest and published in 1994	128-256 bits	∅
RC6	AES finalist developed by RSA Labs	128-256 bits	∅
Rijndael	NIST selection for AES, developed by Daemen and Rijmen	128-256 bits	Ω
Serpent	AES finalist developed by Anderson, Biham, and Knudsen	128-256 bits	∅
Triple-DES	A three-fold application of the DES algorithm	168 bits	Λ
Twofish	AES candidate developed by Schneier	128-256 bits	∅

Key to ratings:

Ω) Excellent algorithm. This algorithm is widely used and is believed to be secure, provided that keys of sufficient length are used.

Λ) Algorithm appears strong but is being phased out for other algorithms that are faster or thought to be more secure.

∅) Algorithm appears to be strong but will not be widely deployed because it was not chosen as the AES standard.

§) Use of this algorithm is no longer recommended because of short key length or mathematical weaknesses. Data encrypted with this algorithm should be reasonably secure from casual browsing, but would not withstand a determined attack by a moderately-funded attacker.

Some of the algorithms that are commonly encountered in the field of computer security are summarized in the following list [14]:

**DES/3DES or TripleDES:** This is an encryption algorithm called Data Encryption Standard that was first used by the U.S. Government in the late 70's. It is commonly used in ATM machines (to encrypt PINs) and is utilized in UNIX password encryption. Triple DES or 3DES has replaced the older versions as a more secure method of encryption, as it encrypts data three times and uses a different key for at least one of the versions.

**Blowfish:** Blowfish is a symmetric block cipher that is unpatented and free to use. It was developed by Bruce Schneier and introduced in 1993.

**AES:** Advanced Encryption Standard or Rijndael; it uses the Rijndael block cipher approved by the National Institute of Standards and Technology (NIST). AES was originated by cryptographers Joan Daemen and Vincent Rijmen and replaced DES as the U.S. Government encryption technique in 2000.

**Twofish:** Twofish is a block cipher designed by Counterpane Labs. It was one of the five Advanced Encryption Standard (AES) finalists and is unpatented and open source.

**IDEA:** This encryption algorithm was used in Pretty Good Privacy (PGP) Version 2 and is an optional algorithm in OpenPGP. IDEA features 64 bit blocks with a 128 bit key.

**MD5:** MD5 was developed by Professor Ronald Rivest and it used to create digital signatures, it is one way hash function and intended for 32 bit machines, it replaced the MD4 algorithm.

**SHA 1:** SHA 1 is a hashing algorithm similar to MD5, yet SHA 1 may replace MD5 since it offers more security

**HMAC:** This is a hashing method similar to MD5 and SHA 1, sometimes referred to as HMAC MD5 and HMAC SHA1.

**RSA Security RC4:** RC4 is a variable key size stream cipher based on the use of a random permutation.

**RC5:** This is a parameterized algorithm with a variable block, key size and number of rounds.

**RC6:** This evolution of RC5, it is also a parameterized algorithm that has variable block, key and a number of rounds. This algorithm has integer multiplication and 4 bit working registers.

## 2.2.2 Symmetric Strengths and Weaknesses

Symmetric key cryptography algorithms have some major advantages, it requiring lesser execution time, it is much faster than asymmetric systems, there are commonly used for long messages, and hard to break if using a large key size. However, these algorithms suffer from the limitations distribution of keys among the users in a secured manner is difficult scalability each pair of users needs a unique pair of keys, so the number of keys grows exponentially.

## 2.3 Asymmetric Encryption

As a solution for the not completely safe symmetric encryption, there is the asymmetric encryption system that uses a pair of keys for added security a private and a public key, the private key is for and the public key is published online for others to see, as shown in Figure 2.3. The public key is used to access the encryption code that corresponds to private key.

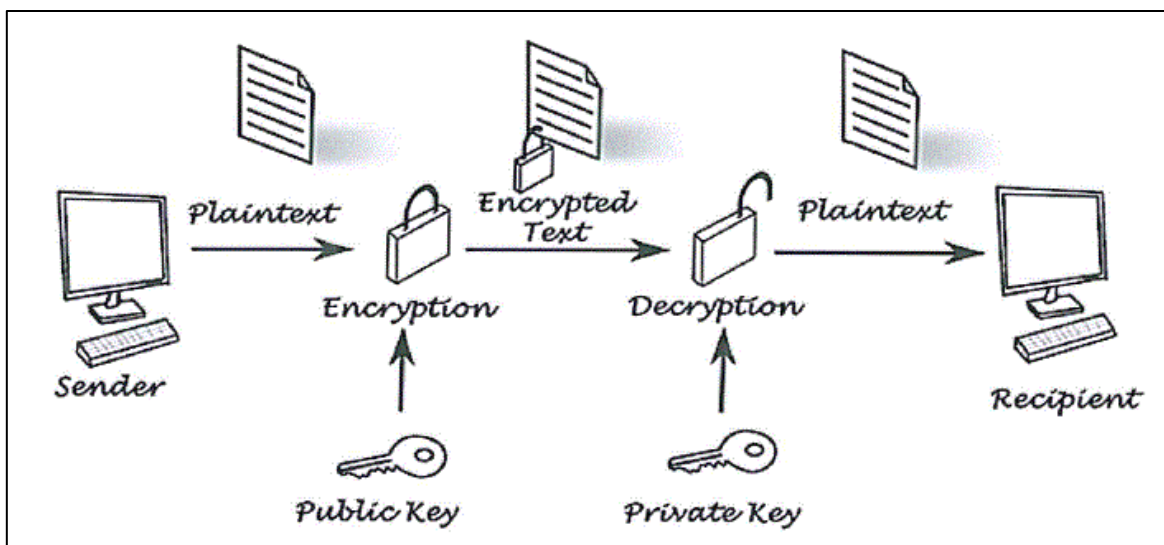


Figure 2.3: Asymmetric Encryption.

So, if you are sending an encrypted message to Susan which do not want others to see, would use her public key to encrypt it. She will be able to decrypt it with her own corresponding private key. And if send message to you, she use your public key to encrypt the message and you would use your private key to decrypt it.

### 2.3.1 Common Public Key Algorithms

The following list summarizes the public key systems common use today: [13].

**Diffie-Hellman key exchange:** A system for exchanging cryptographic keys between active parties, Diffie-Hellman is not actually a method of encryption and decryption, but a method of developing and exchanging a shared private key over a public communications channel. In effect, the two parties agree to some common numerical values, and then each party creates a key. Mathematical transformations of the keys are exchanged. Each party can then calculate a third session key that cannot easily be derived by an attacker who knows both exchanged values.

**DSA/DSS:** The Digital Signature Standard (DSS) was developed by the U.S. National Security Agency and adopted as a Federal Information Processing Standard (FIPS) by the National Institute for Standards and Technology. DSS is based on the Digital Signature Algorithm (DSA). Although DSA allows keys of any length, only keys between 512 and 1,024 bits are permitted under the DSS FIPS. As specified, DSS can be used only for digital signatures, although it is possible to use some DSA implementations for encryption as well.

**RSA:** RSA is a well-known public key cryptography system developed in 1977 by three professors at MIT: Ronald Rivest, Adi Shamir, and Leonard Adleman. RSA can be used both for encrypting information and as the basis of a digital signature system. Digital signatures can be used to prove the authorship and authenticity of digital information. The key can be any length, depending on the particular implementation used.

### 2.3.2 Uses for Public Key Encryption

Two of the most common uses for public key cryptography are encrypted messaging and digital signatures, with encrypted messaging, a person who wishes to send an encrypted message to a particular recipient encrypts that message with the individual's public key, the message can then be decrypted only by the authorized recipient. With digital signatures, the sender of the message uses the public key algorithm and a private key to digitally sign a message, anyone who receives the message can then validate the authenticity of the message by verifying the signature with the sender's public key.

### 2.3.3 Asymmetric Strengths and Weaknesses

Asymmetric key cryptography algorithms have some major advantages, better key distribution than symmetric systems, and better scalability than symmetric systems, can provide confidentiality, authentication, and non repudiation. . However, these algorithms suffer from the limitations, it works much slower than symmetric systems.

## 2.4 Cryptographic Goals

**Confidentiality** is a service used to keep the content of information from all but those authorized to have it, Secrecy is a term synonymous with confidentiality and privacy There are numerous approaches to providing confidentiality, ranging from physical protection to mathematical algorithms which render data unintelligible.

**Data integrity** is a service which addresses the unauthorized alteration of data, to assure data integrity one must have the ability to detect data manipulation by unauthorized parties, data manipulation includes such things as insertion, deletion and substitution.

**Authentication** is a service related to identification, this function applies to both entities and information it self, two parties entering into a communication should identify each other, information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. For these reasons this aspect of cryptography is usually subdivided into two major classes' entity authentication and data origin authentication, data origin authentication implicitly provides data integrity.

**Non repudiation** is a service which prevents an entity from denying previous commitments or actions, when disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary.

## 2.5 Playfair Cipher

The Playfair cipher or Playfair square is a manual symmetric encryption technique and was the first literal digraph substitution cipher, the scheme was invented in 1854 by Charles Wheatstone, but bears the name of Lord Playfair who promoted the use of the cipher [9]. There are several minor variations of the original Playfair cipher.

The Playfair cipher uses (5 X 5) matrix containing a keyword or phrase, memorization of the keyword and 4 simple rules was all that was required to create the (5

X 5) matrix and use the cipher, by assuming a null key the (5 X 5) matrix will be as following in Table 2.2.

Table 2.2: Playfair (5 X 5) matrix, Key = null.

a	B	C	d	e
f	G	H	i/j	k
l	M	N	o	p
q	R	S	t	u
v	W	X	y	z

Playfair cipher has mainly three algorithms, Key Matrix Generation, Encryption and Decryption. These are described below:

### 2.5.1 Key Matrix Generation

To generate the key matrix, one would first fill in the spaces in the matrix with the letters of the keyword (dropping any duplicate letters), then fill the remaining spaces with the rest of the letters of the alphabet in order (usually omitting "J" or "Q" to reduce the alphabet to fit; other versions put both "I" and "J" in the same space), the key can be written in the top rows of the matrix, from left to right, or in some other pattern, such as a spiral beginning in the upper-left-hand corner and ending in the center. The keyword together with the conventions for filling in the (5 X 5) matrix constitutes the cipher key as shown in Table 2.3.

Table 2.3: Playfair 5 X 5 matrix, Key = simple.

s	i/j	m	p	l
e	a	b	c	d
f	g	h	k	n
o	q	r	t	u
v	w	x	y	z

### 2.5.2 Encryption

To encrypt a message, one would break the message into digraphs (groups of 2 letters) such that, for example, "HelloWorld" becomes "HE LL OW OR LD", these digraphs will be substituted using the key table. Since encryption requires pairs of letters, messages with an odd number of characters usually append an uncommon letter, such as

"X", to complete the final digraph, the two letters of the digraph are considered opposite corners of a rectangle in the key table. To perform the substitution, apply the following 4 rules, in order, to each pair of letters in the plaintext:

1. If both letters are the same (or only one letter is left), add an "X" after the first letter, encrypt the new pair and continue, some variants of Playfair use "Q" instead of "X", but any letter, itself uncommon as a repeated pair, will do.
2. If the letters appear on the same row of your table, replace them with the letters to their immediate right respectively (wrapping around to the left side of the row if a letter in the original pair was on the right side of the row).
3. If the letters appear on the same column of your table, replace them with the letters immediately below respectively (wrapping around to the top side of the column if a letter in the original pair was on the bottom side of the column).
4. If the letters are not on the same row or column, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair, the order is important the first letter of the encrypted pair is the one that lies on the same row as the first letter of the plaintext pair.

