XOR BASED OPTICAL ENCRYPTION WITH NOISE PERFORMANCE MODELING AND APPLICATION TO IMAGE TRANSMISSION OVER WIRELESS IP LAN

By

BO ZHANG

A DISSERATION PRESENTED TO THE HIGHER DEGREES COMMITTEE OF PENINSULA TECHNIKON IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF TECHNOLOGY ELECTRICAL ENGINEERING

PENINSULA TECHNIKON

2004

ACKNOWLEDGMENTS

I wish hereby to acknowledge the help and support of the staff of the Department of Electrical engineering at Peninsula Technikon, who have always availed their co-operation since my arrival in South Africa. I also am indebted to the Technikon for providing me with bursary assistance. A special note of thanks to Mr. Lindsay Wicomb, Mr. Marco Adonis and Mr. Kalimulah Mohamed for their patience in assisting me with keeping the computer networks in order and for assisting me in setting up laboratory equipment at a time when my English was still very incomprehensible. I am also indebted to Virginia Elisack from the Public Management department for providing the opportunity for me to study English when I first arrived here. This effort certainly paid off. Lastly I am deeply indebted to my supervisor Prof. MTE Kahn for affording me the opportunity to do this research project and for his invaluable assistance in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF ABBREVIATIONS AND KEYWORDS

DES	Data Encryption Standard
MSE	Mean square error
CE	Consumer electronics
DRM Systems	Digital Rights Management
PDA	Personal Digital Assistant
XOR	Exclusive-OR
CCI	Copy Control Information
CGMS	Copy Generation Management System
Plaintext	The original message
Ciphertext	The encrypted message
Encryption to ciphertext	The process of converting plaintext
Decryption ciphertext to plaintext	The process of converting
Cryptography	The art and science of keeping message secure
Cryptanalysis	The art and science of breaking Ciphertext
CDS	Correlated Double Sampler
ARPA	Advanced Research Projects Agency
TFT matrix	Thin Film Transistor or active-

CRT	Cathode-Ray Tube
QPSK	Quaternary Phase Shift Keying
TCP/IP	Transmission Control Protocol/Internet Protocol
FDMA	Frequency Division Multiple Access

Abstract of Dissertation Presented to the Higher Degrees Committee of Peninsula Technikon in Partial Fulfillment of the Requirements for the Degree of Master of Technology

XOR BASED OPTICAL ENCRYPTION WITH NOISE PERFORMANCE MODELING AND APPLICATION TO IMAGE TRANSMISSION OVER WIRELESS IP LAN

By

Bo Zhang

2004

Supervisor: Prof. MTE Kahn

Faculty: Engineering, Department: Electrical Engineering

Encryption was used whenever someone wanted to send a secret message to someone. The quality of the algorithm and key combination were the factors that ensured the strength of the system. However, until there were some automation one could not use complex methods for encryption because it simply took too long to encrypt and decrypt messages (even worse for images), manually. Optical technologies have recently been employed in encryption. Compared with traditional computer and electrical systems, optical technologies offer primarily two types of benefits, namely optical systems have an inherent capability for parallel processing, that is, rapid transmission of information, and information can be hidden in any of several dimensions, such as phase or spatial frequency. Optical systems therefore have an excellent capability for encoding information.

In this project an image encryption technique was developed using exclusive-OR (XOR) operations in the optical domain before the captured image entered a digital computer network for further processing. A gray-level image of the object to be encrypted was converted a binary format and a pixel by pixel exclusive OR operation

was performed on it with a random key-bit by making use of polarization encoding in LCD technology, before the encrypted binary image was detected by a CCD. The image may also be used as an input to a computer for comparison with a database. However, noise alters the encrypted bit and the value of the noisy encrypted bit is no longer binary. In this research the encryption techniques was evaluated and tested for applicability to encrypt and decrypt successfully. The noise performance was tested and compared.

The technique was applied for image transmission over a wireless IP broadband link. Images (optical and thermal) captured from telemedicine application was transmitted and decrypted with reasonable success in this application.

CHAPTER 1

INTRODUCTION

Since ancient time, people have invented methods to transmit personal messages to one another, but struggled with ideas to keep the message content secret. In the age with communication development, every day a great amount of data is sent through wire (Cable, optical Fiber) and wireless (RF, I/P) channels. It is therefore important to ensure the security of the information including text sound and picture etc. Optical adding algorithm encryption is a good way for supplying security. In this research project, I will test a new kind of optical encryption based on XOR logic, and whether it is applicable to the transmission of images between the two terminals linked by wireless I/P, such as FLR or equivalents such as 3Com.

1.1 Awareness of The Problem

The Oxford English dictionary, defines cryptography as hidden writing. It has been around for a very long time. The Ancient people such as the Egyptians, the Arabs and the Romans developed their own systems.

Cryptography was used whenever someone wanted to send a secret message to someone else, in a situation where spies might be able to intercept the message and read it. Generals used to send orders to their armies, which were encrypted. A famous encryption machine called Enigma was used in the Second World War to send military messages.

A good example of early cryptography is the Caesar cipher, named after the Roman Julius Caesar because he is thought to have used it although he did not actually invent

it. The cipher is described as follows. One may, take a piece of paper and write along the top edge the alphabet. Defeat the alphabet below the first row. There would then be two lines of letter:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ

At this stage the message is formulated, for example: "SEND MONEY TONIGHT."

Move one of the alphabet lines along to the right one or more letters so that they no longer line up. That would now look as follows:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
YZABCDEFGHIJKLMNOPQRSTUVWX

The cipher is now ready. Each time you see a letter of the message in the top line; write down instead the letter on the bottom line.

SEND MONEY TONIGHT becomes "QCLB KMLCW RMLGEFR."

What happened here was cryptographic transformation of the message. To accomplish this we used an algorithm and a key, in this case the value two because we moved A two places forwards on the bottom line.

All we have to do now is make sure that the person receiving our message knows the key and the algorithm. By knowing the Caesar cipher and that the key is two they can put their lower line two places to the right, and by taking each letter of the message and writing down the letter immediately above it, the receiver can re-create the original message.

The quality of the algorithm and key combination were the factors that made the strength of the system. However, until there was some automation cryptographer could not use complex methods because it simply took too long to encrypt and decrypt messages manually.

With modern computers, we are now able to do these things much faster and better. There are many algorithm methods available that are far harder to break than the Caesar cipher. These algorithms have names, such as Rijndahl, Blowfish, RC2, RC4, Triple DES, CAST. They have key sizes that are enormous by comparison to the Caesar cipher.

However, just as computers are able to create such powerful algorithms, computers can also be harnessed to break them. The algorithm DES (Data Encryption Standard) in use for many years to protect banking transactions was considered very strong. The University of Cambridge however published a technique for a custom machine to break the cipher in minutes, for a manufacturing cost of under R7 million.

The expanding Internet bandwidth and availability of digital consumer electronics devices for playback, recording and storage have increased the demand for multimedia services. Namely, many methods were invented to use for encrypting digital information. In the computer world, the binary algorithm is always used. So the logic algorithms are suitable for encryption it. Telecommunication provided closer and closer contact of different countries and it is important problem to supply the safe and fast links.

In privately defined closed systems, security is provided by controlling access to copyrighted content [Eskicioglu, A. M., Town, J. and Delp, E. J. (2003)]. The devices that receive satellite and cable transmissions were equipped with the software and hardware needed to prevent unauthorized access.

The Internet, however, remains to be a public network of networks with a vast potential for data content distribution. Although the first examples of PC-based Digital Rights Management Systems (DRM's) were emerging in the commercial market, the real-time constraints for many multimedia applications cannot be met by an increasing number of smaller client devices (such as PDAs and videophones) with limited processing and communication power. Hence, there is a need to develop techniques that exploit the structure of multimedia data in maximizing the efficiency of encryption algorithms and add-on security devices such as discussed below.

Optical technologies have recently been employed in data security. [Wang R. K., Watson I. A., and Chatwin C. (1996)] Compared with traditional computer and electrical systems, optical technologies offer primarily two types of benefits.

(1) Optical systems have an inherent capability for parallel processing, that is, rapid transmission of information.
(2) Information can be hidden in any of several dimensions, such as phase or spatial frequency; that is, optical systems have excellent capability for encoding information.

In many studies [Javidi B. and Ahouzi E. (1998)] the authors demonstrated different optical verification systems for information security applications, based on optical correlations. These systems correlate two functions: one, the lock, is always inside the correlator, and on the other hand, the user presents the key code to the system in the verification stage. Mostly, the systems determine whether the input is true or false by detecting the correlation peak in the output plane. The next generation of these security systems should offer a higher level of security and more sophisticated services than the simple verification offered by the existing systems. In this research we investigate one such optical security system

In the field of electronic ciphering, there are many encryption algorithms. One such algorithm is the exclusive-OR encryption (XOR encryption). This very popular method for encryption is based on the exclusive-OR Boolean operator.

1.2 Significance of The Research

Image encryption processing, is vital to the economy and the quality of life of people is increasingly affected by the pre-dominantly digital world today. To share this digital information, digital communication networks are required. Thus these systems are increasingly requiring higher levels of security control to allow access to information to authorized personnel only. It is suggested that image encryption based on XOR operations is a cheap and simple encryption method that may be used in digital systems. This technique can be used a variety of security systems for images transmission.

To secure privacy or access, in several modern systems, an image, such as a picture or a face or a fingerprint, can be used to identify individuals. This system could also be integrated into a network. If the memory units of the security system are stolen or if important data is intercepted by monitoring the transmission line in security system network, however, an unauthorized person can know vital information easily. With the rapid advances in electronic systems, it is becoming increasingly simple to reproduce the important data that was previously safe in a system that transmitted data over long distances. Therefore, data protection has become necessary and encryption and decryption layers, external to the digital transmission and communication system need to be added. This highlights the significance of this research.

1.3 Statement of the problem

The aim of the overall study was to develop an optical encryption processing system that was secure before entering the electronic communication system and to decrypt such an image after reception. Optical encryption techniques seemed to be promising for security applications because they take advantage of encryption outside the electronic domain. In this project, we propose an optical method that conceals the usual data of authorized persons by encryption before they are transmitted over wireless, stored, or compared in the pattern recognition system of security systems. This was to be developed as a low cost prototype to test the technique of optical encryption and decryption using XOR pixel by pixel logic operations.

1.4 Objectives of the research

The problem as stated above had two sub-problems: (1) Encryption and decryption of real time image system. (2) And the Transmission of the image over a mobile wireless broadband link. Recently, several optical processing systems for security have been suggested. The objective of this project is to propose a low cost optical image encryption technique based on exclusive-OR (XOR) operations for telemedicine images transmitted over a FLR or other IP over wireless broadband link or a security system that is only possible to be viewed by authorized persons with the necessary hardware.

Performing optical XOR operations with the key bit stream that are generated by a digital encryption algorithm should encrypt the input image. The optical XOR operations between the key data and the bit planes would then be performed by a polarization encoding method.

The research was conducted in accordance with the following objectives.

(i) Technology survey, comparison and evaluation of different image encryption methods as well as that based on XOR operations including its advantages.

(ii) Analysis of security and modeling of XOR optical encryption technique

(iii) Review digital communication principles, and modulation schemes, applicable to image transmission over a FLR or other IP over wireless mobile broadband link.

(iv) To write software for XOR encryption and decryption simulation in Matlab.

(v) Detailed hardware design of image encryption unit and decryption unit as well as setting up a FLR or other IP over wireless mobile broadband link.

(vi) Test the system performance.

1.5 Dissertation outline

The project was solved in the following manner.

A literature survey was performed in chapter 2. In this chapter, I reviewed the spectrum of encryption methods in order solve the problem. According to application, the encryption techniques were divided into still image and Audio/Visual Streams. This is further subdivided what is encoded, and again was sub-divided into frequency domain and spatial domain techniques. A comparison and discussion of the techniques were made in this chapter.

Chapter 3 introduces a discursion on optical encryption methods and in particular a discursion and comparison of double phase encryption and XOR encryption. The XOR encryption is shown to be feasible for implementation low cost implementation using existing LCD and CCD technology in this chapter.

Chapter 4 outlines a detailed analysis and modeling of the XOR technique with the aim of establishing a mathematical proof of the technique as having sufficient rigor for application as an encryption and decryption technique for information security. LCD and CCD implementation of the proposed encryption and decryption then follow.

It was also necessary to develop hardware and build the cryptographic system, and testing whether this system can carry out the optical encryption based on XOR algorithm. This is discussed in chapter 5.

The next part involved modeling and simulation of image encryption using XOR theory, including how to change the color image (RGB, three domain) to the binary image (black-white image). Key-bit generation was then investigated. Software routines were developed and a programs written using Matlab to test the encryption and decryption. The details of this are discussed in chapter 6. A link, controlling wireless IP was set up to transmit the data between the two terminals. Details of this are discussed in chapter 7. The results and a comparison are presented in chapter 8 and the conclusion and recommendations were presented in chapter 9.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter was reviewed the spectrum of encryption methods in order solve the problem. According to application, the encryption technique was divided into still image and Audio/Visual Streams processing. This was further subdivided into what is encoded, and again was sub-divided into frequency domain and spatial domain concerns. Future encryption directives are then outlined.

2.2 Different Encryption methods

Different cryptography methods are used to encrypt different data. In this chapter, I only discuss the encryption of image information. With the Internet development, large volumes of binary data are processed everyday. Text as well as audio was processed on a regular basis. The security of multimedia data in digital distribution networks is commonly provided by encryption. The mathematical process transforms a plaintext message into unintelligible ciphertext. Nevertheless, the classical and modern ciphers have all been developed for the simplest form of multimedia data, and are not appropriate for higher forms such as images and video with very large file sizes. Selective encryption is a recent approach to reduce the computational requirements for huge volumes of multimedia data in distribution networks with different client device capabilities. In this chapter, we provide a survey and classification of the proposed schemes, discuss the current issues and present some future directions.

Presently, encryption appears to be the only technical tool that can be used to provide confidentiality in applications such as video conferencing, DSTV and on-line video games. This is also very true of telemedicine application. The cost of multimedia data compression can be aggravated by the additional need for protecting copyrighted digital content. It is argued that the encryption algorithms, which have been originally developed for text data, are not suitable for securing many real-time multimedia applications because of large data sizes. Software implementations of ciphers are usually too slow to process image and video data in commercial systems. Hardware implementations, on the other hand, add more cost to service providers and consumer electronics device manufacturers. A major recent trend is to minimize the computational requirements for secure multimedia distribution by "selective encryption" where only parts of the data are encrypted. A classification of the proposed schemes from the open literature is given in Table 2-1. To the best of our knowledge, such an extensive classification appears in the relevant literature for the first time.

Table 2-1. Classification of selective encryption schemes

Type of data	Domain	Proposal	Encryption algorithm	What is encrypted?
Image	Frequency domain	Cheng & Li, 2000	No algorithm is specified	Pixel and set related significance information in the two highest pyramid Levels of SPIHT
		Droogenbroeck & Benedett, 2002	DES, Triple DES and IDEA	Bits that indicate the sign and magnitude of the non-zero DCT coefficients
		Pommer & Uhl, 2003	AES	Subband decomposition structure
		Cheng & Li, 2000	No algorithm is specified	Quadtree structure

	Spatial domain	Droogenbroeck & Benedett, 2002	XOR	Least significant bitplanes
		Podesser, Schmidt & Uhl, 2002	AES	Most significant bitplanes
Video	Frequency domain	Meyer & Gadegast, 1995	DES, RSA	Headers, parts of I-blocks, all I-blocks, I-frames of the MPEG stream
		Spanos & Maples, 1995	DES	I-frames, sequence headers and ISO end code of the MPEG stream
		Tang, 1996	Permutation, DES	Permutation, DES
		Qiao & Nahrstedt, 1997	XOR, permutation, IDEA	Every other bit of the MPEG bit stream
		Shi & Bhargava, 1998	XOR	Sign bit of DCT coefficients
		Shi, Wang & Bhargava, 1999	IDEA	Sign bit of motion vectors
		Alattar, A-Regib and Al-Semari	DES	Every n^{th} I-macroblock, headers of all the predicted macroblocks, header of every n^{th} predicted macroblock
		Cheng & Li, 2000	No algorithm is specified	Pixel and set related significance information in the two highest pyramid levels of SPIHT in the residual error

		Zeng & Lei, 2002	Permutation, XOR	Selective bit scrambling, block shuffling, block rotation of the transform coefficients (wavelet and JPEG) and JPEG motion vectors
	Spatial domain	Cheng & Li, 2000	No algorithm is specified	Quadtree structure of motion vectors and quadtree structure of residual errors
	Entropy codec	Wu & Kuo, 2000; Wu & Kuo, 2001	Multiple Huffman tables, multiple state indices in the QM coder	Encryption of data by multiple Huffman coding tables and multiple state indices in the QM coder

The schemes listed in Table 2-1 are used to protect 5 types of bitstreams:

- Uncompressed images
- JPEG compressed stream (images)
- MPEG compressed stream (video)
- Wavelet-based compressed stream (images and video)
- Quadtree-based compressed stream (images and video)

Note that a majority of the schemes are developed for JPEG and MPEG compliant streams. We will give a brief description of the proposed schemes.

2.2.1 Schemes for Still Images

At below, they are many typical encryption methods for still image.

2.2.1.1 Selective Encryption in Frequency Domain

Cheng & Li, 2000: In general, wavelet compression algorithms based on zerotrees transmit the structure of the zerotree with the significant coefficients. The SPIHT algorithm, for example, transmits the significance of the coefficient sets that correspond to trees of coefficients. Among the many different types of bits generated by the SPIHT algorithm, the proposed partial encryption scheme encrypts only the significance information related to pixels or sets in the two highest pyramid levels in addition to the parameter n that determines the initial threshold.

Droogenbroeck & Benedett, 2002: In JPEG compression, the Huffman coder aggregates zero coefficients into runs of zeros and uses symbols that combine the run of zeros with magnitude categories for the non-zero coefficients that terminate the runs. These symbols are assigned 8-bit code words by the Huffman coder. The code words precede the appended bits that specify the sign and magnitude of the non-zero coefficients. In the proposed scheme, the appended bits corresponding to a selected number of AC coefficients are encrypted. The DC coefficients are left unencrypted because, it is argued, and they carry important visible information and are highly predictable.

Pommer & Uhl, 2003: The encoder chooses different decomposition schemes with respect to the wavelet packet subband structure for each image that needs to be protected. Classical best basis selection algorithm is not appropriate to determine a useful wavelet packet basis as it results in trees that share common features for many images, leading to a potential security weakness. Instead, the generation of the decomposition tree is randomized using a pseudo random number generator (PRNG). The tree carrying the subband decomposition structure is then secured for transmission with AES encryption.

2.2.1.2 Selective Encryption in Spatial Domain

Cheng & Li, 2000: Quadtree image compression produces two logical parts: the quadtree and the parameters describing each block in the tree. The only parameter

used by the authors to describe each block is the average intensity. As intensity corresponds to a leaf node in the quadtree, the block intensities are called the *leaf values*. In the proposed partial encryption scheme, only the quadtree structure is encrypted. It can be used for both lossy compression (where each leaf is represented by the same number of bits) and lossless compression (where the number of bits to represent each leaf is different). For the transmission of the leaf values, two orderings are introduced: Leaf Ordering I (in order traversal of the quadtree) and Leaf Ordering II (the leaf values are encoded one level at a time from the highest level to the lowest level). For security reasons, Leaf Ordering I is not recommended for lossy or lossless compression while Leaf Ordering II is reportedly secure for both.

Droogenbroeck & Benedett, 2002: The decomposition of a gray scale image into its 8 bitplanes shows that the highest bitplanes exhibit some similarities with the original image while the least significant bitplanes look random. To exploit this property, some of the least significant bitplanes are encrypted. It is observed that at least 4 or 5 bitplanes need to be encrypted before the degradation becomes visible.

Podesser, Schmidt & Uhl 2002: The gray scale image is decomposed into its 8 bitplanes and the most significant bitplanes are encrypted. After a number of experiments, it is observed that (1) the encryption of the most significant bitplane is not secure enough, (2) selectively encrypting 2 bitplanes is sufficient if severe alienation of the image data is acceptable, and (3) encryption of 4 bitplanes provides high confidentiality.

