

CHAPTER I

INTRODUCTION

1.1 Introduction:

In recent time a high - speed prosperity in e-commerce through the world, with ever increasing popularity of an online shopping.

Online shopping is retrieval of product information via internet and issue of purchase order through electronic purchase request, filling of credit or debit card information and shipping of product by mail order or home delivery by courier [1]. But because of tremendous development in the field of hacking [2], led to common security threats in online shopping like Identity theft and Phishing.

Identity theft is a form of stealing someone's identity in which someone pretends to be someone else. Whereas phishing is the method of stealing personal confidential information such username, password or card details from a victim [3], so the majority of securing information in online payment systems it's very important for customers, merchants and banks. To cure this issue Secure Socket Layer (SSL) encryption inhibits the interference of consumer information in transit between the consumer and the online merchant. However, one must still trust merchant and its employees not to use consumer information for their own purchases and not to sell the information to others.

The use of some cryptographic technique as visual cryptography and steganography minimizes detailed information sharing between consumer and online merchant but enable successful fund transfer from consumer's account to merchant's account thereby safeguarding consumer information and preventing misuse of information at merchant's side.

Steganography [4] is the process of masking sensitive information in any media to transfer it securely over the underlying unreliable and insecure communication network, whereas Visual cryptography [4] (VC) is a method of encrypting a secret information into shares such that stacking a sufficient number of shares reveals the secret image.

1.2 The research problem:

The online payment systems facing new and rapidly phishing or fraud techniques aim to expose customer's payment information and making security breach.

In other words problem statement can be stated as following question: "how to securing customer's payment information from being exposing or misused".

1.3 The objectives of research:

1. To secure the online payment system by applying mechanisms to restrict who see what.
2. To achieve security goals from performance point of view.
3. To applying anti-phishing approach by Certified Authority (CA) verification.

1.4 The methodology of research:

In this thesis the securing of an online payment system is based on the use of two encryption approaches these are secret sharing, AES encryption and steganography.

Firstly, the system generated two shares which is secure the customer's credential information, after that shares are encrypted and then the system generates two stego-image using steganography. Shares are created to minimizing the information shared between merchant and customer and to provide trustworthy to the merchant side.

This system is used third party as a CA to authorized merchant and its customers.

The implementation of the system is done using java programming language under Model View Controller (MVC) architecture.

1.5 The importance of research:

The importance of the study is to provide extra layer of security for customer's payment information by restricting access to that information internally between online payment parties or externally from others.

1.6 The boundaries of research:

This thesis is focusing on how to achieve secure online payment transaction firstly, reducing customer's payment details by restricting access to the customer's information, achieving security goals with suitable performance by using superior encrypting techniques finally, applying Certified Authority approach to add privacy to customer and trustworthiness to the system.

1.7 The content of research:

The research contains the following chapters:

Chapter One: Introduction.

Chapter Two: The state of arts.

Section one: this section discusses the background of some techniques that use in the field of securing data.

Section two: related works in securing e-payment systems.

Chapter Three: Methodology, Techniques and Tools.

Chapter Four: Results and Discussion.

Chapter Five: Conclusion and Recommendations.

Finally, References of the research.

CHAPTER II

RELATED WORK AND LITERATURE REVIEW

2. Overview:

This chapter provides brief overview about popular techniques that use for security issues those are cryptography, visual cryptography and steganography.

The next part discusses some related studies in the field of e-payment and securing e-payment transaction mechanisms.

2.1 Introduction:

Cryptography is probably is the most important aspects of communications security. It concerns with secret writing with goal of hiding meaning of the message.

Encryption is primary method of protecting valuable electronic information as the modern form of cryptography [5]; the common forms of encryption are symmetric and asymmetric cipher.

Cryptography goals:

Confidentiality: Unauthorized parties can't access the information.

Authenticity: Validating the source of the message to insure the user is identifying.

Integrity: Assurance that the message was not identified during the transmission.

Non-repudiation: A sender can't deny sending the message at later date [6].

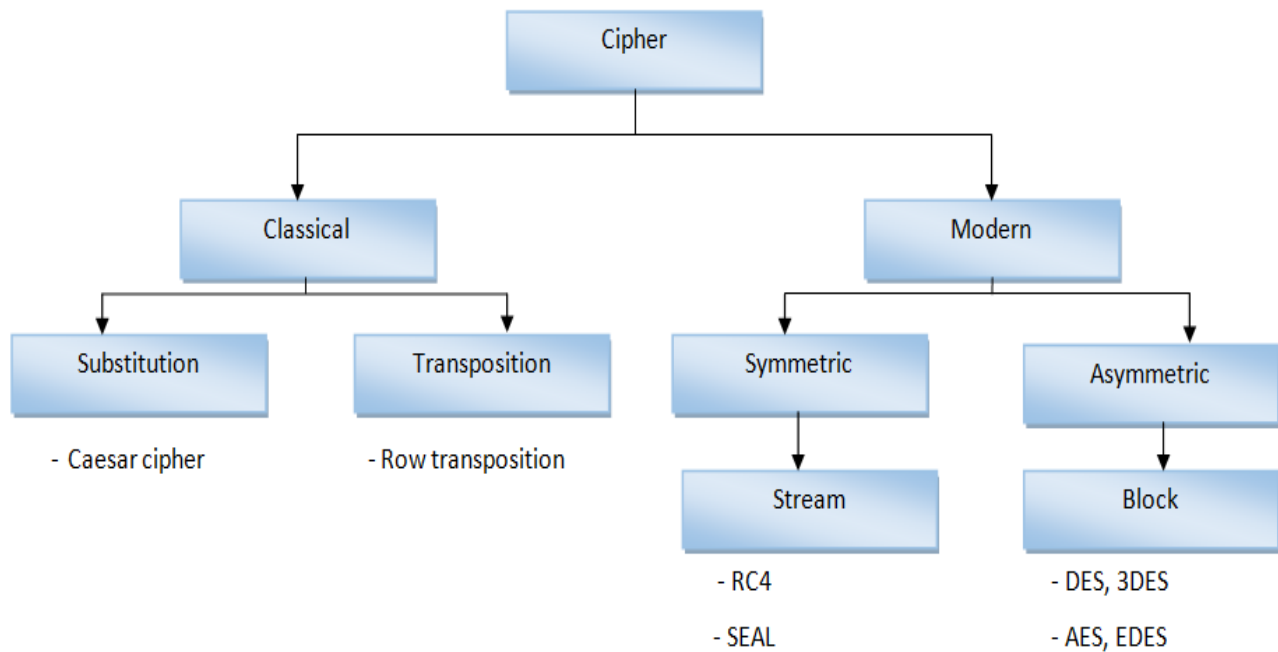


Figure 2.1: cipher model classifications

Above figure illustrated the classifications of cipher models with some examples.

2.1.1 Substitution techniques:

This part examines a sampling of what might be called classical encryption techniques. Two building blocks of all encryption techniques are substitution and transposition.

Substitution technique is one in which the letters of plaintext are replaced by other letters, numbers or by symbols [5].

2.1.1.1 Caesar cipher:

The earliest known use of substitution cipher is the Caesar cipher algorithm, it is simplest way introduced by Julius Caesar. It involves replacing each letter of the alphabet with letter standing three places further down the alphabet [5]. For example:

Plaintext: meet me after the toga party
Cipher text: PHHW PH DIWHU WKH WRJD SDUWB

Then the algorithm can be expressed as flow:

For each plaintext P, substitute the cipher text C:

$$C = E(3, P) = (P + 3) \bmod 26$$

Shift may be for any amount, so that the general Caesar algorithm is:

$$C = E(k, P) = (P + k) \bmod 26$$

Where K take value in the range 1 to 25. The decryption algorithm is:

$$P = D(k, C) = (C - k) \bmod 26$$

2.1.1.2 Playfair cipher:

With only 25 possible keys Caesar cipher is far from secure. The best-known multiple letter encryption cipher is playfair, which treats digrams in the plaintext as single units and translates these units into cipher digrams.

The playfair algorithm is based on the use of a 5 x 5 matrix of letters constructed using a keyword. The matrix is constructed by filling in the letter of keyword from left to right and from top to bottom, and then filling in the remainder of the matrix remaining the letters in alphabetic order.

2.1.2 Transposition techniques:

In transposition cipher the plaintext remains the same, but the order of the characters is shuffled around. The plaintext is written horizontally onto a piece of graph paper of a fixed width and the cipher text is read off vertically [7].

Decryption is the matter of writing the cipher text vertically onto a piece of graph paper of identical width and then reading the plaintext off horizontally.

For example, columnar transposition:

Key:	4	3	1	2	5	6	7
Plaintext:	a	t	t	a	c	k	p
	O	s	t	p	o	n	e
	D	u	n	l	t	l	t
	W	o	a	m	x	y	z

Cipher text: TTNAAPIMTSUOAODWCOTXPETZ

2.2 Symmetric model:

Symmetric encryption is the term that means transforms plaintext into cipher text using secret key and encryption algorithm.

By the using the same key and decryption algorithm the plaintext is recover from cipher text [5] anyone who find the key able to extract the hidden message.

Symmetric encryption ingredients:

1. Plaintext: this is the original message or data, is fed into encryption algorithm as input.
2. Encryption algorithm: algorithm that performs various substitution and transformation on plaintext.
3. Secret key: is the key that substitution and transformation of plaintext depend on it.
4. Cipher text: this is scramble message produced as output.
5. Decryption algorithm: is essentially the encryption algorithm run in reverse it takes the cipher text and the same secret key and produced the plaintext.

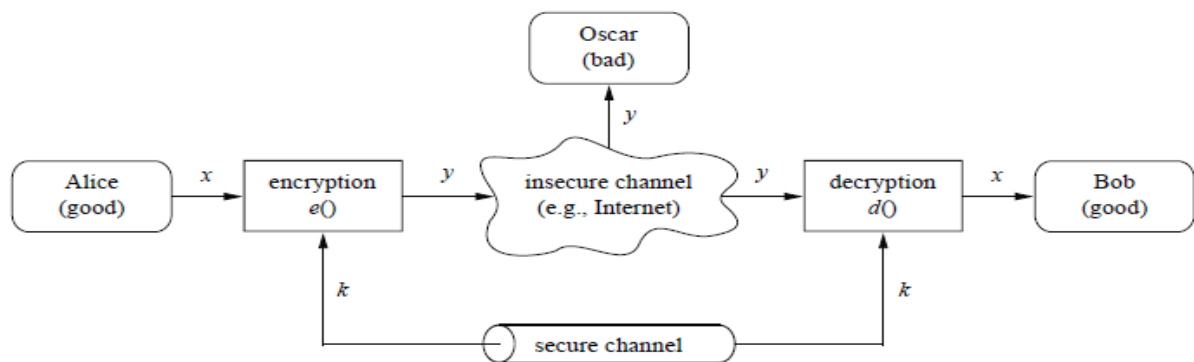


Figure 2.2: symmetric key cryptosystem

X: is called plaintext or clear text, Y: is called cipher text and K: is called key (this key is used for encryption and decryption).

Figure above illustrates the sending secret message from Alice to Bob using symmetric key model.

There are classes of symmetric key encryption schemes which are commonly distinguished: block cipher and stream cipher.

2.2.1 Block cipher model:

Block cipher is an encryption scheme which breakup the plaintext message to be transmitted into strings (called blocks) of fixed length over an alphabet and encrypt one block at a time [8]. This model is important element in many encryption systems because it's providing confidentiality.

2.2.1.1 Data Encryption Standard (DES):

Data Encryption Standard is the most well-known symmetric key block cipher. It set precedent in the mid of 1970s as the first commercial grade modern algorithm with openly and fully specified implementation details. The design of DES is related to two general concepts: product ciphers and Feistel ciphers. Each evolves iterating a common sequence or round of operation [8].

DES is Feistel cipher which processes plaintext blocks with 64 bits, producing 64 bits cipher text blocks and it use secret key with the size of 56 bits and 16 identical operation called round with some functions.

The strength of DES is based on Feistel structure which is depending on substitution and transposition processes to provide confusion and diffusion of message.

At encryption process the input is divided into two halves with 32 bits block size, the right half and secret key use as input of a round function and the output of that function will substituted to be the left half, these processes are done 16 rounds to provide more strength for encryption process.

DES the symmetric cipher so the same secret key and the same algorithm with reverse steps is used for decryption process.

2.2.1.2 Advanced Encryption standard (AES):

AES is the symmetric block cipher that is intended replaces DES as the approved standard for wide range of applications.

AES increase in block size from the old standard of 64 bits up to 128 bits and keys from 128 to 256 bits. This has been driven the public demonstrations of exhaustive key searches of

DES. The triple DES is secured and understood algorithm but it's slow in software. From several algorithms NIST (National Institute for Standard and Technology) select Rijndael as the proposed AES algorithm.

AES is encrypting data as sequence of 4 blocks each of which with 4 bytes. Encryption process isn't same as decryption process so we needed two software and application for encryption and decryption processes [5].

Rijndael was design to resistance against all known attacks, speed, code compactness on a wide range of platforms and design simplicity [9].

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six times faster than triple DES.

The features of AES are as follows:

1. Symmetric key symmetric block cipher
2. 128-bit data, 128/192/256-bit keys
3. Stronger and faster than Triple-DES
4. Provide full specification and design details
5. Software implementable in C and Java

2.2.1.2.1 Operation of AES:

AES is an iterative rather than Feistel cipher. It is based on 'substitution-permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix.

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration:

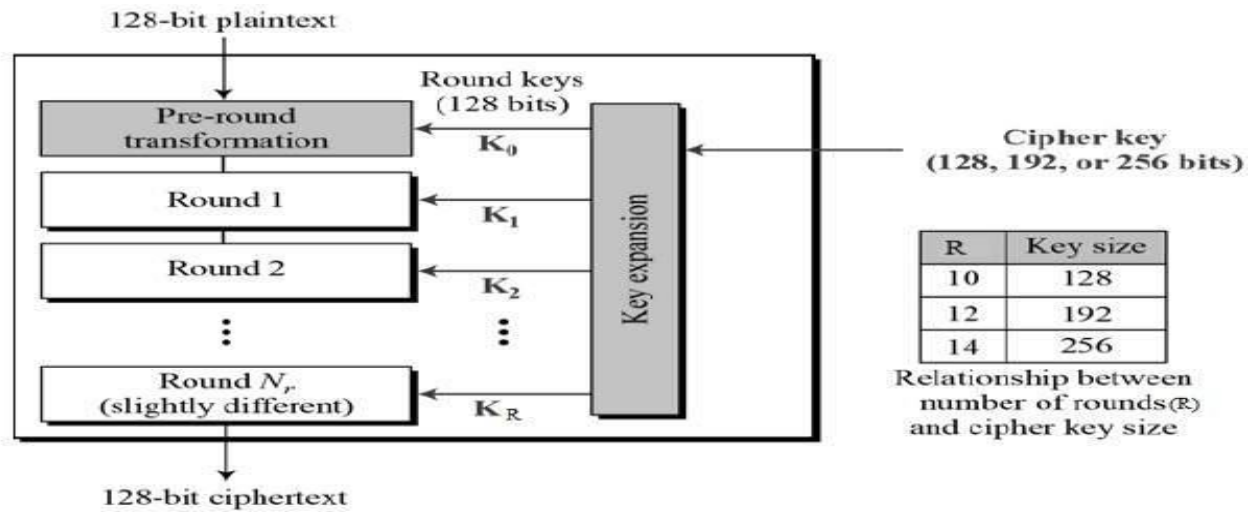


Figure 2.3: AES structure

2.2.1.2.1.1 Encryption Process:

Here, we restrict to description of a typical round of AES encryption. Each round comprises of four sub-processes. The first-round process is depicted below:

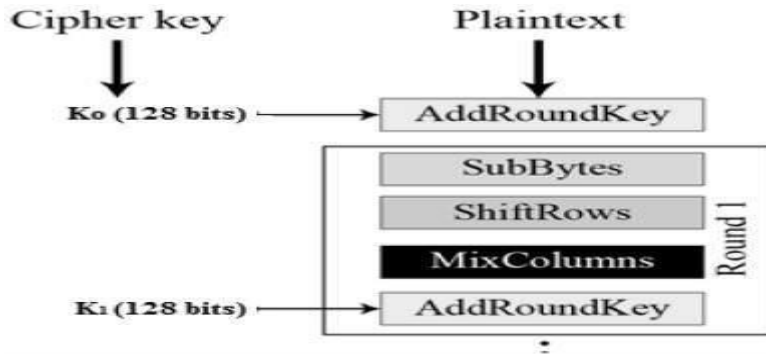


Figure 2.4:

First round process

Byte Substitution (SubBytes):

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shift rows:

Each of the four rows of the matrix is shifted to the left. Any entries that ‘fall off’ are re-inserted on the right side of row. Shift is carried out as follows:

First row is not shifted; the second row is shifted one (byte) position to the left, third row is shifted two positions to the left and fourth row is shifted three positions to the left.

The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

Mix Columns:

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Add round key:

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

2.2.1.2.1.2 Decryption Process:

The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order: add round key, mix columns, shift rows and byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms need to be separately implemented, although they are very closely related [13].

2.2.2 Stream cipher:

Stream ciphers form an important class of symmetric key encryption schemes. They are in one sense, very simple block ciphers having block size equal to one. They encrypt individual characters of a plaintext one at a time.

Stream ciphers are advantages because they have no error propagation [8].

2.2.2.1 RC4 Algorithm:

RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security. It is a variable key-size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation.

Analysis shows that the period of the cipher is overwhelmingly likely to be greater than 10100 runs very quickly in software. RC4 is used in the SSL/TLS (Secure Sockets Layer/Transport Layer Security) standards that have been defined for communication between Web browsers and servers. It is also used in the Wired Equivalent Privacy (WEP) protocol and the newer Wi-Fi Protected Access (WPA) protocol. RC4 was kept as a trade secret by RSA Security. In September 1994, the RC4 algorithm was anonymously posted on the Internet on the Cypher punks anonymous remailers list.

The RC4 algorithm is remarkably simply and quite easy to explain. A variable-length key of from 1 to 256 bytes is used to initialize a 256-byte state vector S , with elements $S[0]$, $S[1]$,..., $S[255]$. At all times, S contains a permutation of all 8-bit numbers from 0 through 255. For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion. As each value of k is generated, the entries in S are once again permuted [5].

The strength of RC4 based on the same factors like reasonable key length, the difficulty of knowing which location on the table is used to select each value in the sequence, and encryption is about 10 times faster than DES and particular RC4 key can be used only once.

2.3 Asymmetric cipher model:

Asymmetric cryptography also known as public key cryptography uses public key and private key for encrypt and decrypt data. The keys are simply large numbers that have been paired together but are not identical. One key in pair can be shared with every one; it is called a public key. The other one key in the pair is kept secret, it is called private key.

Many protocols like SSH, open PGP and SSL/TLS relay on asymmetric encryption and digital signature functions, it is also used in software programs such as browsers which needed to establish secure connection over an insecure network or needed to validate a digital signature.

For asymmetric encryption to deliver confidentiality, integrity, authenticity and non-reputability, users and systems need to be certain that a public key is authentic, that is belongs to the person or entity claimed and that it has not been tampered with or replaced by a malicious third party, Public key infrastructure (PKI). Where trust certificates authorities certify ownership of key pair and certificates.