### **2.5.3 Decryption**

To decrypt a message, use the inverse (opposite) of the last 3 rules, and the first as is (dropping any extra "X"s or "Q"s that do not make sense in the final message when finished).

### **2.5.4 Limitations of classic Playfair**

The existing Playfair technique is based on the use of a (5 X 5) matrix of letters constructed using a keyword, this algorithm can only allow the text that contains alphabets only.

The (5 X 5) matrix can only allow 25 characters, hence the letters I/J count as one, if we encrypt the plaintext which is having the letter I/J and when we decrypt the cipher text at the receive end, the receiver will be under ambiguity whether to consider I or J in his text, because the meaning can be changed with the change of the letters, this algorithm can

only useful for the plain text containing of alphabets but it is failed for the plain text containing of alphanumeric values.

The algorithm encrypts pairs of letters (digraphs), the cryptanalysis of the Playfair cipher is also aided by the fact that a diagram and its reverse will encrypt in a similar fashion. That is, if AB encrypts to XY, then BA will encrypt to YX [3, 4]. So by looking for words that begin and end in reversed diagrams, one can try to compare them with plaintext words that are similar.

## **2.6 3D-Playfair Cipher**

Digraphs (pairs of letters) are being substituted instead of monographs (single letters) in Playfair and this method proposed trigraph (pairs of 3 letters), so this is more complex than playfair cipher. 3D-Playfair cipher is the multiple letter encryption cipher, which encrypts a trigraph of plaintext into corresponding cipher text trigraph, it requires (4 X 4 X 4) matrix to store 26 alphabets, 10 numerals and 28 special symbols, and these letters are arranged in (4 X 4 X 4) matrix based on secret key.

3D-Playfair cipher has mainly three algorithms, Key-Matrix Generation, Encryption and Decryption. These are described below:

### **2.6.1 Key-Matrix Generation**

3D-Playfair Cipher makes use of (4 X 4 X 4) matrix, which is used to store a keyword that becomes the key for encryption and decryption, storing keyword into (4 X 4 X 4) matrix is based on some simple rules as below:

1. Enter the secret (password) which may contain numerals, alphabets and special symbols like: aman2012nitj@gmail.com, cipher, make, fatima, like, 29101989, ravindra\_1987\_singh@nitj.ac.in etc.
2. Find out the keyword by dropping the duplicate letters of key, for example: amn201itj@g.com, ravind\_1987sgh@tj.c, cipher, 29108.
3. Arrange the keyword in 4 matrix, row-wise: left to right and then top-to-bottom.
4. Fill the remaining spaces in the matrix with the rest of numerals (0-9), alphabets (A-Z), special symbols that were not the part of our keyword.



By assuming a null key the (4 X 4 X 4) matrix will be arrange as following in Table 2.4 (Sequence of 64 characters):

Table 2.4: 3D-Playfair 4 X 4 X 4 matrix. Key = null.

<b>Matrix 1</b>	<b>Matrix 2</b>
0   1   2   3	G   H   I   J
4   5   6   7	K   L   M   N
8   9   A   B	O   P   Q   R
C   D   E   F	S   T   U   V
<b>Matrix 3</b>	<b>Matrix 4</b>
W   X   Y   Z	-   .   /   :
!   “   #   \$	;   <   =   >
%   &   ‘   (	?   @   [   \
)   *   +   ,	]   ^   _

For example if the secret is FRIENDS4EVER@NITJ\_2012.CSE, keyword will be FRIENDS4V@TJ\_201.C and Key-Matrix will be as following in Table 2.5.

Table 2.5: 3D-Playfair (4 X 4 X 4) matrix. Key = FRIENDS4V@TJ\_201.C.

<b>Matrix 1</b>	<b>Matrix 2</b>
F   R   I   E	.   C   3   5
N   D   S   4	6   7   8   9
V   @   T   J	A   B   G   H
_   2   0   1	K   J   M   O
<b>Matrix 3</b>	<b>Matrix 4</b>
P   Q   U   W	*   +   ,   -
X   Y   Z   !	/   :   ;   <
“   #   \$   %	=   >   ?   [
&   ‘   (   )	\   ]   ^

### 2.6.2 Encryption

To encrypt a message, one would break the message into trigraph (groups of 3 letters), if any two letters are the same or only one letter is left, add two filler letter X and Z after the first letter in the trigraph. And if any two letter is left, add a filler X after the second letter. So that BALLOON would be treated as {BAL}, {LOX}, {ONX}, and

HELLOWORLDS would be treated as {HEL}, {LOW}, {ORL}, {DSX} and MASTI\_M.TECH @NITJ.2012 would be treated as {MAS}, {TL\_}, {M.T}, {ECH}, {@NI}, {TJ.}, {201}, {2XZ}.

A letter in the trigraph will be replaced by the letter that will lay on the same row of the letter and the column of the next letter and at the matrix of next-to-next letter in circular fashion, this approach can be better understand by the Table 2.6.

Table 2.6: Encryption Process of 3D-Playfair (4 X 4 X 4) matrix.

Plain Text Trigraph	Plain Text Trigraph			Cipher Text Trigraph
	1st Letter	2nd Letter	3rd letter	
1st Letter	Row	Column	Matrix	1st Letter
2nd Letter	Matrix	Row	Column	2nd Letter
3rd letter	Column	Matrix	Row	3rd letter

Circular fashion means if we consider 1st letter for encryption then 2nd letter will be the next letter and 3rd letter will be the next-to-next letter and if we consider 2nd letter for encryption then 3rd letter will be the next letter and 1st letter will be the next-to-next letter and if we consider 3rd letter for encryption then 1st letter will be the next letter and 2nd letter will be the next-to-next letter.

### 2.6.3 Decryption

A letter in the trigraph will be replaced by the letter that will lay on the same row of the letter and at the floor of the next letter and the column of next-to-next letter in circular fashion. This approach can be better understand by the Table 2.7.

Remove the filler letter from the trigraph (Dropping any extra X and Z that don't make sense in the final message when finished) to find out the actual text (plaintext).

Table 2.7: Decryption Process of 3D-Playfair (4 X 4 X 4) matrix.

Plain Text Trigraph	Plain Text Trigraph			Cipher Text Trigraph
	1st Letter	2nd Letter	3rd letter	
1st Letter	Row	Matrix	Column	1st Letter
2nd Letter	Column	Row	Matrix	2nd Letter
3rd letter	Matrix	Column	Row	3rd letter

### 2.6.4 Properties of 3D-Playfair Cipher [3].

3D-Playfair cipher holds these properties for its strength over classical Playfair cipher as following:

- 3D-Playfair cipher shows a great advancement over the mono alphabetic ciphers.
- Like classical Playfair cipher, 3D-Playfair cipher is not case sensitive.
- It uses trigraph rather than using digraph, so the length of plaintext may be even or odd, That's why it is very hard to determine the actual length of plaintext.
- It removes the drawback of diagram & its reverse encryption attack (Chosen Plaintext Attack), by using trigraph 3D-Playfair Cipher eliminates this security loophole of classical Playfair cipher.
- Classical Playfair cipher supports only 25 English alphabets but 3D-Playfair cipher supports all 26 alphabets (including "i" and "j" both), 10 numerals and 28 frequently used special symbols.
- The identification of trigrams is more difficult than individual letters or digraphs. In the mono alphabetic cipher, the attacker searches in 26 letters only while in classical Playfair cipher an attacker has to search in  $26 \times 26 = 676$  digraphs. But by using the 3D-Playfair cipher, the attacker has to search in  $64 \times 16 \times 4 = 4096$  trigraph.

### 2.7 Related Work

Numerous researchers attempt to enhance the Playfair and 3D-Playfair to create variant algorithms.

**3D (4 X 4 X 4) - Playfair Cipher [3].** This research proposed 3D-Playfair Cipher (4 X 4 X 4 Playfair cipher) which works on trigraph rather than using digraph. 3D-Playfair cipher supports all 26 alphabets {A-Z}, 10 digits {0-9} and 28 special characters { ! " # \$ % & ' ( ) \* + , - . / : ; < = > ? @ [ \ ] ^ \_ | }.encrypt all alphabets, numerals and most commonly used special symbols. It uses trigraph rather than using digraph, by using trigraph and ( 4 X 4 X 4 ) matrix it provides high rate of confusion and diffusion rate, there is  $64 \times 16 \times 4 = 4096$  possible trigraph so it is too hard for applying brute force attack on it.

BY uses trigraph rather than using digraph to eliminate the fact that a diagram and its reverse will encrypt in a similar fashion, this cipher is not vulnerable to security attacks. by using trigraph and  $4 \times 4 \times 4$  matrix it provides high rate of confusion and diffusion rate, there is  $64 \times 16 \times 4 = 4096$  possible trigraph so it is too hard for applying brute force attack on it. It works on 64 characters so the probability of occurrence of a character in 3D-Playfair matrix is  $1/16 * 1/4 = 1/64 = 0.0156$ . But it is not case sensitive because it contain only capital letters and some spatial characters.

**A Modified Playfair Cipher for Encrypting Digital Images [4].** in this paper, a new extension of the Playfair cipher algorithm is proposed to encrypt image data more securely. The proposed method constructs a  $(16 \times 16)$  secret key matrix to scramble image data byte by byte. In addition, the algorithm complexity is increased using masking and XOR procedure. That is, the key is used to generate a mask that is subsequently XORed with the scrambled image.

The experimental results showed that the key space of the proposed technique makes it hard for the attacker to perform a frequency analysis based on the used pixel digraphs, and the tests showed that a small change in the key value results in completely different cipher-images.

But digraph and its reverse will encrypt in the same fashion.

**3D - Playfair Cipher with Message Integrity using MD5 [5].** This paper based on two steps securing the message first the plaintext encryption using 3D- Playfair algorithm and generate cipher text and in the second step you create digital signature again cipher text using MD5 algorithm for integrity of the message during transmission.

This cipher is not susceptible to security attacks, by using trigraph and  $(6 \times 4 \times 4)$  matrix it provides high rate of confusion and diffusion, there is  $96 \times 16 \times 4 = 9216$  possible trigraph so it is too hard for applying brute force attack on it.

The key is not contain all RBG value.

**Enhanced Playfair Cipher for Image Encryption using Integer Wavelet Transform [6]**

This paper analyses the performances of four novel, lossless methodologies of implementing the enhanced Playfair cipher in spatial and frequency domains, with integer wavelet transform (IWT) through lifting scheme.

The proposed methods involve a spatial domain algorithm being applied to the frequency domain of the images for encryption which gives out better desired error metrics.

But digraph and its reverse will encrypt in the same fashion.

**Implementation of 3D Approach in Playfair Cipher [7].** 3D playfair cipher is upgraded version of simple 2D playfair cipher, here we are using 125 characters (26 Lower case alphabets, upper case alphabets, 10 digits, 63 special characters). Here we are using 5 matrices of (5 X 5), all matrices are overlap to each other in a 3D manner, we are arranging 125 characters in 5\*5 matrix.

Cryptanalysis of the 3D-Playfair cipher is much more difficult than normal simple substitution ciphers, in 3D 6\*6\*3 Playfair cipher 15625 combinations of trigraph are possible. Here it gives a high rate of confusion and there is 15625 combination possible and very hard to analyze brute force attack and the probability of occurrence of characters is very less. It is  $\frac{1}{25} \times \frac{1}{5} = 0.008$ , its complexity is very high because we use 26 upper and 26 lower characters, 10 numbers and 63 special characters for encryption and decryption.

But the key is not contain all RGB value.

**Modified Playfair Cipher for Encrypting Images [8].** the Playfair cipher algorithm to encrypt images, the new method is created matrix of 16 X 16 based on the key being entered by the user to become more secretive, and to increasing complexity of the algorithm using masking and an XOR. That is, the key is used to generate XORed with the image to encrypt it.