2.2.2 Schemes for Audio/Visual Streams

At below, they are many typical encryption methods for dynamic image.

2.2.2.1Selective Encryption in Frequency Domain

Meyer & Gadegast, 1995: A new bit-stream called SECMPEG is a modified version of MPEG, and incorporates selective encryption and additional header information for bit-error recovery. Four levels of security are implemented: (1) encryption of the headers from the sequence layer down to the slice layer, (2) encryption of parts of the *I*-blocks, (3) encryption of *I* frames and all *I*-blocks, and (4) full encryption.

Maples & Spanos, 1995: A new security mechanism called Aegis is presented. Aegis encrypts the *I* frames of all groups of frames in an MPEG video stream. The encryption of the *I* frames is justified by the fact that they have great significance in the decompression of an MPEG stream whereas *B* and *P* frames represent only translations of the picture information found in adjacent *I* frames. Aegis is also used to encrypt the MPEG video sequence header and the ISO (International Organization for Standardization) end code (last 32 bits of the MPEG video stream) to conceal the identity of the bit stream.

Tang, 1996: The basic idea is to use a random permutation list to replace the zig-zag order in mapping the 8x8 DCT (Discrete Cosine Transform) block to a 1x64 vector. Six experiments were conducted with different variations of permutation: (1) The DC coefficient is mapped to the first element in the 1x64 vector; the 63 AC coefficients are randomly permuted. (2) The DC coefficient is set to zero; the 63 AC coefficients are permuted according to the zig-zag order. (3) All coefficients are randomly permuted and the DC coefficient is not in the first position of the vector. (4) All coefficients are permuted according to the zig-zag order; the last AC coefficient is set to zero. (5) The DC coefficient is split according to the splitting procedure; all coefficients are permuted randomly. The splitting procedure is defined as follows: Let $d_7 d_6 \ldots d_1 d_0$ be the 8-digit binary representation of a DC coefficient. It is split into two numbers $d_7 d_6 d_5 d_4$ and $d_3 d_2\ d_1 d_0$, which are both in the range of [0,15]. Set the value of the DC coefficient to be $d_3 d_2\ d_1 d_0$, and the value of the last AC coefficient to be ${}_7 d_6 d_5 d_4$. (6) The DC coefficient is encrypted by the function f_i $(x_{i1} \ldots x_{i8}) = (k \oplus (x_{11} \ldots x_{18} \ldots x_{81} \ldots x_{88}))_{8*i+1,\ldots,8*i+8}$, where k is a 64-bit secret key,

x_{ij} is the *ij*-th bit of the binary sequence formed by grouping eight DC coefficients together, and $\oplus$ is the binary *XOR* operation; splitting and random permutation procedures are applied.

Qiao & Nahrstedt, 1997: The basic approach is to take the chunk $a_1a_2a_3...a_{2n-1}a_{2n}$ of an *I*-frame and create the two byte streams $a_1a_3...a_{2n-1}$ (odd list) and $a_2a_4...a_{2n}$ (even list). The substreams are then xored to obtain the ciphertext $c_1c_2c_3...c_n$, which is concatenated to E ($a_2a_4...a_{2n}$), where E denotes an encryption function. If $a_2a_4...a_{2n}$ has no repeated pattern, the secrecy depends on the function E as $a_2a_4...a_{2n}$ can be considered to be a onetime pad. Using the basic idea, an algorithm is developed with several keys: *keyM* is used to derive the two byte streams; each key_i (i=1...8) shuffles a chunk of data to obtain a non-repeated pattern with a length of 1/2 frame (it was observed that the non-repeated patterns have a life time of over only one 1/16 chunk); *keyF* is assigned to each frame to change the pattern of choosing even and odd lists (which is applied to *keyM* repeatedly and to key_i to derive new keys for each frame); and *keyE* is the encryption key for the function E. *keyM*, key_i's and *keyF*'s can be encrypted with *keyE*, and *keyE* can be sent via a separate secure channel. Alternatively, *keyM* and key_i can be sent in a separate secure channel.

Shi & Bhargava, 1998: The Video Encryption Algorithm (VEA) uses a secret key to randomly change the sign bits of the DCT coefficients of MPEG video. VEA's secret key $k = b_1b_2...b_m$ is a randomly generated bitstream of length *m*. If the sign bits of DC and AC coefficients are represented by $S = s_1...s_m s_{m+1}...s_{2m}$, VEA's encryption function is $E_k(S) = ...(b_1 \oplus s_1)...(b_m \oplus s_m)(b_1 \oplus s_{m+1})...(b_m \oplus s_{2m})...$, where $\oplus$ is the binary *XOR* operation. VEA does not have a limit on the key length or number of keys. Multiple keys can be used in several ways: 2 keys, one for Y blocks, one for Cb and Cr blocks; 3 keys, one for Y blocks, one for Cb blocks and one for Cr blocks; 3 keys, one for *I* frames, one for *B* frames and one for *P* frames.

Shi, Wang & Bhargava, 1999: Real-time Video Encryption Algorithm (RVEA) is based on the previous work VEA and introduces two improvements: it adopts ciphers to increase security and limits the maximum number of selected bits to bound the computation time. For each 16x16 macroblock in a video slice, RVEA selects at most 64 sign bits. The order of the sign bits in the six 8x8 blocks Y1, Y2, Y3, Y4, Cr and Cb is defined in a specific way with the consideration that DC coefficients are more significant than AC coefficients and lower frequency AC coefficients are more significant than higher frequency AC coefficients. The selected sign bits are encrypted with DES or IDEA and put back in their original positions.

Alattar, A-Regib and Al-Semari, 1999: Three methods are proposed to improve the performance of an earlier work by two of the co-authors. In the first method, encryption is applied to the data associated with every n^{th} *I*-macroblock. In the second method, the headers of all predicted macroblocks are encrypted together with the data associated with every n^{th} *I*-macroblock. To reduce the computational load, the third method encrypts only the headers of every n^{th} predicted macroblock in addition to the data associated with every n^{th} *I*-macroblock.

Cheng & Li, 2000: An extension of the image compression algorithm called the Set Partitioning in Hierarchical Trees (SPIHT) algorithm (which is an implementation of zerotree wavelet image compression) is used to encode the residual error. Since the residual error is an image frame, the partial encryption scheme for wavelet image compression can be directly applied to residual error coding. It is observed that the relative size of the important part of the residual error is similar to the size in image compression.

Zeng & Lei, 2002: The frequency domain scrambling technique divides the transform coefficients into blocks/segments and performs some or all of the following three operations: selective bit scrambling, block shuffling and block rotation of the transform coefficients and motion vectors. Two compression schemes are used to

illustrate the approach: wavelets transform based compression and 8x8 DCT based compression. In the simulations, several combinations of the three scrambling operations are tested in the wavelet and DCT domains.

2.2.2.2 Selective Encryption in Spatial Domain

Cheng & Li, 2000: Video compression algorithms generally address motion compensation and residual error coding. An extension of the quadtree compression algorithm is used to encode both motion vectors and residual errors. Since the residual error is an image frame, partial encryption scheme for quadtree image compression can be directly applied to residual error coding. It is observed that the relative size of the important part of the residual error is similar to the size in image compression. In encoding motion vectors, the leaf values become the motion vectors. As in the original scheme for images, the encrypted part is the quadtree decomposition, not the motion vectors. For low-resolution videos where the maximum height of the tree may be small enough to allow exhaustive tree enumeration, the motion vectors are completely encrypted.

2.2.2.3 Entropy Codec Design

Wu & Kuo, 2000; Wu & Kuo, 2001: Base on a number of observations, the authors argue that selective encryption (i.e., the selection of the most important coefficients from a compression system and their encryption with a conventional cipher) may not be effective in two situations: (1) The media compression system is based on an orthogonal transform followed by quantization, and (2) The media compression system contains entropy coding at the last stage. Using multiple statistical models, they investigate a different encryption methodology that turns entropy coders into ciphers. This methodology allows the construction of two selective encryption schemes by application to the Huffman coder and the QM coder, two of the most popular entropy coders in multimedia compression. Both coders have very simple statistical models; the model of the Huffman coder is usually a fixed-size non-

adaptive binary tree, and the initial state of the QM coder includes only three integer numbers. Since hiding the Huffman coding table or the initial state of the QM coder does not provide sufficient secrecy, it is argued that the problem of a limited key/model space can be overcome by using m statistical models instead of only one. The first of the proposed encryption schemes makes use of multiple Huffman tables and the second multiple indices in the QM coder estimation state machine.

2.3 Security Definitions

It is clear that a minimal requirement of security would be that: any adversary who can see the ciphertext and knows which encryption and decryption algorithms are being used, cannot recover the entire original text. But, many more properties may be desirable. Four of these are listed below:

1. It should be hard to recover the messages from the ciphertext when the messages are drawn from arbitrary probability distributions defined on the set of all strings (i.e arbitrary message spaces). A few examples of message spaces are: the English language, the set {0,1}. We must assume that the adversary knows the message space.

2. It should be hard to compute partial information about messages from the ciphertext.

3. It should be hard to detect simple but useful facts about traffic of messages, such as when the same message is sent twice.

4. The above properties should hold with high probability.

In short, it would be desirable for the encryption scheme to be the mathematical analogy of opaque envelopes containing a piece of paper on which the message is written. The envelopes should be such that all legal senders can fill it, but only the legal recipient can open it.

2.4 Current Issues

There are a lot of the factors that affect security even if one use cryptography. The computing power of machines, key used, randomness of key and the algorithm itself can affect the security of a cryptography scheme. The transmission channel, or type of application also plays a role. This will be discussed in 2.5 below. An algorithm is computationally secure if it cannot be broken with computing resources. However, computing power has advanced quite well in the last 30 years. One should note that most algorithm attacks nowadays also deploy parallel computer processing power. Therefore an algorithm that is secure today might not be secure tomorrow or next year. The key length is also an important factor that affects the performance of an algorithm. We can use longer keys to make attack unfeasible. Keys that use a variety of combinations and avoid using English words or names are more secured. A key should also be changed frequently and old keys should not be reused. Most cryptographic applications demand truly random sequence generator. Computers can, however only produce pseudo-random sequences. People think that by keeping the algorithm secret it will make the scheme secure. But in actual fact any computer programs can be decomposed. A good algorithm can be made public without any concern. It can even be published but without the decryption key, even its designers cannot decrypt the encrypted message. [Eskicioglu, A. M. and Delp, E. J. (2001)]

As noted earlier, most of the modern encryption algorithms were developed for text data, i.e., the simplest form of multimedia. The more complex forms (especially image and video) may vary substantially in their cost of creation as well as their storage and communication channel requirements. When compared with bank account information, for example, the value of a movie is much lower while the bit rate is much higher.

There is a wide spectrum of secure multimedia applications with different requirements. They range from military applications that mandate total data obscurity

to applications where a part of the multimedia data needs to be visible to allow searching in a shared database. Hence, it is desired to develop an encryption-based technology that is appropriate for many of such scenarios. The desirable attributes of such a technology include [Doerr, G. and Dugelay, J. -L. (2003)]:

- That it should provide sufficient security for a range of multimedia applications,
- That it should preserve the size of the original unencrypted bit stream,
- That it should result in substantial computational reduction with respect to total encryption,
- That it should produce a bit stream that is compliant to the standard formats,
- That it should not create a key whose size is much longer than those of commonly used modern ciphers,
- That it should identify the portions of the multimedia data to be encrypted.

The transmission channel also plays a role in the application of a successful image or media encryption process.

2.5 Wireless Technologies Evolution

In this part of the review, concerns about the transmission medium, whether landline based or wireless is discussed. In February 1896, Guglielmo Marconi had developed the first operational wireless telegraph apparatus. He filed a British patent application on June 2 of that year. Signals were sent in July 1896 over a distance of one-and-three-fourths miles on Salisbury Plain. [David G. Leeper (2003)]

The technology development in communications and communication channels are listed below. Only what has been considered the most appropriate in terms of this project are presented.

1896 - Guglielmo Marconi developed the first wireless telegraph system

1970s - Packet switching emerges as an efficient means of data communications, with the X.25 standard emerging late in the decade

1992 - One-millionth host connected to the Internet, with the size now approximately doubling every year

2000 -IEEE 802.11(b)-based networks became successful and were in popular demand

2000 January - Wired Equivalent Privacy (WEP) Security is broken. The search for greater security for IEEE 802.11(x)-based networks increases

2.6 Conclusion

One can reliably conclude from the above that several initiatives influenced the direction of image and multimedia encryption.

Since key management (i.e., the generation, storage and replacement of keys) was a critical issue in all encryption based security systems, it cannot be separated from the design of secure multimedia distribution. In most distribution architectures, multimedia content was encrypted with a symmetric key, which also needed to be protected in transmission to the receiver. A common tool of achieving the protection of the decryption key is public-key cryptography. The difficulty in cryptanalizing public-key ciphers would provide reasonable security in most applications. In some of the selective encryption schemes surveyed in this paper, key generation is not properly addressed. To be able to maintain entropy-coding efficiency, these proposals avoid using standard ciphers, and instead employ key-based permutation, resulting in key lengths much longer than 64 or 128 bits needed for cryptographically strong symmetric ciphers. Furthermore, the security of operations such as coefficient shuffling may not be as strong as that of hybrid modern ciphers that employ both

permutation and substitution. Hence, the storage and security requirements of key management need to be discussed in greater detail in future proposals.

The classification as shown above in Table 2-1 indicates that a common approach in selective image encryption is to integrate compression and encryption processes whereby only a subset of the transform coefficients (or some of their bits) was scrambled. Selective encryption is the process of encrypting only parts of an image content to reduce the computational requirements of client devices in real-time applications. However, it has been observed that energy concentration via transform domain compression does not imply intelligibility [Wu, C.-P. and Kuo, C.-C. J. (2001)]. A consequence of this is that selective encryption does not meet criteria for orthogonal transform based compression (today's compression standards - JPEG for image compression, MPEG for video compression, and MP3 for audio compression). It may therefore be desirable to keep compression and encryption as two separate processes. Development of a general selective encryption scheme for still images and video with the listed desirable attributes is a complex problem and outside the scope of this research.

In this chapter was presented a survey and classification of a number of encryption schemes in the literature. The major problems associated with most of these schemes were insufficient security, decrease in the compression performance and insignificant computational reduction with respect to total encryption. There were also a lack of bit stream compliance and increase in encryption key size. Future directions of research include more emphasis on key management, resolving the conflict between compression and encryption, and finding ways to change the selection criteria dynamically. With the increasing availability of digital distribution and storage technologies, the demand for multimedia services is on the rise. The most effective methods of cryptography are separated in frequency and spatial domain. All of these methods does however indicate that a development of purely optical techniques in encryption, being external to the digital computer domain have not yet made sufficient impact. There is a wide gap for investigations into purely optical techniques

to enhance image and multimedia encryption and security over communication and computer network links. The consideration of an optical technique, such as light polarization encoding using XOR logic in the light transmission path from an image to a capturing device, will be the topic of the next chapter.

CHAPTER3

OPTICAL ENCRYPTION METHODS

In this chapter, purely optical encryption methods will be focused on and the object would be to discuss an effective XOR encryption scheme for practical demonstration.

3.1 Introduction

The subject of optical encryption techniques was until recently not a very popular one, but is presently enjoying increasing prominence. Conventional reproduction techniques such as photocopiers have been increasing in quality steadily for the last thirty years, and therefore necessitate innovative countermeasures to counterfeiting. As mentioned in chapter 2, in the last decade the importance of cryptography has risen greatly with the expansion of public communication systems. The coincident rise of optical holographic memories has created a burgeoning field of optical and optoelectronic encryption. Figure 3-1 shows a typical encryption and decryption channel with a possible noise perturbation added in.

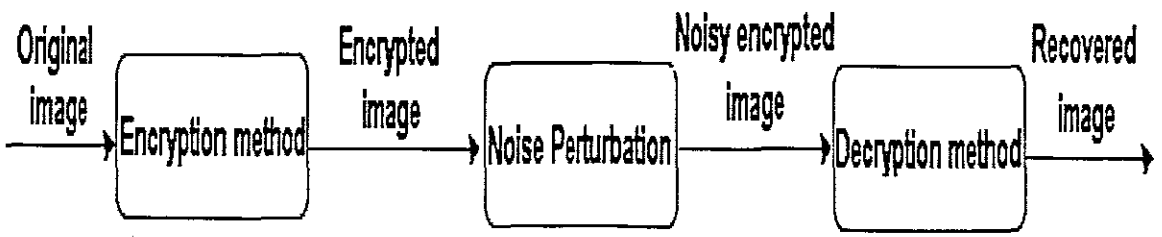


Figure 3-1 Block principle diagram of the process used to test the robustness to noise of the encryption methods. The noise is introduced in the encrypted data.

In this section are discussed two current optical encryption methods, and a comparison is made to each other.

3.2 Current Encryption

Current cryptography abandons the assumption that the adversary or code cracker has available infinite computing resources, and assumes instead that the adversary's computation is resource bounded in some reasonable way. In particular, the adversary is a probabilistic algorithm who runs in polynomial time. Similarly, the encryption and decryption algorithms designed are probabilistic and run in polynomial time.

Modern optical cryptography is based on a gap between efficient algorithms for encryption for the legitimate users versus the computational infeasibility of decryption for the adversary; it requires that one have available primitives with certain special kinds of computational hardness properties. Optical addition logic algorithm encryption is a rising subject in this domain. Optical cryptography has some unique characteristics, which allows that complex algorithms can be avoided. This encryption method is invariably faster than computational methods as discussed in chapter 2. The important point to relies is that the method is difficult to break and that and aspect of hardware dependability is required. The techniques explored in optical encryption and decryption utilizes light and changes in the path of light, which either use some logic algorithms or inflect the spectrum to achieve the objective.

3.3 Double-phase Encryption Technique

Double phase encoding [Goudail F., Bollaro F., Javidi B., and Refregier P. (1998)] is a technique that transforms the input information to a white Gaussian noise. As a result, it is difficult to decode the original information without knowing the codes used in the encoding process. So it is a good and effective encryption and decryption method for images.

Double phase encoding redistributes the energy of the input image such that the encoding data is white Gaussian noise-like.

Let the input to the double phase encoder be a complex signal $H(x,y)$. $b1(x,y)$ and $b2(x,y)$ are two statistically independent uniformly distributed random variables from 0 to 1. Here we denote (x,y) as the spatial coordinates. Let $\psi_1(x,y) = e^{2j\pi b1(x,y)}$ and $\psi_2(\xi,\gamma) = e^{2j\pi b1(\xi,\gamma)}$ be two phase functions. The double phase encoding signal, $H_d(x,y)$ is then given by:

$$H_d(x,y) = \{H(x,y)\psi_1(x,y)\} \oplus IFT[\psi_2(\xi,\gamma)] \tag{3.1}$$

where (ξ,γ) are the coordinates in the Fourier plane and *IFT* stands for inverse Fourier transform. And the symbol $\oplus$ stands for convolution. The decoding process is straight forward [Refregier P. and Javidi B. (1995)].

Using an analysis similar to [Joseph Rosen, Bahram Javidi (2001)] it can be shown that the encoded double phase encoded complex signal is a white Gaussian noise having a variance given by:

$$\sigma^2 = \frac{1}{N.M} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} H(x,y)H^*(x,y) \tag{3.2}$$

3.3.1 Implementations and Uses

Two-lens classical processors and Vander Lugt correlator [Goodman J. (1996)] have been very effectively used in optical information processing for the last number of years. Recently, a number of novel cryptographic algorithms incorporating these devices have been proposed. The most mature technique, double-phase encoding, will be explained, and its security will be analyzed.