Figure 2.5: Public Key Cryptography

2.4 Steganography:

Steganography coming from Greek words stegos, meaning roof or covered and graphia which means writing is the art and science of the fact that communication is taking place. Using steganography, you can embed a secret message inside a piece of unsuspecting information and send it without anyone knowing of the existence of the secret message [11].

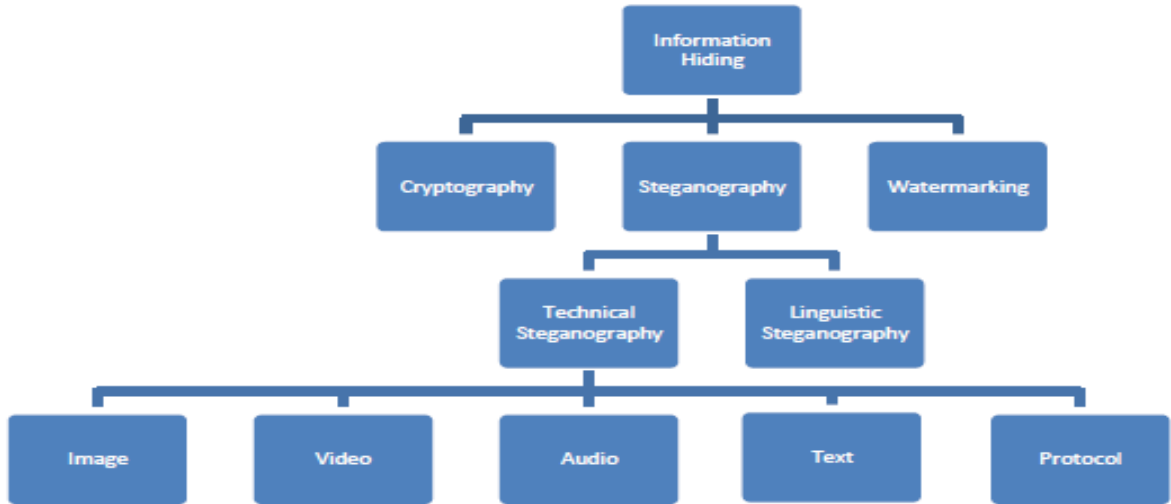


Figure 2.6: information hiding classification

Hiding information into a media requires some elements:

Cover media (C): that will hold the hidden data.

Secret message (M): may be plaintext, cipher text or any type of data.

Stego function (F): and its inverse (F^{-1}).

Stego key (K): used to hide and unhide message.

The stego function operates over cover media and the message along with a stego-key to produce a stego media (S).

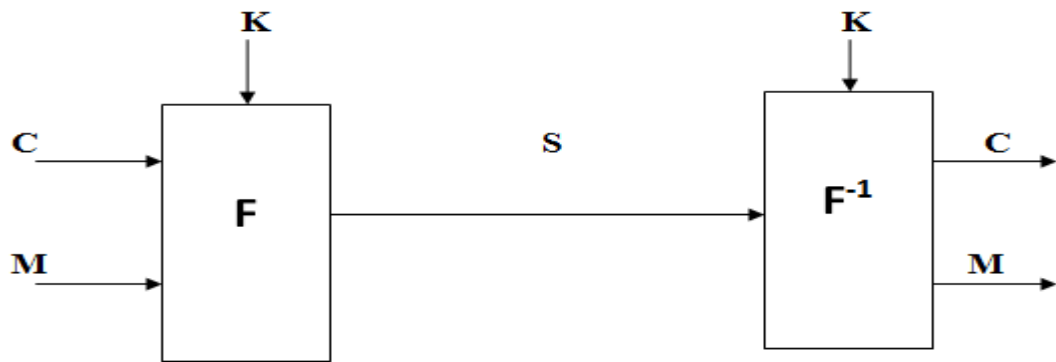


Figure 2.7: steganographic operation

2.4.1 Modern techniques of steganography:

The common modern techniques of steganography exploit the property of the media itself to convey a message.

2.4.1.1 Plaintext steganography:

In this technique the message is hidden within a plaintext file using different schemes.

Use of selected character of cover text:

Sender sends series of integer number (key) to recipient with a prior agreement that the secret message is hidden within the respective position of subsequent words of cover text.

For example:

The series is: '1, 1, 2, 3, 2, 4'

The secret text is: "a team of five men joined today"

The hidden message is "a t f v o a"

Use of extra white space characters of cover text:

A number of extra blank spaces are inserted between consecutive words of covert text. This numbers are mapped to a hidden message through an index of a lookup table.

2.4.1.2 Image steganography:

The most widely used technique today is the hiding secret messages into a digital image. A picture can be represented by a collection of color pixels. The individual pixels can be represented by their optical characteristics like brightness. Each of these characteristics can be presented in terms of 1s or 0s.

In picture each pixel represents three values (red, green, blue) as 8 bits the change in one bit from 0 to 1 and vice versa is undetectable by human visual system (HVS) this way can be used as carrier of hidden message.

2.4.1.2.1 Modification of LSB of cover image in bitmap format:

In this method a binary equivalent of the message -to be hidden- is distributed among the LSBs of each pixel.

For example:

When try to hide the character 'A' into 8-bit color image we take 8 consecutive pixels from left corner of the image. The equivalent binary bit pattern of those pixels like this:

```
00100111  11101001      11001000  00100111      11001000      11101001
11001000  00100111
```

Then each bit of binary equivalence of letter 'A' i.e. **01100101** are copied (from left to right) to the LSB's of equivalent binary pattern of pixel like this:

```
00100110  11101001      11001001  00100110      11001000      11101001
11001000  00100111
```

The problem of this technique is that it's very vulnerable to attack such as image compression and formatting.



Figure 2.8: image steganography

In this figure the left image is the original cover image whereas right one dose embedding a text file into the cover image makes the stego-image is undetectable by HVS.

2.4.1.3 Audio and Video steganography:

In audio steganography secret message is embedded into digitalized audio signal which result slight altering of binary sequence of the corresponding audio file. Several method of audio steganography is available like:

2.4.1.3.1 LSB coding:

Sampling technique followed by quantization, converts analog audio signal to digital binary sequence.

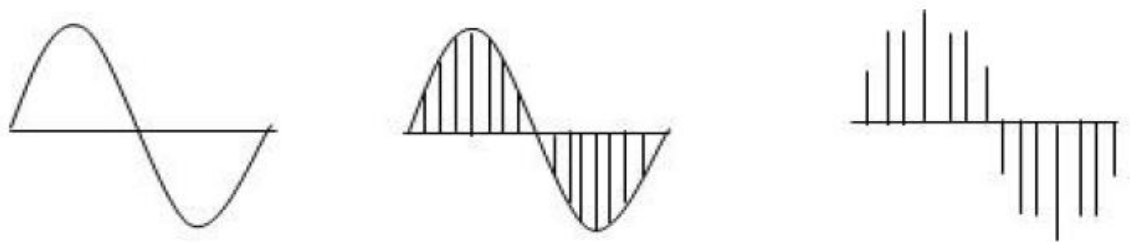


Figure 2.9: LSB sampling

Figure 2.7 illustrate the sampling of the sine wave followed by quantization process.

In this technique LSB of binary sequence of each sample of digital audio file is replaced with binary equivalent of secret message.

2.4.1.4 IP datagram steganography:

The technique which employs hiding data in the network datagram level in a TCP/IP based network like internet.

In this approach information to be hide is placed in IP header of TCP/IP datagram. Some of the fields of IP header and TCP header in an IPV4 are chosen for data hiding.

Version	Hd. Len.	TOS	Total Packet Length	
Identification			flags	Fragment Offset
TTL	Protocol		Header Checksum	
Source IP Address				
Destination IP Address				
Options				Padding

Figure 2.10: IPv4 Header

2.4.1.4.1 Cover channel communication using identification:

The 16bit identification field in IPv4 header is used to identifying the fragment packet of an IP datagram; if there is no fragmentation of datagram, then this identification field can be used to embed sender specified information.

2.4.2 Steganalysis:

Steganalysis is the process of identifying steganography by inspecting various parameters of stego media.

In steganalysis the suspected media may or may not be with hidden message. The steganalysis process start with set of suspected information streams.

2.4.3 Steganography attacks:

Steganographic attacks consist of deleting, extracting and destroying hidden object of stego media, based on the information available for analysis there are several types of attack:

1. **Known carrier attack:**

The original cover media and stego media are available for analysis.

2. **Steganography only attack:**

In this type only stego media is available for analysis.

3. **Known message attack:**

Hidden message is known in this case.

4. **Known steganography attack:**

The cover media, stego media as well as steganography algorithm or tool are known.

2.5 Visual cryptography (VC):

Visual cryptography is the new type of cryptographic scheme that allows us to share secret between numbers of trusted parties effectively [16].

The main instantiation of VC realizes a cryptography protocol called secret sharing (SS). In a conventional SS scheme, a secret data is shared among n participants in such a way that subsets of qualified participants can pull their shares and recover the secret but subsets of forbidden participants can obtain no information about it [12].

Visual cryptography (VC), proposed by Naor and Adi Shamir [13], is a method for protecting image-based secrets that has a computation-free decryption process.

The main concept of original visual cryptography scheme is encrypting the secret image into shares. Secret information cannot be revealed with a few shares, all shares are necessary to combined to reveal the secret image. Visual cryptography is simple, secure and effective cryptographic scheme [14].

2.5.1 Traditional Visual Cryptography schemes (VCS):

2.5.1.1 Traditional secret sharing Scheme:

A secret is something which is kept from knowledge of any, but for initiated or privileged. In secret sharing a secret can be distributed between a group of participants, each participant is allocated a piece of secret called share. The reconstruction of the secret can be possible when sufficient numbers of shares are combined together.

The traditional secret sharing scheme which was produced by Shamir [15] and Blakley [16] is actually, separated. Concerning the nature of the secret sharing scheme, it is supposed to be a bank vault that must pass by a secret key. The bank has three tellers, but the system of bank does not trust any of them individually. Therefore, they must be to design a system such that any two of the three tellers can pass the vault together. This problem can be known as a (2, 3) secret sharing scheme.

- (k, n) Secret sharing scheme is a method to share a Secret k amongst t participants one for each participant such that the following conditions should be hold: 1- No participant knows the share given to another participant. 2- k together can reveal the secret data by superimposing k (transparencies). 3- Any t transparencies, $t < k$, the secret cannot be decoded.

2.5.1.2 Visual Secret Sharing Scheme (VSSS):

In the Visual Secret sharing scheme, there is a secret picture to be shared among n participants. The picture is divided into n transparencies (shadows) such that if any m transparencies are placed together, the picture becomes visible. However, if fewer than m transparencies are placed together, or analyzed by any other means; nothing can be seen. Visual Secret Sharing scheme uses mathematical secret sharing but implements in hardware, printed on transparencies. It once created, it requires no technology, and however resolution and contrast are lost. Depending on how the decryption of secret image, the difference between a VSSS and a traditional secret sharing scheme is usually, the traditional secret sharing scheme requires computation over a finite field. In a VSSS, however, the computation is simply performed by the human visual system of the users.

To construction of a secure VSSS is difficult. Assume that a particular pixel P on a share s_i is black. Whenever a set of shares (including s_i) is stacked together, the result must be black. It means that in the secret image, the pixel P must be black. That is mean that by

examining one of the shares we can get some information about the secret image, but the security condition does not allow this. Naor and Shamir [13] introduced a VSSS that solved this problem by expanding each original pixel into m sub pixels. Suppose that the image secret is a collection of black and white pixels, or a binary image, and each pixel is encrypted individually. Each original pixel encodes into n shares, and each share is a set of m black and white sub pixels, which are printed near to each other such that human visual system averages their individual black/white contribution.

Visual Secret Sharing is based on the access structure schemes specified as follows:

2.5.1.2.1 2 out of 2 Scheme (2 sub pixels):

In this scheme, a binary image pixel is divided into two sub-pixels out of which black or white is randomly chosen depending on the current pixel as following in figure:

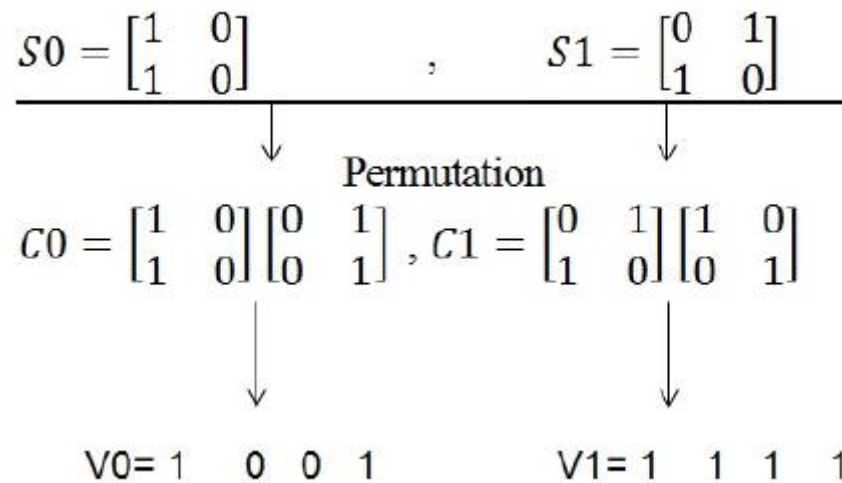


Figure 2.11: a random column-permutation of white pixel and black pixel is done from S0 and S1

If the image pixel is white, then chose one of the two rows for white, else if it is black, then choose between one of the two rows for black. A random column-permutation of white pixel and black pixel is done from S_0 and S_1 to generate a given matrix as C_0 and C_1 represents white and black pixel respectively which produces two vectors V_0 and V_1 corresponding to occurrence of the pixel i.e. either white or black in a secret image as shown in figure (2.11). If the pixel is white then V_0 will be outputted which is of either 1 0 or 0 1 with gray-level $\frac{1}{2}$ and if black pixel then V_1 will be outputted with either [1 1] or [1 1].

2.5.1.2.2 2 out of 2 Scheme (4 sub pixels):

In this scheme, each black/white pixel is encoded as a 2x2 cell into two shares where each share has 2 black, 2 white sub pixels. When stacked, shares combine to give solid black or half black (seen as grey) as shown in figure2.10.

$$C_1 = \left\{ \begin{bmatrix} 0101 \\ 1010 \end{bmatrix} \begin{bmatrix} 1010 \\ 0101 \end{bmatrix} \begin{bmatrix} 0011 \\ 1100 \end{bmatrix} \begin{bmatrix} 1100 \\ 0011 \end{bmatrix} \begin{bmatrix} 0110 \\ 1001 \end{bmatrix} \begin{bmatrix} 1001 \\ 0110 \end{bmatrix} \right\}$$

$$C_0 = \left\{ \begin{bmatrix} 0101 \\ 0101 \end{bmatrix} \begin{bmatrix} 1010 \\ 1010 \end{bmatrix} \begin{bmatrix} 0011 \\ 0011 \end{bmatrix} \begin{bmatrix} 1100 \\ 1100 \end{bmatrix} \begin{bmatrix} 0110 \\ 0110 \end{bmatrix} \begin{bmatrix} 1001 \\ 1001 \end{bmatrix} \right\}$$

Figure 2.12 C0 and C1 are the two column matrices of white and black pixel respectively which is selected on random basis.

2.5.1.2.3 3out of 3 Scheme:

This scheme encodes the secret image into three shares based on pixel expansion such that when all the three shares are combined, the secret image will be revealed.

2.5.1.3 Visual cryptography for general access structures:

The drawback of (k, n) Basic model that is any " k " shares will decode the secret image which reduces security level. To overcome this issue G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson [17] extended the basic model to general access structures by, where an access structure is a specification of all qualified and forbidden subsets of 'n' shares. Any subset of ' k ' or more qualified shares can decrypt the secret image but no information can be obtained by stacking lesser number of qualified shares or by stacking disqualified shares. Construction of k out of n threshold visual cryptography scheme for general access structure is better with respect to pixel expansion.















To illustrate the idea behind the extension of basic model to general access structure, suppose that a bank has a vault. In this time, the bank employs three senior tellers and a manager. They would like to design a system such that one of the three senior tellers together with the manager can open the vault. However, two of the three senior tellers cannot obtain the permission. This problem can be viewed as a general access structure scheme. In (k, n) basic model, the secret image is decrypted by stacking any k shares together. In a general access structure, however, we can specify some qualified subsets of shares that can decrypt the secret image, but other forbidden subsets of shares have no information about it.

2.5.1.4 Black and White Visual Cryptography Schemes:

2.5.1.4.1 Sharing Single Secret:

Naor and Shamir's [13] introduced encoding scheme to share a binary image into two shares Share1 and Share2. If pixel is white one of the above two rows of the table (2.1) is chosen to generate Share1 and Share2. Similarly, if pixel is black one of the below two rows of the following table is chosen to generate Share1 and Share2. Here each share pixel p is encoded into two white and two black pixels each share alone gives no clue about the pixel p whether it is white or black. Secret image is shown only when both shares are superimposed.

Table 2.1: Naor and Shamir's scheme for encoding to share a Binary image into two shares

Pixel	Probability	Share ₁	Share ₂	Share ₁ ⊗ Share ₂
	50%			
	50%			
	50%			
	50%			

The disadvantage of the above schemes is that only one set of confidential messages can be embedded, so to share large amounts of confidential messages several shares have to be generated.

2.5.1.5 Gray Level Visual Cryptography Scheme:

Previous works done in visual cryptography were restricted to binary images which is insufficient in real time applications. Visual cryptography for gray level images by dithering

techniques was suggested by Chang- ChouLin, Wen-Hsiang Tsai [18]. Instead of using gray sub pixels directly to constructed shares, a dithering technique is used to convert gray level images into approximate binary images. Then existing visual cryptography schemes for binary images are applied to accomplish the work of creating shares. The effect of this scheme is still satisfactory in the aspects of increase in relative size and decoded image quality, even when the number of gray levels in the original image still reaches 256.

2.5.1.6 Analysis of Visual Cryptography:

2.5.1.6.1 Robustness in Visual Cryptography:

Traditional visual cryptography schemes typically use black and white pixels to represent an image in its binary format. These black and white pixels are very resilient due to the fact that white pixels will always be white and black pixels will always be black. There is no change in potential pixel values after the image has been altered or tampered with.

Binary images are very robust against attacks which are commonly used on images. Such attacks come in the form of image resizing, cropping, scaling, skewing and compression. After these attacks, the black pixels remain black and the white pixels remain white. There are no intermediate values that these pixels can take; therefore, binary images are a very good choice when it comes to protecting specific types of data.

Scaling, cropping and image compression could also be attack vectors from the point of view of making the secret leak out.

2.5.1.6.2 Security in Visual Cryptography:

The security within VC, as with a lot of cryptographic schemes is heavily based upon randomness [12]. This is the core security feature within visual cryptography. This means that while the shares are separate, no cryptographic analysis will yield the original secret based on analysis of one of the shares.

2.5.1.6.3 Complexity within VC:

Many of the proposed schemes within VC result in share sizes that grow very large, depending on the image type and size. Typically, as the contrast improves, the share size also increases quite dramatically. This increases an image processing time which increases the overall complexity of the schemes.