The results showed that the use of slightly different secret keys, and the resulting encoded images makes it a completely different picture, and also encrypts the data that contain alphanumeric characters, integers, and most symbols.

But digraph and its reverse will encrypt in the same fashion.

## CHAPTER III

### RESEARCH METHODOLOGY

#### 3.1 Introduction

In this research, an enhanced 3D-Playfair cipher is introduced in order to apply it on image. It is known that any image is matrix of pixels, each pixel is components of three color Red, Green and Blue, their values range is between 0 and 255, to achieve this type of encryption we are need to using a matrix of size (16 X 4 X 4) filled with values between 0 and 255 can be a perfect solution to encrypt color values directly into other intensity levels, the key is expected to be sequence of integer numbers between 0 and 255 in a random order use to construct matrix, then the 3D-Playfair encryption process can be applied on three of the pixels color components in the plain image, the resultant scrambled image is not the final output yet, the proposed system adopts an XOR operation as an additional step to improve security, the secret key is used once more to generate a random mask, this random mask is XORed with the scrambled image in order to produce the cipher-image, this additional step process guarantees that the resultant cipher image is completely different from the plain image even if two similar keys were used.

#### 3.2 Digital Images

Digital image is electronic snapshots taken of a scene or scanned from documents, such as photographs, manuscripts, printed texts, and artwork, the digital image in the computer is an array of numbers that represent light intensities at various points (pixels). Each pixel is assigned a tonal value (black, white, shades of gray or color).

##### 3.2.1 Type of Digital Images

1. **Binary image** is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black and white (0, 1).
2. **Grayscale image** is an image in which the value of each pixel represent also as black-and-white, are composed exclusively of shades of gray, varying from black and white.

3. **Color images** are stored in either 24-bit (true color images) or 8-bit per pixel files. A common image size is  $640 \times 480$  pixels and 256 colors (or 8 bits per pixel) Represent colors RGB.

### 3.3 Requirements of Images Encryption

1. Ability to get the pixels of the original image.
2. Create a strong encryption image such that it cannot be hacked easily.
3. Faster encryption time such that encrypted image is transferred faster to the person.
4. The original image clarity we get after decrypting it.

### 3.4 Proposed Method

The proposed method is to have 16 matrix each of them is  $(4 \times 4)$  matrix. Using  $(4 \times 4)$  bigger matrix each cell contain  $(4 \times 4)$  matrix, each of the smaller  $(4 \times 4)$  matrix will have 16 value, and then we have similarly 16 matrix which all together contain 256 value.

The main contribution in this research is enhance 3D-Playfair algorithm to encrypt image which is based on replacement technique and the introduction of all the 256 value in that process for more confusion.

3D-Playfair cipher has mainly 3 algorithms, Key Matrix Generation, Encryption and Decryption. These are described below:

#### 3.4.1 Key Matrix Generation Algorithm

**Input:** Secret key.

**Output:** Key Matrix.

3D-Playfair Cipher use  $(16 \times 4 \times 4)$  matrix, which is used to store a keyword that becomes the key for encryption and decryption, storing keyword into  $(16 \times 4 \times 4)$  matrix is based on some simple rules, as below:

**STEP 1:** Enter the secret key(password) which may contain numerals, alphabets (caps and small letters are accepted) special symbols and any ASCII character like: Passion, dsf:j5%d2@9pLJFSWRY32HH\*, hayyTj\*\$=uP,m247@\_, aakj@mj/u+.

**STEP 2:** Find out the keyword by dropping the duplicate letters of key.

- Passion => Pasion.
- dsf:j5%d2@9pLJFSWRY32HH\* => dsf:j5%2@9pLJFSWRY3H\*.
- hayyTj\*\$=uP => hayTj\*\$=uP.
- aakj@mj/u+ => akj@mj/u+.

**STEP 3:** Convert the resultant keyword letters into their respective decimal values as show in Table 3.1.

Table 3.1: Keyword= dsf:j5%2@9pLJFSWRY3H\* .

<b>Character</b>	d	s	f	:	j	5	%	2	@	9	p
<b>Decimal</b>	100	115	102	058	106	053	037	050	064	057	112
<b>Character</b>	L	J	F	S	W	R	Y	3	H	*	
<b>Decimal</b>	076	074	070	083	087	082	089	051	072	042	

**STEP 4:** Fill the matrix in linear mode first using the decimal values of keyword, and the rest with the remaining decimal values from (0 - 255) as following:

**STEP 4.1:** Fill the even column in the first row in first matrix.

**STEP 4.2:** Repeat step 4.1 to all matrix.

**STEP 4.3:** Repeat step 4.1 and 4.2 to remaining rows.

**STEP 4.4:** Fill the odd column in the first row in first matrixes.

**STEP 4.5:** Repeat step 4.4 to all matrixes.

**STEP 4.6:** Repeat step 4.4 and 4.5 to remaining rows.

Figure 3.1 show the process of Key-Matrix generation algorithm.

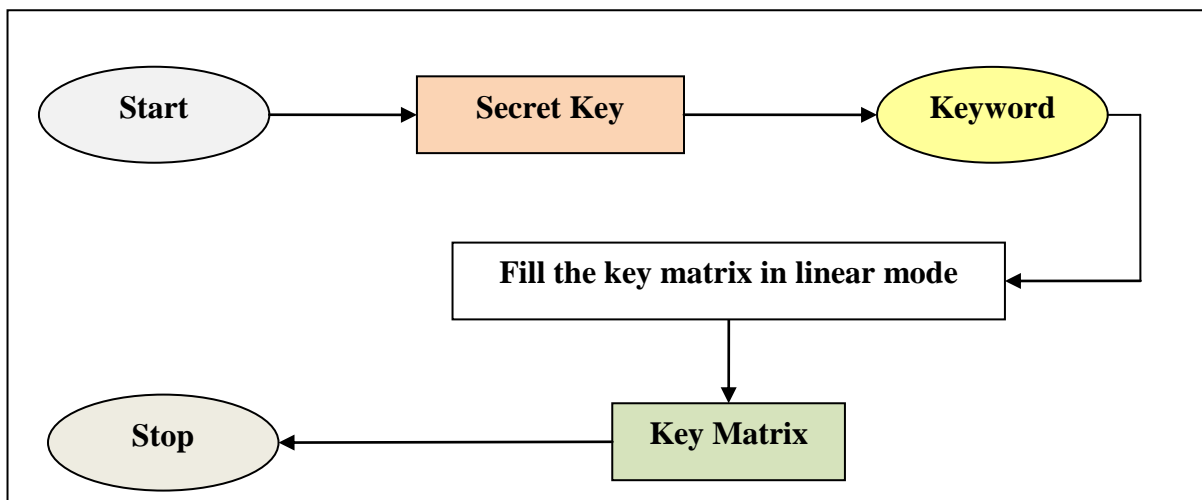


Figure 3.1: The Key-Matrix generation process



By assuming a null secret key Tables 3.2 show the key matrix (4 X 4 matrix each cell in this matrix contains another 4 X 4 sub matrix) (16 X 4 X 4) .This will lead to 16 units in each cell which will lead to a total of 256 units, these 256 units will have a range of decimal values (000-255) .

Table 3.2: 3D-Playfair 16 X 4 X 4 matrix. secret key = null.

	<b>C0</b>				<b>C1</b>				<b>C2</b>				<b>C3</b>			
<b>R0</b>	<b>Matrix 0</b>				<b>Matrix 1</b>				<b>Matrix 2</b>				<b>Matrix 3</b>			
	000	032	016	048	001	033	017	049	002	034	018	050	003	035	019	051
	128	160	144	176	129	161	145	177	130	162	146	178	131	163	147	179
	064	096	080	112	065	097	081	113	066	098	082	114	067	099	083	115
	192	224	208	240	193	225	209	241	194	226	210	242	195	227	211	243
<b>R1</b>	<b>Matrix 4</b>				<b>Matrix 5</b>				<b>Matrix 6</b>				<b>Matrix 7</b>			
	004	036	020	052	005	037	021	053	006	038	022	054	007	039	023	055
	132	164	148	180	133	165	149	181	134	166	150	182	135	167	151	183
	068	100	084	116	069	101	085	117	070	102	086	118	071	103	087	119
	196	228	212	244	197	229	213	245	198	230	214	246	199	231	215	247
<b>R2</b>	<b>Matrix 8</b>				<b>Matrix 9</b>				<b>Matrix 10</b>				<b>Matrix 10</b>			
	008	040	024	056	009	041	025	057	010	042	026	058	011	043	027	059
	136	168	152	184	137	169	153	185	138	170	154	186	139	171	155	187
	072	104	088	120	073	105	089	121	074	106	090	122	075	107	091	123
	200	232	216	248	201	233	217	249	202	234	218	250	203	235	219	250
<b>R3</b>	<b>Matrix 12</b>				<b>Matrix 13</b>				<b>Matrix 14</b>				<b>Matrix 15</b>			
	012	044	028	060	013	045	029	061	014	046	030	062	015	047	031	063
	140	172	156	188	141	173	157	189	142	174	158	190	143	175	159	191
	076	108	092	124	077	109	093	125	078	110	094	126	079	111	095	127
	204	236	220	252	205	237	221	253	206	238	222	254	207	239	223	255

By assuming the secret key is dsf:j5%2@9pLJFSWRY32H\* the (16 X 4 X 4) matrix will be arrange as following in Table 4.3:

Table 3.3: 3D-Playfair 16 X 4 X 4 matrix, secret key = dsf:j5%2@9pLJFSWRY32H\*.

	<b>C1</b>				<b>C2</b>				<b>C3</b>				<b>C4</b>			
<b>R1</b>	<b>Matrix 0</b>				<b>Matrix 1</b>				<b>Matrix 2</b>				<b>Matrix 3</b>			
	100	011	082	027	115	012	089	028	102	013	051	029	058	014	072	030
	128	160	144	176	129	161	145	177	130	162	146	178	131	163	147	179
	045	091	067	110	046	092	068	111	047	093	069	113	048	094	071	114
	192	224	208	240	193	225	209	241	194	226	210	242	195	227	211	243
<b>R2</b>	<b>Matrix 4</b>				<b>Matrix 5</b>				<b>Matrix 6</b>				<b>Matrix 7</b>			
	106	015	042	031	053	016	000	032	037	017	001	003	050	018	002	034
	132	164	148	180	133	165	149	181	134	166	150	182	135	167	151	183
	049	095	073	116	052	096	075	117	054	097	077	118	055	098	078	119
	196	228	212	244	197	229	213	245	198	230	214	246	199	231	215	247
<b>R3</b>	<b>Matrix 8</b>				<b>Matrix 9</b>				<b>Matrix 10</b>				<b>Matrix 11</b>			
	064	019	003	035	057	020	004	036	112	021	005	038	076	022	006	039
	136	168	152	184	137	169	153	185	138	170	154	186	139	171	155	187
	056	099	079	120	059	101	080	121	060	103	081	122	061	104	084	123
	200	232	216	248	201	233	217	249	202	234	218	250	203	235	219	251
<b>R4</b>	<b>Matrix 12</b>				<b>Matrix 13</b>				<b>Matrix 14</b>				<b>Matrix 15</b>			
	074	023	007	040	070	024	008	041	083	025	009	043	087	026	101	004
	140	172	156	188	141	173	157	189	142	174	158	190	143	175	159	191
	062	105	085	124	063	107	086	125	065	108	088	126	066	109	090	127
	204	236	220	252	205	237	221	253	206	238	222	254	207	239	223	255

### 3.4.2 Image Encryption Algorithm

**Input:** Plain image and secret key.

**Output:** Cipher image.

**STEP 1:** Read the plain image as Red, Green and Blue matrixes.

**STEP 2:** Transform color matrixes to one dimension arrays.

**STEP 3:** Break the arrays into trigraph (groups of 3 value).

**STEP 4:** If the plain image arrays has dimension number we cannot break into trigraph, append zero or zero and one to the end.