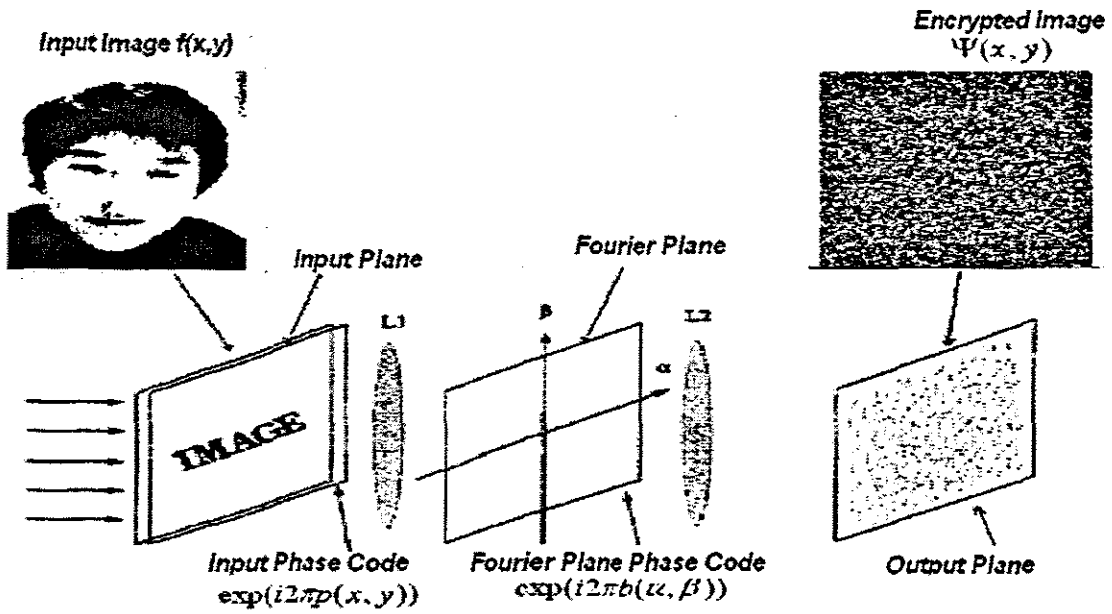
The discrete 1-D notation is used in this analysis. Let l denote the coordinate in the spatial domain and v the coordinate in the Fourier domain. Let $o(l)$ denote the image to be encrypted and $e(l)$ is the encrypted image. Let $k_1(l)$ and $k_2(v)$ be two independent white noise sequences, which are uniformly distributed in [0,1]. The double-phase encryption of the input image $o(l)$ is obtained by two operations. First $o(l)$ is multiplied by a phase mask function $e^{i2\pi k_1(l)}$. Second, the product $o(l)e^{i2\pi k_1(l)}$ is convolved by a function $h(l)$, which is the impulse response of a phase-only transfer function, that is,

$$FT[h(l)](v) = e^{i2\pi k_2(v)}. \tag{3.3}$$

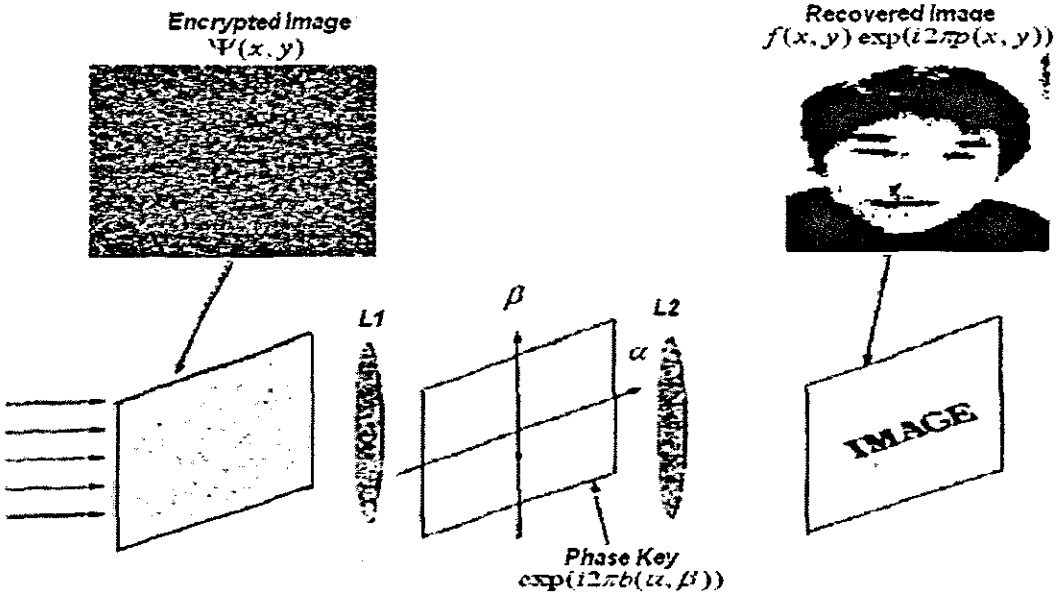
Here, FT denotes the Fourier transform. Thus the encrypted image $e[l]$ is given by the following equation:

$$e(l) = [o(l)e^{i2\pi k_1(l)}] * h(l), \tag{3.4}$$

where * denotes the convolution operation. The [Refregier P. and Javidi B. (1995)] encrypted image $e(l)$ is a wide-sense stationary white sequence.



(a)



(b)

Figure 3-2 Encryption-Decryption of a Double-Phase system

(a) Optical implementation of the double-phase encryption

(b) Optical decryption for the double-phase encryption technique

Figure 3-2(a) illustrates the process of encryption. For the decryption, we have

$$o(l) = [e(l) * h^*(-l)]e^{-i2\pi k_1(l)}. \tag{3.5}$$

When wide-sense stationary noise is added to the encrypted image [Javidi B., Sergent A., Zhang G., and Guibert L. (1997)], the additive noise for the recovered image is a wide-sense stationary white noise regardless of the type of the noise added to the encrypted image. Figure 3-2(b) illustrates the decryption process.

3.3.2 Analysis of Security

The security of this method is sadly poor [Javidi B., Zhang G., and Li J. (1998)]. While there are some statements as to its apparent security, there is nothing in the literature that approaches a formal proof. Furthermore, its linearity opens several avenues of attack.

The few statements of security of this algorithm centre on the fact that the encrypted data has the form of stationary white noise (the random input phase is required for this property). While this may be true, it is no guarantee of security. Almost every major cipher has this property, and some of them are quite insecure. Also, tests showing that a very small number of keys chosen from the key space fail to properly decrypt an encrypted image are of little value, since this offers no assurances that other randomly chosen keys will similarly fail.

The key flaw in this algorithm is that it is a linear system, which allows a number of attacks that would not be possible in non-linear algorithms. For example, if the same key is used to encode multiple images (as suggested in [Refregier P. and Javidi B. (1995)]), attackers can take advantage of the correlation between two (or more) encrypted images to find the key (this assumes only a modicum of structure in the original images). Worse yet, if one of the unencrypted images is somehow disclosed, the key can be found as follows. First, take the Fourier transform of the encrypted

image and conjugate it. Place this in the Fourier plane of a two-lens processor and place the unencrypted image in the input plane.

Note that we now have the key and can decrypt any other message that was encoded with it. Therefore, this algorithm is only secure if key material is never reused, which is a fairly stringent requirement. Also, note that this technique is to some extent immune to multiplicative and additive white noise [Javidi B. (1999)]. This implies that each output pixel is not a function of every input pixel, which would tend to make iterative key guessing attacks much less complex.

3.4 A Brief Review XOR Encryption Technique

This encryption technique is appropriate for binary signals and it encrypts each pixel (bit) individually. The method made use of the XOR Boolean operator, defined for binary inputs. The applied XOR logic returns True if one of the values is True and the other is False. It returns False if both arguments are True or both are False. For example, True XOR True is False, but True XOR False is True. The simulation processing is shown in Figure 3-3.

A	B	A xor B
T	T	F
T	F	T
F	T	T
F	F	F

(3.6)

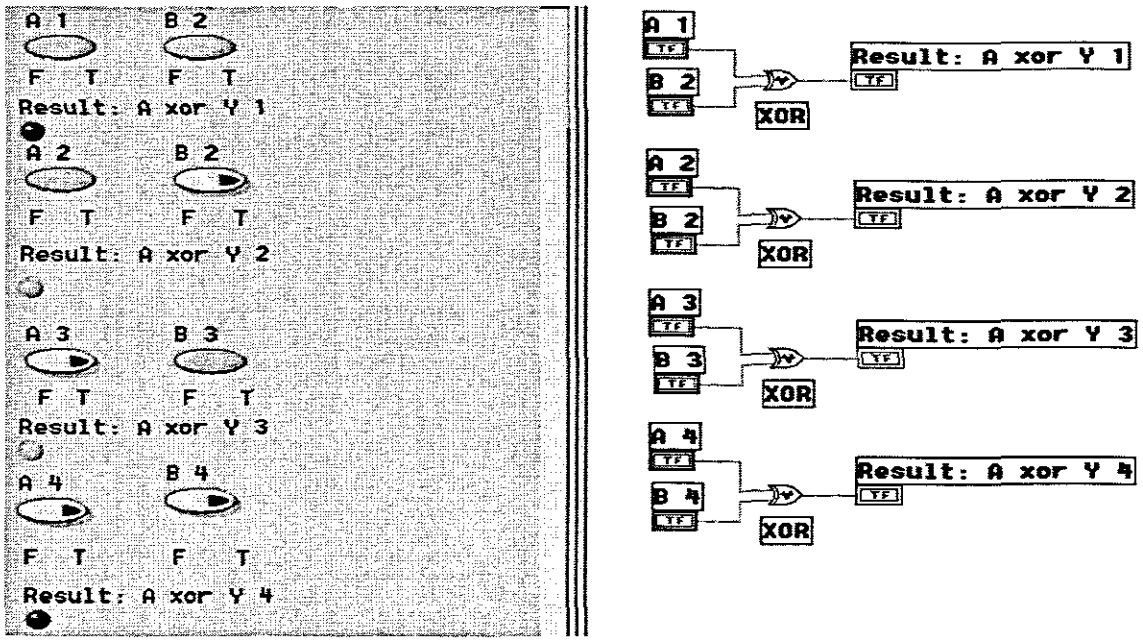


Figure 3-3 Simulation XOR Operations in LabVIEW

The XOR encryption is based on the following property: let $o(l)$, which is original bit stream or image, be the bit we wish to encrypt, and $k(l)$ the corresponding bit from the key. The key $k(l)$ is random binary stream. To make the key difficult to decode, the distribution of 1's and 0's should be equal and they should be randomly located. The encrypted bit is given by

$$e(l) = o(l) \oplus k(l). \tag{3.7}$$

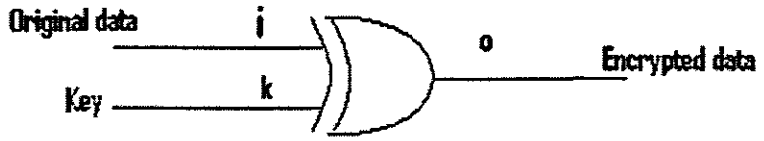
Where $e(l)$ is encrypted bit stream.

To decrypt, we simply use the fact that

$$o(l) = e(l) \oplus k(l). \tag{3.8}$$

The operations for the encryption and decryption are identical.

(a) XOR Encryption



(b) XOR Decryption when no noise occurs

(c) XOR Decryption in the noisy case

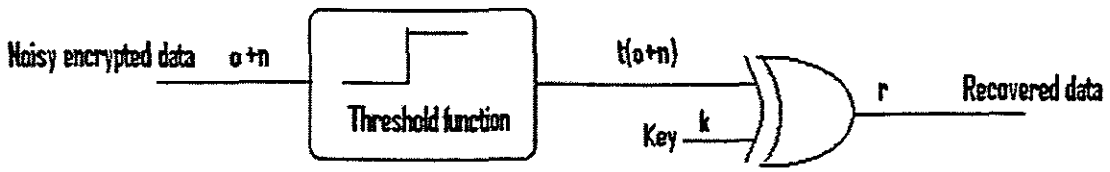


Figure 3-4 Model of XOR encryption and decryption

3.4.1 Implementations and Uses

A quick survey of major symmetric encryption algorithms will show that they tend to be composed of a bit-wise logical operators (mostly exclusive-or), bit-permutations and lookup tables. Each of these can be implemented with varying degrees of ease.

A two-input exclusive-or (XOR) gate can be implemented very simply using two polarizers and two liquid crystal displays (Figure 3-5) [Han J-W., Park C-S., Ryu D-H., and E-S. Kim (1999)] as follows. Note first that an LCD that is switched on will rotate light by 90°, while one that is switched off will simply pass the light. One

arbitrary polarization (in this case, horizontal) is designated to be 1, while the other (vertical) is zero. Unpolarized light is then passed through a "0" polarizer, then through two LCD's which are either on or off depending on their input value, and then outputted through a "1" polarizer. If the light passes through exactly one LCD, which is switched on, it will be able to pass through the output polarizer and is considered a 1 output. If it passes through either zero or two polarizers which are switched on, it will not be able to pass through the output polarizer and will be considered a 0. It is evident that this scheme can be trivially extended to an arbitrary number of inputs. All other logic gates can also be implemented optically. The reader is directed to [Khan A. and Nejib U. (1987)] for more details.

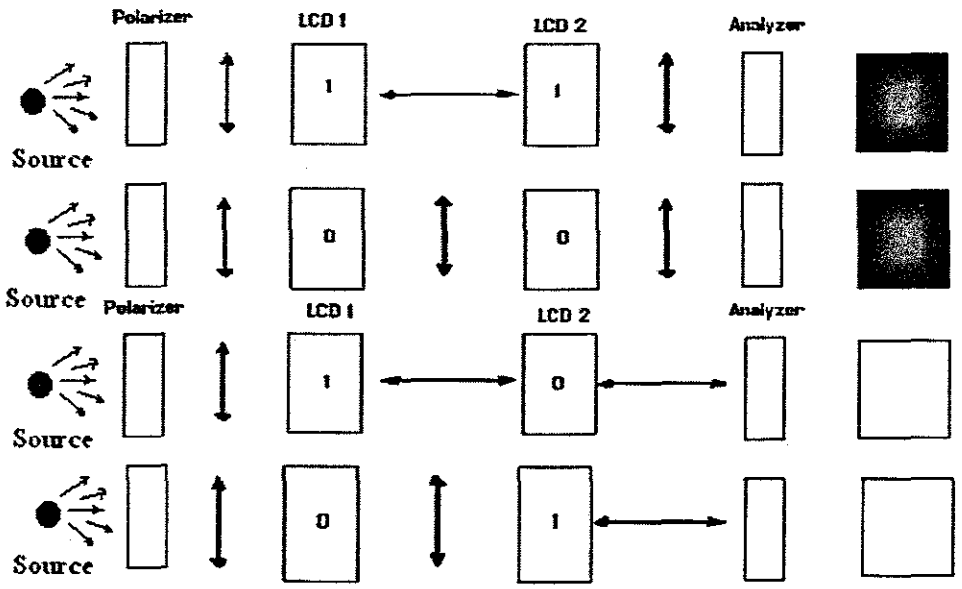


Figure 3-5 XOR optical logic operation using two LCD's

The optical structures to produce look-up tables are fairly involved, and are described in [Schmalz M. (1994)]. Here is also describes how to create bit-permutations, which are simply appropriate lens structures, and details an optical implementation of the U.S. Data Encryption Standard (DES).

The majority of asymmetric (public key) encryption algorithms are based around modular exponentiation. While this operation can be decomposed into a sequence of logical operators (in fact, any computer operation can), it is likely that the algorithm will become very cumbersome to implement in optoelectronic hardware at this point.

The major benefits of optoelectronically implemented cryptography is that a vast number of operations can be processed in parallel very quickly in a relatively straight-forward and cost-effective manner compared to similar operations in electronics.

Since these algorithms are identical to those already in use, and there is no obvious change in security attributable to optical implementation, they will have the same security. Specifically, XOR-only encryption schemes (as with double-phase encoding) will be perfectly secure if and only if the key data is perfectly random and never reused (ie. a one-time pad); otherwise, correlations between blocks encoded with the same key can be exploited to easily break the encryption. DES itself is reasonably securing again most attackers, however, those with medium resources can decrypt a message through brute force in an average of 2 days [Gilmore J. (1998)].

3.5 Conclusion

In this chapter double phase encryption and optical XOR encryption was discussed. Double phase encoding is a optical technique that transforms the input information to a white Gaussian noise. As a result, it is difficult to decode the original information without knowing the codes used in the encoding process. So it is reasonably good and effective encryption and decryption method for images but security is not infallible. XOR techniques are not as simple to implement as double phase encryption and has the potential for pixel-by-pixel parallel processing, therefore more complex key bit arrangements. The major benefits of optoelectronically implemented cryptography is that a vast number of operations can be processed in parallel very quickly in a relatively straight-forward and cost-effective manner compared to similar operations

in electronics. The development of an optical XOR encryption scheme is covered in the next chapter.

CHAPTER 4

DEVELOPMENT OF THE XOR ENCRYPTION SYSTEM

4.1 Introduction

With the rapid advances in electronic and optical systems, it is becoming increasingly simple to reproduce the important data of the security system. Therefore, data protection has become necessary. A novel optical technique for recording data for identification, verification, or validation was proposed. In this section, we propose an optical method that conceals the data of authorized persons by encryption before they are stored or compared in the pattern recognition system for security system. In this section, we recommend a new optical encryption method based on XOR technique. It is a simple and effective method, which can conceal the data of authorized persons for access to a restricted area. It can encode the data before those are transmitted through Internet or other else networking of wireless. This technique can apply into long distant security communication including still image and active image between the sender and receiver.

In this research is proposed an optical image encryption technique based on exclusive-OR (XOR) operations for information security system. The basic idea is that we convert a gray-level image to binary image for image encryption. We used the XOR operation that is commonly used such as the well-known encryption method. Performing optical XOR operations with the key bit streams, which are generated by digital encryption algorithms, encrypts the input image. The gray-level image is converted to binary image, which is represented on a liquid crystal device (LCD). The key data represented on different LCD is converted binary bit stream by random performing. The optical XOR operation between the key data and the binary image

are performed by the polarization encoding method. The results of XOR operation are detected by a CCD camera and display on computer or LCD.

4.2 XOR Algorithm

In the field of electronic ciphering, there are many encryption algorithms, one such algorithm is the exclusive-OR encryption (XOR encryption). This very popular method for encryption is based on the exclusive-OR Boolean operator as explained before in chapter 3.

In this algorithm, one encrypts data by XORing (exclusive-OR) the bits of the message with the bits of a string of key data.

The Boolean operator XOR acts as follows:

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0 \qquad (4.1)$$

To get a flavor for how a secret key encryption algorithm would work, let's look at one of the simplest: XOR. XOR is the name of an operation that can be performed on any binary number, including binary numbers that represent characters in computers.

The XOR takes two of these numbers and produces a third number using a bit-by-bit mapping. For example, consider the assignment of $C = A \oplus B$. In any bit position, if either **A** or **B** (but not both) has a one in that position, then **C** gets a one in that position. If both **A** and **B** have a one, or both have a zero, the **C** gets a zero in that position.

4.2.1 XOR Encryption

To encrypt a message to be sent, the characters in message are XORed with the key (Encrypted = Message XOR Key).

For example, you could use the key 10110101 10101101 to encrypt the following binary data (each key sequence is underlined for grouping).

Plain	01010011 01100101 01101001 01111010 01100101 00100000 01110100 01101000
Key	01001110 01100111 01001110 01100111 01001110 01100111 01001110 01100111
Encrypted	00110111 00000110 00101010 01000111 00101011 00001111 00111010 01000111

Notice that if the bit in the key sequence is a 1, then the data is inverted, but that if the bit is 0, then the data stays the same. It is certainly a very easy scheme.

4.2.2 XOR Decryption

To decrypt a received message, characters of the encrypted message are XORed with the same key (Message = Encrypted XOR Key).

Decryption is equally easy. Simply XOR the encrypted data with the key to get the plain data:

Encrypted	00110111 00000110 00101010 01000111 00101011 00001111 00111010 01000111
Key	01001110 01100111 01001110 01100111 01001110 01100111 01001110 01100111
Plain	01010011 01100101 01101001 01111010 01100101 00100000 01110100 01101000

This is easily proven using Boolean algebra. Let **P**, **K**, **C**, and **D** be variables representing the **p**lain data, **k**ey, **c**ipher data, and **d**eciphered data bits. We compute $C = P \oplus K$. Then compute $D = D \oplus K = (P \oplus K) \oplus K$, which is equivalent to

$D = P \oplus (K \oplus K)$ by the associative law. Since any variable *xor* itself equals 0, $D = P \oplus 0 = P$. Thus we see that $P = C \oplus K$.