2.5.2 Color Visual Cryptography Scheme (CVCS):

Though some researches in field of visual cryptography applied on the binary images, it left some limitations in the quality of the decoded binary images, which makes it inapplicable for protection of color image. As a result of these limitations, F.Liu, C.K. Wu X.J. Lin [19] introduced a new approach on visual cryptography for colored images. They suggested three approaches as follows:

1. The first approach to realize color VCS is to print the colors in the secret image on the shares directly similar to basic model. It uses larger pixel expansion which reduces the quality of the decoded color image.
2. The second approach converts a color image into black and white images on the three-color channels (red, green, blue or equivalently cyan, magenta, yellow), respectively, and then apply the black and white VCS to each of the color channels. This results in decrease of pixel expansion but reduces the quality of the image due to halftone process.
3. The third approach utilizes the binary representation of the color of a pixel and encrypts the secret image at the bit-level this results in better quality but requires devices for decryption.

2.5.3 Extended Visual Cryptography Scheme (EVCS):

Extended visual cryptography schemes (EVCS) allow the construction of visual secret sharing schemes within which the shares are meaningful as opposed to having random noise on the shares. After the sets of shares are superimposed, this meaningful information disappears and the secret is recovered.

2.5.3.1 Halftone Visual Cryptography Scheme (HVC):

Mizuho Nakajima and Yasushi Yamaguchi [20] introduced extended visual cryptography which was of poor quality of the generated meaningful shares which again increases the suspicion of data encryption. Halftone Visual Cryptography introduced in [21], [22], [23] which increases the quality of the meaningful shares, it actually, based upon the basis matrices collections available in conventional visual cryptography. A secret binary pixel p is encoded into an array of $q=v_1*v_2$ called a halftone cell, in each of the n shares. The selection of the secret information pixels (SIPs) in a halftone cell is important as it affects the visual quality of the resultant halftone shares. However, as long as the positions of the secret information pixels are independent of the secret information, the arrangement of the modified

pixels satisfies the security requirements. By using halftone cells with an appropriate size, visually pleasing halftone shares can be obtained. Also maintains contrast and security.

2.5.4 Dynamic Visual Cryptography Scheme (DVCS):

The core idea behind dynamic visual cryptography is increasing the overall capacity of a visual cryptography scheme. This means that using a set of two or more shares can potentially hide two or more secrets. Multiple secret sharing is very useful when it comes to hiding more than one piece of information within a set of shares [12].

2.5.5 Applications for Visual Cryptography:

2.5.5.1 Moiré Patterns:

Moiré patterns [12] are induced when a revealing layer such as a dot screen or line grating is superimposed on top of a periodically repeating shape.

The revealing layer contains horizontal black lines; between those lines is transparent white space. When the revealing layer is superimposed, the shapes that appear are the magnified versions of the repeating pattern. This magnifying property could be used as a method of locating hidden VC shares within a Moiré pattern.

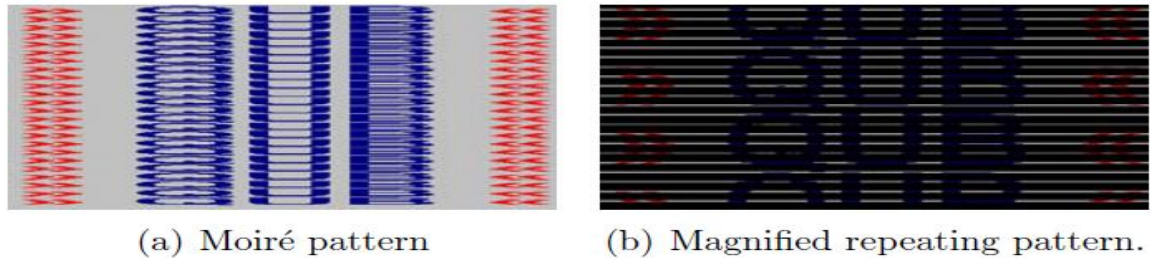


Figure 2.13: Example of Moiré patterns.

Visual cryptography has been implemented using Moiré patterns. Desmedt and Le [24] provide a scheme by which secrecy and anonymity are both satisfied. Moiré patterns occur when high frequency lattices are combined together to produce low frequency lattice patterns. It is the difference in these high frequencies that give the Moiré patterns. Figure 7.2 shows an example of generation these Moiré patterns.

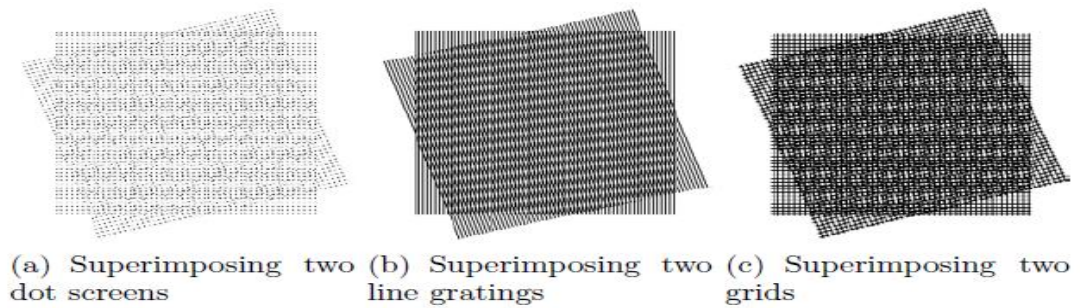


Figure 2.14: Generation of Moiré patterns.

The Moiré cryptography model is as follows: The embedded secret image is randomized into two shares, known as pre-shares. These are independent of the original image. XORing these pre-shares will recover the original. Next, the hiding algorithm takes the cover image and combines it with each of the pre-shares separately. Its output is the original two shares that are used to reveal the original embedded image. These resulting shares look the same as the input cover image that is used.

2.5.5.2 Watermarking:

Watermarking in actual fact is an important content in information hiding, however it emphasizes the little tags consisting of random binary numbers.

Watermarking is grouped into two basic categories according its imperceptible to human visual system: visible and invisible. Visible watermark such as logo can be seen on the visual media such as images, photos and videos. Although invisible watermarks cannot be visually touched, invisible watermarking is the validate way to identify original authors, ownerships, distributors. They all are applied to protect the ownership.

2.5.5.2.1 Watermarking in Visual Cryptography:

Practical uses for visual cryptography come in the form of watermarking. Memon and Wong [25] propose various techniques by which these watermarks can be applied to images. A simple watermark insertion scheme is watermark embedded within the least significant bit. This scheme is not robust because it could easily be destroyed. A more robust scheme should be able to deal with loosely image compression, altering, and scanning. The idea of random noise [26] is employed on color images to make removal of the watermark very difficult. Cryptographic functions such as the MD5 hash have also been employed to improve the security features when it comes to embedding data within images.

2.6 Related studies:

Brief studies of related work in the area of securing online payment transactions based on steganography and visual cryptography is presented in this section.

2.6.1: Santosh Kumbhar, Saumya Sahu [27], proposed approach for secure an online transaction. Here information submitted by the customer to the online merchant is minimized by providing only minimum information that will only verify the payment made by a customer from its bank account by using central Certified Authority and combined application of steganography and visual cryptography.

Their securing transaction hidden an authentication password in connection to the bank is inside a cover text using the text-based steganography method. Customer authentication information (account no) in connection with merchant is placed above the cover text in its original form. Then a snapshot of two texts is taken. From the snapshot image, two shares are generated using visual cryptography, but text-based steganography not suitable for large data representation and it makes use of features of English Language like inflexion, fixed word order and use of periphrases for hiding data rather than using properties of a statement.

2.6.2: Vaishnavi Deshmukh, Dr. Alvi [28], proposed techniques used to achieve security by combing of Bit Plane Complexness Segmentation (BPCS) steganography and 2-out-2 visual cryptography and central certified authority, when customer adds the item to the card; customer will be entering the card number and unique authentication password. This information will be created as a stego image using BPCS Steganography this technique converts the carrier image (of any file-format) from PBC (Pure Binary Code) to CGC (Canonical gray Code) system and in Portable Network Graphics(.png) format. 2-out-2 algorithm of visual cryptography creates two shares out of the stego image -Customer's share and CA's share - CA browses user's share and generates the card number which is sent to the bank so as to extract the customer's PIN (de-steganography). Finally, fund will be transferred from the bank to the merchant with minimum customer's information these techniques encrypt image in (.png) format only with low performance.

2.6.3: Nikita Chaudhari, Priya Parate [29], this thesis introduces secure approach that used several techniques to avoid phishing and other kinds of attacks like Quick Response Code, On Time Password (OTP) and visual cryptography. This study suppose customer and merchant side have account in bank and bank verify them, when customer request item from the merchant server then the information of request sends to bank server to verify both of customer and merchant if verification is done correctly then bank server generate an OTP and applied VC process on it to generate two shares, one share send to merchant and other one sends to customer after that merchant server send its share to the customer to combine it with his share and generate an OTP if its correct then the payment process is done. Here merchant side will be misused customer information (more detail about customer).

2.6.4: Ketan Raju Kundiya, Ram Joshi [30], proposed system uses symmetric key cryptography for security issues. To add more security to secret sharing of the image, encryption is done before creation of shares. If intruder get all the shares, since secret image itself is encrypted, intruder might not get any of the information about secret image.

For maintain more strengthen security they applied Lossless image compression methodology before encryption that less redundancy than the original image so cryptanalysis is difficult. In this system secret information applied to image using steganography then compression process is done to prevent against cryptanalysis after that encryption process is applied and shares are generated and transmitted to enhanced security of data. This System supports gray scale as well as colored images encryption.

2.6.5: Suganya Devi, Srinivansan, vaishave [31], proposed system that used steganography, visual cryptography using Discrete Cosine Transformation (DCT) to secure information shared online. The secret information of customer is hidden in cover image using LSB algorithm using random hide technique based on seed key that find random pixel position to encode the message within, after that the system applied DCT on stego image that use to prevent against intruders to find the steganography type that been performed on image after that system generate two share using VC for secret image. This system wasn't prevented against phishing attack, and secret information may be misused.

CHAPTER III

METHODOLOGY, TECHNIQUES AND TOOLS

3. Overview:

This chapter focuses on the proposed solution from technical view, techniques and tools that used to achieve thesis goals.

This chapter has three parts: the first one about thesis methodology, second one for techniques that has been used and the last one for system tools.

3.1 PART ONE: METHODOLOGY:

The proposed framework provides solution to online payment process against security threats like phishing and credential information exposing.

The main idea is based on the use of secret sharing and certified authorities as trusted third party between customers-who purchase an item- and merchant-who, provide it-. The third party provides authentication and privacy for sensitive customer's information behind anti-phishing solution.

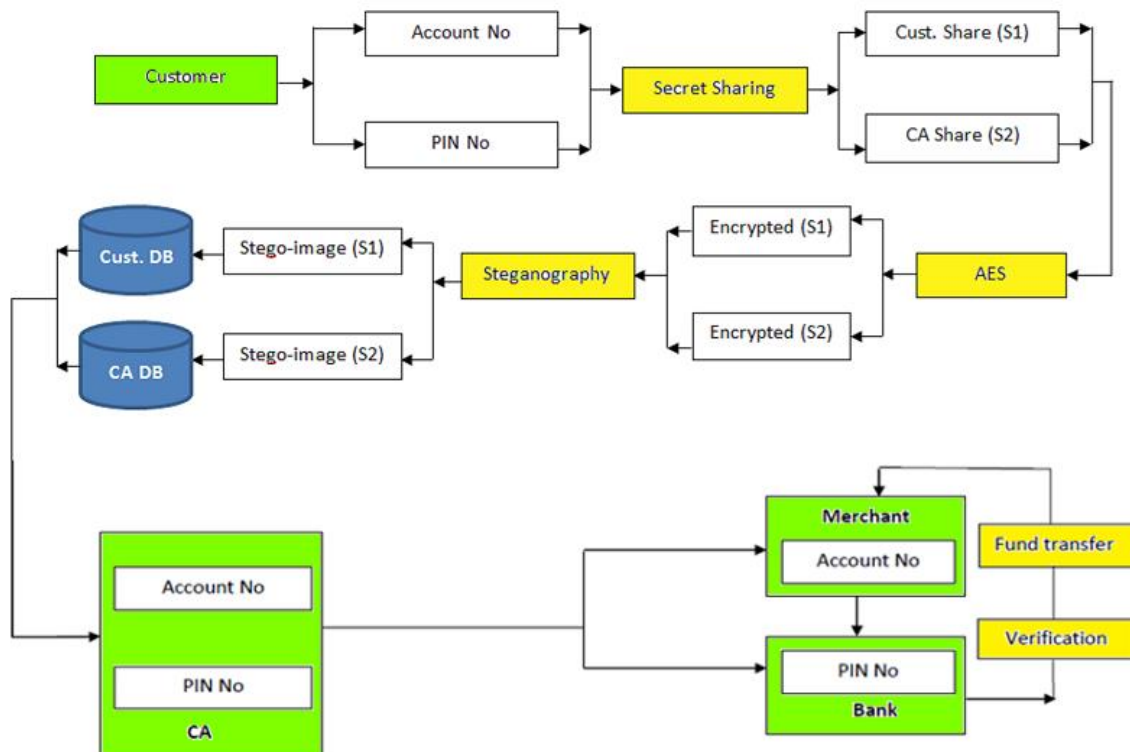


Figure 3.1: System's Block Diagram

The previous figure describes solution which will provide security for online payment transaction. When customer selects item(s) and filling payment info, the system generate two secret shares via secret sharing scheme and encrypt these shares using AES encryption algorithm to prevent shares from any potential internal attack like cheating attack, that may occurs because secret sharing scheme depend on N number of total number of threshold to recover the secret, after shares are encrypted the next process is hiding those shares using image steganography and two stego-images are generated. The steganography mechanism provides extra security layer against external attack like statistical attack. One of these images is send to CA and the other one is saving at customer DB.

In this approach the third party Certified Authority (CA) authorized the customer by getting customer's stego-image and extract the share and reconstruct two shares-customer (S1) and CA (S2) - to obtain the customer secret info, this step is verify the customer if reconstruction process is done in a right way that means the customer is authorized, after CA getting customer's credential info, CA send only the account number of customer to the merchant, and PIN number to the Bank. This mechanism provides least credential information needs to complete payment process at merchant side, provide privacy to customer's secret info and provide authority to customer at merchant, after this step verification code (4-digit number) is send to the customer's e-mail that refer to accomplishes of payment process. Finally, after the code verified then fund will be transfer from bank to merchant's account securely.

The following diagrams explain the functionality of the system.

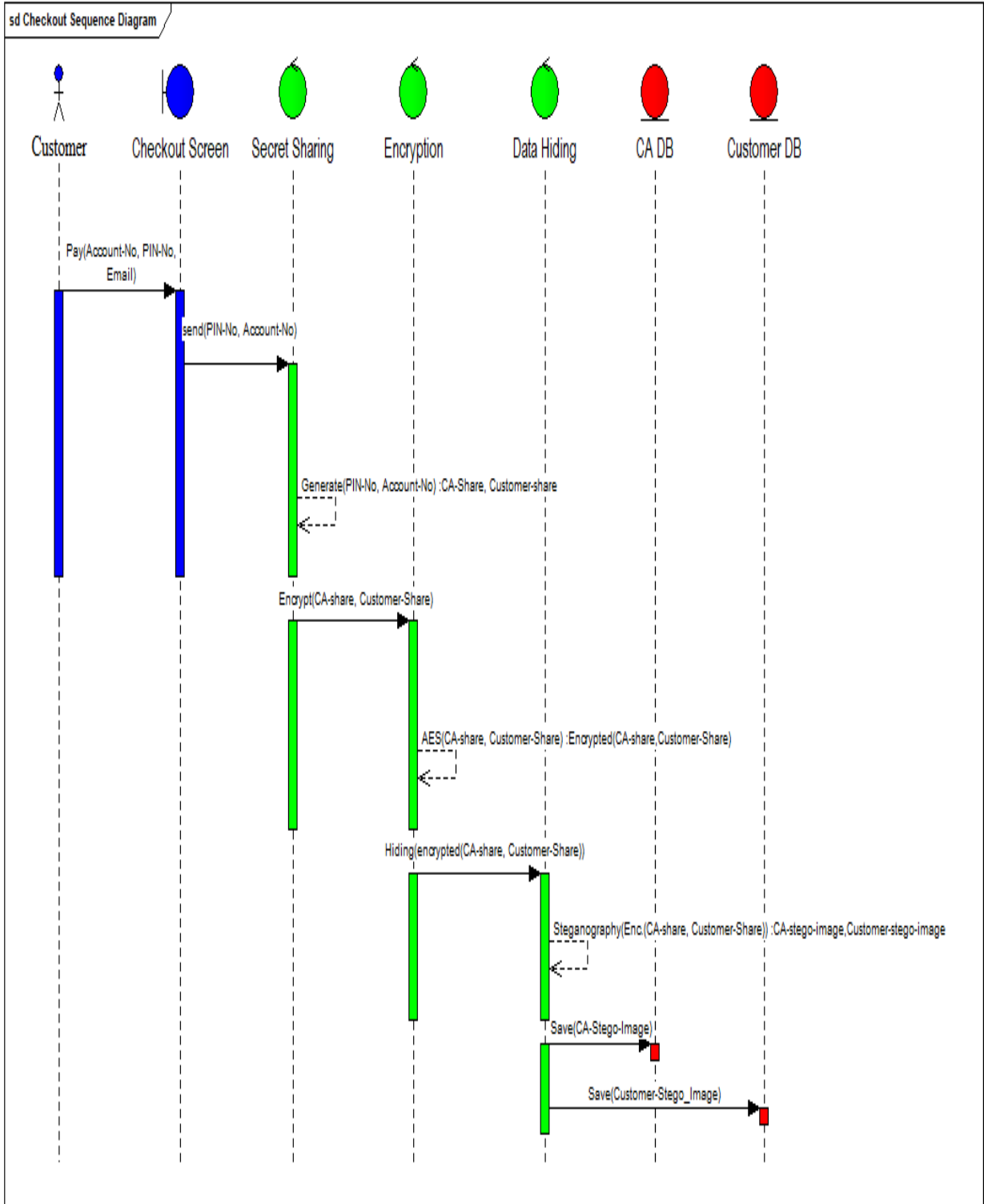


Figure 3.2: Sequence Diagram for Checkout Process

The above figure describe the process of securing the secret info of a customer by applying secret sharing algorithm, encrypt these shares using AES and finally, hiding them into images via steganography, after hiding process images ware saved into DB.