**STEP 5:** Construct the Key Matrix (16 X 4 X 4) using the secret key and Key Matrix Generation algorithm.

**STEP 6:** Pre processing step for each trigraph of color components in the Red plane of the plain image do the following:

**STEP 6.1:** XOR the first value with the multiplication of the other two values.

**STEP 6.2:** Repeat step 6.1 to second and third value.

**STEP 7:** Encryption step for each trigraph of color components in the Red plane of the plain image do the following:

A value in the trigraph will be replaced by the value that will lay on the same row of the value and the column of the next value and at the matrix of next-to-next letter in circular fashion, this approach can be better understand by the following Table 3.4:

Table 3.4: Encryption Process of 3D-Playfair (16 X 4 X 4) matrix.

Plain Image Trigraph	Plain Image Trigraph			Cipher Image Trigraph
	1st value	2nd value	3rd value	
1st value	Row	Column	Matrix	1st value
2nd value	Matrix	Row	Column	2nd value
3rd value	Column	Matrix	Row	3rd value

**STEP 8:** Use the secret key to generate a mask made up with a random permutation of the numbers between 0 and 255.

**STEP 9:** XOR the resultant scrambled image with the generated random mask.

**STEP 10:** Repeat step 6 to 9 for Green and Blue color planes of the plain image.

**STEP 11:** Transform the arrays to color matrixes.

**STEP 12:** Return the resultant image as the cipher image.

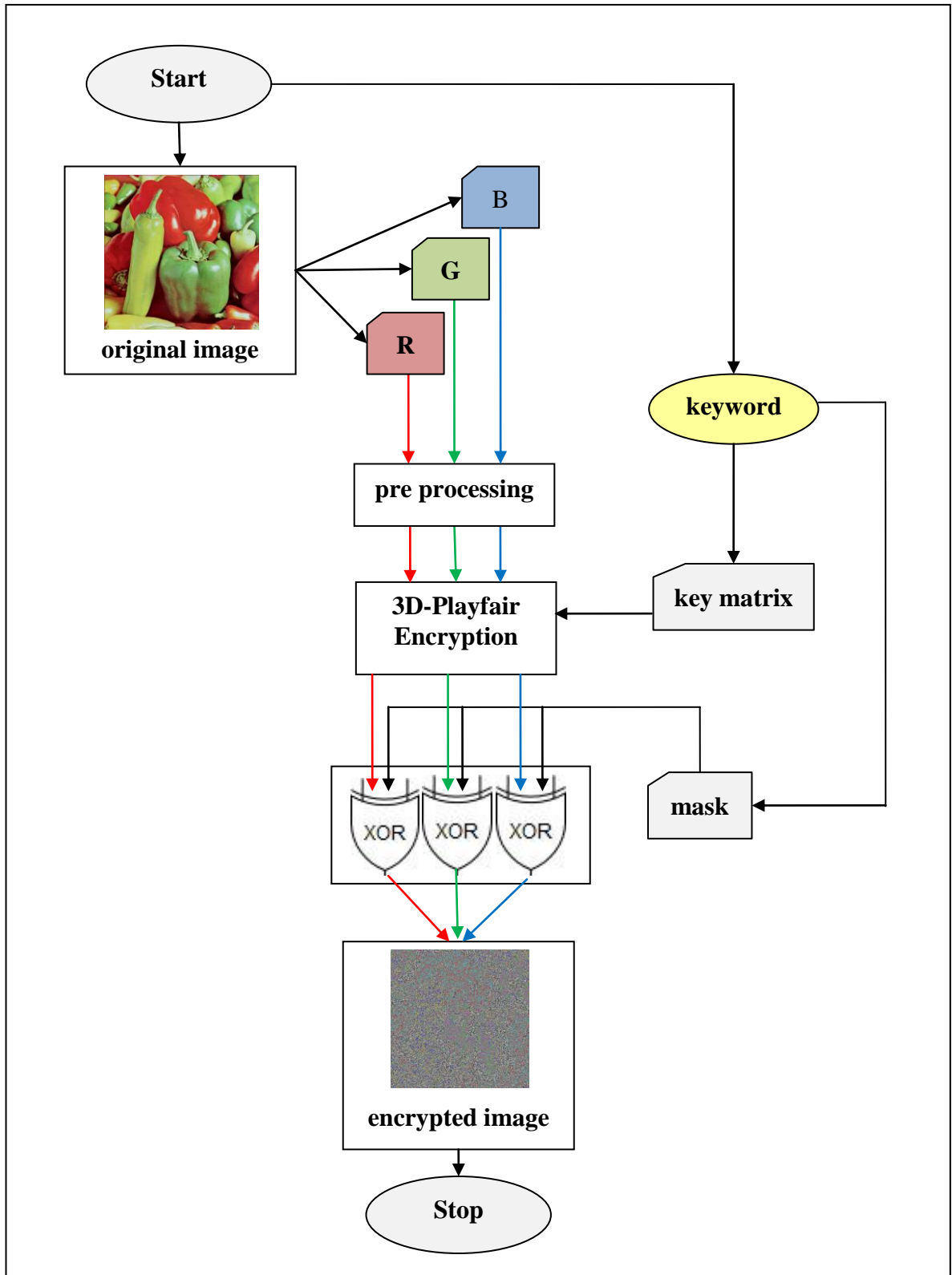


Figure 3.2: Show the process of encryption algorithm.

### 3.4.3 Image Decryption Algorithm

**Input:** Cipher image and secret key.

**Output:** Plain image.

**STEP 1:** Read the Cipher image as Red, Green and Blue matrixes.

**STEP 2:** Transform color matrixes to one dimension arrays.

**STEP 3:** Use the secret key to generate a mask made up with a random permutation of the numbers between 0 and 255.

**STEP 4:** XOR the RED color plane of the cipher image with the generated random mask.

**STEP 5:** Construct the Key Matrix (16 X 4 X 4) using the secret key and Key Matrix Generation algorithm.

**STEP 6:** Decryption step for each trigraph of the resultant XORed RED plane of the Cipher image do the following:

A value in the trigraph will be replaced by the value that will lay on the same row of the value and at the matrix of the next value and the column of next-to-next value in circular fashion, this approach can be better understand by the following Table 3.5:

Table 3.5: Decryption Process of 3D-Playfair (16 X 4 X 4) matrix.

Plain Image Trigraph	Plain Image Trigraph			Cipher Image Trigraph
	1st value	2nd value	3rd value	
1st value	Row	Matrix	Column	1st value
2nd value	Column	Row	Matrix	2nd value
3rd value	Matrix	Column	Row	3rd value

**STEP 7:** Inverse pre processing step for each trigraph of color components in the Red plane of the plain image do the following:

**STEP 7.1:** XOR the first value with the multiplication of the other two values.

**STEP 7.2:** Repeat step 7.1 to second and third value.

**STEP 8:** Repeat step 5 and 6 for Green and Blue color planes of the cipher image.

**STEP 9:** Remove the filler value from the trigraph (Dropping any extra 0 and 1 in the end of arrays) to find out the actual image (plain image).

**STEP 10:** Transform the arrays to color matrixes.

**STEP 11:** Return the resultant image as the reconstructed image.

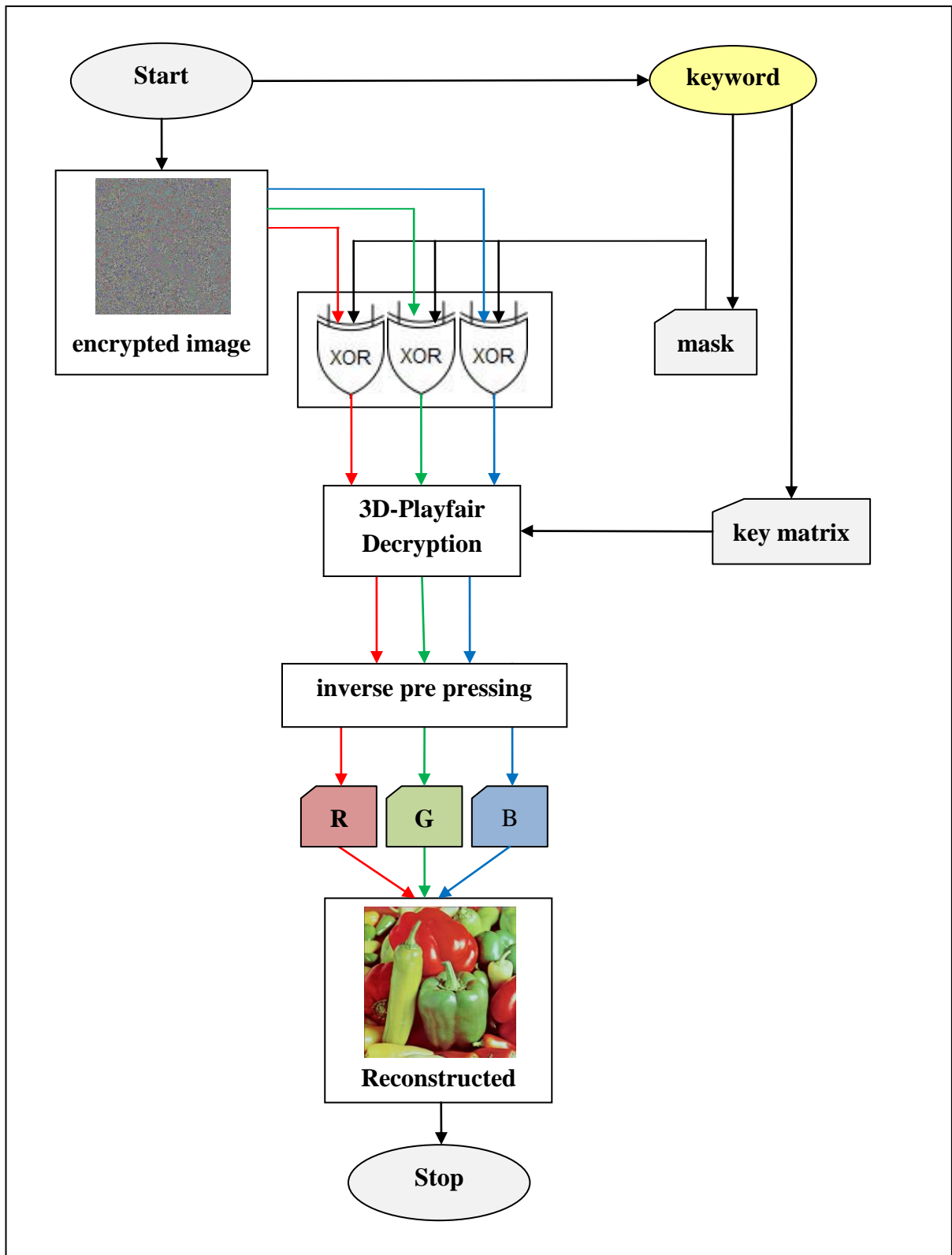


Figure 3.3: Show the process of decryption algorithm.

## **3.5 Techniques and Tools**

### **3.5.1 Java**

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture, java is one of the most popular programming languages in use, particularly for client server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them [16].