As an exercise, derive a mathematical proof that the same key can be used for both encryption and decryption. Start by investigating the interesting property of the XOR function that it is self-reciprocating, meaning that the expression $(A \oplus B) \oplus B = A$ holds true for any **A** and **B** values.

4.3 The XOR Scheme Analyzed for Authentication of Information Transmission

4.3.1 Analysis

The veracity of the XOR Technique needs analysis. With *XOR* schemes a certain block size l is say given. We choose a parameter m, this time $1 \le m \le l-2$. We view the message M as being divided into blocks of size $l-m-1$, and denote the $i-th$ block by x_i. We denote by $\langle i \rangle$ the encoding of integer i as an $m-bit$ binary string. We assume that $|M| \le (l-m-1)(2^m - 1)$ so that the index of any block can be written as an $m-bit$ string [Bellare M., Guerin R., and Rogaway P. (1995)].

Both to define the schemes and to analyze their security it is helpful to introduce an auxiliary function, which we call $XOR-Tag^n(.,.)$. It takes an oracle for a function $f : \{0,1\}^l \to \{0,1\}^L$. It also takes two inputs. The first is an $l-1$ bit string which we call s, and whose role will emerge later. The second is the message M discussed above. It processes these inputs using f f as indicated below and returns a value we call τ.

Algorithm $XOR-Tag^f(s, M)$ (4.2)

Divide M into $l-m-1$ bit blocks, $M = x_1 ... x_n$

$y_0 \leftarrow f(0 \| s)$ (4.3)

For $i = 1,...,n$ do $y_i \leftarrow f(1 \| \langle i \rangle \| x_i)$

$\tau \leftarrow y_0 \oplus y_1 \oplus ... \oplus y_n$ (4.4)

Return τ

Our auxiliary function applies f at $n+1$ points, each of these points being an $l-bit$ string. The first point is $0\|s$. Namely we prefix the $(l-1)-bit$ string s with a zero bit, which brings the total to $l-bit$, and then apply f to get y_0. The other n points on which f is applied are all prefixed with the bit 1. That is followed by an encoding of the block index and then the data block itself, for a total length of $1+m+(l-m-1)=l$ bits.

We are now ready to describe the scheme. We fix a family of functions $F:\{0,1\}^k \times \{,01\}^l \rightarrow \{0,1\}^L$, and the key for the message authentication scheme is simply a $k-bit$ key K for the family F, which specifies a specific function F_K. The parties will use F_K in the role of f above. There are actually two versions of the scheme. One is deterministic and state, making use of a global counter; the other is stateless and randomized. The difference is only in how s is chosen. We begin with the counter version. Here the sender maintains an integer counter $\langle ctr \rangle$, initially 0. We denote by $\langle ctr \rangle$ its encoding as an $(l-1)-bit$ integer. (The counter thus ranges from 0 to $2^{l-1}-1$. Note that when i is a block index, $\langle i \rangle$ also denotes its binary encoding, but as an m bit string, so that the notation $\langle \cdot \rangle$ is a bit overloaded in that the length of the string returned depends on the context of its argument, but hopefully this will not cause too much confusion.) The counter-based XOR scheme using F, denoted $C-XOR^F=(K,T,V)$, works as follows:

Algorithm $T_K(M)$ (4.5)

$\tau \leftarrow XOR-Tag^{F_K}(\langle ctr \rangle, M)$

$\sigma \leftarrow (\langle ctr \rangle, \tau)$ (4.6)

$ctr \leftarrow ctr+1$ (4.7)

Return σ

Algorithm $V_K(M,\sigma)$ (4.8)

Parse σ as (s,τ)

$\tau' \leftarrow XOR-Tag^{F_K}(s,M)$ (4.9)

If $\tau = \tau'$ then return 1 else return 0

In other words, the tag for message $M = x_1...x_n$ is a pair consisting of the current counter value $\langle ctr \rangle$ encoded in binary, and the subtag τ, where

$$\tau = F_K(0\|\langle ctr\rangle) \oplus F_K(1\|\langle 1\rangle\|x_1) \oplus ... \oplus F_K(1\|\langle n\rangle\|x_n) \quad (4.10)$$

To verify the received tag $\sigma = (\langle ctr\rangle, \tau)$ the verification algorithm recomputes the correct subtag, calling it τ', as a function of the given counter value, and then checks that this subtag matches the one provided in σ. The randomized version of the scheme, namely the randomized XOR scheme using F, is denoted $R-XOR^F = (K,T,V)$. It simply substitutes the counter with a random $(l-1)-bit$ value chosen anew at each application of the tagging algorithm. In more detail, the algorithms work as follows:

Algorithm $T_K(M)$ (4.11)

$r \xleftarrow{R} \{0,1\}^{l-1}$ (4.12)

$\tau \leftarrow XOR-Tag^{F_K}(r,M)$ (4.13)

$\sigma \leftarrow (r,\tau)$ (4.14)

Return σ

Algorithm $V_K(M,\sigma)$ (4.15)

Parse $\sigma\_$ as (r,τ)

$$\tau' \leftarrow XOR - Tag^{F_K}(r, M) \tag{4.16}$$

If $\tau = \tau'$ then return 1 else return 0

In other words, the tag for message $M = x_1 ... x_n$ is a pair consisting of a random value r and the subtag τ where

$$\tau = F_K(0 \| r) \oplus F_K(1 \| \langle 1 \rangle \| x_1) \oplus ... \oplus F_K(1 \| \langle n \rangle \| x_n) \tag{4.17}$$

To verify the received tag $\sigma = (r, \tau)$ the verification algorithm recomposes the correct subtag, calling it τ', as a function of the given value r, and then checks that this subtag matches the one provided in σ.

4.3.2 XOR Security Considerations

Before we consider security it is important to clarify one thing about possible forgeries. Recall that a forgery is a pair M, σ consisting of a message M and a value σ, which purports to be a valid tag for M. In the *XOR* schemes, a tag σ is a pair (s, τ) where s is an $(l-1)-bit$ string and τ is an $L-1-bit$ string. Now, we know that the tagging algorithm itself generates the first component in a very specific way. For concreteness, take the counter-based *XOR* scheme; here s is the value of a counter, and thus for legitimately tagged messages, a value that never repeats from one message to the next. We assume no more than 2^{l-1} messages are authenticated so that the counter does not wrap around. This does not mean that the adversary is forced to use a counter value in the role of s in its attempted forgery M, (s, τ), The adversary is free to try any value of s, and in particular to re-use an already used counter value. Remember that the adversary's goal is to get the verification algorithm to accept the pair M, (s, τ), subject only to the constraint that M was not a query to the tagging oracle. Look at the code of the verification algorithm: it does not in any way reflect knowledge of s as a counter, or try to check any counter-related property of s. In fact the verification algorithm does not maintain a counter at all; it is

stateless. So there is nothing to constrain an adversary to use the value of the counter in its attempted forgery.

A similar situation holds with respect to the randomized version of the *XOR* scheme. Although the legitimate party chooses r at random, so that legitimate tags have random values of r as the first component of their tags, the adversary can attempt a forgery M, (r,τ) in which r is quite non-random; the adversary gets to choose r and can set it to whatever it wants. This freedom on the part of the adversary must be remembered in analyzing the schemes.

We will look at the attack in the context of the counter-based *XOR* scheme. Remember that the attack, specified by adversary above, requested the tags of three related messages and XORed the returned values to get the tag of the third message, exploiting commonality between the values to get cancellations. For the same three messages under the new scheme, let us look at the subtags returned by the tagging oracle. They are:

$$XOR-Tag^{F_K}(\langle 0\rangle, x_1x_2) = F_K(0\|\langle 0\rangle) \oplus F_K(1\|\langle 1\rangle\|x_1) \oplus F_K(1\|\langle 2\rangle\|x_2) \tag{4.18}$$

$$XOR-Tag^{F_K}(\langle 1\rangle, x_1x'_2) = F_K(0\|\langle 1\rangle) \oplus F_K(1\|\langle 1\rangle\|x_1) \oplus F_K(1\|\langle 2\rangle\|x'_2) \tag{4.19}$$

$$XOR-Tag^{F_K}(\langle 2\rangle, x'_1x_2) = F_K(0\|\langle 2\rangle) \oplus F_K(1\|\langle 1\rangle\|x'_1) \oplus F_K(1\|\langle 2\rangle\|x_2) \tag{4.20}$$

Summing these three values yields a mess, something that does not look like the subtag of any message, because the values corresponding to the counter do not cancel. So this attack does not work.

It seems hard to come up with one, but that does not mean much; maybe the attack is quite clever. This is the point where the kind of approach we have been developing, namely provable security, can be instrumental. We will see that the *XOR* schemes can be proven secure under the assumption that the family F is a PRF. This means that simple attacks like the above do not exist. And our confidence in this stems from

much more than an inability to find the attacks; it stems from our confidence that the underlying family F is itself secure.

4.3.3 Results on the security of the XOR schemes

We state the theorems that summarize the security of the schemes, beginning with the counter-based scheme. We call the (integer) parameter m in the scheme the block-indexing parameter in the following. We also let Plaintexts (l,m) denote the set of all strings M such that the length of M is $n\cdot(l-m-1)$ for some integer n in the range $1\le n\le 2^m-1$; this is the message space for the XOR message authentication schemes.

The theorem below has (what should by now be) a familiar format. It upper bounds the insecurity of the counter-based XOR message authentication scheme in terms of the insecurity of the underlying PRF F. In other words, it upper bounds the maximum (over all strategies an adversary might try) of the probability that the adversary can break the XOR scheme (namely, successfully forge a correct tag for an as yet unauthenticated message), and the upper bound is in terms of the (assumed known) maximum ability to break the PRF F that underlies the scheme. It is another example of the kind of "punch line" we strive towards: a guarantee that there is simply no attack against a scheme, no matter how clever, as long as we know that the underlying tool is good. In particular we are assured that attacks like those we have seen above on our example schemes will not work against this scheme.

Also as usual, the bounds are quantitative, so that we can use them to assess the amount of security we will get when using some specific PRF (say a block cipher) in the role of F. The bounds are rather good: we see that the chance of breaking the message authentication scheme is hardly more than that of breaking the PRF.

Suppose $F:\{0,1\}^k\times\{,01\}^l\to\{0,1\}^L$ is a PRF, and let $C-XOR^F=(K,T,V)$ be the corresponding counter-based XOR message authentication scheme as described

above, with block-indexing parameter $m \le l-2$ and message space Plaintexts (l,m). Then for any t, q, μ with $q \le 2^{l-1}$ we have

$$Adv^{uf-cma}_{C-XOR^F}(t,q,\mu) \le Adv^{prf}_{F}(t',q') + 2^{-L}, \tag{4.21}$$

where $t' = t + O(\mu)$ and $q' = q + 1 + \mu/(l-m-1)$. (4.22)

The result for the randomized version of the scheme is similar except for accruing an extra term. This time, there is a "collision probability" type term of $q^2/2^l$ in the upper bound, indicating that we are unable to rule out a breaking probability of this magnitude regardless of the quality of the PRF. We will see later that this is an inherent feature of the scheme, which is subject to a sort of birthday attack.

Suppose $F:\{0,1\}^k \times \{,01\}^l \to \{0,1\}^L$ is a PRF, and let $R-XOR^F = (K,T,V)$ be the corresponding randomized XOR message authentication scheme as described above, with block-indexing parameter $m \le l-2$ and message space Plaintexts (l,m). Then for any t, q, μ

$$Adv^{uf-cma}_{C-XOR^F}(t,q,\mu) \le Adv^{prf}_{F}(t',q') + \frac{q^2}{2^l} + 2^{-L} \tag{4.23}$$

where $t' = t + O(\mu)$ and $q' = q + 1 + \mu/(l-m-1)$. (4.24)

These results are stated without further proof.

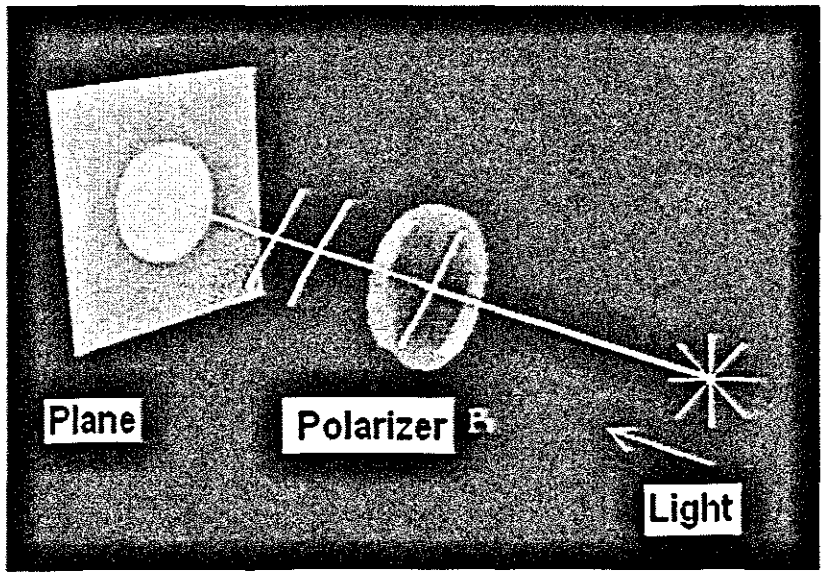
4.4 The Polarization of Light as an Encoding Method

Etienne Louis Malus Law (1775 - 1812): The power of light, which is linear polarization light, is variant I, after it passes the polarizer. The power of transmission

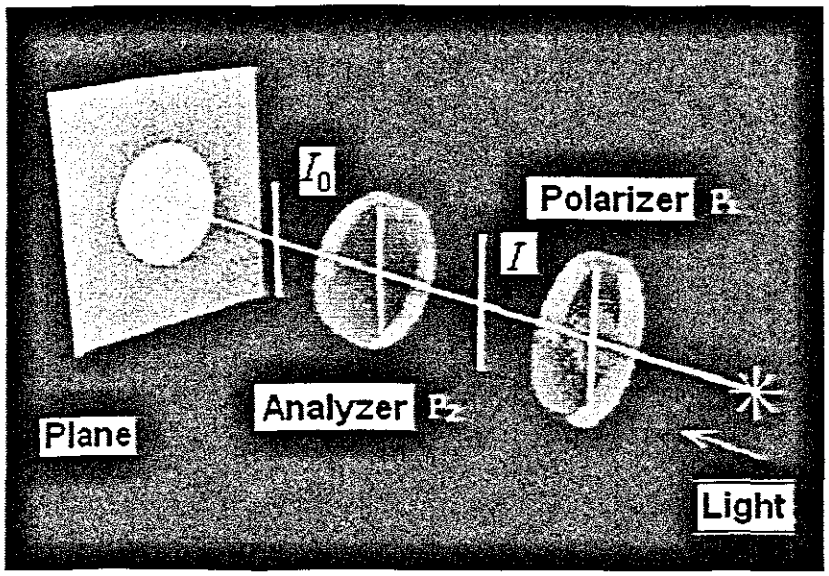
light is $I = I_0 \cos^2 \alpha$. Where α is angle of between the direction of linear polarization light and the direction of the polarizer.

When the angle α between p_1 and p_2 is equal to 0^0 or 180^0. I is maximize: $I = I_0$.

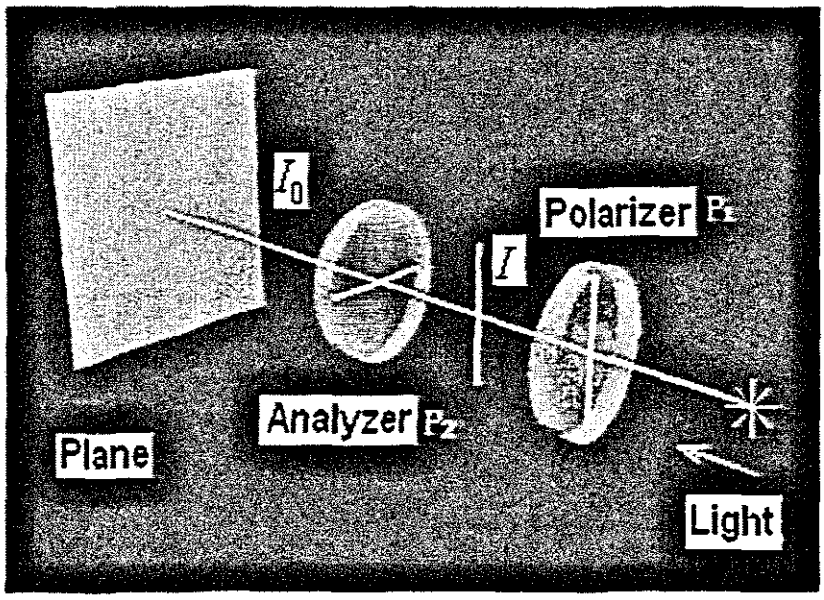
When the angle α between p_1 and p_2 is equal to 90^0 or 270^0. I is minimize: $I = 0$.



(a)



(b)



(c)

Figure 4-1 Etienne Louis Malus Law

p_1 Polarizer, p_2 Analyzer

It is interesting to note that the cathoderay tube (CRT) which is using as one of the most common display was born when Braun invented an electron tube (a cathoderay tube, Braun tube) in 1887. LCD was originated as Reinitzer an Austrian discovered Liquid Crystal in 1888 one year after the CRT. In deed, CRT and LCD were invented at the almost same time. Shwenberg in England invented a vacuum tube TV with CRT in 1934, while on the other hand LCD's was not been applied to displays until RCA a US company used it in 1968.Even though it took a pretty long time until (liquid crystal) was adapted as a display device, STN-LCD and small TFT-LCD was developed to the practical use level in 1986 while an electronic calculator and an electronic watch were fitted for display. LCD became a representative display device of note in the 1990's when 10.0" TFT-LCD was initiated as mass-production, Nowadays, LCD's are widely applied to Computer monitors, Digital TV's, Mobile phones, etc. as a substitution of CRT.

Liquid crystal molecules have an especial feature. Putting the liquid crystal molecules between the two polarizers, when no electric field is applied, the plane of polarization for linearly polarized light is rotated through 90 degrees by the liquid crystal molecules. Thus, the light can be transmitted through the analyzer. However, under an electric field, all the liquid crystal molecules align in the direction of the applied field. The light will not be rotated, and cannot pass through the analyzer.

When you put the twisted liquid crystal molecules between the polarizations and when no electric field is applied, the linearly polarized light is rotated 180 deg by the twisted liquid crystal molecules. Thus, no light will be transmitted through the analyzer. However, under an electric field, the twist and tilt of the molecules are altered, and the liquid crystal molecules attempt to align parallel with the applied field. This results in partial transmission of light through the analyzer. As the electric field increases further, all the liquid crystal molecules align in the direction of the applied field. This property of the LCD can be used to implement XOR Boolean logic. Great progresses have been made in the logic operations using polarization encoding of the LCD.

4.5 How to Encrypt Image Based on Optical XOR And LCD Technology

With the facts mentioned above, it is seen that the XOR as a logic algorithm, which applies for binary operation can be investigated for use with LCD's. In our system, LCD is important part to perform optical XOR encryption. A liquid crystal display (LCD) is controlled by voltage, which can display binary images. Passive display technology like LCD does not emit light nor any radiation. Instead, they use the ambient light in the environment. By manipulating this light, displaying image occur using very litter power.