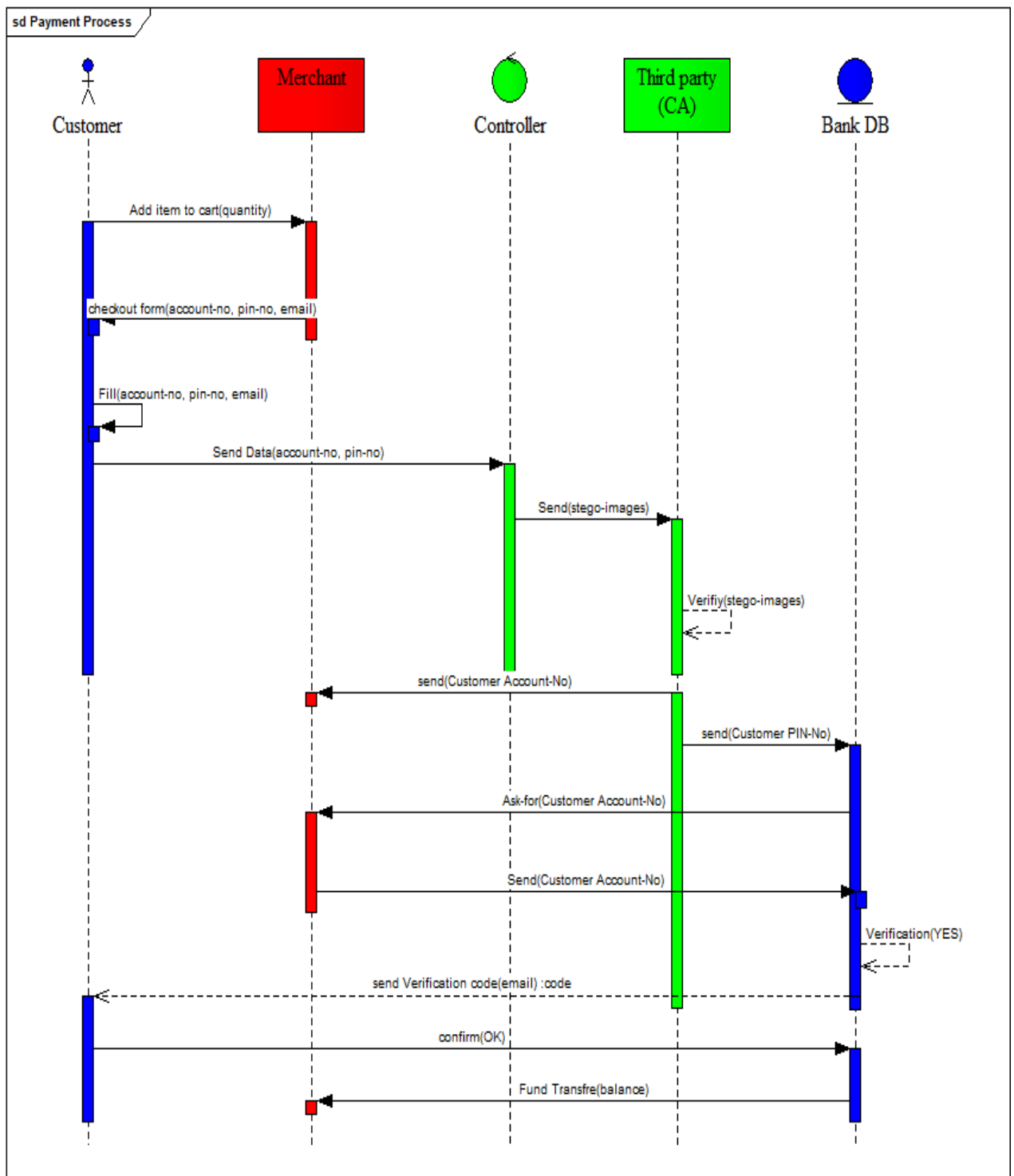


Figure 3.3: Sequence Diagram for Payment Process

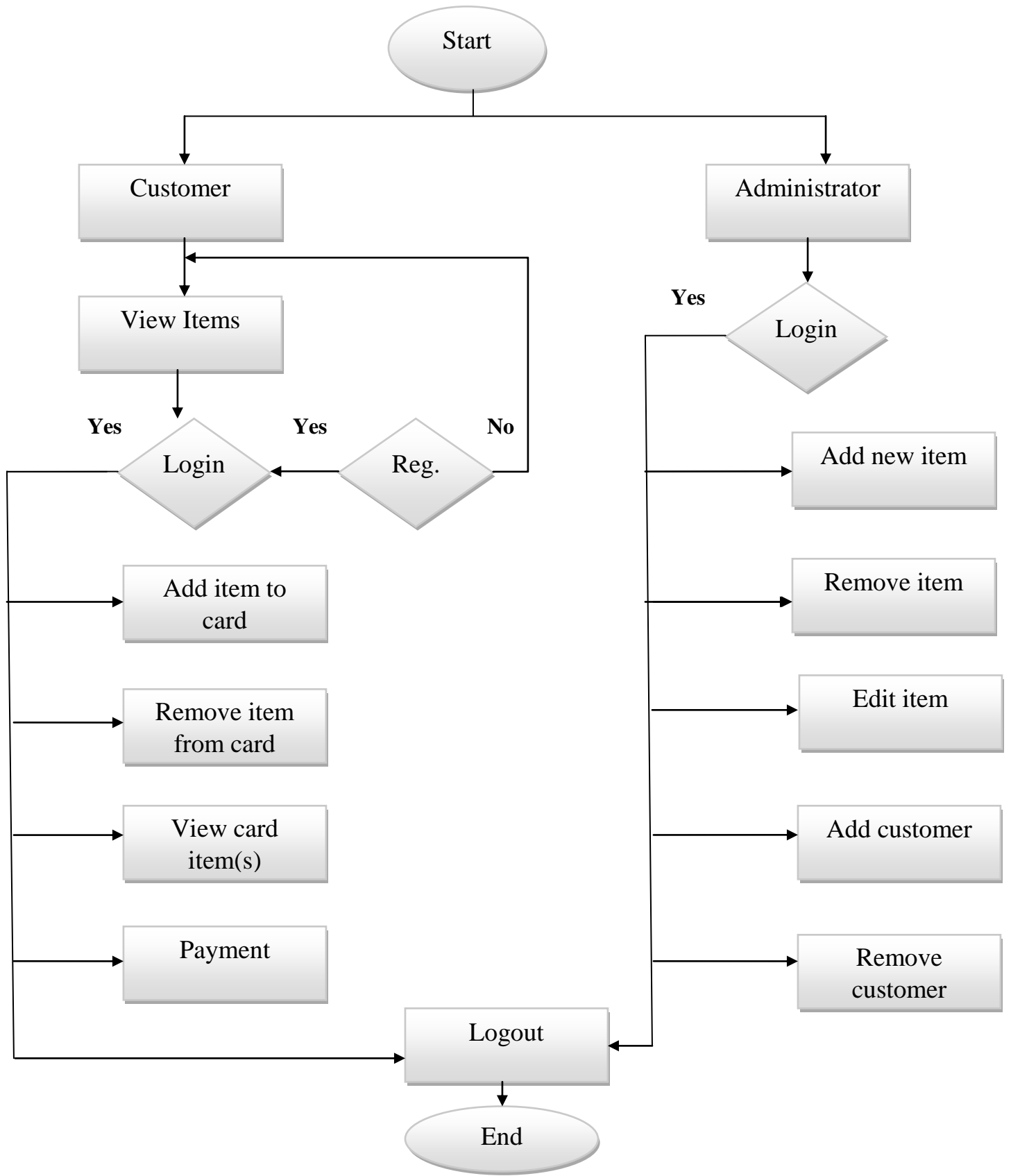


Figure 3.4: Flowchart for System processes

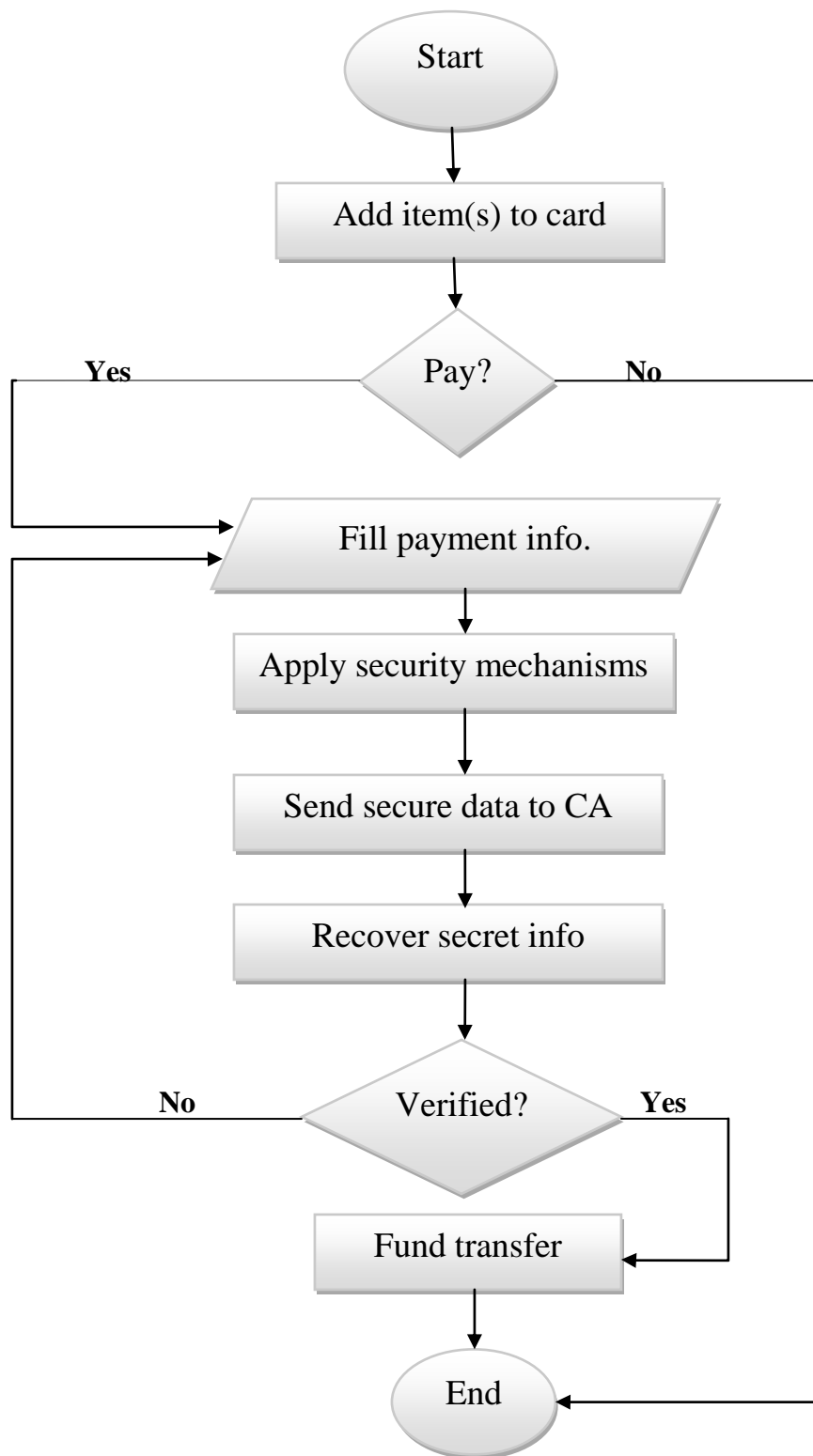


Figure 3.5: Flowchart for payment process

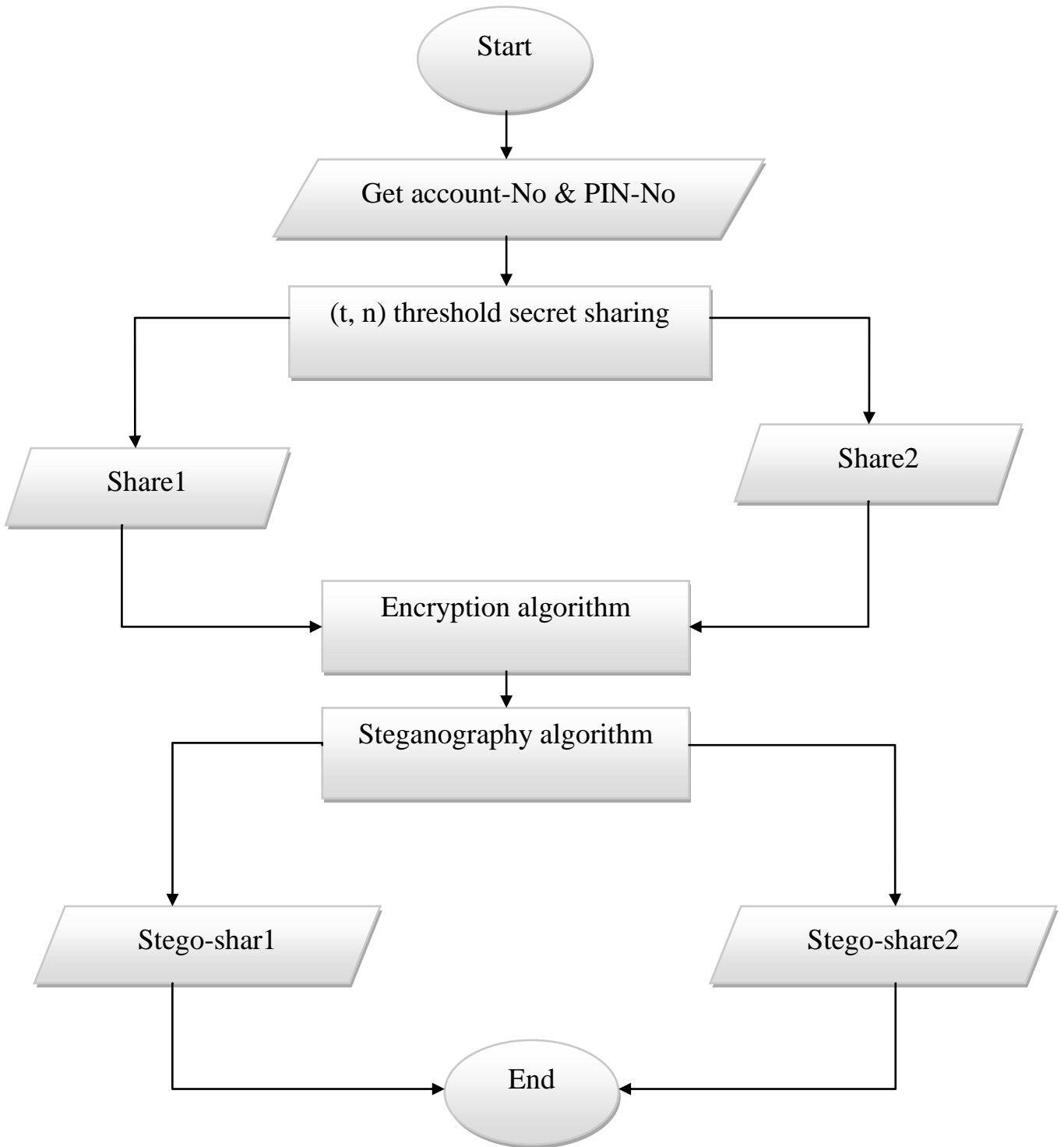


Figure 3.6: Flowchart for Generate secure shares

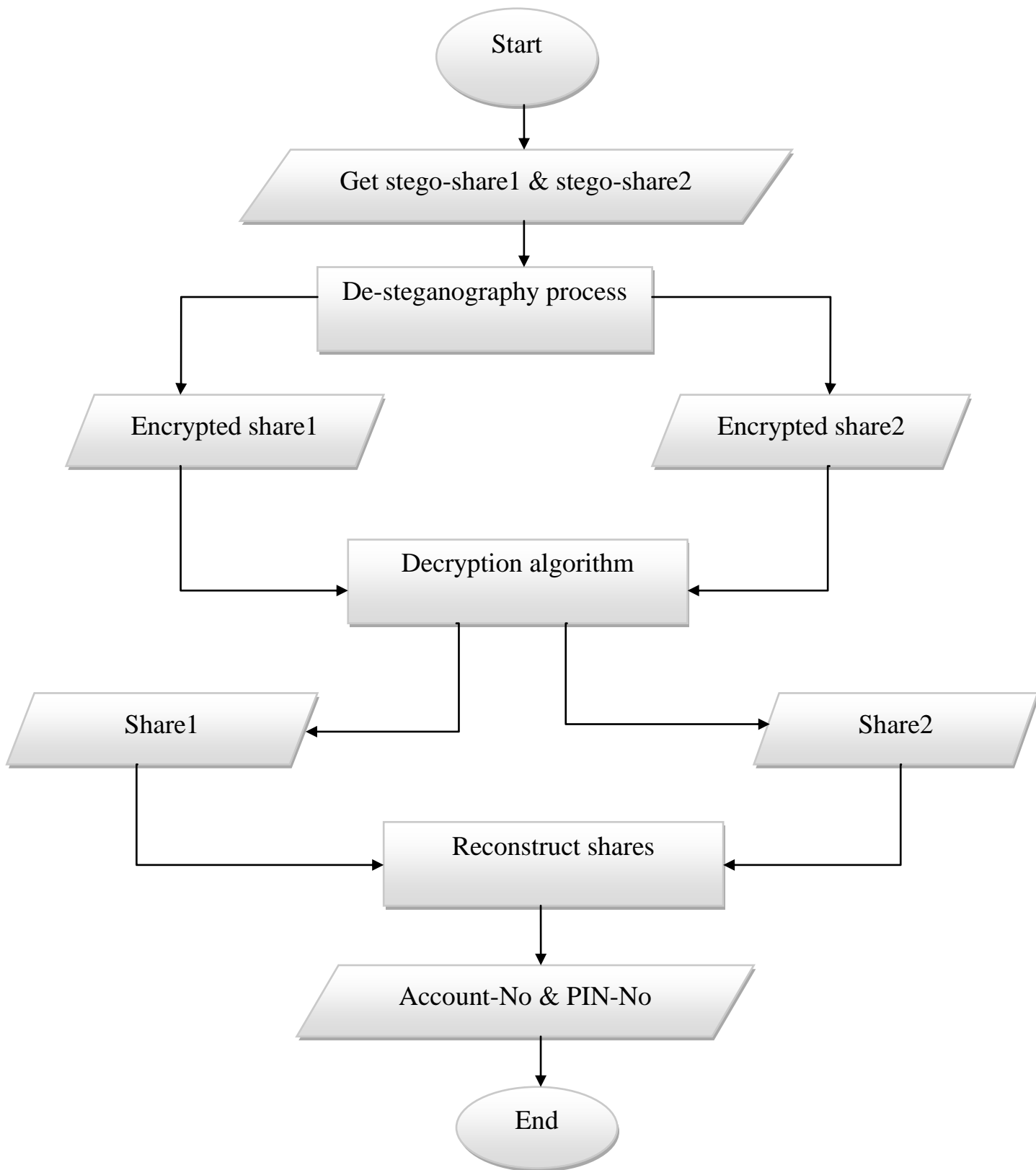


Figure 3.7: Flowchart for Recover shares

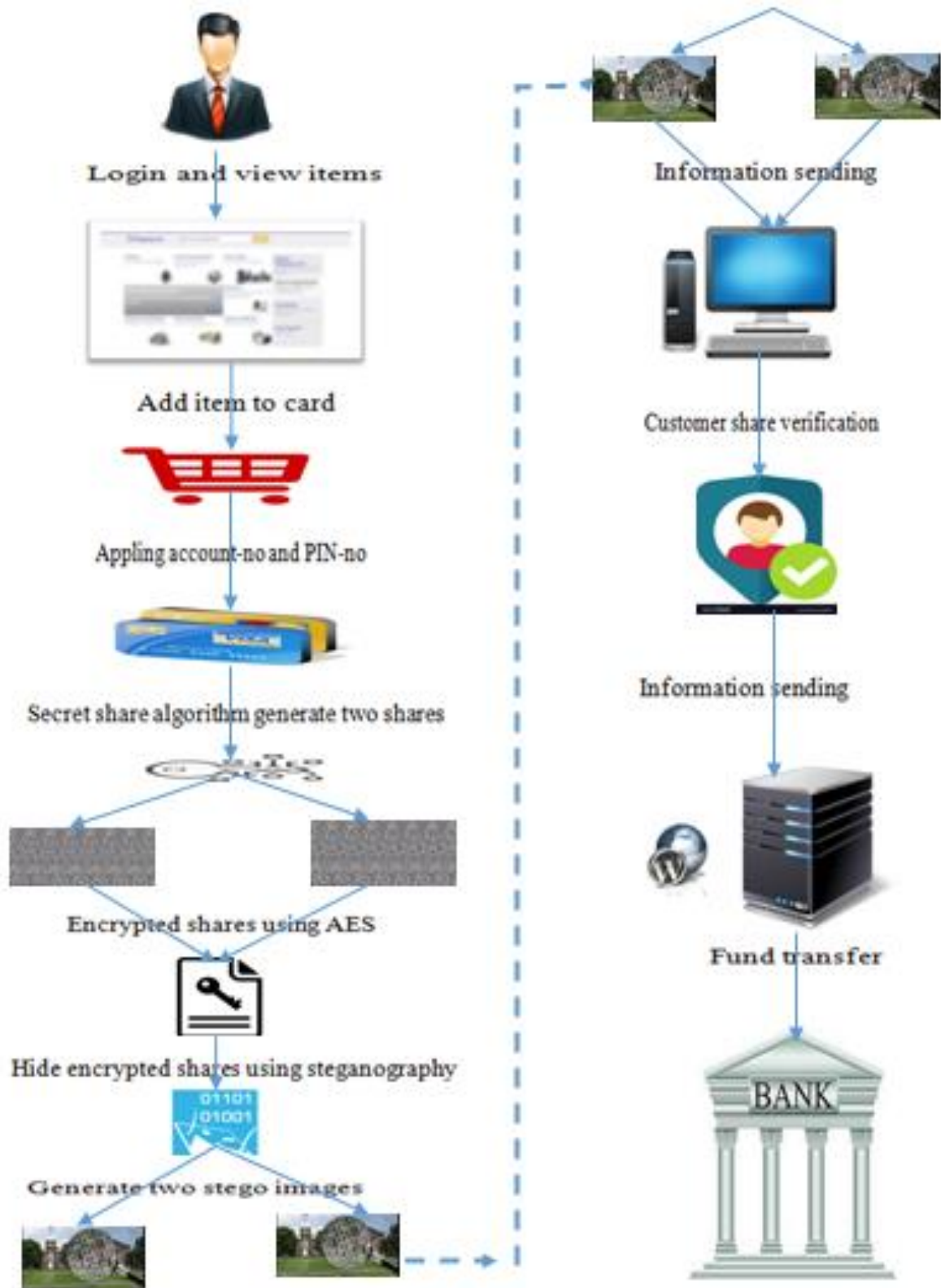


Figure 3.8: Workflow diagram of the system

3.2 PART TWO: TECHNIQUES:

3.2.1 Secret Sharing Scheme:

3.2.1.1 Introductions:

Secret sharing scheme plays a significant role in protecting important information from getting lost, annihilated or in to wrong hands. In a secret sharing scheme one secret value is distributed into shares among a set of participants in such a way that only the authorized subsets of participants can reconstruct the secret from their shares, while the participants in a forbidden subset cannot obtain any information about the secret value [32]. In 1979, the first (t, n) threshold secret sharing scheme is proposed by Shamir [15] and Blakley [16] independently.

3.2.1.2 Threshold schemes:

A (k, n) -threshold scheme is a method of sharing a secret \mathbf{K} among a set of n participants in such a way that any k participants can compute the value of the secret, but no group of $k - 1$ or fewer can do so.

Let the set of participants be denoted by \mathbf{P} . The value of the secret \mathbf{K} is chosen by the dealer, denoted \mathbf{D} , who is a special participant not in \mathbf{P} . When \mathbf{D} wants to share the secret \mathbf{K} among the participants in \mathbf{P} , \mathbf{D} gives each participant some partial information, called a share. The shares are distributed secretly, so no participant knows any other participant's share.

At a later time, some qualified subset of participants $\mathbf{B} \subseteq \mathbf{P}$ would like to compute the secret \mathbf{K} . They will then pool their shares together in an attempt to compute \mathbf{K} .

The most famous construction of a (k, n) -threshold scheme, called the Shamir Threshold Scheme, invented in 1979.

3.2.1.2.1 Shamir Threshold Scheme:

First some notations: Let $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$ be the set of participants, \mathbf{K} the set of possible secrets, and \mathbf{S} the set of possible shares. In the Shamir Threshold Scheme $\mathbf{K} = \mathbb{Z}_p$, where $p \geq n + 1$ is a prime. Also, $\mathbf{S} = \mathbb{Z}_p$. Hence both the secret and the shares are elements of \mathbb{Z}_p . The construction is now as follows.

Initialization:

D chooses n distinct non-zero elements from \mathbf{Z}_p , denoted x_i , $1 \leq i \leq n$. All these values are public.

Share Distribution:

1. **D** wants to share the secret $\mathbf{K} \in \mathbf{Z}_p$. **D** randomly chooses $k - 1$ elements of \mathbf{Z}_p , denoted a_1, a_2, \dots, a_{k-1} . Furthermore, $a_0 = \mathbf{K}$

2. **D** computes $y_i = a(x_i)$, for $1 \leq i \leq n$, where

$$a(x) = \sum_{j=0}^{k-1} (a_j x^j) \text{ mod } p$$

3. **D** gives participant \mathbf{P}_i the share y_i . In short, the dealer constructs a random polynomial of degree at most $k - 1$ in which the constant term is the secret \mathbf{K} , i.e.,

$\mathbf{a}_0 = \mathbf{K}$. Every participant obtains a point (x_i, y_i) on this polynomial. We now have to verify two different properties, firstly that any k participants can reconstruct the polynomial $a(x)$ and, hence, calculate the secret, and secondly that any group of $k - 1$ participants cannot do so.

Reconstruction:

Let us start by looking at how k participants can reconstruct the polynomial $a(x)$. This is basically done by means of **polynomial interpolation**. Without loss of generality we can assume that $B = \{P_1, P_2, \dots, P_k\}$ is the set that is to reconstruct the secret. The participants in B know

$$y_i = a(x_i), 1 \leq i \leq k,$$

where $\mathbf{a}(x) \in \mathbf{Z}_p[x]$ is the secret polynomial chosen by **D**. The polynomial $a(x)$ has degree at most $k - 1$ and, hence, can be written

$$a(x) = a_0 + a_1 x + \dots + a_{k-1} x^{k-1},$$

Where the coefficients a_0, a_1, \dots, a_k are unknown elements of \mathbf{Z}_p and $a_0 = \mathbf{K}$ is the secret. Each participant knowing $y_i = a(x_i)$ can obtain a linear equation in the k unknowns a_0, a_1, \dots, a_{k-1} . So,

the group \mathbf{B} has \mathbf{k} linear equations at its disposal. If the equations are all linearly independent, there will be a unique solution, and \mathbf{a}_0 will be revealed as the key.

The system of linear equations is the following:

$$\begin{aligned} a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_{k-1}x_1^{k-1} &= y_1, \\ a_0 + a_1x_2 + a_2x_2^2 + \cdots + a_{k-1}x_2^{k-1} &= y_2, \\ &\vdots \\ a_0 + a_1x_k + a_2x_k^2 + \cdots + a_{k-1}x_k^{k-1} &= y_k, \end{aligned}$$

This can be written in matrix form as follows:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{k-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^{k-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{pmatrix}.$$

The coefficient matrix, we call it A , is a so-called Vandermonde matrix. There is a well-known formula for the determinant of a $k \times k$ Vandermonde matrix, namely

$$\det A = \prod_{1 \leq i < j \leq k} (x_i - x_j) \pmod{p}.$$

Since the x_i 's are all distinct, no term in the product equals 0. Because of the fact that Z_p is a field, the product of non-zero elements are again a non-zero element, and we have that determinant of A $\det A \neq 0$. Since the coefficient matrix has a non-zero determinant, the system of linear equations has a unique solution over \mathbf{Z}_p . This establishes that any k participants can recover the polynomial $a(x)$ and, hence, the secret \mathbf{k} [33].

Security Analysis: Because scheme of thesis is based on Shamir's scheme, at least \mathbf{k} or more participants combining their shares will make it easy to reconstruct the secrets, but only $\mathbf{k}-1$ or fewer participants will not do. Scheme is a perfect threshold scheme in which knowing only $\mathbf{k} - 1$ or fewer shares provide no more information about the secrets to an opponent than knowing no pieces.

In the proposed solution the number of generated shares is two shares that mean the reconstruction of the secret is impossible from one share.