### **3.5.2 Advantages of Java**

Java is characterized by special features, making it the most used programming languages, which distinguishes them as follows:

- **Simple:** Java was designed to be easy to use, write, compile, debug, and learn than other programming languages.
- **Object-Oriented:** Allows to create modular programs and reusable code.
- **Platform-Independent:** Ability to move easily from one computer system to another
- **Distributed:** Designed to make distributed computing easy with the networking capability that is inherently integrated into it.
- **Secure:** The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.
- **Allocation:** Java has the feature of Stack allocation system. It helps the data to be stored and can be restored easily.
- **Multithreaded:** The capability for a program to perform several tasks simultaneously within a program.

### **3.5.3 Disadvantages of Java**

- Performance: Significantly slower and more memory-consuming than natively compiled languages such as C or C++.
- Look and feel: The default look and feel of GUI applications written in Java using the Swing toolkit is very different from native applications.
- Single-paradigm language: The addition of static imports in Java 5.0 the procedural paradigm is better accommodated than in earlier versions of Java.

### **3.5.4 MATLAB**

MATLAB is a programming platform designed specifically for engineers and scientists. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

Using MATLAB, you can, analyze data, develop algorithms, create models and applications.

### **3.5.5 Advantages of MATLAB**

Ease of use, platform independence, predefined functions, Matlab makes use of highly respected algorithms, confident about your results, can build up set of functions for a particular application.

### **3.5.6 Disadvantages of MATLAB**

Can be slow and expensive.



## CHAPTER IV

### IMPLEMENTATION, RESULTS AND DISCUSSIONS

#### 4.1 Implementation

The implantation of the algorithm was done using java as show in Figure 4.1, and tested on sample of images. The image size not fixed since the algorithm work on (m x n) image size.



Figure 4.1: System home screen

The Figure 4.2 illustrates key matrix when the secret key = maxk(123\*).

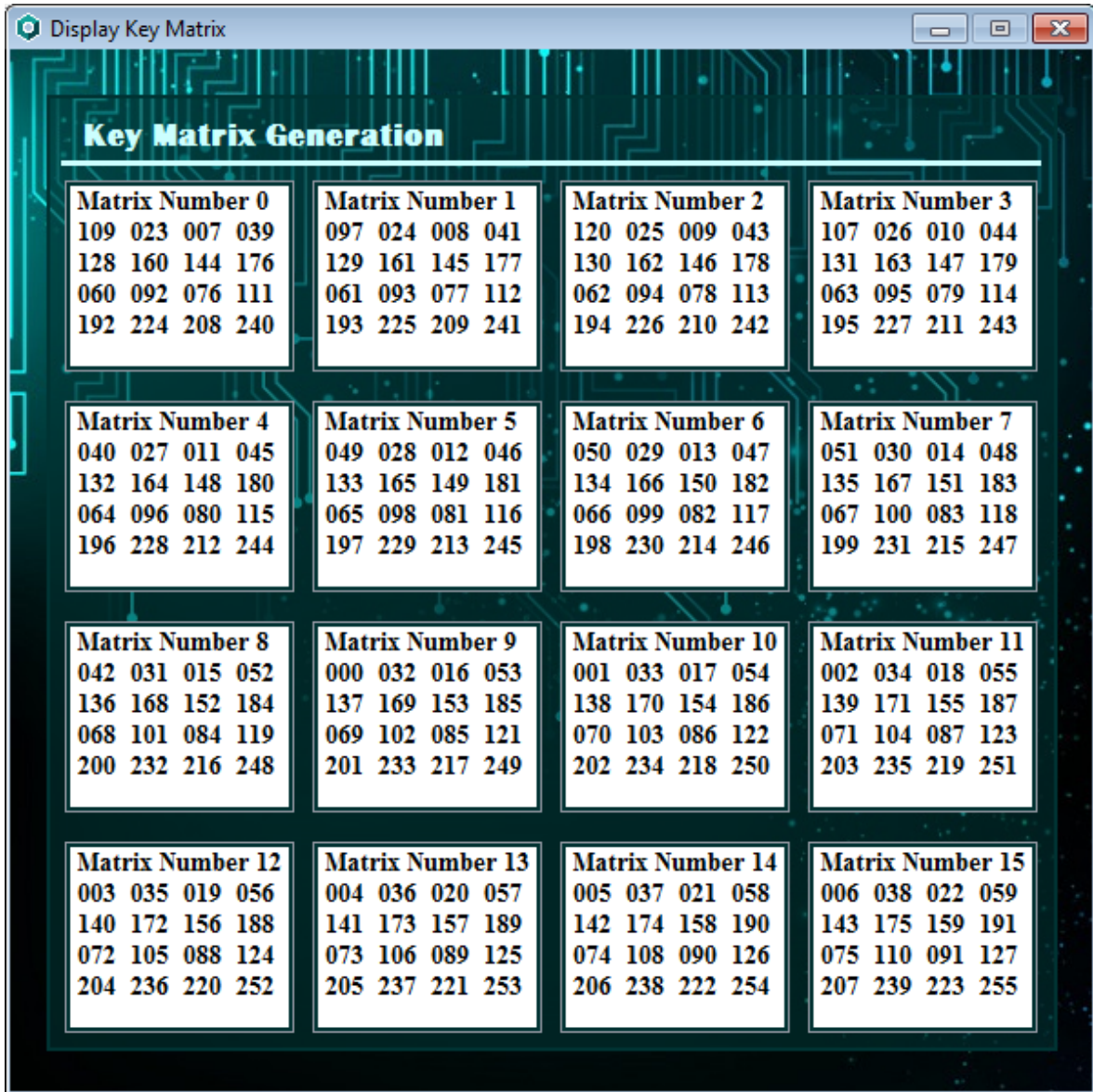


Figure 4.2: Key matrix, secret key=maxk(123\*)

The Figure 4.3 illustrates the graphical user interface of executing the application to encrypt such images. Text filed for enter secret key and two buttons (Select Image) which loads the image that the user selected, and (Encrypt) which complete encryption process after click it.



Figure 4.3: Graphical user interface for encryption process

Steps of encryption:

1. Enter secret key.
2. Select image (original image) from disk.
3. Press (Encrypt) button.
4. Save or open the decrypted image.

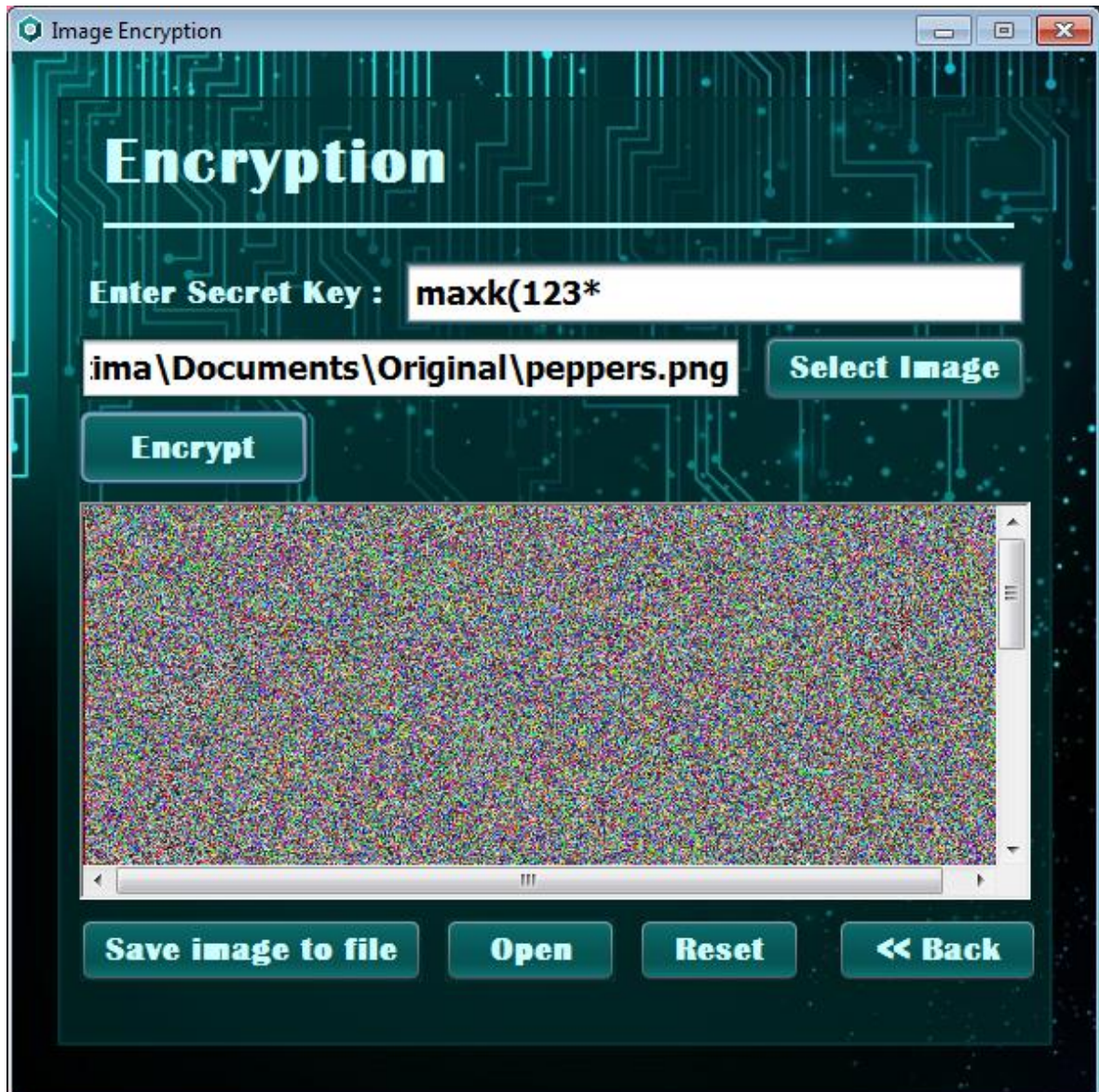


Figure 4.4: Encryption process while it done



Figure 4.5: Image after encryption process (encrypted image)

The Figure 4.6 illustrates the graphical user interface of executing the application to decrypt such images. Text field for enter secret key and two buttons (Select Image) which loads the image that the user selected, and (Decrypt) which complete decryption process after click it.



Figure 4.6: Graphical user interface for decryption process

Steps of decryption:

1. Enter secret key.
2. Select image (encrypted image) from disk.
3. Press (Decrypt) button.
4. Save or open the reconstructed image.