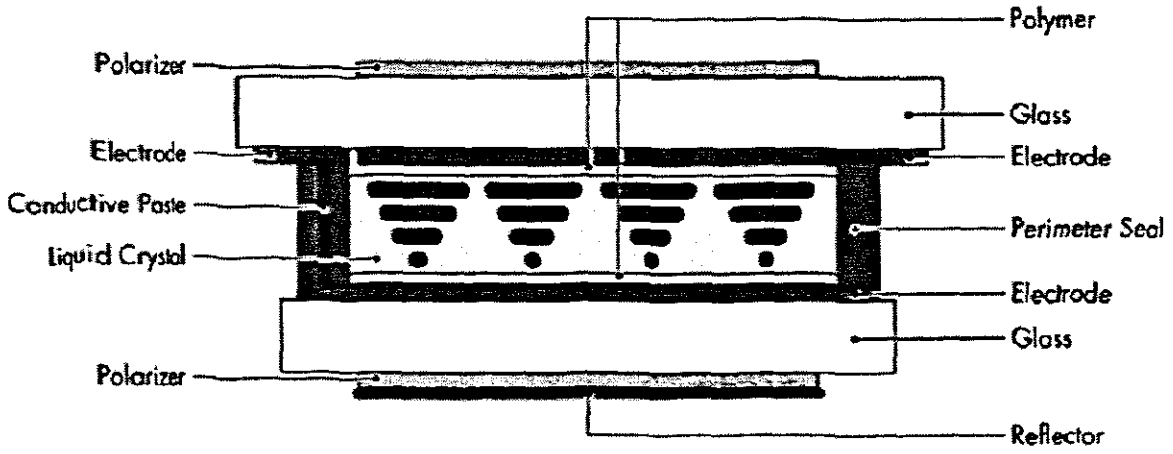


Figure 4-2 The Structure of LCD

The structures of LCD consist of two pieces of glass with transparent electrodes printed on the internal surfaces. [Sheng Zhen Jing Hua Displays (2001)] An alignment layer on each glass surface is used to twist the liquid crystal material in a helical or "twisted" pattern. Polarizers are used on the outside front and rear surface. When the LCD is "off", no voltage is applied to the electrodes, and light passes through the LCD. When it's "on", voltage is applied and the LC molecules align themselves in the direction of the electric field. This causes the light to be out of phase with the polarizers and to be blocked, creating a dark area on the LCD. By selectively applying voltage to the electrodes, a variety of patterns can be achieved. Many advance in TN LCD's have been produced. This processing is as shown in Figure 4-2.

Field Effect of LCD

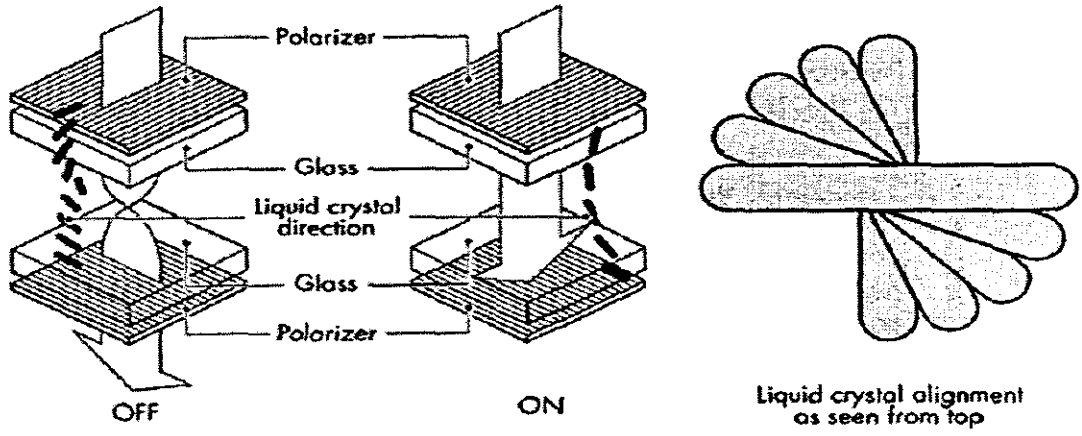


Figure 4-3 Cross-Sectional View of A Typical Display

In our encryption method, we used two LCD performing Optical Encryption Image based on XOR. The applied electric field that decides the rotation of the input polarization is controlled by a gray-level or binary value represented on the LCD. The gray-level or binary value, which represents a logic state, can be determined by measuring the rotation angle of the polarization. In this paper, the value that changes the direction of input polarization by 90 deg is define as logic 1 and one that does not rotate the input polarization is defined as logic 0.

An XOR function using two LCD's is illustrated in the Figure 3-5. The original polarizer of two LCD's should be removed. The orientation of the polarizer is perpendicular to that of the analyzer and input light has vertical polarizer. The analyzer, which is placed after LCD 2, converts the polarization information to intensity information LCD's labeled 1, which rotate the polarization state, are equivalent to logic 1 and LCD's labeled 0, which do not rotate the polarization state, indicate logic 0. If the state of the LCD is logic 1, the output polarization of the LCD becomes the horizontal due to the twisted liquid crystal molecules. And if the state of the LCD is logic 0, the output polarization of the LCD remains the same, the result is 0 (no light) when same logic states are represented on two LCD's and is 1 (light) when different logic states are represented on two LCD's. As a result, a combination

of two LCD's is capable of performing XOR logical operations, as shown in Figure 3-5.

The optical XOR operations between the key bit stream and the binary bit of image are performed by the polarization encoding method. The input image is converted to binary bit and all bits are represented on LCD1. The random key bit stream generated by digital algorithms is represented on LCD2. So we can perform XOR between the binary bit of image and random key bit on the LCD's. Finally a CCD captures the result. I have done simulation of XOR encryption in Matlab (in Chapter 6).

4.6 CCD Detector In The XOR Scheme

The CCD or Charge Couple Device is used for the detector in my system. [Baier S. (1993)] With developing optical and imaging techniques, researchers invented a lot of opto-electronic products. CCD (Charge Couple Device) is one of these revolutionary products. Technology advances with CCD, like increased resolution at lower manufacturing costs, have fueled the growth in the electronic imaging industry. However, some of the typical constraints of CCD remain unchanged, such as the very low output signal level and the inherent noise sources. Furthermore, increased resolution generally equals higher read-out speed, which in turn dictates the requirements for the subsequent electronics. The CCD is the central element in an imaging system. Designers need to be aware of the special requirements for the signal conditioning of the CCD in order to achieve the maximum performance. The output signal of the CCD is a constant stream of the individual pixel "charges" and this results in the typical form of stepped DC voltage levels. This output signal also contains a DC-bias voltage, which is in the order of several volts. The signal is then passed through a capacitor to block the DC voltage before going into the preamplifier. To maintain the necessary relationship between the pixel information and the baseline, a clamp or DC-restore circuit is usually situated in the first processing stage. The next stage is used as a noise reduction circuit specific to CCD based systems: the correlated double sampler (CDS). Following is another gain stage, which could be an

automatic gain control amplifier (AGC), or a fixed gain stage with offset adjustment. Before going into the A/D converter it usually passes through a dedicated buffer or driver circuit optimized for the selected converter type. Further baseline stabilization can be achieved by having a D/A converter in a digital control loop. So the CCD is good at transforming the light signal to the electric signal. Through this way, the CCD can act as a substitute for the eyes to detect the light signa. In the following discussion the CCD itself is looked at and design techniques are explored.

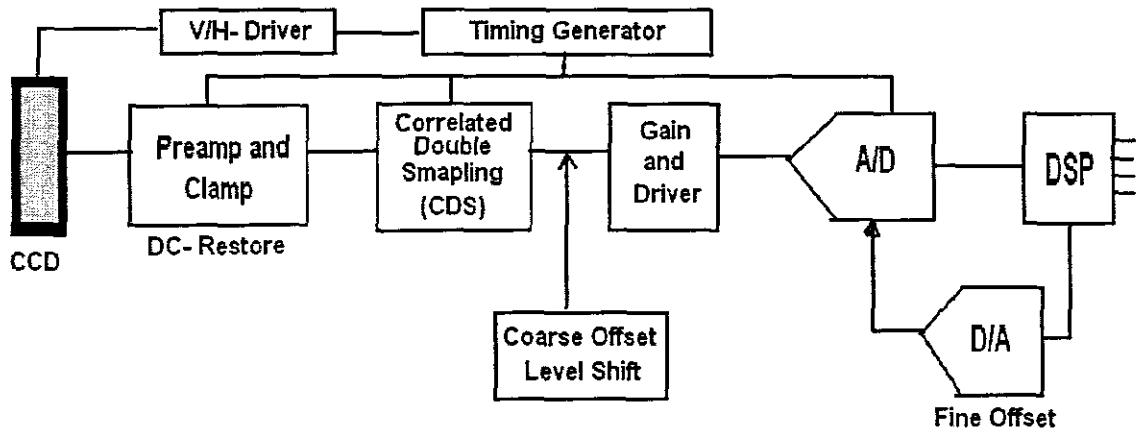


Figure 4-4 CCD Imaging Systems

4.6.1 Basic CCD Theory

In its principle, the operation of a CCD array is quiet simple. A common analogy is shown here (Figure 4-5), using an array of buckets on conveyor belts. During a rain shower the raindrops will fill the lined up buckets more or less. Then the conveyor belts transport the buckets to the front belt and dump their content into another row of buckets. As they move forward the rainwater is spilled into the metering glass. The scale on the metering glass indicates how much water was collected in the individual bucket. When relating this model to a real CCD element, the "raindrops" are the light (photons) falling onto the CCD surface, the buckets are the many pixels of a CCD array and the "conveyor belts" are the shift registers that transport the pixel charge to

the output stage. This output stage is mainly the sense capacitor, here the "metering glass", and an output source follower is used to buffer this sense capacitor.

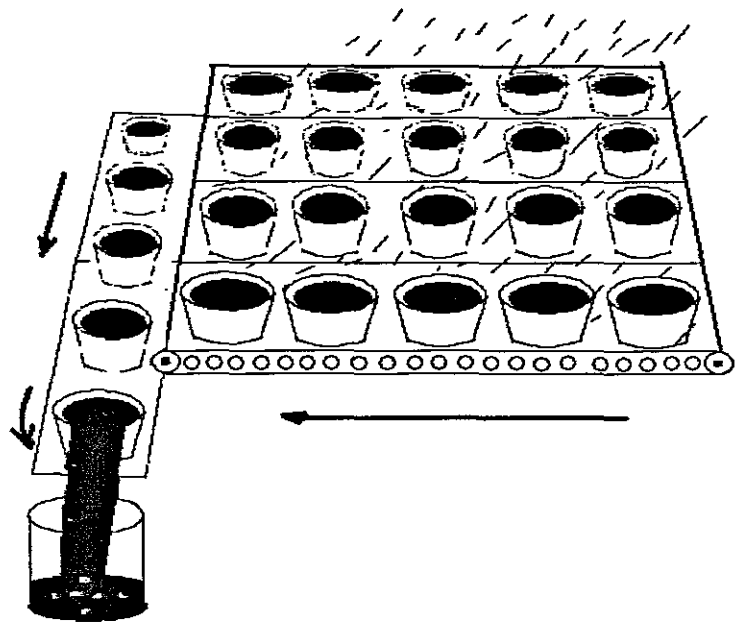


Figure 4-5 Basic CCD Theories

Raindrops=Photons Conveyor Belts=CCD Shift Register Buckets=Pixels Metering Glass=Sense Capacitor

The CCD array is configured into multiple vertical shift registers and usually one horizontal shift register, both requiring different clock patterns. The flow is as follows: the pixel converts the light (incoming photons) into electrons, which are stored as electrical charge. Then the charge is transferred down the vertical register in a conveyor-belt fashion to the horizontal shift register. This register collects one line at a time and transports the pixel charges in a serial manner to the on-chip output stage. The on-chip output converts the charge into a voltage. This voltage is then available at the output in the typical CCD pulse form.

With the standard CCD, most of the pixels can detect the light. The CCD also has small sections at the beginning and at the end of each vertical segment that are covered and therefore "optically black". Those pixels will always have the voltage

level representing black. Some image circuits use those as reference pixels to adjust the signal offset. As is shown in Figure 4-5.

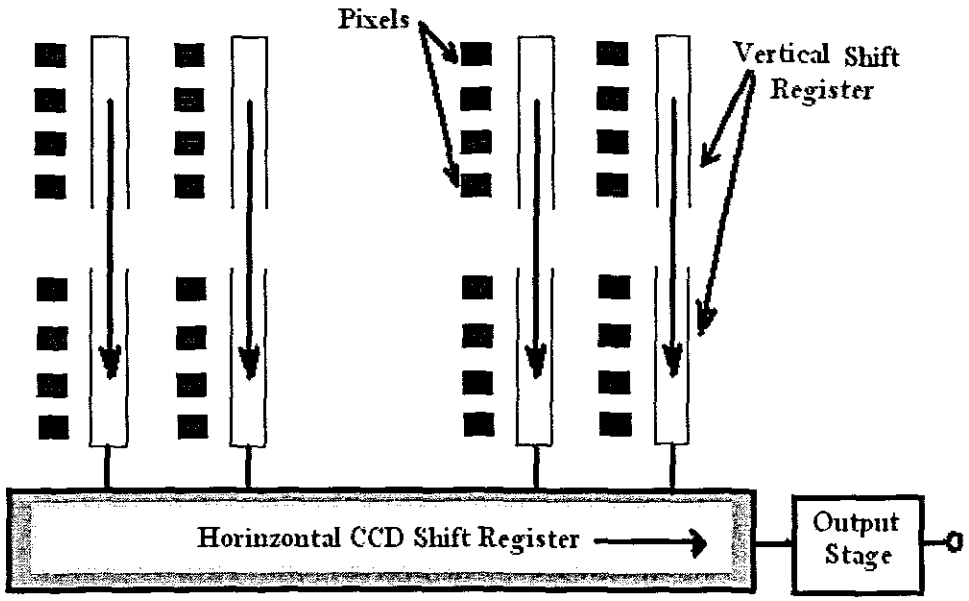


Figure 4-6 The CCD Array Configuration

The horizontal read-out speed for systems with up to 12-bit resolution is up to 10MHz. For higher resolutions (£16-bit) the clock speed is around 1MHz. Typical pixel dimensions are: 27mm2 for a 512x512 array or 12mm2 for a 1024x1024 array.

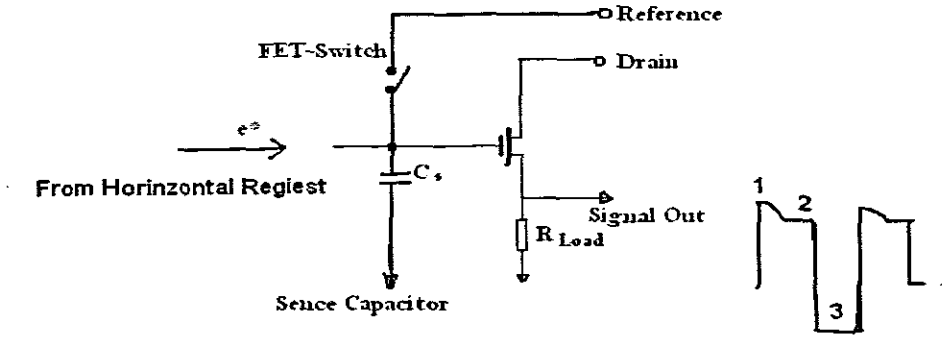


Figure 4-7 Built-In CCD Output Stage—Charge Detection

Shown here (Figure 4-6) is the conceptual schematic of the output stage inside the CCD element. This stage is responsible for the so called 'charge detection'. As discussed earlier, the charge e* generated is moved into the horizontal shift register. The charge of each individual pixel is controlled by the horizontal clock and stored onto the Sense Capacitor (CS). A typical value for such a capacitor is 0.1pF to 0.5pF. According to V = Q/C, the charge will develop a voltage across capacitor CS, representing the light intensity for the particular pixel. The CCD MOSFET transistor configured as a source follower buffers the capacitor from the output node, which connects to the load resistor, R_{Load}. At this point, the image (video) signal becomes available at VOUT for further signal processing.

As indicated in the figure above, the output voltage is a series of stepped DC voltages.

One pixel period is composed of three different levels: (1) the "reset feedthrough", (2) the "reference level", (3) and the "pixel level". A readout sequence begins with the reset. Where the FET-switch is closed, set the sense capacitor to the initial reference voltage. The reference voltage can be relatively high, up to +12V. The closing of the switch causes the reset feedthrough, a result of capacitive coupling through the MOSFET. After the decay of this feedthrough the capacitor will reflect the reference voltage level. Once the capacitor has been reset, the switch opens and pixel charge is transferred to the capacitor, altering its voltage.

An important specification for CCD elements is the sensitivity. This is a measure of the achievable output voltage per electron, $SV = VOUT/e^*$. With a 0.1pF capacitor, the output voltage would be -1.6mV per electron. Unfortunately, the source follower has a gain of less than 1 (~0.8).

The typical output voltage waveform from a CCD element can be described by five characteristics: the Reset Feedthrough, the Reset Level, the Signal Amplitude, the Pixel Period and the actual Pixel Width. As mentioned before, this CCD signal is not

a continuous sinusoidal waveform, but rather is a sequence of stepped DC levels. The sequence for one pixel is as follows:

Reset Feedthrough: This can be a relatively large pulse, as a result of capacitive coupling through the FET.

Reset Level: The "Sense Capacitor" will be charged to this final reset voltage. This level can be in the order of +10V or more, creating the requirement for a DC-decoupling capacitor at the output of the CCD element.

Pixel Level: After the reset period, the pixel is transferred. The amplitude corresponds to the charge representing the incident light level of the addressed pixel. Because of the electron charge (e^*) the CCD output signal is inherently unipolar (negative).

4.7 Conclusion

The XOR algorithm has been thoroughly analyzed and security analysis have been performed in this chapter to proof the usefulness of the method. The result of the XOR security investigation showed that the bounds are quantitative, so that we can use them to assess the amount of security we will get when using some specific PRF in the role of F input image functions. In this chapter, I also described the structure and working principles of LCD and CCD technology. They are todays key opto-electronic devices. This technology forms the basis of my project. In this encryption method, the LCD acts an important part to complete the XOR algorithm computing. The CCD is used to detect light signal. The method by which the LCD performed XOR encryption, is polarization encoding.

CHAPTER 5

MOBILE BROADBAND LINK OVER WIRELESS IP

5.1 The Development History of Internet Technology

Now the Internet has exploded in popularity on a worldwide scale. The Internet was conceived in the 1960s as a tool to link university and government research centers via a nationwide network that would allow a wide variety of computers to exchange information and share resources. The engineering challenges were manifold and complex, beginning with the design of a packet switching network-a system that could make computers communicate with each other without the need for a traditional central system. Other challenges included the design of the machines, data exchange protocols, and software to run it. What eventually grew out of this endeavor is a miraculous low-cost technology that is swiftly and dramatically changing the world. It is available to people at home, in schools and universities, and in public libraries.

The Internet is not owned or controlled by any company, corporation, or nation. It connects people in 65 countries instantaneously through computers, fiber optics, satellites, and phone lines. It is changing cultural patterns, business practices, the consumer industry, and research and educational pursuits. It helps people keep up to date on world events, find a restaurant in Oregon or a cheap flight to Paris, play games, and discuss everything from apples to zoology. It has marshaled support for human rights in suppressed nations, saved the life of a child in Beijing.

On the heels of Sputnik and the onset of the Cold War, President Eisenhower thought it wise to create the Advanced Research Projects Agency (ARPA) in 1958, to keep the United States at the forefront of technology. ARPA would soon begin the

researches that eventually lead to the Internet. However, before ARPA began supporting networking research seriously, Leonard Kleinrock had already invented the technology of the Internet in 1962 while an MIT graduate student. The packet switching technology he proposed was a dramatic improvement over the circuit-switched telephone network in which the entire path connecting a voice call between two parties was dedicated only to their conversation, even when they were silent. Typically, silence occupies about one-third of speech patterns, but in the transmission of data, silence can occupy as much as 99.9 percent of the data stream. Packet switching avoids this inefficiency by chopping messages into packets, and sending these packets of data independently through the network as if they are electronic letters passing through an electronic post office.

Communication links are assigned to packets, providing a highly efficient technology for sending packets over links (fiber, copper, radio) from one network node to another. These nodes are routers (or switches), which collectively share the job of directing the packets from node to node on the way to their destinations. The selection of the routes, the management of the packet flow, and the general rules for running the network are governed by protocols that are typically implemented in software and hardware. [Agi I. and Gong L. (1996)]

In 1963, JCR Licklider, head of the computer research effort at ARPA, articulated a vision of a network that would connect machines and people worldwide. In the mid-1960s, ARPA determined that it needed a network to connect together the research computers and programs it funded. Larry Roberts was brought to ARPA in 1966 to manage the program to create the packet-switched ARPAnet. This network was to form the foundation of the Internet.