3.2.1.3 Lagrange interpolation polynomial:

Lagrange polynomials are used for polynomial interpolation. For a given set of points (x_j, y_j) with no two x_j values equal, the Lagrange polynomial is the polynomial of lowest degree that assumes at each x_j value the corresponding value y_j (i.e. the functions coincide at each point). The interpolating polynomial of the least degree is unique, however, and since it can be arrived at through multiple methods, referring to "the Lagrange polynomial" is perhaps not as correct as referring to "the Lagrange form" of that unique polynomial.

Interpolation polynomial: is the polynomial that calculates the value of a function between the values already known [34].

3.2.2 Steganography using Alpha channel:

Steganography techniques aimed at hiding data secretly in a carrier such as text, audio, image or video, without raising eavesdropper's suspicion. The original carrier is referred to as the *cover object* [35].

A steganography system is expected to meet three key requirements, namely transparency, capacity and robustness. Transparency: evaluates the image distortion due to signal modifications like message embedding or attacking. Capacity: It is the maximum amount of information that a data hiding scheme can successfully embed without introducing any perceptual distortion in the marked media. Robustness: measures the ability of embedded data or watermark to withstand against intentional and unintentional attacks [36].

Steganographic methods can be broadly classified based on the embedding domain into spatial, transform and compression domain. In spatial domain image Steganography: cover image is first decomposed in to its bit's planes and then LSB's of the bit's planes are replaced with the secret data bits. Advantages are high embedding capacity, ease of implementation and imperceptibility of hidden data. The major drawback is its vulnerability to various simple statistical analysis methods. Frequency domain embedding techniques, which first transforms the cover image into frequency domain, secret data is then embedded in frequency coefficients instead of hiding in image pixels directly. Advantages include higher level of

robustness against simple statistical analysis methods. Unfortunately, it lacks high embedding. In compression domain, secret data is embedded into compression codes of the cover-image which is then sent to the receiver. It is of paramount importance where bandwidth is a major concern [37].

3.2.2.1 Pixel representation in technique:

The most direct way to represent pixel's color is by giving an ordered triple of numbers: red (R), green (G), and blue (B) that comprises that particular color. The other way is to use a table known as palette to store the triples, and use a reference into the table for each pixel. For transparent images, extra channel called the Alpha value is stored along with the Red, Green, and Blue (RGB) channels.

3.2.2.1.1 RGBA Color image:

The most used image format contains three channels Red, Green, and Blue. Each color (channel) stored in one byte with value from 0 to 255. In 32-bit representation of graphic images, there are four channels of 8 bits as shown in figure 3.9, three color channels Red, Green, and Blue (RGB), and the fourth channel called the alpha channel. The alpha channel specifies how the pixel's colors should be merging with another pixel when the two overlaid and control the transparency or opacity of an image.

RGBA extends the RGB color model with the alpha value. Alpha channel was invented by Catmull and Smith in 1971. The alpha channel used as an opacity channel; where its value varies from 0 to 255, which 0 means completely transparent while 255 means opaque. Not all RGB images contain an alpha value. If RGB images are used to hide the information, it can lead to suspicion because the default value of the alpha in the RGB images is 255. In RGBA images alpha value is not same in all the pixels of the image. Therefore, the proposed technique gives much better results if RGBA images are used [38].

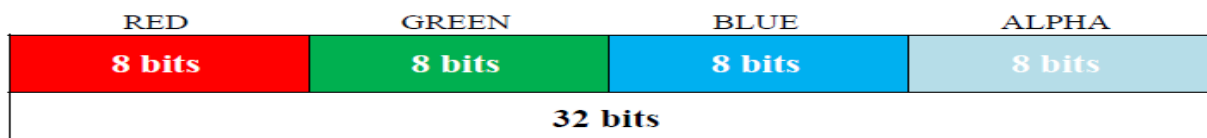


Figure 3.9: RGBA Image Channels

3.2.2.1.2 Alpha channel:

In computer graphics, alpha compositing is the technique of mixing an image with a background to create the appearance of the partial transparency. This process is useful to render images in separate passes and then combine them into a final image.

3.2.3 Advanced Encryption Standard (AES):

AES is a symmetric encryption algorithm. The algorithm was developed by two Belgian cryptographers Joan Daemen and Vincent Rijmen. AES was designed to be efficient in both hardware and software, and supports a block length of 128 bits and key lengths of 128, 192, and 256 bits.

The encryption/decryption algorithm consists of several rounds of processing; the number of rounds depends on the key length: 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Except for the last round, all other rounds are identical.

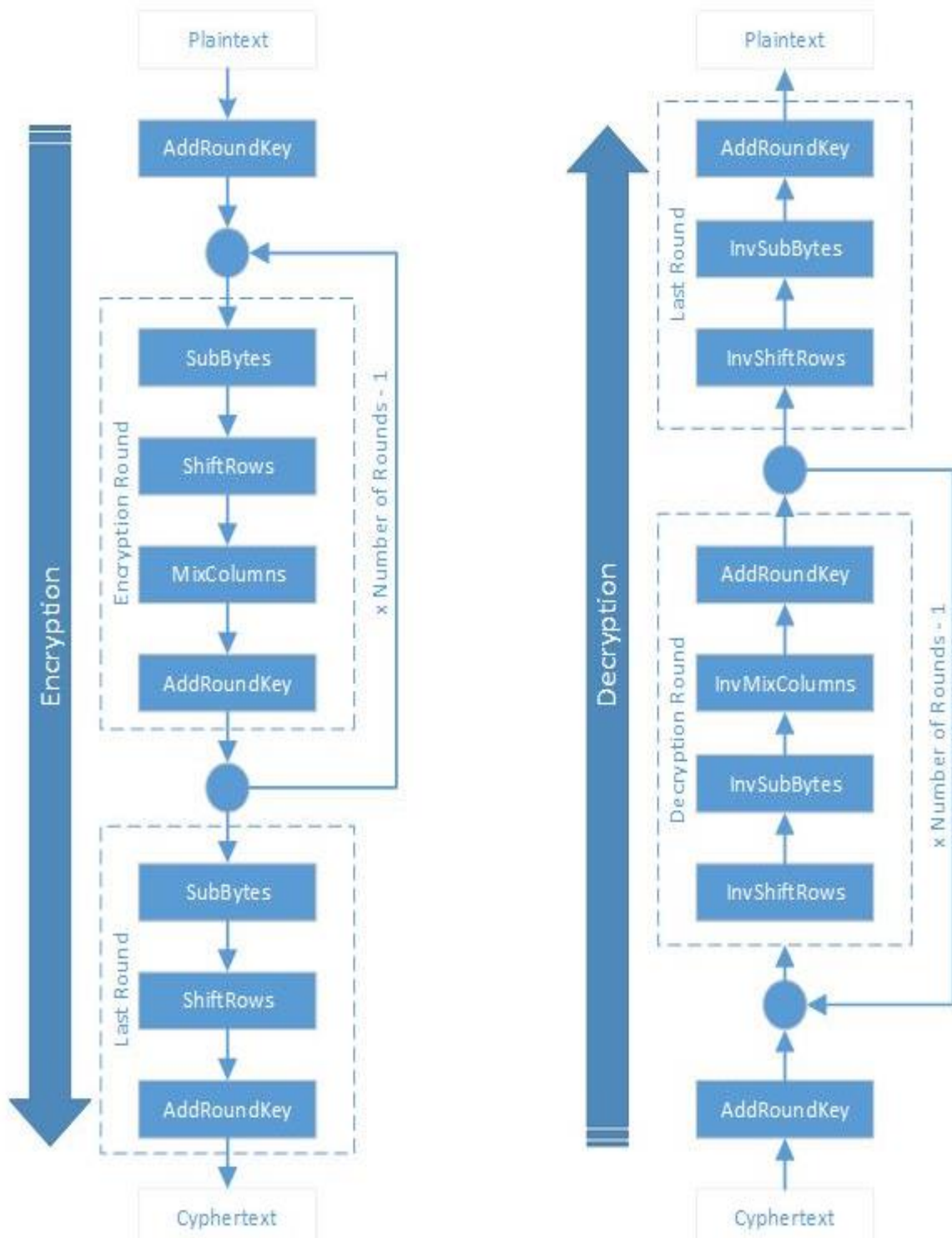


Figure 3.10: AES encryption/decryption algorithm

3.2.4 Model View Controller (MVC):

Model View Controller or **MVC** as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

Model: The lowest level of the pattern which is responsible for maintaining data.

View: This is responsible for displaying all or a portion of the data to the user.

Controller: Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response.

1- The Model:

The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

2- The View:

It means presentation of data in a particular format, triggered by a controller's decision to present the data. They are script-based template systems like JSP, ASP, PHP and very easy to integrate with AJAX technology.

3- The Controller:

The controller is responsible for responding to the user input and performs interactions on the data model objects. The controller receives the input; it validates the input and then performs the business operation that modifies the state of the data model [39].

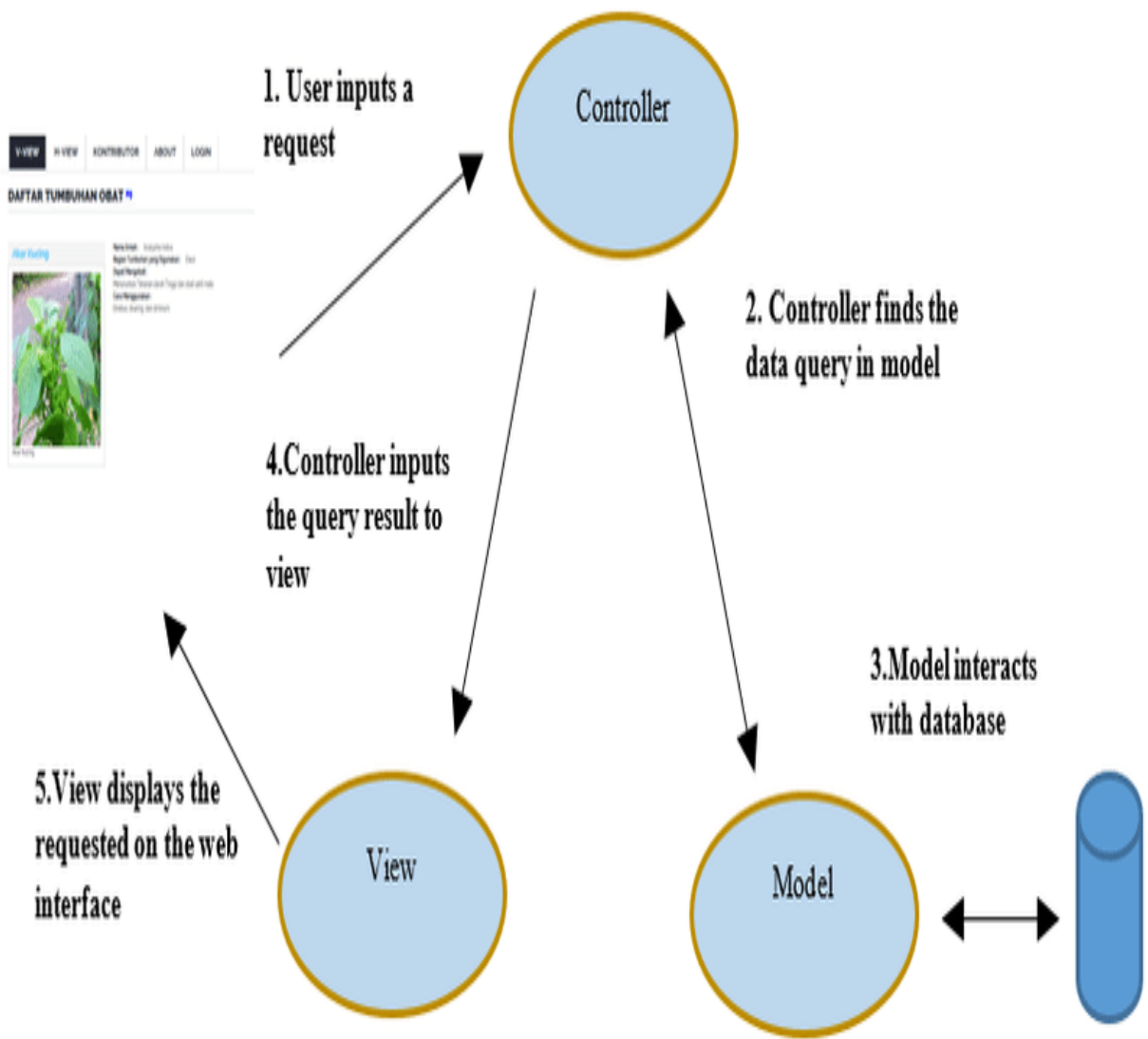


Figure 3.11: MVC Architecture

3.3 PART THREE: SYSTEM TOOLS:

3.3.1 Java Enterprise Edition (J2EE):

Java is both a programming language and a platform. The Java programming language is a high-level object-oriented language that has a particular syntax and style. A Java platform is a particular environment in which Java programming language applications run.