Figure 4.7: Decryption process while it done

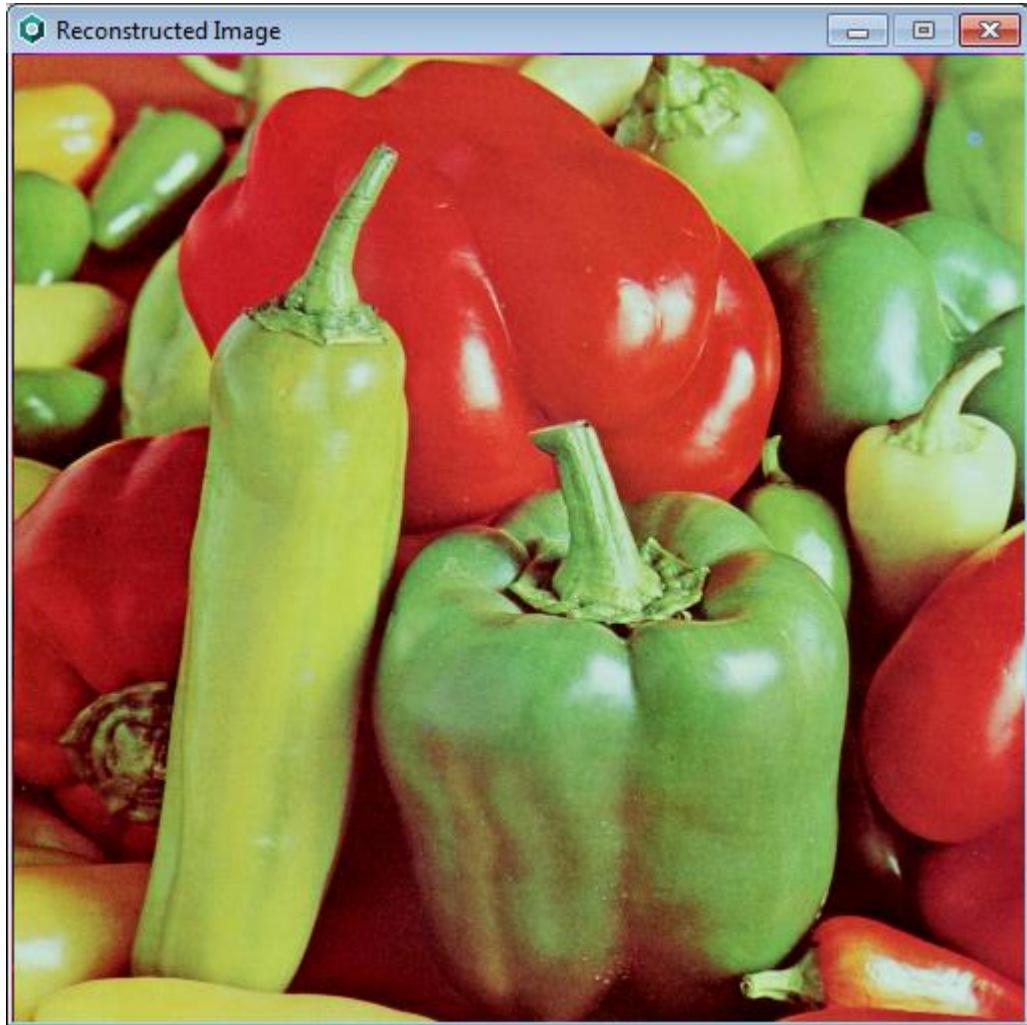


Figure 4.8: Image after decryption process (reconstructed image)



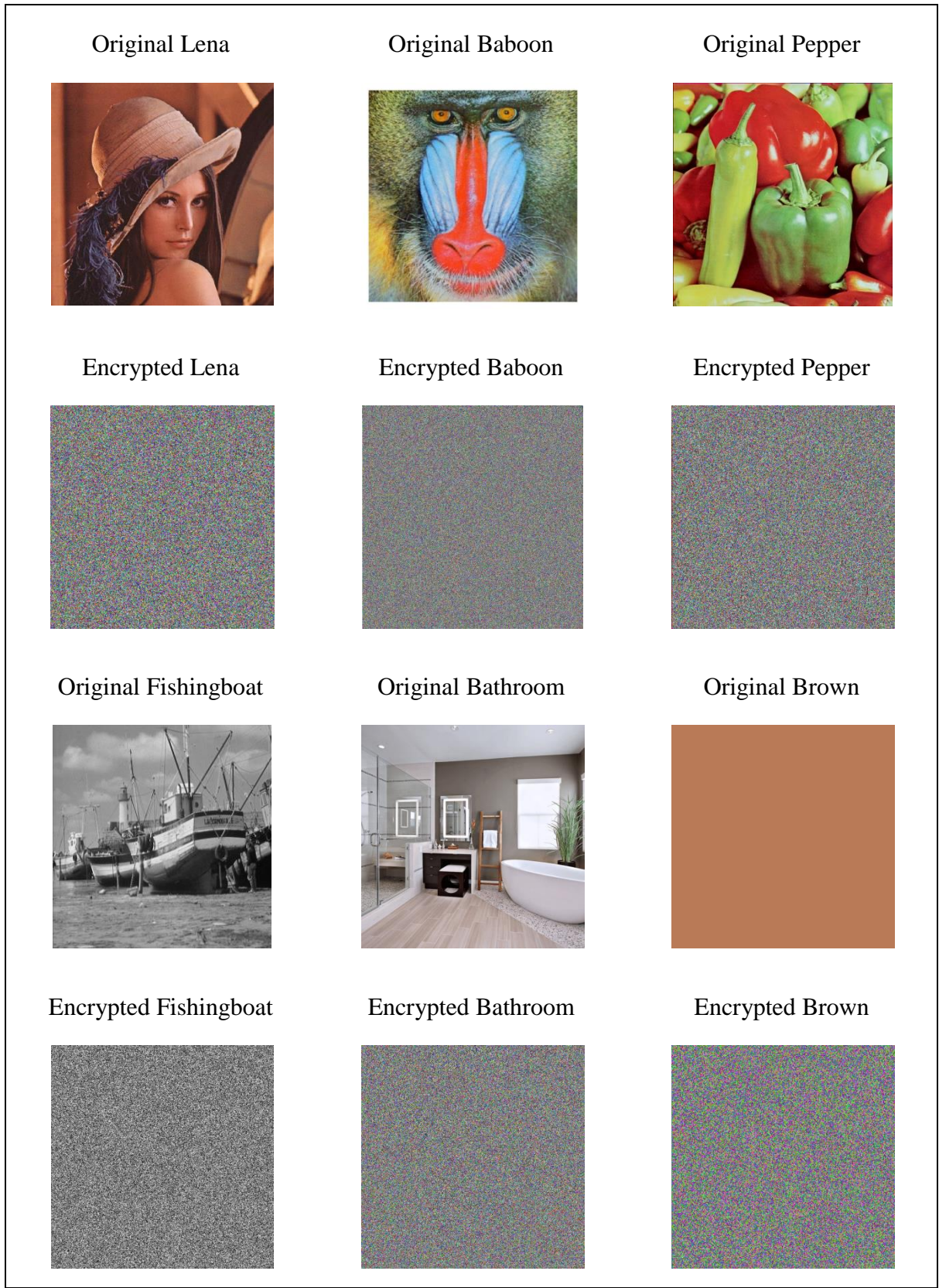


Figure 4.9: The result of applying 3D-Playfair encryption for images sample.

## **4.2 Results and Discussion**

### **4.2.1 Key Space Analysis**

Cryptanalysis is a field that attempt to find techniques to decrypt a message without prior knowledge on it ciphering method. Cryptanalysis is what the layperson calls "breaking the code". Together with cryptography they are called cryptology.

In classic Playfair cipher, obtaining the key is relatively straightforward if both plain-text and cipher-text are known. However, usually only the cipher text will be available. Thus, guessing some of the words based on knowledge about the origin of the message can be of a great help in reconstructing the substitution matrix, it should be recognized that guessing some of the plain-text and using that to reconstruct the key square is by far the easiest way to crack this cipher.

Cryptanalysis of the 3D-Playfair cipher for image is much more difficult than normal simple Playfair substitution cipher, because in this case digraphs replace with trigraph, and represent pixels instead of letters.[15]

### **4.2.2 Key Sensitivity Test**

Several key sensitivity tests were performed using a number of close key values. Figure 4.10 shows the resultant cipher images using the key values: 21985, 21986, 21987, 21988 and 21989 respectively. Figure 4.11 shows the corresponding reconstructed images using more closed faked key value while the correct key is 21987. The results show that the roposed method is sensitive to the key. That is, a small change of the key value will result in a completely different image.

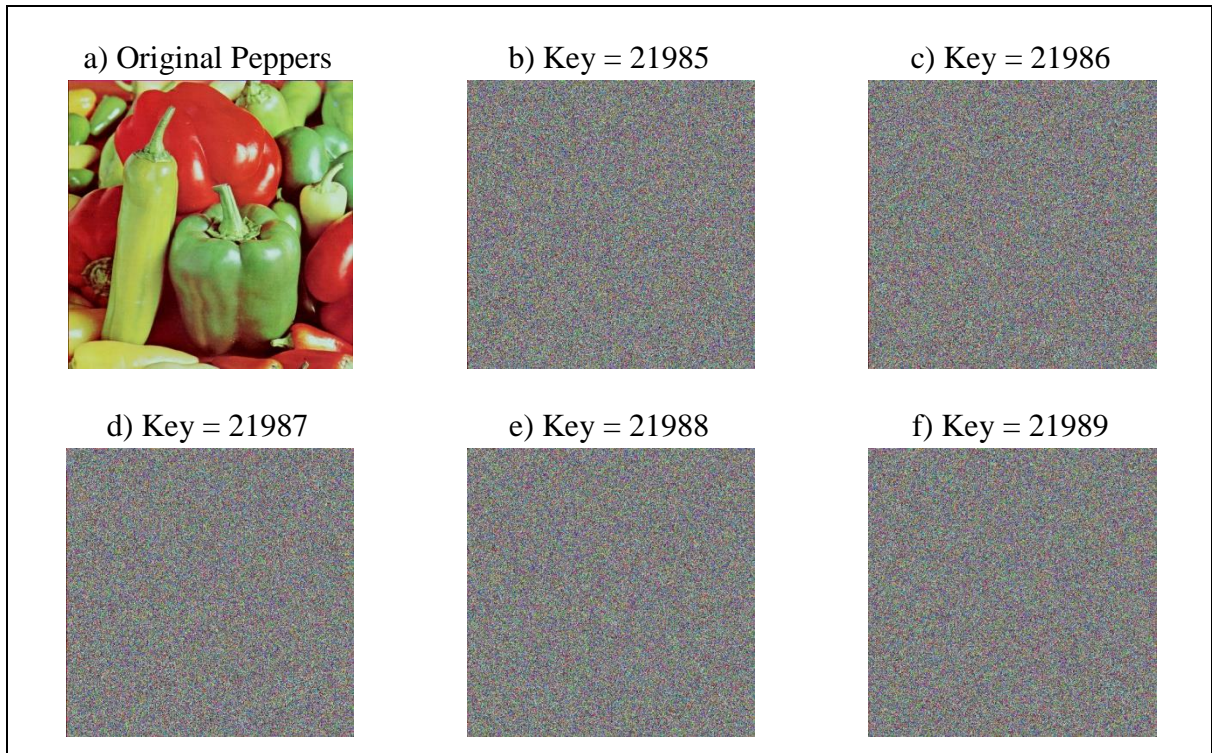


Figure 4.10: The original Peppers and Ciphered images using different keys.