A contract was let to Bolt, Beranek, and Newman in Cambridge, Mass., in January 1969 to implement the design. Supervised by Frank Heart, they designed small machines called Interface Message Processors (IMPs), specifically for packet-switching technology. A new communications protocol for packet switching was

needed and they came up with the Network Control Protocol (NCP). The new network was ready for unveiling at UCLA in September 1969.

Universities and research organizations were among the first to join the network in order to exchange information. Electronic mail was introduced in 1972 by Ray Tomlinson. NCP was phased out by a new communications protocol technology -- Transmission Control Protocol/Internet Protocol (TCP/IP), which was created by Bob Kahn and Vint Cerf in 1973. It was accepted by the U.S. government in 1978, and became the de facto networking standard in 1983. More networks began to pop up in the 1980s. Educational and commercial organizations that fell outside the original charter wanted to use the same packet-switching technologies, and the system came to be known as the Internet during this period. It had far exceeded its original purpose, and was providing the impetus for a vast technological revolution that was just ahead.

Major innovations in software were necessary before the Internet could function as a global information utility. In 1989 Tim Berners-Lee, a scientist at the European Laboratory for Particle Physics in Geneva, proposed the World Wide Web project, and a new language for linked computers known as HTML (Hyper-Text Markup Language). Simple tools to retrieve information from the Web and communicate would be the focus of much activity in the next few years. [Qiao L. and Nahrstedt K. (1998)] In 1991, the University of Minnesota developed "Gopher," the first successful Internet document retrieval system. In the spring of 1993, a group of graduate students at the University of Illinois computer laboratories created a "browser" program called Mosaic, and distributed it free. Netscape and then Microsoft followed with browsers that greatly simplified a computer user's ability to search the Internet in search of information.

Today people can search thousands of databases and libraries worldwide in several languages, browse through hundreds of millions of documents, journals, books, and computer programs, and keep up to the minute with wire-service news, sports, and

weather reports. An increasing number of people shop, bank, and pay bills on the Internet. Many invest in stocks and commodities online. It's a powerful symbol of society's expectations about the future fast-moving technology that adds convenience and efficiency to their lives.

Beyond convenience, as people consider the philosophical ramifications of the Internet, some view it as a tool of unity and democratization. In the 1960s, long before the Internet, futurist and author Sir Arthur C. Clarke predicted that by 2000 a vast electronic "global library" would be developed. Recently, a judge cited it as "the single most important advancement to freedom of speech since the writing of the Declaration of Independence." Marshall McLuhan coined the phrase "the global village" when he spoke of how radio and television had transformed the world in the course of the 20th century. In the 21st century, it seems the Internet is destined to have even more profound effects.

5.2 IP Wireless Technique

For 25 years we have been analyzing the global wireless industry. An evolution is occurring in wireless and portable computing: Wireless Internet. Wireless Internet & Mobile Computing is known for its global perspective. We understand the dynamics of international wireless data efforts. Whether it's in Europe, Asia or Latin America, we've studied the products, spoken to the wireless data leaders and presented corporate policy sessions.

Wireless Internet is already here. The way many people think of wireless Internet. That is, people want speeds that are close to that of a dial-up telephone line at a reasonable price. The cellular industry is trying to do that for wide area wireless networks, and might succeed this year. The airtime (packet) pricing could be a big issue, though. That is, don't count on a low-cost, flat rate price for higher-speed cellular data services. Data rates for the new generation of cellular systems now

average about 20K - 30K bps for GSM GPRS networks and 40K - 60K bps for CDMA RTT networks.

The wireless LAN technology 802.11, also called WiFi, offers much high-speed wireless Internet access for local area environments. WiFi use is exploding in offices and in and homes of the technologically savvy). [Rappaport T. S. (1995)]

Now, the cellular operators, in the United States and abroad, are getting into the 802.11 businesses. This year will be a very interesting, and confusing, time for corporate users and consumers as higher-speed cellular networks and high-speed WiFi networks are implemented. These networks will offer true value for existing and new services. Accessing e-mail and corporate files is getting much easier. Wireless offers more fun, such as enabling streaming of audio and video (with WiFi, not cellular, being much more appropriate for streaming video). 802.11b offers speeds with a theoretically maximum rate of 11M bps at in the 2.4 GHz spectrum band. 802.11a offers speeds with a theoretically maximum rate of 54M bps in the 5 GHz band; 802.11a hardware is just beginning to be introduced. 802.11g is a new standard for data rates of up to a theoretical maximum of 22M bps at 2.4 GHz. As with other wireless technologies, you won't get the theoretically maximum speeds. The cellular wireless data industry is notorious for hyping the theoretical data rates, rather than the typical data speeds. 802.11 typically provides real speeds of at least 500K bps, and often much faster. Pricing models for 802.11 networks as well as 2.5G/3G cellular networks are in a state of flux, and remain that way for months. There are many variables. For example, pricing can be per-minute, per-packet, per location, per day, per month.

It is no doubt that 802.11 are changing the telecommunications environment in corporations, the home and in public areas. It's occurring right now, and the advantages are significant. As a consulting firm, we are making sure to explore and leverage the power of 802.11 technologies.

Table 5-1. Current Wireless LAN Technique Comparing

Item / Specification	Wireless LAN 802.11	HOME RF 1.09	BLUETOOTH
Application	High speed wireless data networking (long distance)	Wireless communication in home & SOHO	Wireless communication in short range
Technology	FHSS, DSSS	FHSS	FHSS
Frequency	RF 2.4GHz	RF 2.4GHz	RF 2.4GHz
Power	+18dbm	+18dbm	+18dbm
Data rate	11Mbps	11Mbps	1Mbps
Distance	150M	50M	10M
Transmission	DSSS: Data FHSS: Data & Voice	Data & Voice	Data & Voice
Specification	IEEE	Home RF group	Bluetooth SIG
Interface	USB, ISA, PCI, PCMCIA	N/A	Module
Main structure	MAC, RF, Baseband	MAC, RF, Baseband	RF, Baseband, HCI, Ling manager
Power Consumption	250mA	100mA	40mA
Cost	High	Middle	Low

5.3 The IEEE 802.11 Protocol

IEEE 802.11 refers to a family of specifications developed by the IEEE for wireless LAN technology. 802.11 specify an over-the-air interface between a wireless client and a base station or between two wireless clients. The IEEE accepted the specification in 1997.

There are several specifications in the 802.11 families:

802.11 -- applies to wireless LANs and provides 1 or 2 Mbps transmission in the 2.4 GHz band using either frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS).

802.11a -- an extension to 802.11 that applies to wireless LANs and provides up to 54 Mbps in the 5GHz band. 802.11a uses an orthogonal frequency division multiplexing encoding scheme rather than FHSS or DSSS.

802.11b (also referred to as 802.11 High Rate or Wi-Fi) -- an extension to 802.11 that applies to wireless LANS and provides 11 Mbps transmission (with a fallback to 5.5, 2 and 1 Mbps) in the 2.4 GHz band. 802.11b uses only DSSS. 802.11b was 1999 ratification to the original 802.11 standard, allowing wireless functionality comparable to Ethernet.

802.11g -- applies to wireless LANs and provides 20+ Mbps in the 2.4 GHz band.

5.3.1 Architecture Components

An 802.11 wireless LAN is based on a cellular architecture where the system is subdivided into cells, where each cell (called Basic Service Set or BSS, in the 802.11 nomenclature) is controlled by a Base Station (called Access Point, or in short AP).

Even though that a wireless LAN may be formed by a single cell, with a single Access Point, (and as will be described later, it can also work without an Access

Point), most installations will be formed by several cells, where the Access Points are connected through some kind of backbone (called Distribution System or DS), typically Ethernet, and in some cares wireless itself.

The whole interconnected Wireless LAN including the different cells, their respective Access Points and the Distribution System, is seen to the upper layers of the OSI model, as a single 802 network, and is called in the Standard as Extended Service Set (ESS). [IEEE 802.11 (1997)]

The following picture shows a typical 802.11 LAN, with the components described previously:

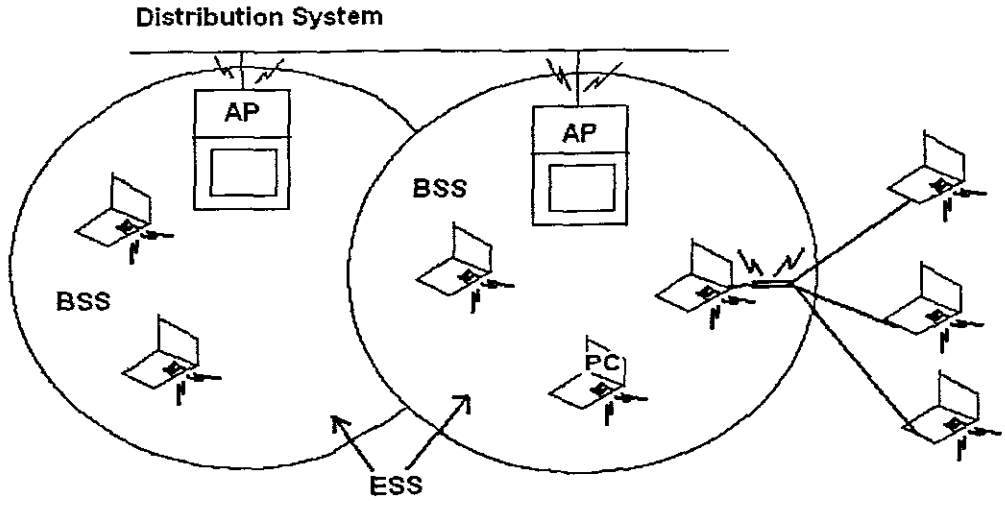


Figure 5-1 A Typical 802.11 Wireless LAN

5.3.2 The IEEE 802.11 Layers Description

As any 802.x protocol, the 802.11 protocol covers the MAC and physical Layer, the Standard currently defines a single MAC, which interacts with three physical Layer (all of them running at 1 and 2 Mbit/s):

- Frequency Hopping Spread Spectrum in the 2.4GHz Band

- Direct Sequence Spread Spectrum in the 2.4GHz Band
- InfraRed

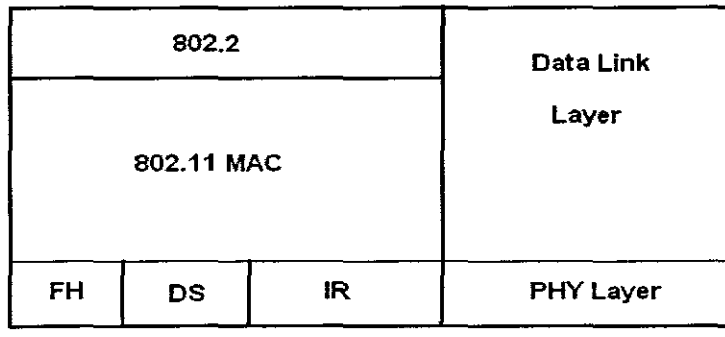


Figure 5-2 The IEEE 802.11 Layers Description

Beyond the standard functionality usually performed by MAC Layers, the 802.11 MAC performs other functions that are typically related to upper layer protocols, such as Fragmentation, Packet Retransmissions, and Acknowledges.

The MAC Layer defines two different access methods, the Distributed Coordination Function and the Point Coordination Function.

5.4 OFDM Modulation

The modulation used in the IEEE 802.11 standard is an Orthogonal Frequency Division Multiplexing (OFDM). The IEEE 802.11 standard specifies an OFDM physical layer (PHY) that splits an information signal across 52 separate sub-carriers to provide transmission of data at a rate of 6, 9, 12, 18, 24, 36, 48, or 54 Mbps. The 6, 12, and 24 Mbps data rates are mandatory. Four of the sub-carriers are pilot sub-carriers that the system uses as a reference to disregard frequency or phase shifts of the signal during transmission.

In the 802.11g standard, a pseudo binary sequence is sent through the pilot sub-channels to prevent the generation of spectral lines. In the 802.11g, the remaining 48 sub-carriers provide separate wireless pathways for sending the information in a

parallel fashion. The resulting sub-carrier frequency spacing is 0.3125 MHz (for a 20 MHz with 64 possible sub-carrier frequency slots). [Meng-Han Hsieh and Che-Ho Wei (September 1999)]

Orthogonal Frequency Division Multiplexing (OFDM) is a multi-carrier transmission technique, which divides the available spectrum into many carriers, each one being modulated by a low rate data stream. OFDM is similar to FDMA in that the multiple user access is achieved by subdividing the available bandwidth into multiple channels, which are then allocated to users. In Frequency Division Multiple Access (FDMA), the available bandwidth is subdivided into a number of narrower band channels. Each user is allocated a unique frequency band in which to transmit and receive on. During a call, no other user can use the same frequency band. Each user is allocated a forward link channel (from the base station to the mobile terminal) and a reverse channel (back to the base station), each being a single way link. The transmitted signal on each of the channels is continuous allowing analog transmissions. [Nee R. and Ramjee Prasad, (2000)] The bandwidths of FDMA channels are generally low (30kHz) as each channel only supports one user. FDMA is used as the primary breakup of large allocated frequency bands and is used as part of most multi-channel systems. However, OFDM uses the spectrum much more efficiently by spacing the channels much closer together. This is achieved by making all the carriers orthogonal to one another, preventing interference between the closely spaced carriers.

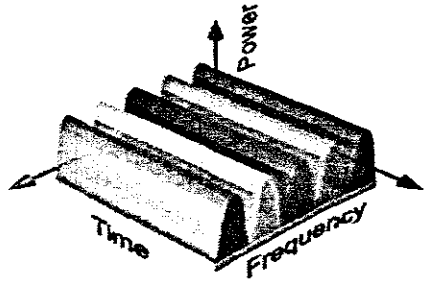


Figure 5-3 FDMA showing that the each narrow band channel is allocated to a single user

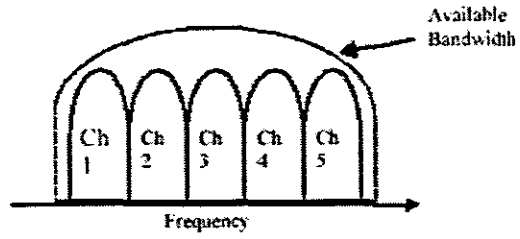


Figure 5-4 FDMA spectrums, where the available bandwidth is subdivided

OFDM overcomes most of the problems with both FDMA and TDMA. OFDM splits the available bandwidth into many narrow band channels (typically 100-8000). The carriers for each channel are made orthogonal to one another, allowing them to be spaced very close together, with no overhead as in the FDMA example. Because of this there is no great need for users to be time multiplex as in TDMA, thus there is no overhead associated with switching between users.

Table 5-2 Key Parameters of the OFDM Standards

Data Rate	*6, 9, 12, 18, 24, 36, 48, 54 Mbps*
Modulation	BPSK, QPSK, 16-QAM, 64-QAM
Coding Rates	1/2, 9/16, 2/3, 3/4
Number of Subcarriers	52
Number of Pilot Tones	4
OFDM Symbol Duration	4 μ sec
Guard interval	800 μ sec, 400 μ sec (optional)
Subcarrier Spacing	312.5 kHz
Signal Bandwidth	16.66 MHz
Channel Spacing	20 MHz

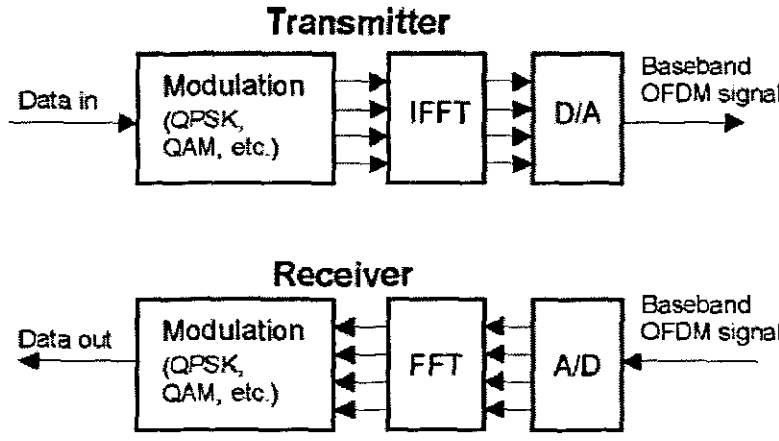


Figure 5-5 Basic FFT, OFDM transmitter and receiver

5.4.1 Quaternary Phase Shift Keying (QPSK)

Up to now all of our bandpass modulation schemes have been binary with one sinusoid for 1 and another with a different amplitude, frequency or phase for 0. The objective of this and the next Investigation is to introduce Quaternary Phase Shift Keying (QPSK) in which we use four sinusoids of the same amplitude and frequency but different phases. We will show, in particular, how to express QPSK signals in terms of orthonormal basis functions and how this facilitates their generation and detection. [FELZER A.P., (2004)]

1. As we said in the introduction Quaternary Phase Shift Keying (QPSK) uses four sinusoids of the same amplitude and frequency but with different phases. We refer to these sinusoids as symbols.

a. How many bits of information are contained in a QPSK symbol.

b. How does the amount of information transmitted by a QPSK system in a given time compare with that of a binary system like BASK, BFSK and BPSK.

c. We refer to the rate at which symbols are being transmitted as the baud rate. Memorize this term and then find the bit rate of a QPSK signal with a baud rate of 1000 symbols/sec.

2. Given that the sinusoids of QPSK signals are as follows

$$S_i(t) = \sqrt{\frac{2E}{T}}\cos[2\pi f_c t + (2i-1)\frac{\pi}{4}] \qquad 0 \le t \le T \tag{5.1}$$

$$S_i(t) = S_{i1}\sqrt{\frac{2}{T}}\cos(2\pi f_c t) + S_{i2}\sqrt{\frac{2}{T}}\sin(2\pi f_c t) \tag{5.2}$$

$$\text{with } S_{i1} = \sqrt{E}\cos[(2i-1)\frac{\pi}{4}] \text{ and } S_{i2} = \sqrt{E}\sin[-(2i-1)\frac{\pi}{4}] \tag{5.3}$$

$$S_i(t) = \sqrt{\frac{2E}{T}}\cos[2\pi f_c t + (2i-1)\frac{\pi}{4}] = S_{i1}\phi_1(t) + S_{i2}\phi_2(t) \tag{5.4}$$

$$\text{where } \phi_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t) \text{ and } \phi_2(t) = \sqrt{\frac{2}{T}}\sin(2\pi f_c t). \tag{5.5}$$

a. Show that $\phi_1(t)$ and $\phi_2(t)$ are orthogonal, which they satisfy

$$\int_0^T \phi_1(t)\phi_2(t)dt = 0 \tag{5.6}$$

b. Show that $\phi_1(t)$ and $\phi_2(t)$ are normal, which they satisfy

$$\int_0^T \phi_1^2(t)dt = 1 \text{ and } \int_0^T \phi_2^2(t)dt = 1 \tag{5.7}$$

for $i = 1,2,3,4$ where E is the energy and T the duration of $S_i(t)$.