The J2EE platform is built on top of the Java Standard Edition (JSE) platform. The J2EE platform provides an API and runtime environment for developing and running large-scale, multi-tiered, scalable, reliable, and secure network applications [40].

3.3.2 Java Servlet:

Java Servlet provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server and for accessing existing business systems. A servlet can almost be thought of as an applet that runs on the server side--without a face. Java servlets make many Web applications possible.

Servlets provide a component-based, platform-independent method for building Web-based applications, without the performance limitations of CGI programs.

Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection [41].

A servlet is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes [42].

3.3.3 Java Server Pages (JSPs):

JSPs are a Sun Microsystems specification for combining Java with HTML to provide dynamic content for Web pages. When you create dynamic content, JSPs are more convenient to write than HTTP servlets because they allow you to embed Java code directly into your HTML pages, in contrast with HTTP servlets, in which you embed HTML inside Java code.

JSPs are Web pages coded with an extended HTML that makes it possible to embed Java code in a Web page. JSPs can call custom Java classes, called taglibs, using HTML-like tags. The WebLogic appc compiler `weblogic appc` generates JSPs and validates descriptors. You can also precompile JSPs into the `WEB-INF/classes/` directory or as a JAR file under `WEB-INF/lib/` and package the servlet class in the Web archive to avoid compiling in the server. Servlets and JSPs may require additional helper classes to be deployed with the Web application.

JSPs enable you to separate the dynamic content of a Web page from its presentation. It caters to two different types of developers: HTML developers, who are responsible for the graphical design of the page, and Java developers, who handle the development of software to create the dynamic content [43].

3.3.4 NetBeans IDE:

NetBeans IDE is a free and open source integrated development environment for application development on Windows, Mac, Linux, and Solaris operating systems.

The IDE simplifies the development of web, enterprise, desktop, and mobile applications that use the Java and HTML5 platforms. The IDE also offers support for the development of PHP and C/C++ applications [44].

CHAPTER IV

RESULTS AND DISCUSSION

4. Overview:

This chapter focuses on outcomes of the proposed solution as the system interfaces, system analysis and the results that was been achieved through the applied of the proposed methodology, additional to benefits of the system. Finally, this chapter shows comparison between the proposed solution and other previous work.

4.1 System Interfaces:

The system as mention at the previous is implemented under MVC, which is used java for building the control and model classes, whereas the JSPs is used for view.

4.1.1 Registration Process:

Figure 4.1 shows the registration process for a new customer.

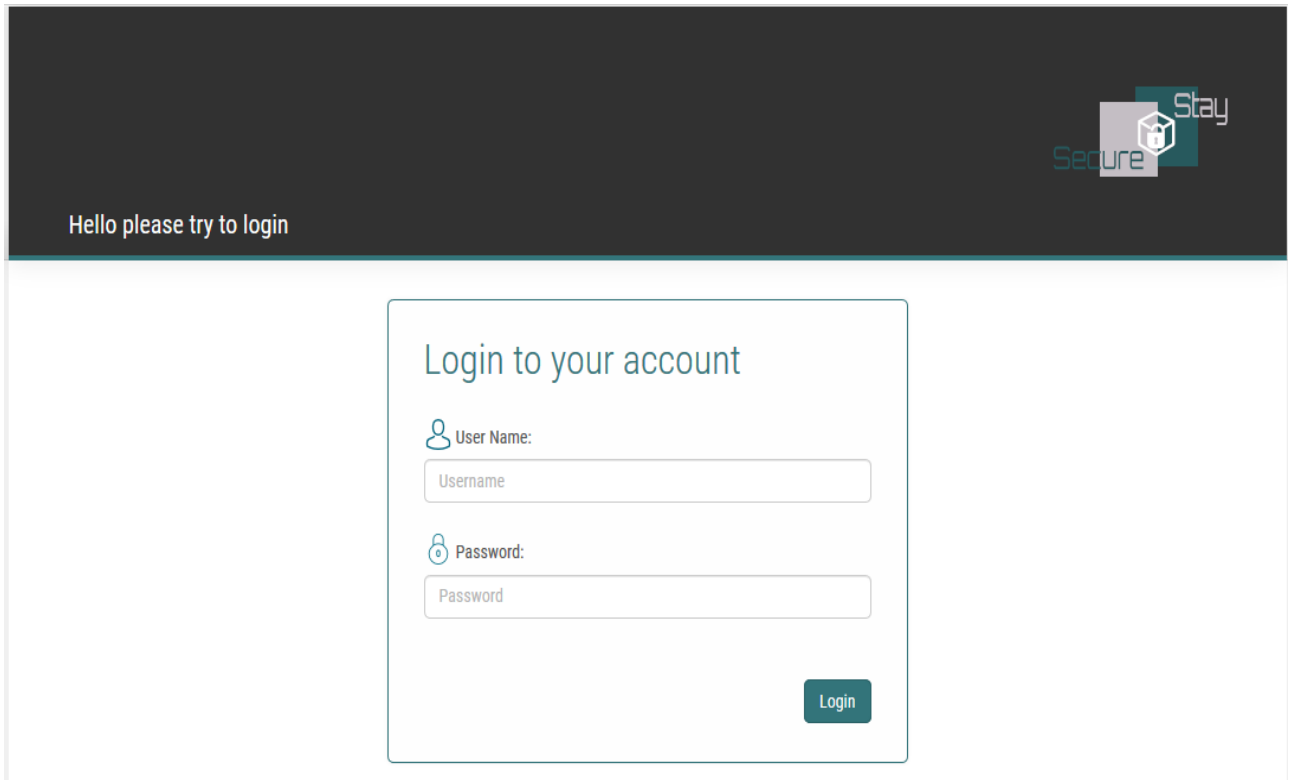
The image shows a registration form titled "Register a new account". The form contains the following fields and buttons:

- User Name**: A text input field.
- Address**: A text input field.
- E-mail**: A text input field.
- Phone Number**: A text input field with a dropdown arrow on the right side.
- Password ***: A text input field.
- Confirm Password ***: A text input field.
- RESET**: A dark teal button.
- REGISTER**: A dark teal button.

Figure 4.1: Registration Process

4.1.2 Login Process:

After sign-up is completed, customer can login to the system and accesses to all services in the system.



The image shows a login interface on a dark-themed website. At the top right, there is a logo with the text "Secure Stay" and a shield icon. Below the logo, the text "Hello please try to login" is displayed. The main content is a white-bordered box containing the following elements:

- Title: "Login to your account"
- Label: "User Name:" with a user icon
- Input field: "Username"
- Label: "Password:" with a lock icon
- Input field: "Password"
- Button: "Login"

Figure 4.2: Login Process

When a customer press login the system compares the inserted data to data that saved in DB, if the username and password is corresponding then the system is sending greeting message to the customer and view the service page.

4.1.3 View and Select Service(s) Processes:










		
<p>Digital single-lens reflex</p> <ul style="list-style-type: none">● 20.3 mega pixels● Sensor type: 1/2.3" / 6.17 x 4.55 mm● Optical zoom: 50 x● 3" / 76.2 mm LCD screen● Built-in WiFi / GPS / NFC● Built-in Viewfinder with LCD <p>Price: 500\$</p> <p>Quantity: <input type="text"/></p> <p>Add to cart view cart</p>	<p>HXR-MC2500 Shoulder Mount</p> <ul style="list-style-type: none">● 8.85MP Super 35mm CMOS Sensor● 4K DCI (4096x2160) Up to 59.94p● Built-in WiFi / GPS / NFC● Dual DIGIC DV 6 Image Processors● ISO 160-25,600 Expandable to 102,400● Built-in Viewfinder with LCD <p>Price: 900\$</p> <p>Quantity: <input type="text"/></p> <p>Add to cart view cart</p>	<p>High Definition Video Camera</p> <ul style="list-style-type: none">● Dual DIGIC DV 6 Image Processors● 4K DCI (4096x2160) Up to 59.94p● Full HD (1920x1080) Up to 120p● Built-in Viewfinder with LCD● ISO 160-25,600 Expandable to 102,400● 8.85MP Super 35mm CMOS SensorB <p>Price: 500\$</p> <p>Quantity: <input type="text"/></p> <p>Add to cart view cart</p>

Figure 4.3: Service Page

This figure shows the service page, here customer can browse all the products. This page shows an image of each item, category of it, brief description and the price of each product.

The customer can determine the quantity of selected product(s) and then add it into cart.

4.1.4 View cart Process:

IMAGE	IMAGE NAME	QUANTITY	TOTAL_PRICE	CATEGORY	DELETE
	Digital single-lens reflex	1.0	500.0	canon	
	HXR-MC2500 Shoulder Mount	2.0	1800.0	canon	
	Digital single-lens reflex	2.0	1000.0	canon	
Total					
3300.0					

[BACK](#) [PAY](#)

Figure 4.4: View cart Process

After customer add products to cart, customer can view all products that was been added to the cart. As shown figure4.3 view cart: views an image of all added items, name, quantity, total price base on the quantity of each product.

4.1.5 Checkout Process:

Previous figure show view card in the above figure PAY button is used for completing the purchase request.

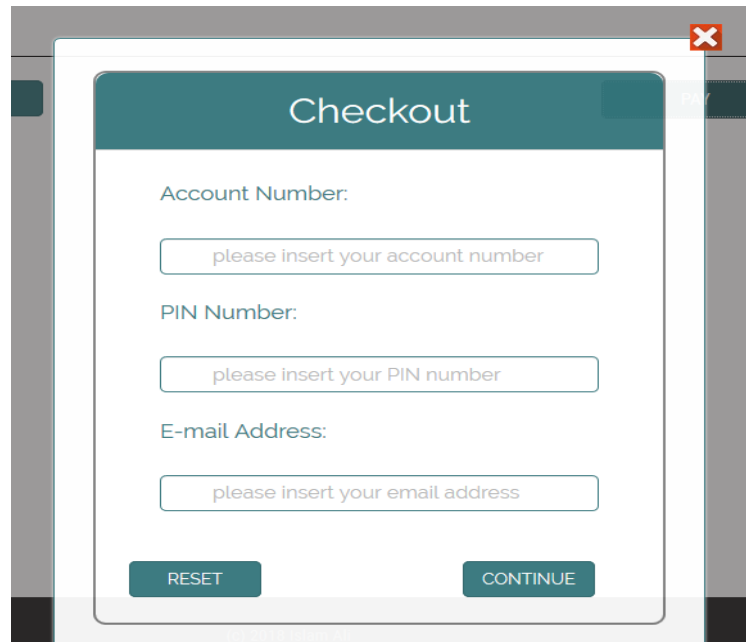
A screenshot of a web application's checkout process. The form is titled "Checkout" and is contained within a modal window. It features three input fields: "Account Number:" with a placeholder "please insert your account number", "PIN Number:" with a placeholder "please insert your PIN number", and "E-mail Address:" with a placeholder "please insert your email address". At the bottom of the form, there are two buttons: "RESET" and "CONTINUE". The modal window has a close button (an 'X' icon) in the top right corner.

Figure 4.5: Checkout Process

In this step customer asked to fill credential information as account number, PIN number and customer's e-mail.

After customer filling this fields and press continue button, the system firstly generates two secret shares for account and pin numbers, then those shares are encrypted using AES algorithm. Finally, encrypted shares are embedding into images, one of these images is saved in customer's DB and the other one is saved at CA DB, after that CA getting shares and extract the secret data then CA sends account number to the Merchant and PIN to the bank.

The customer's account number is sends to the bank from Merchant DB and the bank verify the sent account number by comparing it with the account that corresponding to the PIN number sent from CA DB, when the verification is successful, the system is generate four-digit random number as a verification code and send it to the customer as below figure.

4.1.6 Sending Verification Code process:

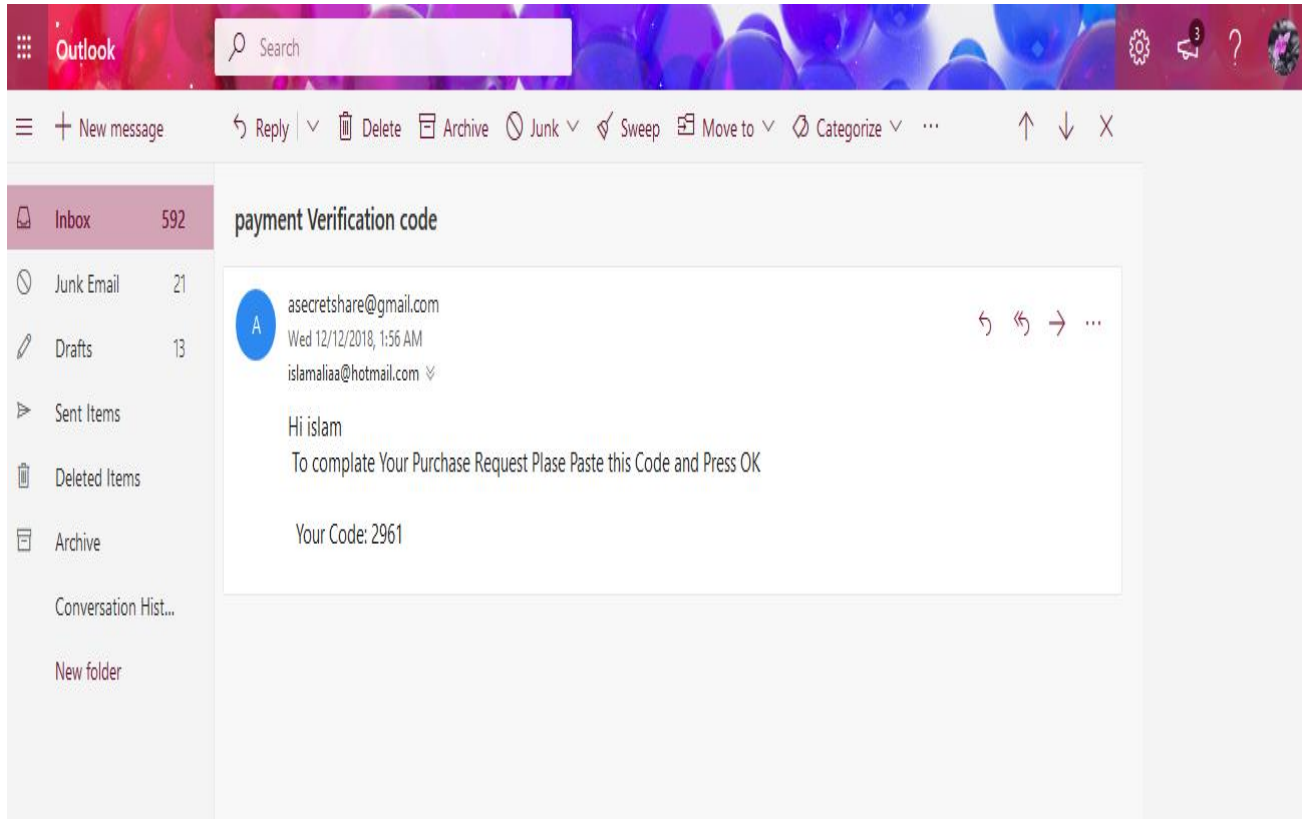


Figure 4.6: Sending Verification Code process

The above figure shows the code that was been sent to the email of a customer, this message tells the customer about process of purchase some products and the code uses to verifying a customer.