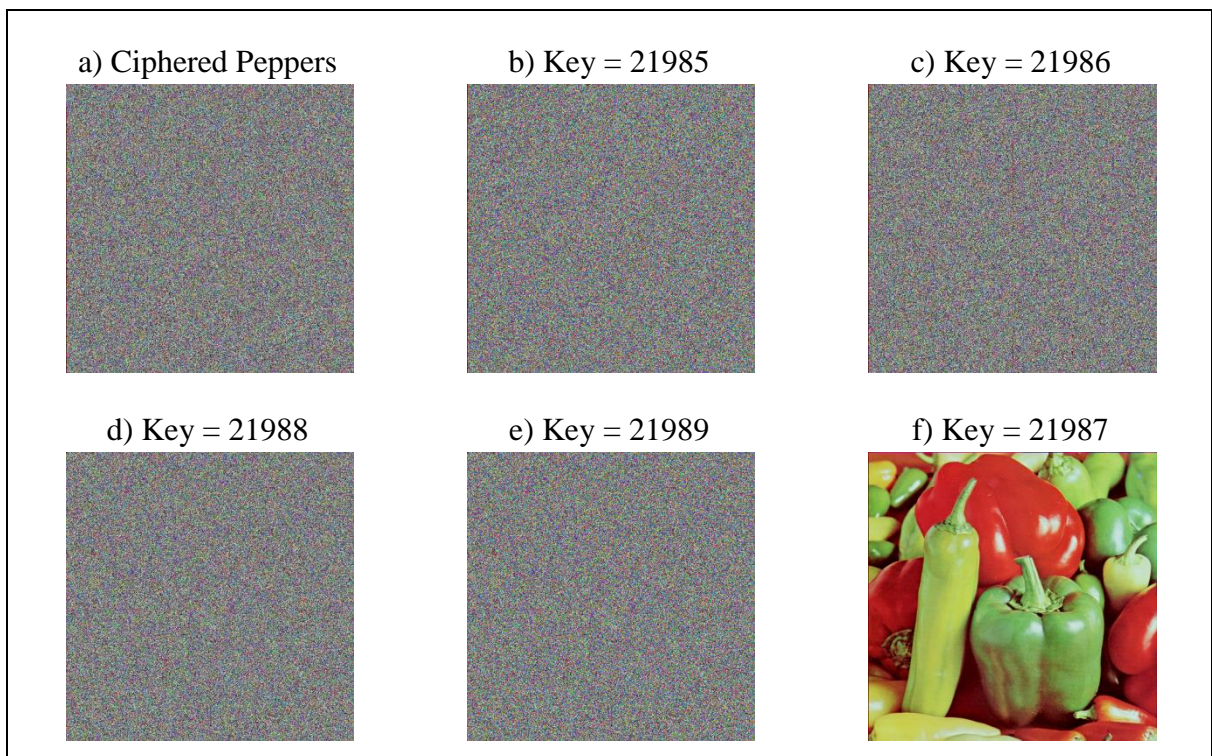


Figure 4.11: The ciphered Peppers and reconstructed images using different keys.

### 4.2.3 Visual Diffusion Test

More experimentation has been conducted to visually judge the diffusion in the resulted images using similar key values. The popular Peak signal-to-noise ratio (PSNR ) metric was employed as a similarity measure. PSNR can be computed using the following formula:

$$PSNR = 10 \times \log\left(\frac{(\max f(x,y))^2}{MSE}\right) \quad (1)$$



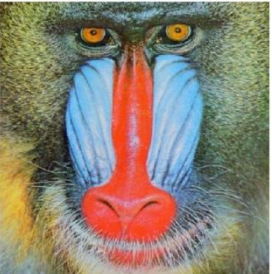
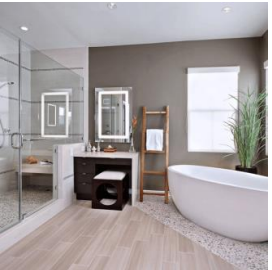
$$MSE = \frac{1}{X \times Y} \sum_{x,y} (f(x,y) - p(x,y))^2 \quad (2)$$

Where  $f(x, y)$  and  $p(x, y)$  are the compared images of size  $X \times Y$  and MSE denotes the Mean Square Error. PSNR values are often expressed in decibels (dB) where the values will run to infinity if the two examined images are identical. Table 4.1: show PSNR, MSE values for original images, cipher images and reconstructed image, and Table 4.2 compares the PSNR values showing further information on the diffusion aspect using different keys on various original images.

Table 4.1: PSNR, MSE values for original images, cipher images and reconstructed image.

	Original	Encrypted	Reconstructed
<b>Original Pepper</b> 	MSE = 0.0 PSNR = Inf dB	MSE = 8202.089 PSNR = 9.026 dB	MSE = 0.0 PSNR = Inf dB
<b>Encrypted Pepper</b> 	MSE = 8202.089 PSNR = 9.026 dB	MSE = 0.0 PSNR = Inf dB	MSE = 8202.089 PSNR = 9.026 dB
<b>Reconstructed Pepper</b> 	MSE = 0.0 PSNR = Inf dB	MSE = 8202.089 PSNR = 9.026 dB	MSE = 0.0 PSNR = Inf dB
<b>Original Bathroom</b> 	MSE = 0.0 PSNR = Inf dB	MSE = 9103.035 PSNR = 8.573 dB	MSE = 0.0 PSNR = Inf dB
<b>Encrypted Bathroom</b> 	MSE = 9103.035 PSNR = 8.573 dB	MSE = 0.0 PSNR = Inf dB	MSE = 9103.035 PSNR = 8.573 dB
<b>Reconstructed Bathroom</b> 	MSE = 0.0 PSNR = Inf dB	MSE = 9103.035 PSNR = 8.573 dB	MSE = 0.0 PSNR = Inf dB

Table 4.2: PSNR, MSE values for original Images and various cipher images using different secret keys.

Image	Key	19907S	Make?/	at@m*
Pepper 	19907S	MSE = 0.0 PSNR = Inf dB	MSE = 10928.05 PSNR= 7.779 dB	MSE = 10853.83 PSNR= 7.808 dB
	Make?/	MSE = 10928.05 PSNR= 7.779 dB	MSE = 0.0 PSNR = Inf dB	MSE = 10905.07 PSNR= 7.789 dB
	at@m*	MSE = 10853.83 PSNR= 7.808 dB	MSE = 10905.07 PSNR= 7.789 dB	MSE = 0.0 PSNR = Inf dB
Lena 	19907S	MSE = 0.0 PSNR = Inf dB	MSE = 10888.92 PSNR=7.794 dB	MSE = 10872.44 PSNR= 7.802 dB
	Make?/	MSE = 10888.92 PSNR=7.794 dB	MSE = 0.0 PSNR = Inf dB	MSE = 10896.17 PSNR= 7.792 dB
	at@m*	MSE = 10872.44 PSNR= 7.802 dB	MSE = 10896.17 PSNR= 7.792 dB	MSE = 0.0 PSNR = Inf dB
Baboon 	19907S	MSE = 0.0 PSNR = Inf dB	MSE = 10854.16 PSNR= 7.808 dB	MSE = 10872.181 PSNR= 7.802 dB
	Make?/	MSE = 10854.16 PSNR= 7.808 dB	MSE = 0.0 PSNR = Ind dB	MSE = 10881.193 PSNR= 7.798 dB
	at@m*	MSE = 10872.181 PSNR= 7.802 dB	MSE = 10881.19 PSNR= 7.798 dB	MSE = 0.0 PSNR = Inf dB
Bathroom 	19907S	MSE = 0.0 PSNR = Inf dB	MSE = 10877.34 PSNR= 7.799 dB	MSE = 10874.21 PSNR =7.801 dB
	Make?/	MSE = 10877.34 PSNR= 7.799 dB	MSE = 0.0 PSNR = Inf dB	MSE = 10907.39 PSNR= 7.789 dB
	at@m*	MSE = 10874.21 PSNR= 7.801 dB	MSE = 10907.39 PSNR= 7.789 dB	MSE = 0.0 PSNR = Inf dB

#### 4.2.4 Histogram

Histogram is a representation of the distribution of colors in an image, for digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the image's color space, the set of all possible colors, original image and cipher image histograms using same key is shown in Table 4.4.

In the histogram form image properties tested three colors (RGB), comparison of plain image and cipher image histograms is shown in Table 4.3.

Table 4.3: Histogram comparison between the original and cipher images use different color

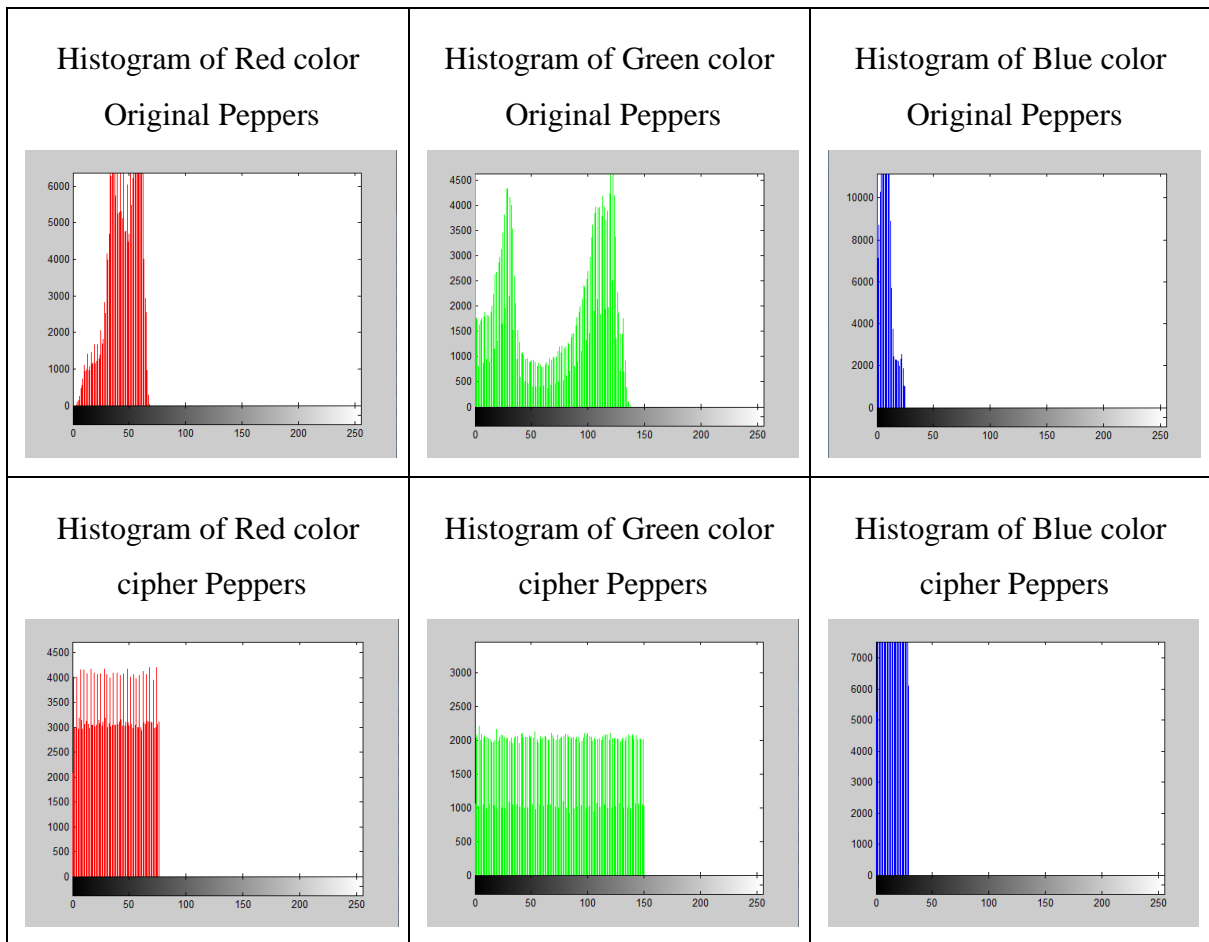

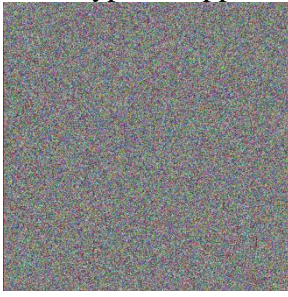

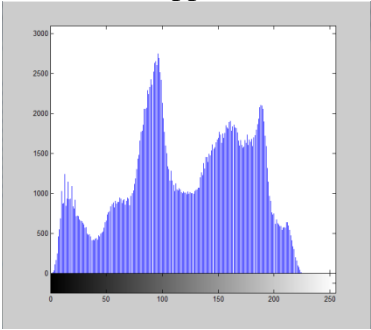
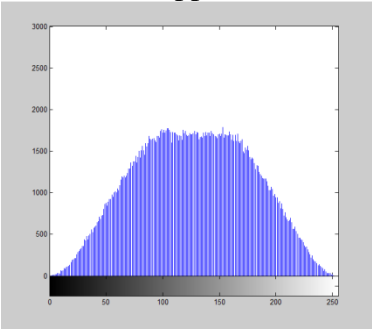
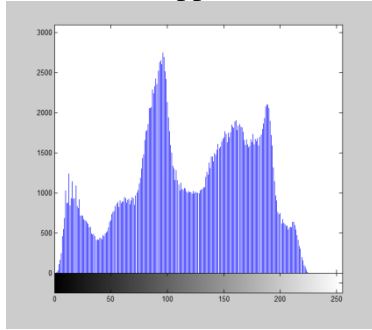