3. Given the following assignment of phase with data

Table 5-3.QPSK Signal Vector

i	phase	data
1	$\pi/4$	1 0
2	$3\pi/4$	0 0
3	$-3\pi/4$	0 1
4	$-\pi/4$	1 1

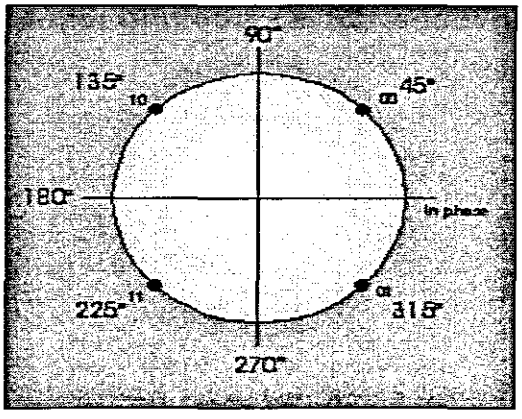


Figure 5-6 Quadrature Phase-Shift Code Keying

5.5 The Structure Wireless LAN

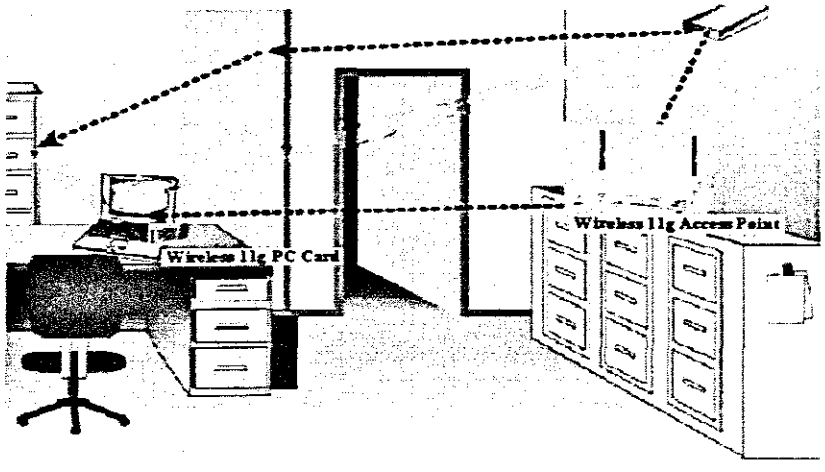


Figure 5-7 Using PC Card and Wireless Accessing Point to set up LAN

In this project, I choose the PC Card of 3COM Company, which is link between the sender and receiver. They are the 3Com Connect Wireless 11g Access Point and 3Com Connect Wireless 11g PC Card (Product Number: 3CRWE154G72).

5.5.1 Access Point and PC Card Features & Benefits

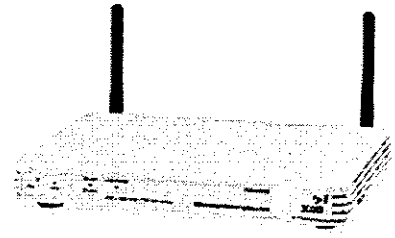


Figure 5-8 3Com Connect Wireless 11g Access Point

With the 3Com Connect Wireless 11g Access Point, users can access network resources, the Internet, and e-mail at speeds up to 54 Mbps and at distances up to 100 meters (328 feet). That is almost five times the speed of existing 802.11b wireless products, which makes this product ideal for small offices that run demanding audio, video and multimedia applications. Because the 802.11g standard uses the 2.4 GHz radio spectrum, the 11g access point is backward compatible with 802.11b wireless products. This flexibility preserves your network investment and allows you to upgrade or scale your network according to your budget and time frame. The 3Com Connect Wireless 11g Access Point supports 802.11g and 802.11b notebooks, PCs, and other wireless client devices.

Like all award-winning Connect products, the access point is affordable and easy to install. It delivers performance and reliability features that automatically select the best channel and connection speed, so connections stay clear and open. It also employs advanced 256-bit WPA (Wireless Protected Access) encryption as well as 40/64- and 128-bit shared-key WEP (Wireless Encryption Protocol) encryption to help protect data on the wireless LAN. And because the access point is Wi-Fi certified, it should work seamlessly with Wi-Fi certified products from other vendors.

Consistently recognized as the normally computer networking brand for small offices, the 3Com Office Connect family offers a broad range of products, including access points, LAN switches and hubs, gateway routers and firewalls, and PC Cards and network adapters.

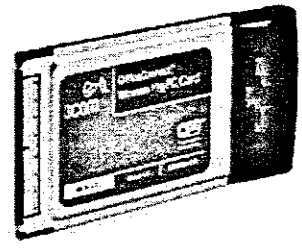


Figure 5-9 3Com Connect Wireless 11g PC Card

With the 3Com Connect Wireless 11g PC Card, notebook users can access network resources, the Internet, and e-mail at speeds up to 54 Mbps and at distances up to 100 meters (328 feet). That's almost five times the speed of existing 802.11b wireless products, which makes this product ideal for small offices that work with demanding audio, video, and multimedia applications. The 11g PC card works with the 3Com Wireless 11g Access Point to create a powerful, high-speed all-wireless network in minutes. The 11g products are compatible with 11b products, so both 11b and 11g clients can reside on the same network. This flexibility preserves your network investment and allows you to upgrade or scale your network according to your budget and time frame.

Setup and operation are extraordinarily easy, making these products great choices for first-time wireless users. Like all award-winning Office Connect products, the 11g PC card delivers reliability features such as dynamic rate shifting, which automatically matches the best connection speed to room conditions so connections stay clear and open. The PC card also employs advanced 256-bit WPA (Wireless Protected Access) encryption as well as 40/64- and 128-bit shared-key WEP (Wireless Encryption

Protocol) encryption to help protect data on the wireless LAN. And because it is Wi-Fi certified, the PC card should work seamlessly with Wi-Fi certified products from other vendors.

Consistently recognized as the computer networking brand for small offices, the 3Com Office Connect family offers a broad range of products, including access points, LAN switches and hubs, gateway routers and firewalls, and PC Cards and network adapters.

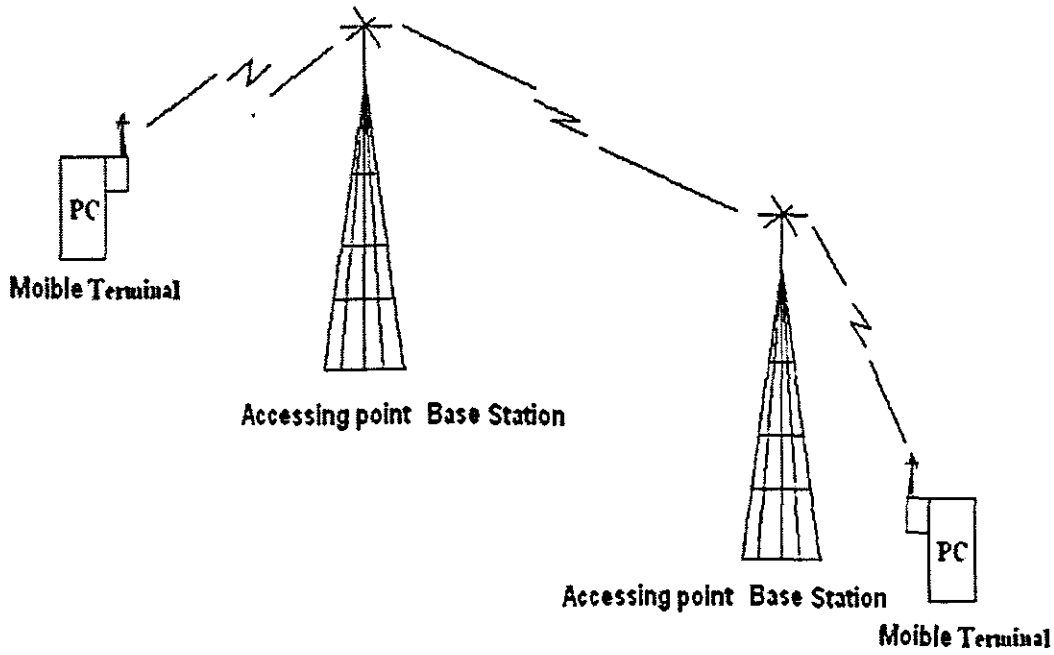


Figure 5-10 Wireless IP Networking

5.6 Conclusion

In this chapter, I explained the outline of the Internet and IP wireless technology. The Internet has been one of most popular communication system in the world. We also looked at a short distance wireless link. In this way, one can implement or access the Internet with wireless (using cell phone, PDA, personal computer etc.). This technique used the PC card and a wireless access point to connect to the Internet. The

PC card and wireless access point are based on the IEEE 802.11 protocol to access Internet.

CHAPTER 6

SIMULATION OF ENCRYPTION PROCESSES IN MATLAB

6.1 Introduction

In this chapter, the simulation of the XOR encryption process on computer is performed using Matlab.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB was chosen for this simulation because it is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

The MATLAB software stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and

EISPACK projects. Today, MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users in both the entirety as well as industry environment.

The main feature of MATLAB is a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow the user to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

6.2 Simulating The Problem

The problem was followed the description of the XOR encryption simulation. How to change color images to binary images, and XOR encryption and decryption. The simulation was separated into two parts, namely the encryption and decryption of still images, and then that of dynamic images. The results are shown in Figure 6-1.

6.2.1 Simulation XOR Encryption For Still Image processing

Simulations were performed to verify the validity of the proposed scheme for image encryption. The simulations were carried out on a PC Matlab platform. The validity of the proposed encryption method was investigated using the "*Board.tif*" image. Figure 6-1 describes the input images for computer simulations. The full software listing to perform this simulation is shown in Appendix 1.

Simulations Software Routines:

1. Display an image in Matlab (all types).

```
RGB= imread('board.tif');
imshow(RGB)
% 1. Display original image. Using function "imread" to load data of image and show
it as Figure.
```

2. Turn the color image into a binary image.

```
I= rgb2gray(RGB);
BW= im2bw(I,0.25);
%2. Using function "rgb2gray" to change the format of image from color to binary
image.
```

3. According to large of image, giving key-bit (random key-bit stream).

```
randn('state',0)
k= randn(648,306);
k1= im2bw(k,0.25);
%3. Create key-bit, using function "randn" to create random key-bit
```

4. Cause key-bit XOR the binary image. Getting an encryption image (complete encryption processing).

```
E= XOR(k1,BW);              %4. Encrypt image
%4. Encrypt image. Using function "XOR" to perform xor operation
```

5. After transmitting. Cause the transformed image XOR key-bit again, you can get the binary image again. This processing is decryption.

D= XOR(k1,E);

%5. Decrypt image. Using "XOR" function to decrypt the encrypted image

6. Compare the original image with the decryption image (Figure 6-1 (a)(g)).

7. Conclusion [Full programme sees in Appendices].

For example:

One time pad

Binary Image plaintext = 0100101010101111010011100 10...

To encrypt: XOR

Key-bit = 1100110100101010000101010 00...

Cipher text = 1000011110000101010110110 10... (6-1)

To decrypt: XOR

Key-bit = 1100110100101010000101010 00...

Decrypted = 0100101010101111010011100 10... (6-2)

The mechanism of the one time pad [Frank Stajano (1998)] is very simple: XOR every plaintext bit with the corresponding random key bit to yield a ciphertext bit: $c = p \oplus k$.

To decrypt, XOR the ciphertext with the key once more: $d = c \oplus k$ The two keys will cancel out and you'll get the plaintext.

The one time pad is the only demonstrably secure cipher, in the sense that even an infinite amount of ciphertext will not leak any bits of information to the attacker about the plaintext (except the length). This is because, given a ciphertext, you can choose any plaintext you want and there will always exist a key that generates that ciphertext from that plaintext. Well, the one time pad is not very practical because it requires a lot of key material (as many key bits as message bits). You can not ever

reuse the same key bits and you must use a truly random source; otherwise, cryptanalytic attacks become possible. If you use a pseudo-random number generator, you instead obtain what is known as a stream cipher.

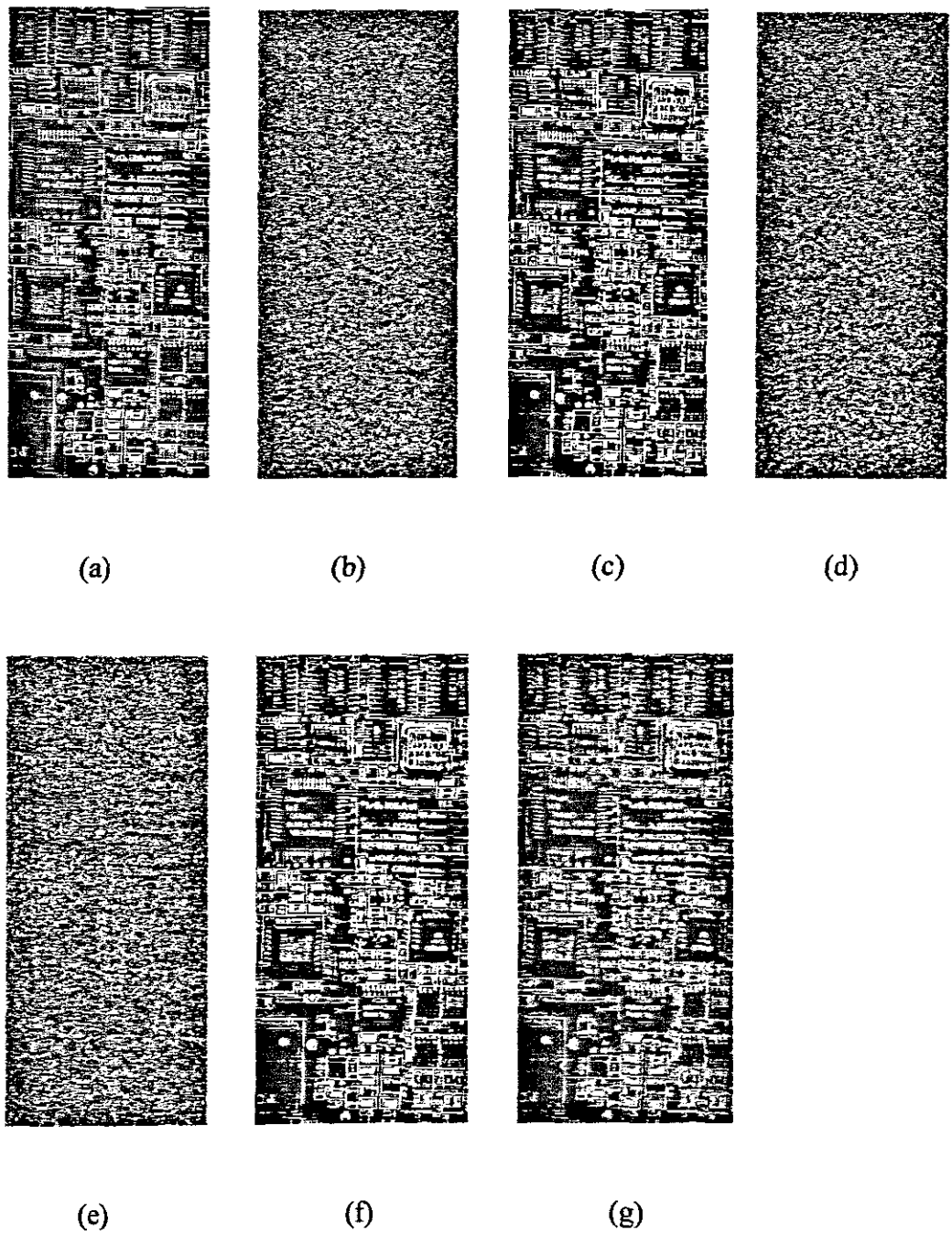


Figure 6-1 Testing Image and Result Image (a)~(f)
(a) Input image to be encrypted: gray-level image

(b) Random key bit stream

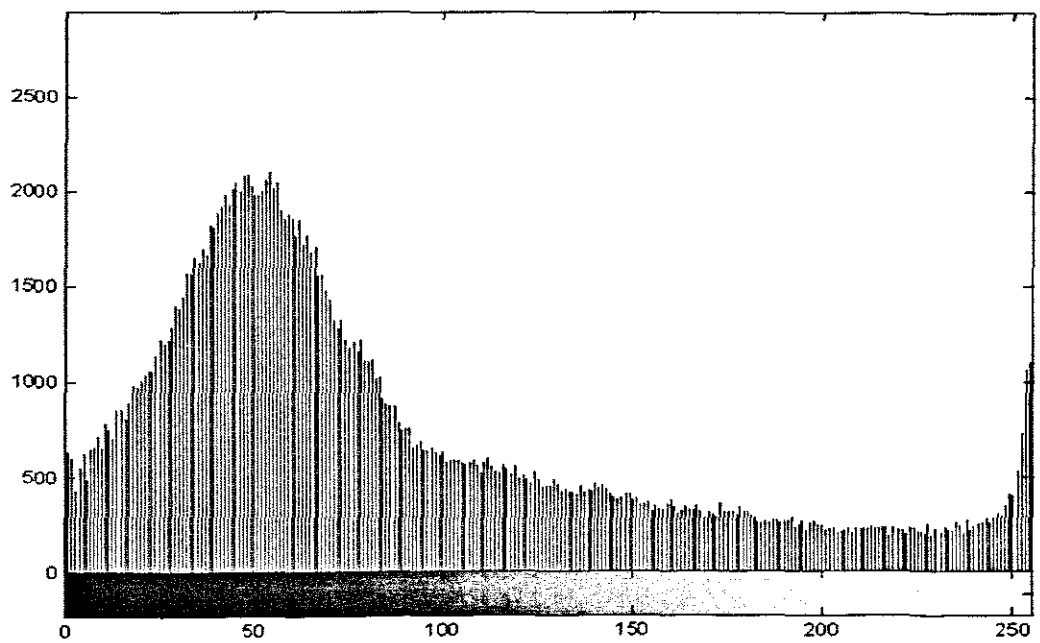
(c) Binary image

(d) Binary key-bit

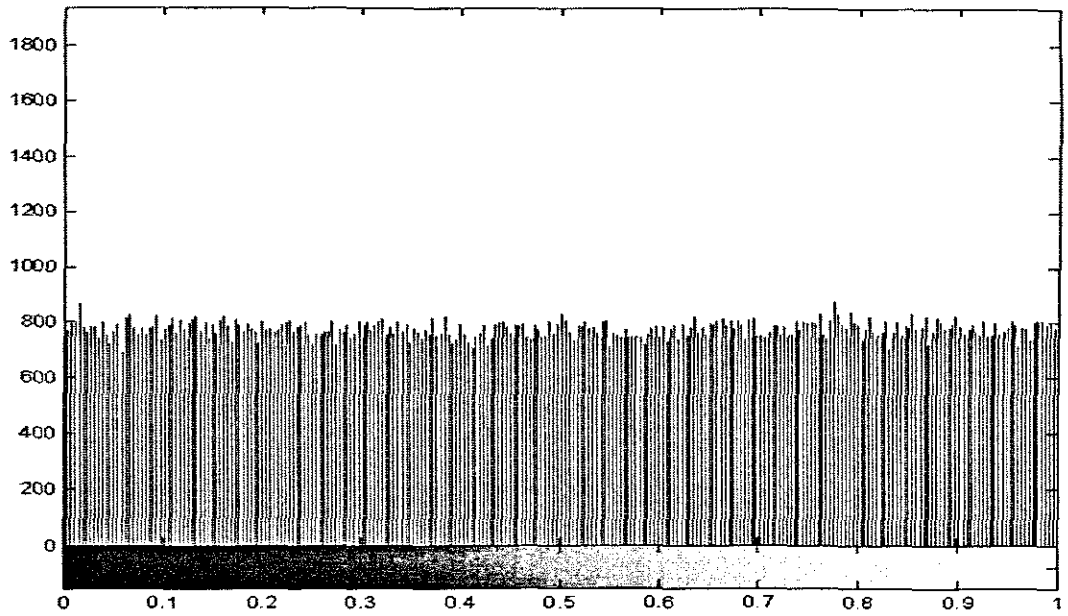
(e) Encrypted image: Binary image XOR Binary key-bit