4.1.7 Verification Process:

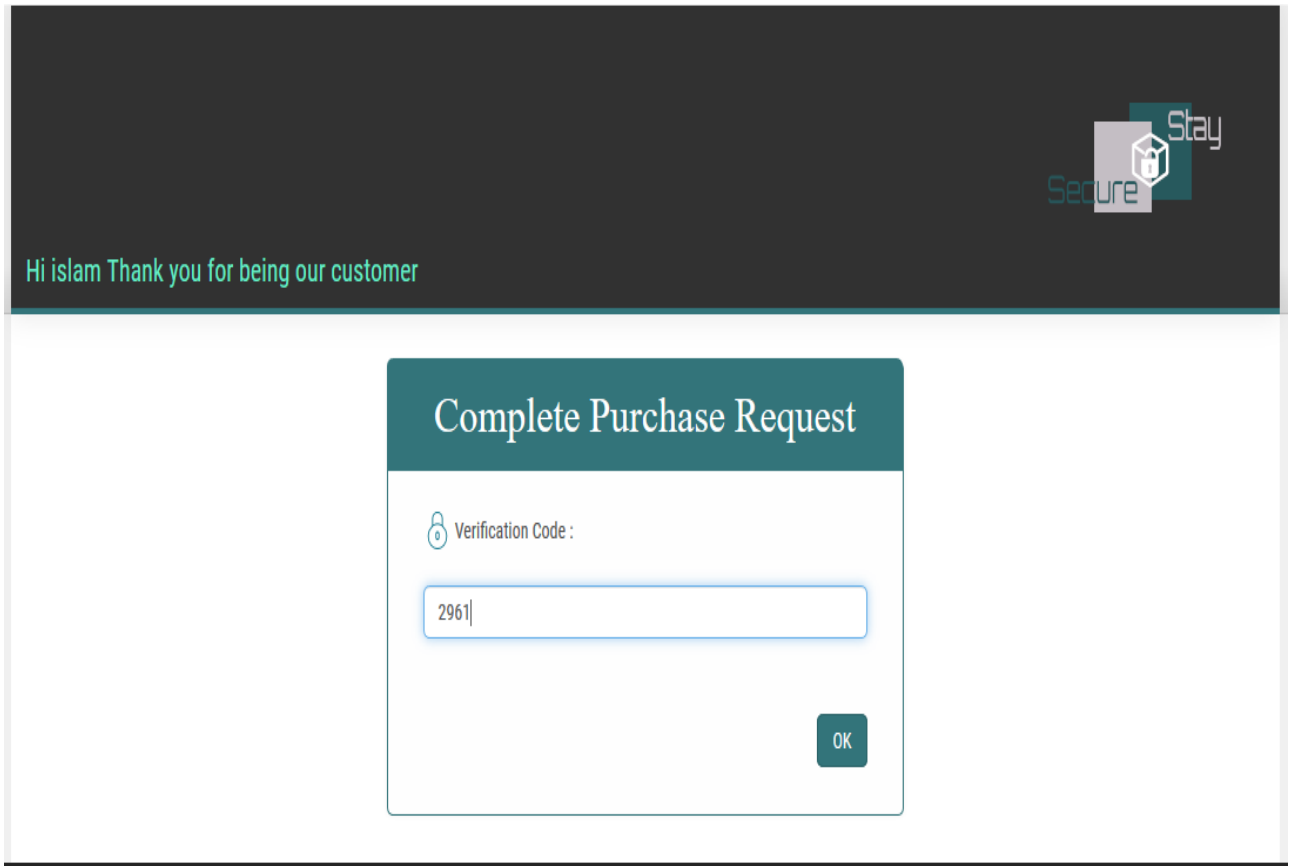
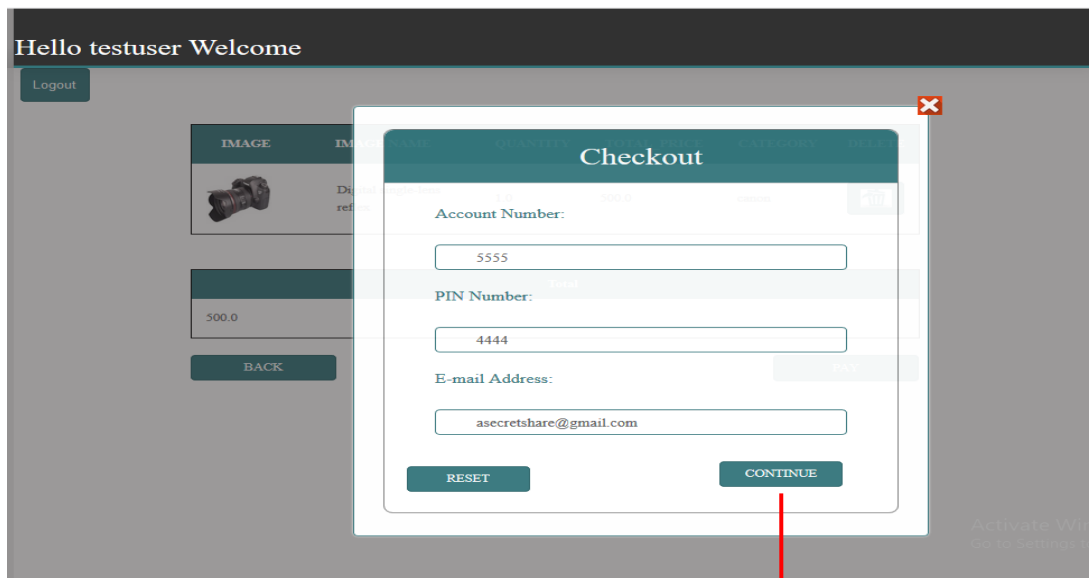
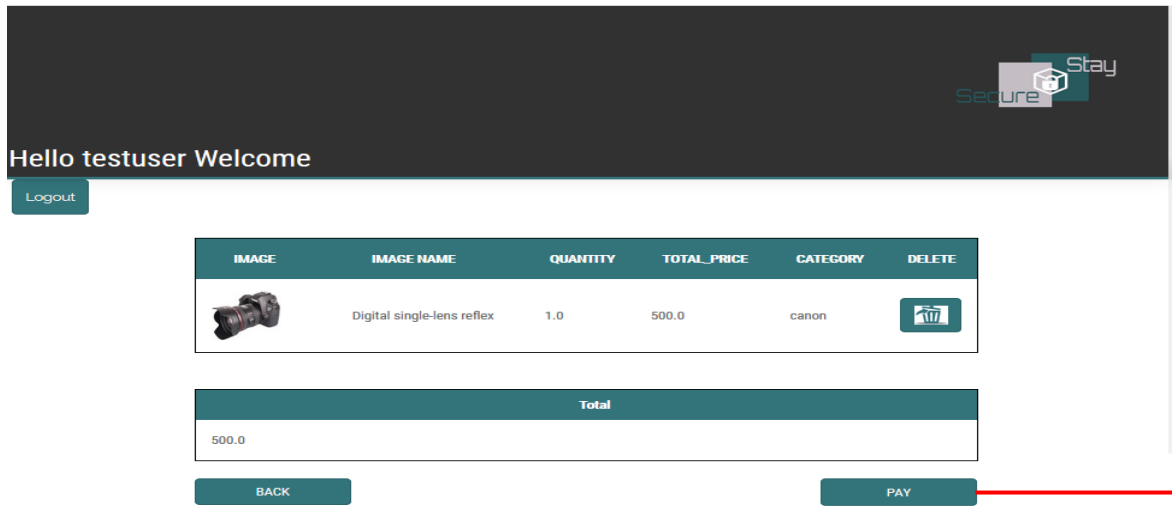


Figure 4.7: Verification Process

Finally, when a customer inserts the code - that was received above -, the funds will be transferred from a customer account to the merchant account.

4.2 System Analysis:

Step_1:

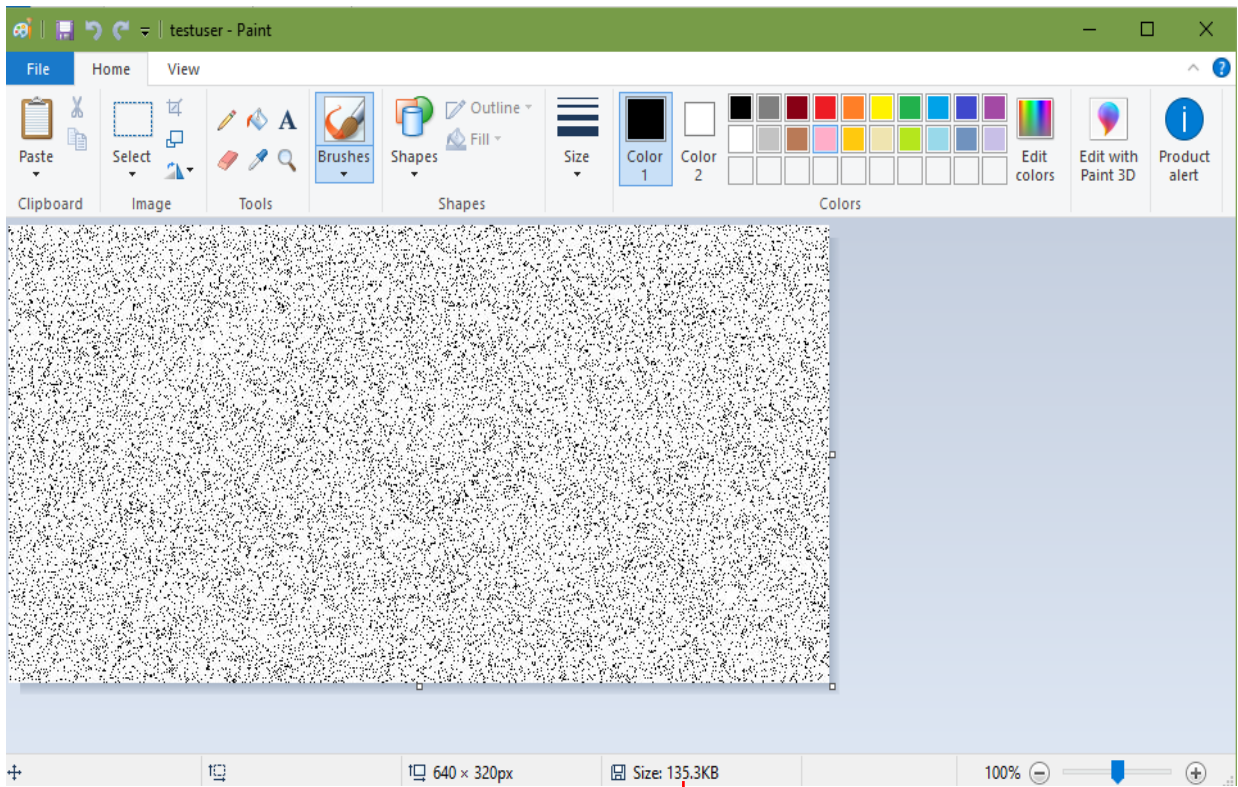
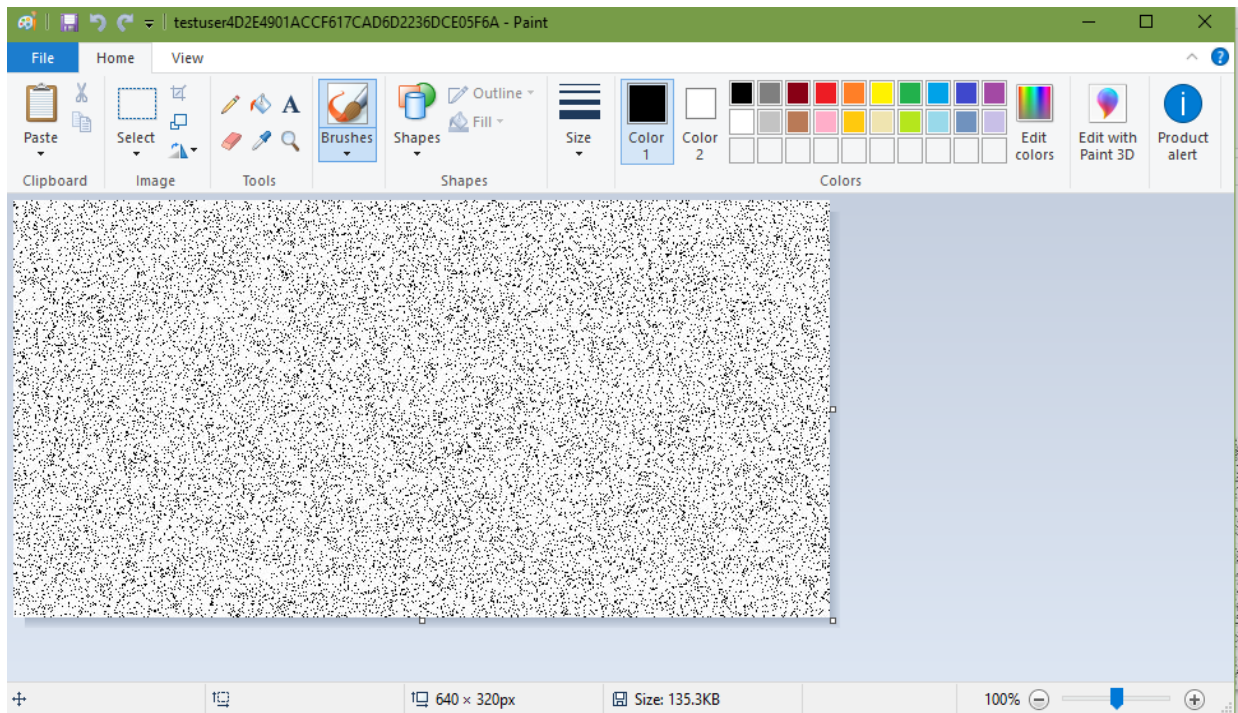


Step_2

Step_3:

 testuser	06-Jan-19 1:05 PM	PNG File	136 KB
 testuser4D2E4901ACCF617CAD6D2236D...	06-Jan-19 1:05 PM	PNG File	136 KB

Step_3 cont....



Step_4:

Step_5:

Server: localhost ▶ Database: secretshare ▶ Table: ca "InnoDB free: 11264 kB"

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Operations](#) [Empty](#) [Drop](#)

Showing rows 0 - 0 (1 total, Query took 0.0003 sec)

SQL query:

```
SELECT *
FROM `ca`
LIMIT 0, 30
```

[Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

	ca_id	share2	cust_id	image_name
<input type="checkbox"/>	11	[BLOB - 30 Bytes]	46	testuser.png

Check All / Uncheck All With selected:

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

[Insert new row](#) [Print view](#) [Print view \(with full texts\)](#) [Export](#)

Bookmark this SQL query

Label: Let every user access this bookmark

[Bookmark this SQL query](#)

Server: localhost ▶ Database: secretshare ▶ Table: customer_share "InnoDB free: 11264 kB"

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Operations](#) [Empty](#) [Drop](#)

Showing rows 0 - 0 (1 total, Query took 0.0006 sec)

SQL query:

```
SELECT *
FROM `customer_share`
LIMIT 0, 30
```

[Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

	cus_sh_id	cust_id	share1	image_name
<input type="checkbox"/>	11	46	[BLOB - 31 Bytes]	testuser4D2E4901ACCF617CAD6D2236DCE05F6A.png

Check All / Uncheck All With selected:

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

[Insert new row](#) [Print view](#) [Print view \(with full texts\)](#) [Export](#)

Bookmark this SQL query

Label: Let every user access this bookmark

[Bookmark this SQL query](#)

In the above steps, at step_1 when a customer press Pay, Checkout window it'll be appeared after that customer fills the payment information as shown as step_2, then secret shares are generated, first share for CA and other one for customer, the system generate these two shares with the same name of the customer but one of this image contain the session-ID to differ from CA image. When shares are generated it'll be encrypted and embedded into images as step_3. Finally, images were saved in CA DB as step_4 and customer share DB as step_5.

```
The secret is: 55554444
55554444 resultcoml
SecretShare [num=0, share=60454055] share ***0
SecretShare [num=1, share=65353666] share ***1
60454055 share ///0
65353666 share ///1
```

Figure 4.8: generate shares Process

In figure 4.8 two shares are generated for secret represented by 55554444 as same as concatenated account and pin number.