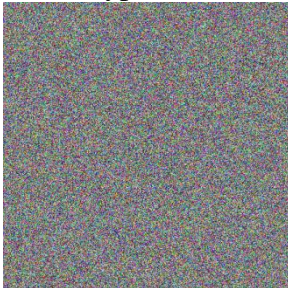

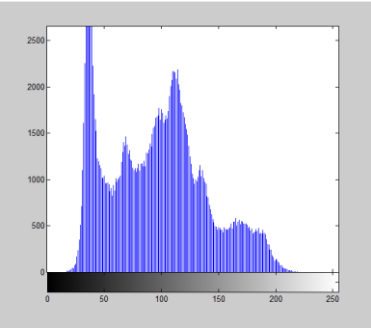
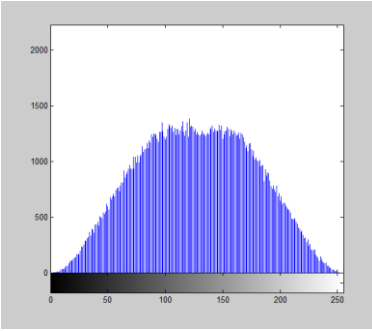
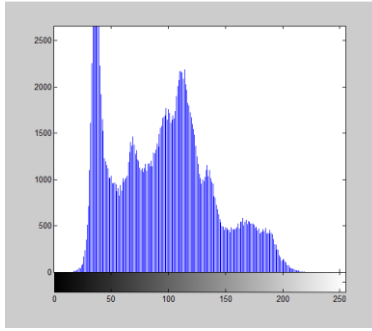

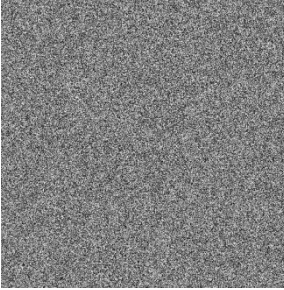

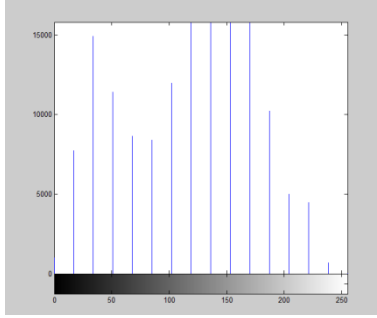
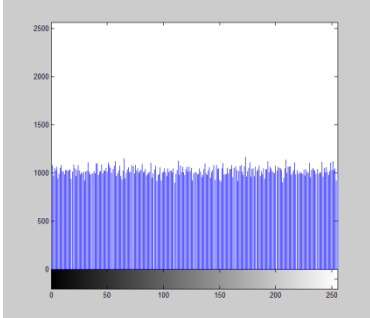
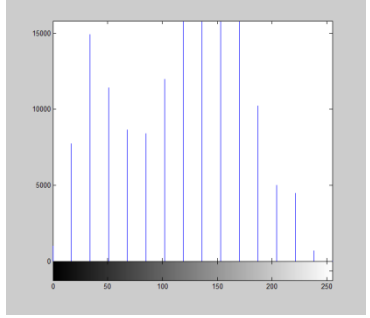

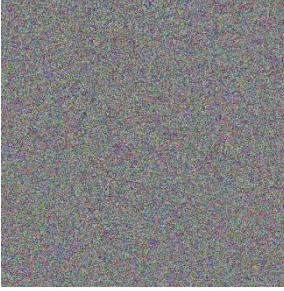

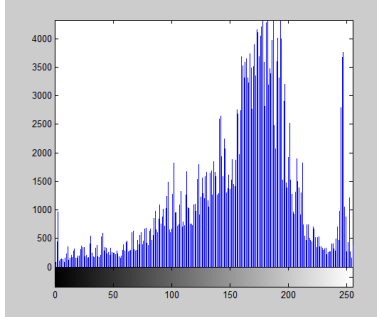
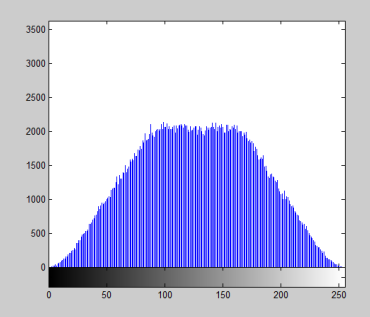
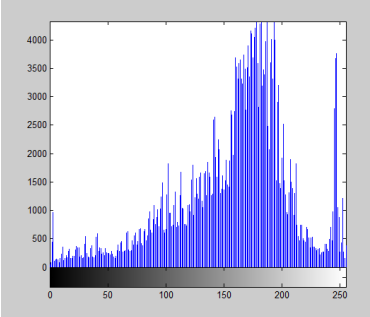


Table 4.4: Original image and cipher image histograms using same key.

<p>Original Pepper</p> 	<p>Encrypted Pepper</p> 	<p>Reconstructed Pepper</p> 
<p>Histogram of original Pepper</p> 	<p>Histogram of encrypted Pepper</p> 	<p>Histogram of reconstructed Pepper</p> 
<p>Original Lena</p> 	<p>Encrypted Lena</p> 	<p>Reconstructed Lena</p> 
<p>Histogram of original Lena</p> 	<p>Histogram of encrypted Lena</p> 	<p>Histogram of reconstructed Lena</p> 



<p><b>Original Fishingboat</b></p> 	<p><b>Encrypted Fishingboat</b></p> 	<p><b>Reconstructed Fishingboat</b></p> 
<p><b>Histogram of original Fishingboat</b></p> 	<p><b>Histogram of encrypted Fishingboat</b></p> 	<p><b>Histogram of reconstructed Fishingboat</b></p> 
<p><b>Original Bathroom</b></p> 	<p><b>Encrypted Bathroom</b></p> 	<p><b>Reconstructed Bathroom</b></p> 
<p><b>Histogram of original Bathroom</b></p> 	<p><b>Histogram of encrypted Bathroom</b></p> 	<p><b>Histogram of reconstructed Bathroom</b></p> 

### **4.3 Security Aspects of Cipher**

Security is main aspects for any encryption algorithm while time complexity and space complexity also play roles in the selection of any cryptographic algorithms but security is the sole parameter. So some security aspects are discussed here [10].

#### **4.3.1 Brute Force Attack**

A brute force attack systematically attempts every possible key. It is most often used in a known plaintext or cipher value-only attack [10]. In the proposed system we use  $16 \times 4 \times 4$  matrix for encryption and decryption purpose.

So attackers will get  $256 \times 16 \times 4 = 16384$  trigraph for brute force attack.

#### **4.3.2 Frequency Analysis**

Frequency analysis is the study of the frequency of value or groups of value in a cipher text [11]. Frequency analysis is based on the fact that, in any given stretch of written language, certain letters and combinations of letters occur with varying frequencies. Moreover, there is a characteristic distribution of letters that is roughly the same for almost all samples of that language [9].

The probability of occurrence of any particular character in classical Playfair matrix is  $1/26 = 0.0384$ . Whereas the probability of occurrence of a character in 3D-Playfair matrix is  $1/16 * 1/16 = 1/256 = 0.00391$ .

Applying the same analogy of frequency analysis requires inspecting 16777216 pixel trigraph.

#### **4.3.3 Confusion and Diffusion**

Confusion involves making the statistical relation between plain image and cipher image as complex as possible. Diffusion refers to the property that the redundancy in the statistics of the plain image is dissipated in the statistics of the cipher image [15].

In 3D-Playfair cipher  $16 \times 4 \times 4$  matrix provides better confusion ratio. As it works with trigraph so any cipher image value could be determined by combination of three value, it ensures the high diffusion rate comparing to classical Playfair cipher in which combination of two value could determines the cipher image value.

## CHAPTER V

### CONCLUSIONS AND RECOMMENDATIONS

#### 5.1 Conclusion

3D-Playfair cipher is a symmetric encryption technique which is rich enough to encrypt image. It uses trigraph rather than using digraph to eliminate the fact that a diagram and its reverse will encrypt in a similar fashion. This cipher is not vulnerable to security attacks, by using trigraph and  $16 \times 4 \times 4$  matrix it provides high rate of confusion and diffusion rate, there is  $256 \times 16 \times 4 = 16384$  possible trigraph so it is too hard for applying brute force attack on it. It works on 64 characters so the probability of occurrence of a character in 3D-Playfair matrix is  $1/16 * 1/16 = 1/256 = 0.00391$ .

The experimental results showed that the key space of the proposed technique makes it hard for the attacker to perform a frequency analysis based on the used pixel trigraph. Furthermore, further tests showed that a small change in the key value results in completely different cipher images. PSNR values and histogram comparisons were also deployed to show the robustness of the proposed cipher.

#### 5.2 Recommendations

This thesis suggested in the near future, combine 3D-Playfail with other algorithm to become stronger.

## References

- [1]. A.J. Menezes, P.C. Van Oorschot, S.A. Vanstone, Hand-book of applied cryptography (CRC press, 2010).
- [2]. W. Stallings, Cryptography and Network Security: Principles and Practice 5th ed (Pearson, 2011).
- [3]. Kaur, A., Verma, H.K. and Singh, R.K., 2012. 3D (4 X 4 X 4)-Playfair Cipher. International Journal of Computer Applications.
- [4]. A Hamad, S., Khalifa, A., Elhadad, A. and Rida, S.Z., 2013. A modified playfair cipher for encrypting digital images. Modern Science.
- [5]. Chaturvedi, A.K., Rajput, V. and Richarya, V., 2016. 3D-Playfair Cipher with Message Integrity using MD5. International Journal of Computer Applications.
- [6]. Chakravarthy, S., Venkatesan, S.P., Anand, J.M. and Ranjani, J.J., 2016. Enhanced Playfair Cipher for Image Encryption using Integer Wavelet Transform. Indian J Sci Technol.
- [7]. Ashish, Anil Kumar Pandey, 2017. Implementation of 3D Approach in Playfair Cipher. International Journal of Advances in Electronics and Computer Science.
- [8]. Faisal Mohammed Abdalla1, Khadiga Mohammed Adam Babiker, 2018. Modified Playfair Cipher for Encrypting Images, International Journal of Advances in Electronics and Computer Science.
- [9]. William Stallings, Cryptography and Network Security Principles and Practice. Second edition, Pearson Education
- [10]. Behrouz A. Forouzan, 2007. Cryptography and Network Security. Special Indian Edition, The McGraw- Hill companies, New Delhi.
- [11]. Dhiren R.Patel, 2008. Information Security Theory and Practice. First Edition, Prentice-Hall of India Private Limited.
- [12]. March 9 ,2015. Clark How does Encryption Work, and Is It Really Safe.
- [13]. , eTutorials.org, 2008-2018. Ppractical unix & internet security.
- [14]. Jupitermedia Corporation, Retrieved October 25, 2006. What is encryption algorithm?  
A Word Definition From the Webopedia Computer Dictionary.
- [15]. Schnier B, 1996. Applied cryptography: protocols, algorithms and source code in C. New York: John Wiley and sons.

[16]. Computer Weekly, 2009. "Write once, run anywhere?".

[17]. Altius Directory, Retrieved December 2010. "MATLAB Programming Language".