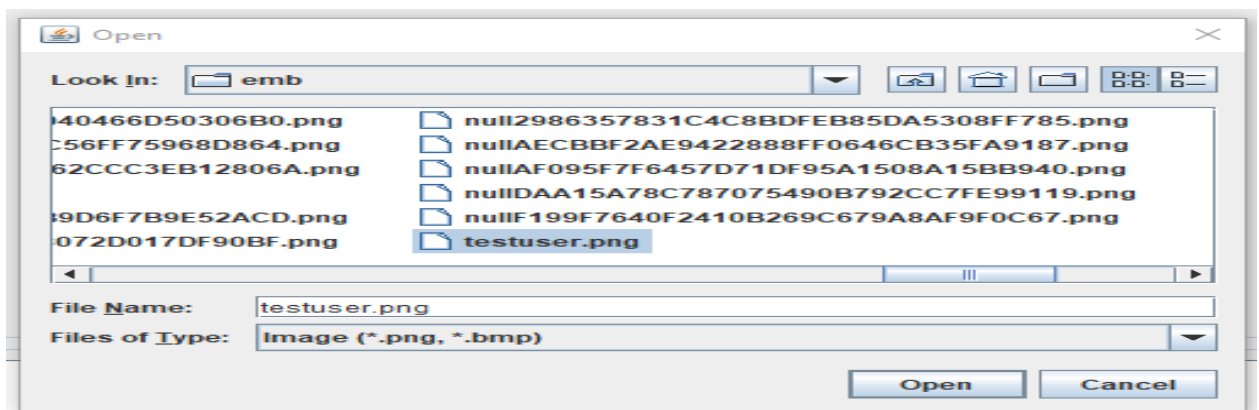


Figure 4.9: Open Encrypted CA Image

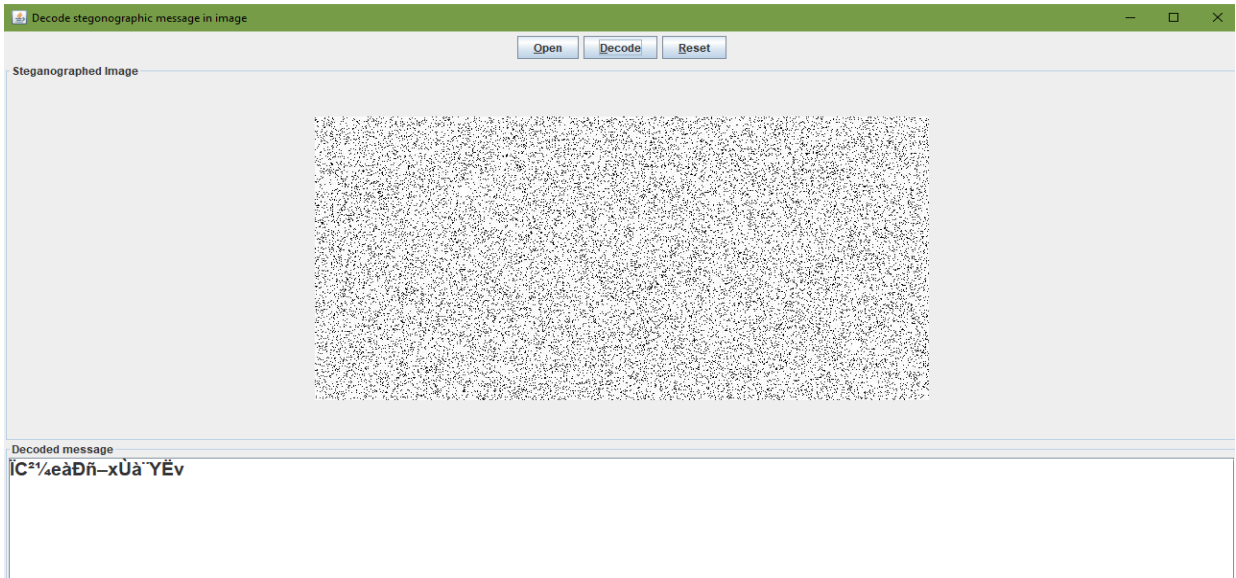


Figure 4.10: View Embed and Encrypted Data

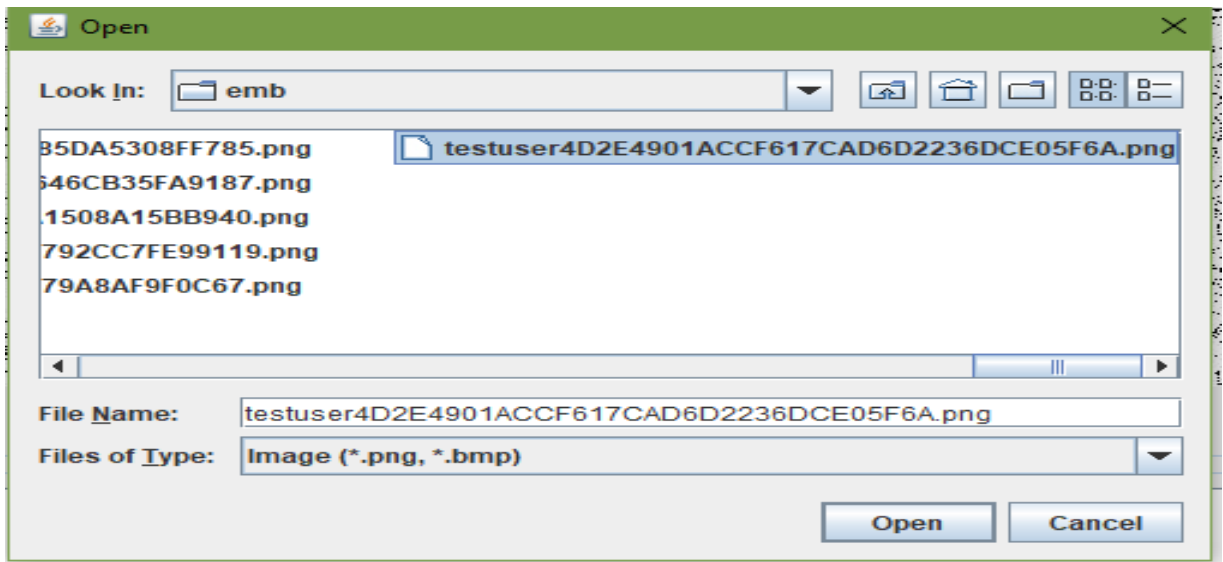


Figure 4.11: Open Encrypted Customer Image

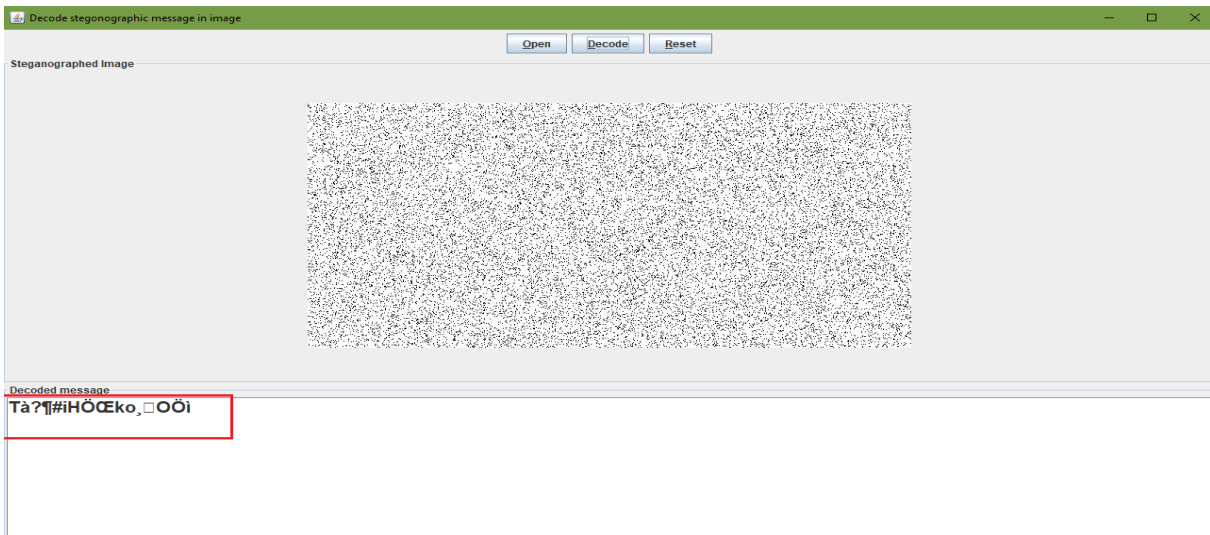


Figure 4.12: View Embed and Encrypted Data

Figures 4.11 and 4.12 show encrypted data in CA share and Customer share.

```

encrypt_Model
here is key generation
AES Symmetric key = 0oáËT68!^10b~|K^
key = 0oáËT68!^10b~|K^ step to check key
Encrypted message ÌC^4eáDñ-xÜà"YËv encrypted data in CA image
Decrypted message 38929102
embedMessage1Tà?#iHÖEko, OÖiget encrypted message2 to be embed
embedMessage1Tà?#iHÖEko, OÖiget encrypted message2 to be embed
Tà?#iHÖEko, OÖiencrypted message2 encrypted data in Customer image
embedmessage2ÌC^4eáDñ-xÜà"YËvget encrypted message to be embed un an image
ÌC^4eáDñ-xÜà"YËvencrypted message

Findshamir Model83564718coeff
Prime Number: 91877389
a1: 83564718
Share SecretShare [num=0, share=47241773]
Share SecretShare [num=1, share=38929102]
den: 91877388, num: 91877388, inv: 91877388
value: 47241773, tmp: 2606157, accum: 2606157
den: 1, num: 1, inv: 1
value: 38929102, tmp: 52948287, accum: 55554444
The secret is: 55554444 reconstructed secret

```

Figure 4.13: Reconstruction Process

Figure shows the process of recover customer info from secret images. Firstly, figure show the encrypted messages as same as figure 4.10 and 4.12 after set of processes the data it'll be clear –account and PIN number- as shown in figure 4.8, when data reconstructed successfully and all verification process is well the fund will be transferred.

4.3 Results:

This solution is providing secure environment for the process of a payment by:

- 1- CA is providing privacy to the customer's info by enforcing privilege on the access to that info. In other words, merchant get account number only that used to complete the payment process.
- 2- Previous step also prevents merchant from misused of customer information.
- 3- Provides confidentiality and anti-cheating by using AES algorithms, that means getting the secret info by uses part of shares is impossible and CA only can extract and getting the secret.
- 4- The use of threshold secret sharing enhanced the performance of the system because the shares is generated as number format rather than images from capacity point of view.
- 5- The customer doesn't expose to phishing attack because:

Verification mechanism is done before the process of transfer a funds from customer's account to the merchant's account, this mechanism is used to assure that the customer is approved and known this transaction because the verification code is sending to the email of the customer, after customer enter this code to the system and approved then funds will be transferred.

This mechanism is guarantee that no other can access to customer's email, and no phishing website can get money with absences of customer's knowledge.

4.4 Discussion:

The significant use of trusted third party is to provide a privacy and authority to the secret customer's information, really when we use trusted third party that means it has an authorized certificate from certified Authority, so the bank and customer trust this party and can getting all information about it till the main root who provided that certificate.

The absence of this party from solution that means all customer information is sending to the merchant here two scenarios are arise:

Merchant is genuine website then it'll get all customer's data so misused of it can be done, or someone else try to getting it.

Merchant side is phishing website that meant credential information is exposed.

4.5 Security analysis:

The proposed system resists to some kinds of attacks, in the case of Masquerade attack, when attacker acts as an authorized entity in the system.

The scenario of internal attack, attacker may try to get the credential information of the customer, the proposed solution generates two shares and encrypts these shares using AES so the extracting data is hard for attacker.

In the case of external attack, vulnerable authentication is the way to attack the system, proposed system provides good authentication mechanism that prevents this kind of attack by the use of third party, this party authenticate customer by getting share that saved in customer's database and merge it with its own share if the third party can extract info that mean the customer is legitimate and vice versa. Other mechanism is the verification code that is sends to the email of the customer to complete the payment process, so the customer only can get this code from his email.

In Man in the Middle attack, attacker tries to intercept the transmitted data. In case of DNS spoofing attacker may diverts data to fake website, in this case the proposed system sends least information about customer to merchant website, if this website is fake it can gain account number only, no other info was sent to the merchant, because the decryption process of customer's data is done at CA and it has certificate and it'll be trusted party for both bank and customer.

The attacker also can spoof the IP in this case attacker may spoof CA IP or merchant IP, suppose attacker spoofed CA IP, the proposed system uses encrypted shares embedding into images so when attacker get secret data it's difficult to decrypts it, and attacker needs the other share to extract secret data so customer info is prevented, merchant IP spoofing least info like account number may send.

The proposed system use session with limited time out after timer expired session was been destroyed. Other mechanism is the use of verification code that is generated and changed with each new request to complete payment process these mechanisms may prevent Replay attack.

4.6 comparisons:

The below table shows the comparison between proposed solution and other solution mentioned as related works in chapter 2:

Table 4.1: Comparison between proposed solution and related works

Paper Name	Techniques	Results	Open issues
The proposed solution	<ul style="list-style-type: none"> - Threshold secret sharing. - AES encryption. - Steganography. - Verification code generator 	<ul style="list-style-type: none"> - providing privacy to the customer's info. - prevents merchant from misused of customer information. - Provides confidentiality and anti-cheating. - enhanced the performance of the system. - provide anti-phishing solution. 	Used black and white image in steganography.

<p>A New Framework for Online Transaction using Visual Cryptography & Steganography [27]</p>	<ul style="list-style-type: none"> - Text-based steganography. - Visual cryptography. 	<ul style="list-style-type: none"> - Provide privacy to the customer. - Preventing against anti-phishing attack. 	<ul style="list-style-type: none"> - text-based steganography not suitable for large data representation and it makes use of features of English Language like inflexion, fixed word order and use of periphrases for hiding data rather than using properties of a statement.
<p>Secure e-payment system using steganography and visual cryptography [31]</p>	<ul style="list-style-type: none"> - Steganography. - DCT Visual cryptography. 	<ul style="list-style-type: none"> - securing customer information. 	<ul style="list-style-type: none"> - It isn't prevented against phishing attack, and secret information may be misused. - Merchant side can get all customers' data.

In A New Framework for Online Transaction using Visual Cryptography & Steganography [27] study used text based steganography this mechanism increase performance because the manipulating of text it's a simple rather than image, the study also used visual cryptography and trusted third party to provide anti-phishing solution, but the use of text-based steganography is not suitable for large data representation and exposed to statistical attack because the features of English language like fixed word and use of phrases rather than using properties of statements.

The proposed solution color image steganography based on the use of alpha channel that provide security against statistical attack and the uses of secret sharing algorithm in secret sharing enhanced the performance of the system.

Secure e-payment system using steganography and visual cryptography [31] this study used Bit Plane Complexness Segmentation steganography after that applied Discrete Cosine Transformation for providing security to avoid steganalysis this study used (2 out of 2) visual cryptography for hiding secret data of a customer, in this study the process of extracting customer's secret data is done at merchant side so that the info of customer may be exposed or misused.

Here the proposed solution providing privacy to the customer's info by restricting the access to that info by the use of CA who authenticates the customer and provide anti-phishing and anti-cheating attacks, and prevent customer's info from being exposed or misused.

CHAPTER V

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion:

The responsibility of protecting customer's information from being misused or being exposed is a major problem over internet; cryptographic techniques used to prevent customer's data against any potential attack.

Some of these techniques like secret sharing technique that allow secret to be share securely and the recovering of it is difficult, Advanced Encryption Standard (AES) that provide confidentiality and steganography are used in this thesis.

The proposed solution in this thesis was implemented to achieve the goal of securing the customers over an online payment system.

Appling this solution provides significant benefits to the customers, as keeping the privacy of customer information, immunizes against phishing, statistical and cheating attacks, and support confidentiality to the customer to pay online. The proposed solution is also implemented to countermeasures MITM, Replay and Masquerade attacks.

The third party that used in this solution playing a main role by gives the system trustiness, and it was been as an effected solution to applying privileges on the access of data.

Uses of secret sharing mechanisms for securing data is old and continued research area, several mechanisms are available and may be used as future work.

5.2 Recommendations:

The proposed solution used to provide confidentiality to the customer's info and to countermeasure some types of attack, in this solution black and white images are used to avoid some attacks like an image compression or resizing attacks, so finding mechanism to prevent these attacks is recommended.

New secret sharing approach provides anti-cheating solution without the use of AES encryption is recommended.

Enhanced this approach to prevent merchant from getting any info about customer is effective secure solution and the use of digital signature to provide confidentiality is recommended.

REFERENCES:

- [1] Dheeraj Agarwal, Shrinivas Deshmukh, “Online Payment System using BPCS Steganography and Visual Cryptography”, International Journal of Science and Research, Impact Factor 3.358, October 2012.
- [2] Priya Sonawane, Prof. Ramesh patole, “A Survey on Online Payment System Using Steganography and Visual Cryptography”, Global Journal of Advanced Engineering Technologies, Volume 5, 2016.
- [3] Priyanka More, Pooja Tiwari, Leena Waingankar, Vivek Kumar, “Online Payment System using Steganography and Visual Cryptography” International Journal of Latest Technology in Engineering Management and Applied Science, vol. 4, October 2015.
- [4] Neha Jain, Suraj Gupta, Ajaykumar Prajapati, Manoj Verma, “Secure Online Transaction Using Text Steganography and Visual Cryptography”, International Journal of Recent Trends in Engineering & Research ,Vol. 2, February 2016.
- [5] William Stallings,” Cryptography and Network Security Principles and Practices”, Fourth Edition, Prentice Hall, 16 November, 2005, ISBN-10: 0-13-187316-4, P. 983 Pages.
- [6] Prof. Christof Paar, Dr. Jan Pelzl, “Understanding Cryptography”, Second Edition, Springer-Verlag Berlin Heidelberg, 2010, ISBN 978-3-642-04100-6, P. 382 Pages.
- [7] Bruce Schneier, “Applied Cryptography”, Second Edition, John Wiley & Sons, Inc, ISBN: 0471128457, 01 January, 1996, P. 784 Pages.
- [8] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, “Handbook of Applied Cryptography”, First Edition, CRC Press, ISBN 9780849385230, 16 October, 1996, P. 810 Pages.

- [9] repository.sustech.edu, 'Enhancement of RC4 Algorithm for Images Encryption'.
[Online] Available: <http://repository.sustech.edu/handle/123456789/13936> .
[Accessed: 13- Jul - 2017 at 03:44 pm].
- [10] tutorialspoint.com, 'Advanced Encryption Standard'. [Online] Available:
https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm .
[Accessed: 11- Oct - 2018 at 08:53 pm].
- [11] repository.sustech.edu, 'The effect of Encryption on Audio Steganography'. [Online]
Available: <http://repository.sustech.edu/handle/123456789/11260> . [Accessed: 13-
Jul - 2017 at 04:19 pm].
- [12] Jonathan Weir, WeiQi Yan, "Visual Cryptography and Its Applications", Ventus
Publishing ApS, ISBN 978-87-403-0126-7, 2012, P. 144 Pages.
- [13] Moni Naor and Adi Shamir, "Visual Cryptography", in Proceedings of Euro crypt
1994, lecture notes in Computer Science, vol. 950,1994.
- [14] Bhagate, Kulkarni, "an overview of various Visual cryptography schemes, vol.2,
September 2013.
- [15] Adi Shamir, "How to share a secret", in Communications of the ACM 22, pp. 612-
613, 1979.
- [16] G. R. Blakley, "Safeguarding cryptographic keys", in Proceedings of the National
Computer Conference, American Federation of Information Processing Societies
Processing Societies (AFIPS'79) National Computer Conference, 25-28 February
1979, California, pp. 313-317.
- [17] G.Ateniese, C.Blundo, A.DeSantis, D.R.Stinson, "Visual cryptography for general
access structures", Proc.ICALP96, Springer, Berlin, 1996, pp.416-428.

- [18] Chang-Chou Lin and Wen-Hsiang Tsai, "Visual cryptography for gray-level images by dithering techniques", Elsevier Science, Pattern Recognition Letters 24, 2003, pp. 349–358.
- [19] F. Liu, C.K. Wu, and X.J. Lin, "Color visual cryptography schemes", IET Information Security, Vol. 2, No. 4, 4th July 2008, pp. 151–165.
- [20] Annalisa De Bonis and Alfredo De Santis, "Randomness in secret sharing and visual cryptography schemes ", in Theory Computer Science, 314, 2004, pp. 351- 374.
- [21] Z. Zhou, G. R. Arce, and G. Di Crescenzo, "Halftone visual cryptography via Error Diffusion," in IEEE Trans. On Information Forensics and Security, Vol. 4, No. 3., Sep. 2009, pp. 383 - 396.
- [22] Z. Zhou, G. R. Arce, and G. Di Crescenzo, "Halftone visual cryptography", in Proc. IEEE Int. Conf. Image Process., Barcelona, Spain, Sep. 2003, pp. I - 521-4.
- [23] Z. Zhou, G. R. Arce, and G. Di Crescenzo, "Halftone visual cryptography", in IEEE Trans. Image Process., vol.15, no. 8, Aug. 2006, pp. 2441– 2453.
- [24] Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 49–56, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [25] Nasir Memon and Ping Wah Wong, "Protecting digital media content", Communications of the ACM, vol.41,1998.
- [26] Roger David Hersch and Sylvain Chosson, "Band moire images", In SIGGRAPH '04: ACM SIGGRAPH 2004 Papers, pages 239–247, New York, NY, USA, 2004. ACM.

- [27] Santosh Kumbhar, Saumya Sahu, "A New Framework for Online Transaction Using Visual Cryptography & Steganography", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, November 2015.
- [28] Vaishnavi J. Deshmukh, Dr. A. S. Alvi, "An Advance Technique for Online Payment Security in E-Commerce for Developing Countries", International Journal of Science and Research, Vol. 4, March 2015.
- [29] Nikita Chaudhari, Priya Parate, "Implementing Visual Cryptography for Secure Online Payment System", Proceedings of IEEE forum International Conference, July 2017.
- [30] Ketan Raju Kundiya, Ram Joshi, "Enhanced security providing using visual cryptography", International Journal of Science and Research, vol. 3, December 2014.
- [31] Suganya Devi, Srinivansan, vaishave, "Secure e-payment system using steganography and visual cryptography", International Journal of Mathematical and Computational Sciences, vol. 11, 2017.
- [32] Abbas Cheraghi, "Sharing Several Secrets based on Lagrange's Interpolation formula and Cipher Feedback Mode", International Journal of Nonlinear Analysis and Applications (**IJNAA**), vol.5, February 2014
- [33] eit.lth.se, 'Secret Sharing Schemes'. [Online] Available: https://www.eit.lth.se/fileadmin/eit/courses/edi051/lecture_notes/LN8.pdf . [Accessed: 16- Oct - 2018 at 08:53 am].
- [34] en.wikipedia.org, 'Lagrange polynomial'. [Online] Available: https://en.wikipedia.org/wiki/Lagrange_polynomial . [Accessed: 16- Oct - 2018 at 10:00 am].

- [35] Anderson, Kuhn, "Information hiding-a survey", Proceedings of the IEEE, volume 87, issue 1, pp 1062-1078, July 1999.
- [36] Veerdeep Kaur Mann, Harmanjot Singh Dhaliwal, "32×32 Color Image Steganography ", International Journal of Engineering Trends and Technology (IJETT), Volume 4, August 2013.
- [37] shuchi Sharma, uma kumara, "A high capacity data-hiding Technique using steganography", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), volume 3, June 2013.
- [38] Ghaith Salem Sarayreh," Text Hiding in RGBA Images Using the Alpha Channel and the Indicator Method", Faculty of Information Technology Middle East University, Jan, 2014.
- [39] tutorialspoint.com, 'Basic MVC Architecture'. [Online] Available: https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm . [Accessed: 11-Oct - 2018 at 01:54 pm].
- [40] Oracle.com, 'Fusion Middleware Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server'. [Online] Available: https://docs.oracle.com/cd/E14571_01/web.1111/e13712/basics.htm#WBAPP125 . [Accessed: 11- Oct - 2018 at 01:24 pm].
- [41] oracle.com, 'The Java EE 5 Tutorial'. [Online] Available: <https://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html> . [Accessed: 11- Oct - 2018 at 01:03 pm].
- [42] Oracle.com, 'what is the servlet'. [Online] Available: <https://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html> . [Accessed: 11- Oct - 2018 at 01:13 pm].

- [43] Oracle.com, 'Fusion Middleware Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server'. [Online] Available: https://docs.oracle.com/cd/E14571_01/web.1111/e13712/basics.htm#WBAPP124 . [Accessed: 11- Oct - 2018 at 01:18 pm].
- [44] oracle.com, 'NetBeans IDE'. [Online] Available: <https://www.oracle.com/technetwork/developertools/netbeans/overview/index-1600855.html> . [Accessed: 11- Oct - 2018 at 12:17 pm].