(f) Decrypted binary image

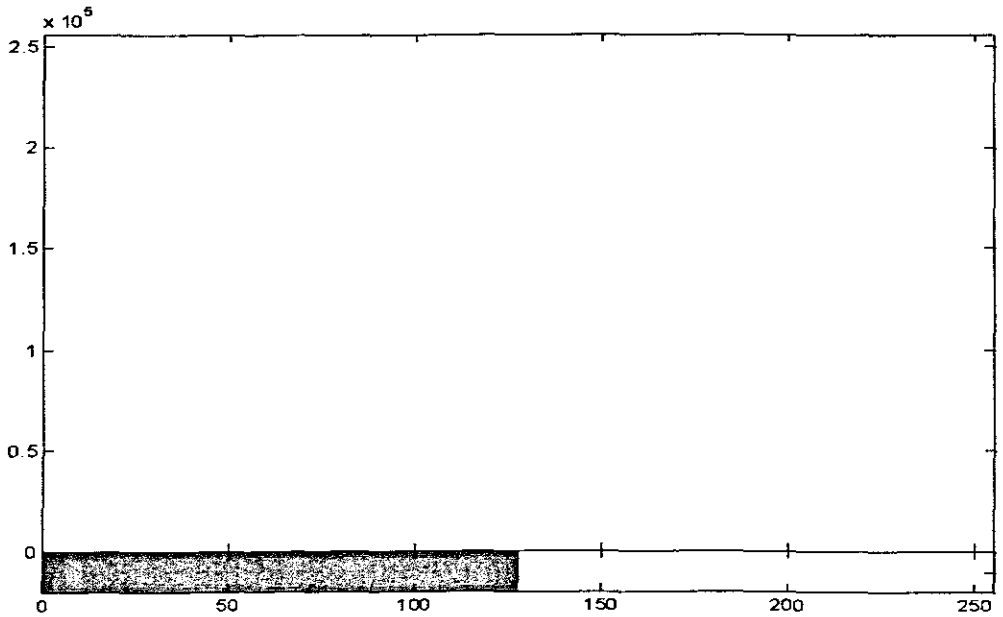
(g) Decrypted gray-level image



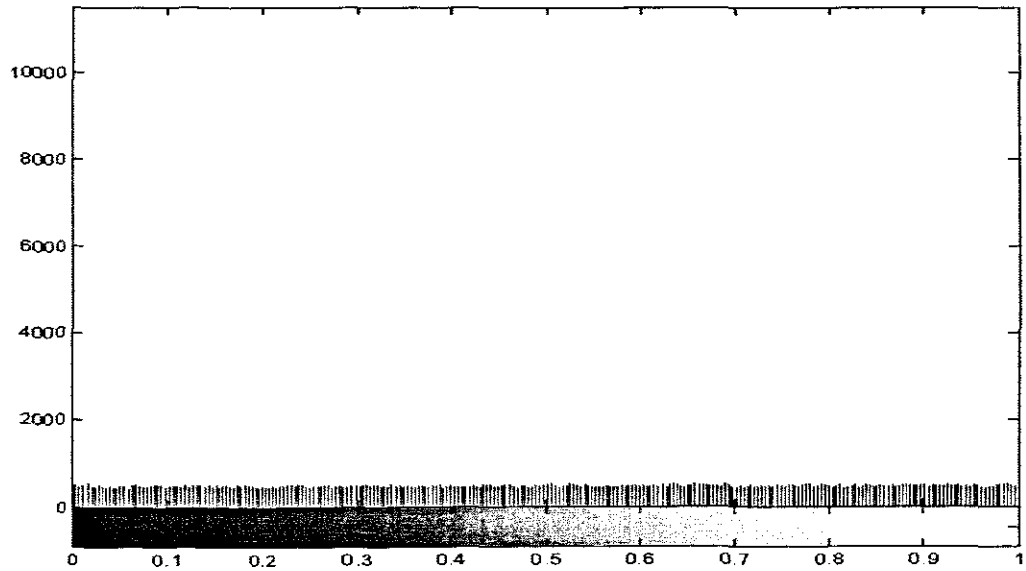
(a)


1800
1600
1400
1200
1000
800
600
400
200
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(b)


x 10^5
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250

(c)



(d)

Figure 6-2 X-axis denotes gray-level; Y-axis denotes intensity range

(a) The original image

(b) The key bit stream arrays

(c) The binary image

(d) The decrypted gray-level image

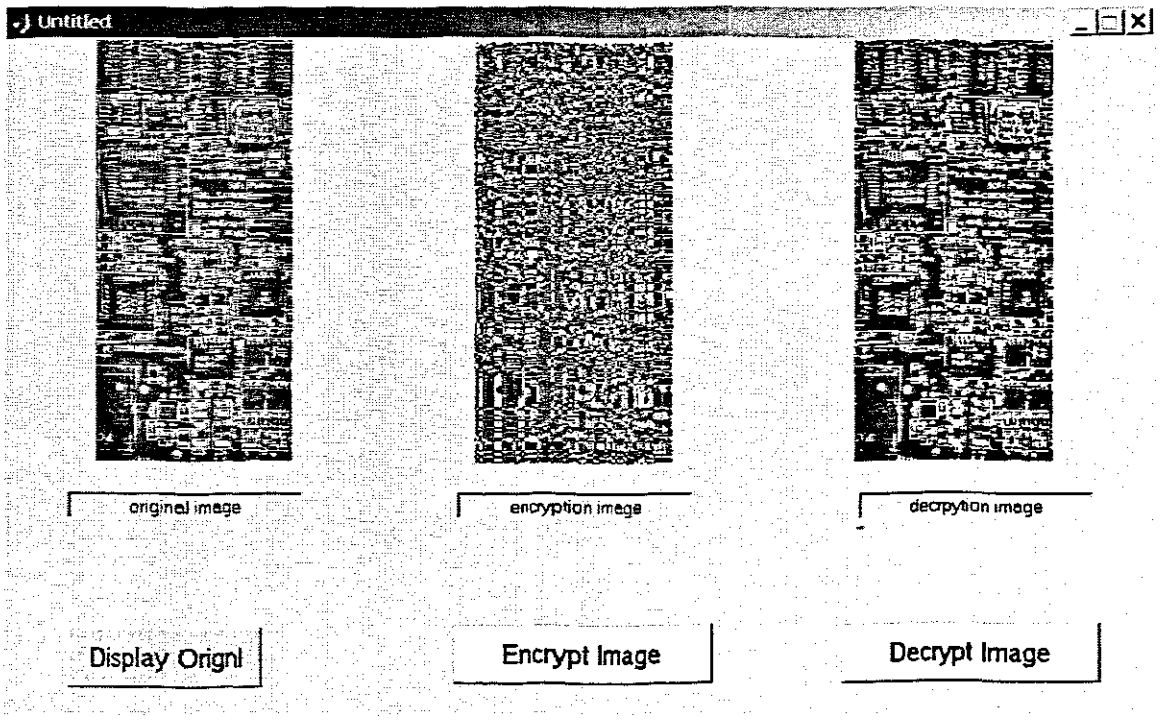


Figure 6-3 Desktop of Simulating XOR image

Figure 6-1 (a) is the input image and Figure 6-1 (b) is the key-bit stream. The size of the input image was 648x306 pixels and the size of the key-bit stream was also bit arrays. A digital generator generated the random key-bit stream. The input gray-level image and key-bit stream arrays should be converted to the binary image, as shown in Figure 6-1 (c) (d). As mentioned, the proposed method is that performing XOR operation with the random key-bit stream encrypted the input image. We can see the result in Figure 6-1 (e) that the binary image has been well encrypted into the random key-bit stream arrays. The decrypted binary image was also obtained using the same key bit stream as shown in Figure 6-1 (f). The resultant output image was converted to the decrypted gray-level image, as shown in Figure 6-1 (g). It is obvious from comparing Figure 6-1 (a) with Figure 6-1 (g) that they are some different. Because original gray-level image is converted binary image, a lot of information have been lost. The reason is that gray-level image has plenty gray level, which contains a great many light and shade information. While binary image is special of gray level image, which has merely two levels: white and black. So binary image contains information

less than gray-level image. But the binary images are the same and after XOR with key bit stream, as shown in Figure 6-1 (c) (f). Therefore, decrypted gray-level image are some different with input gray-level image, as shown in Figure 6-1 (a) (g). In Figure 6-1 (g) there is less gray level than in Figure 6-1 (a). You can see the result in Figure. 6-2 (d).

Figure 6-2 shows the histogram number of gray levels used to represent the images shown in Figure. 6-1 (a) (b) (c) (g). Figure 6-2 (a) is the histogram of the input gray level image. Figure 6-2 (c) is the binary image gray levels. It is obvious that processing caused the loss of lot of information about details of the gray levels. When the image was decrypted it showed less gray levels than the original image grayscale.

The simulations were carried out based on an ideal system. Computer simulations, do not consider problems such as a shift, scale change, rotation, etc, which occur in real time. However, these problems can occur in real situations. Using a variety of filters that have been widely used in image recognition one can reduce these various errors in the input image.

6.2.2 Simulation XOR Of Encryption In Dynamic Image processing

In fact, normally we hope to transmit dynamic encrypted images between two different terminals. For the next step in the development of optical XOR encryption, I also simulate the encryption of video in Matlab. Full software listing is shown in Appendix 11.1. The result is shown below.

The processing is very similar to encryption that of with still images. As we know, dynamic images consisted of many still images. When we show series images in short time, the dynamic pictures are displayed, somewhat like a cartoon show. After decryption, I chose to show only five frames. It is possible to play the series pictures and to get a normal moving pictures effect. For brevity and to illustrate by sample only five pictures was chosen. In practice, the signal is a series of data, and is to

sample in data packets, using the key-bit to XOR every packet. The routines for encryption of dynamic images are shown below.

```
i0= imread('picture 19.jpg');
i1= imread('picture 20.jpg');
i2= imread('picture 21.jpg');
i3= imread('picture 22.jpg');
i4= imread('picture 23.jpg');
%1. Display color images, only five frames

randn('state',0)
m= randn(480,640);
m1= im2bw(m,0.08);
%6. Create random key-bits, the size is 480X640 the same with images.

e0= XOR(t0,m1);
e1= XOR(t1,m1);
e2= XOR(t2,m1);
e3= XOR(t3,m1);
e4= XOR(t4,m1);
%7. Encrypt every binary images and key-bits, using "XOR" function.

d0= XOR(e0,m1);
d1= XOR(e1,m1);
d2= XOR(e2,m1);
d3= XOR(e3,m1);
d4= XOR(e4,m1);
%8. Decrypt images, XOR the encrypted images with the same key-bit.

M1= cat(3,a0,a1,a2,a3,a4);
s1= size(M1);
```

```
for i9= 1:s1(1,1)
   for j9= 1:s1(1,2)
      for k9= 1:s1(1,3)
         D9(i9,j9,1,k9)= M1(i9,j9,k9);
      end
   end
end

figure,mov= immovie(D,map)
figure,imshow(e0)
figure,mov1= immovie(D9,map1)
```

%10. Consist of decryption images, and display. For moving effecting, five frames pictures must be series shown. Using "cat" function to consist of five pictures, and using "mov" function to display series pictures to perform moving effect.

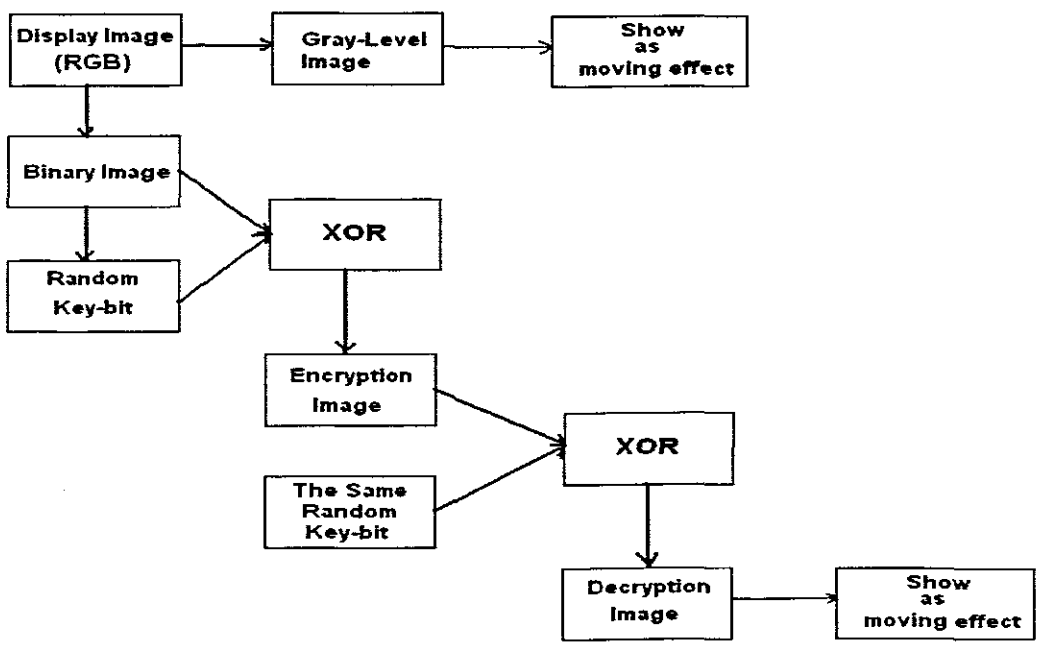


Figure 6-4 Encryption and Decryption Dynamic Images Flow Chart

For simulating the encryption video, five frames pictures were chose to show moving effect. The main problem is to series display the still image to simulation video. Through Matlab simulating, individual of five pictures was encrypted with key-bit, and then each of encrypted images was decrypted with the same key-bit. Finally, five decrypted images were series shown to perform moving effecting.

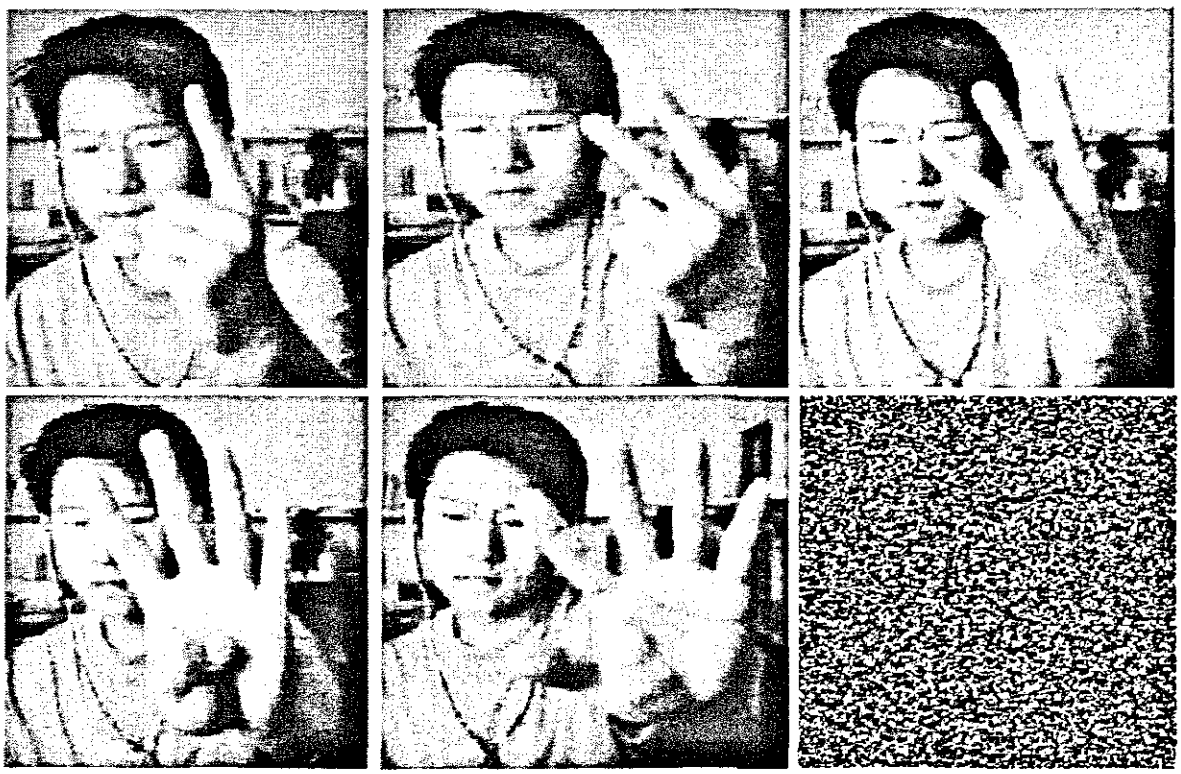


Figure 6-5 XOR Encryption-Decryption Five frames picture

6.3 Conclusion

From result of the simulation, it was educed that XOR algorithm can be use in encryption binary or black and white images. While using XOR to encrypt an image, it is difficult to reconvert. The reasons are that encryption method is not using complex mathematics with programme to code the data of the image again. The XOR encryption has been based on simulation, but the propose of this project is to develop the optical XOR system. The optical XOR process would be fast as it does require computation and an encryption program. The drawback that is extirpated is the light detection by CCD, which may impair the quality of the image. This will be discussed in the next chapter.

CHAPTER 7

DESIGN OF ENCRYPTION AND DECRYPTION SYSTEM WITH WIRELESS IP LINK

In this section is shown the development of a system to complete the optical XOR encryption-decryption process. The system was designed for transmitting an encrypted image through IP wireless, which is mentioned in chapter five. The project used the wireless card from 3COM Company to set up a wireless LAN.

7.1 Hardware Implementation of The Optical XOR Encryption

In my hardware system, I made use of two CCD units and two LCD units. The LCD units fixed from Samsung digital cameras (Table 7-1). The two LCD's are placed the same plane back to back. For proper operation we ensure that there are only two polarizes on the front and back of two LCD's. In this way the optical XOR system was accomplished. When I use the voltage signal to control the direction of liquid crystal array, the binary image would be displayed on the LCD. Different codes were sent to the LCD's, and because liquid crystal changed the light path, on the analyzer you would get a "new" code image, which are the result of an XOR operation between the code of image and random key-bit.

Table 7-1 Samsung digital Camera Function Handbook

IMAGE SENSOR	Type	1/2.0± CMOS
	Effective Pixels	Approx. 1.3 Mega pixels

	Total Pixels	Approx. 1.4 Mega pixels
LENS	Focal Length	Samsung lens f=9.0mm (35mm film equivalent: 46mm)
	F No.	F3.0
	Digital Zoom	Still Image mode: 2X, Play mode: 2X
VIEWFINDER	Optical Viewfinder	Optical viewfinder
	LCD Monitor	1.6" color TFT LCD
FOCUSING	Type	Fixed focus
	Range	1.0m ~ infinity
SHUTTER	Type	Electronic shutter
SHUTTER	Speed	1/15 ~ 1/2,000 sec.
EXPOSURE	Control	Program AE
	Compensation	¡¾2EV (0.5EV steps)
	ISO Equivalent	Auto
FLASH	Modes	Auto/ Auto & Red – eye reduction / Flash off
	Range	1.0 ~ 3.0 m
	Recharging Time	Approx. 10 sec
WHITE BALANCE		Auto / Daylight / Cloudy / Sunset / Tungsten / Fluorescent
SHOOTING	Movie Clip	Size: 320x240 pixels Recording time: Memory capacity dependent
SELF-TIMER		10 sec.

STORAGE	Media	Internal memory: 8MB flash memory External memory (Optional): SD / MMC card (Up to 256MB guaranteed)
	File Format	Still Image: JPEG (DCF) / DPOF Movie Clip: AVI(MJPEG)
	Image Size	Large: 1280x1024 pixels Small: 640x 512 pixels
	Capacity (16/32MB)	Large: Super fine 15, Fine 22, Normal 44 Small: Super fine 30, Fine 60, Normal 121 * These figures are measured by internal memory based. * These figures are measured under Samsung standard
IMAGE PLAY		Single image / Thumbnails / Slide show / Movie Clip
INTERFACE		Digital output connector: USB Video output: NTSC / PAL (User selectable) DC power input connector: 3.3V
POWER SOURCE		2 x AA alkaline / 2 x Ni-MH / CR-V3 batteries AC adapter (Optional)
DIMENSIONS (WxHxD)		108x56x37mm / 4.3x2.2x1.5in
WEIGHT		Approx 125g / 4.4oz (without batteries and card)

SOFTWARE	Camera Driver	Storage Driver (Windows98/98SE/2000/ME/XP, Mac OS 8.6 ~ 10.2)
	Application	MGI PhotoSuite, Digimax Viewer
For Windows		PC with processor better than MMX Pentium 266MHz (XP: Pentium II 300MHz) Windows 98/98SE/2000/ME/XP Minimum 32MB RAM (XP: 128MB) 110MB of available hard-disk space USB port CD-ROM drive 800X600 pixels, 16-bit color display compatible monitor (24-bit color display recommended)
For Macintosh		Power Mac G3 or later Mac OS 8.6 ~ 10.2 Minimum 32MB RAM 110MB of available hard-disk space USB port CD-ROM drive QuickTime 4.0 or later for Movie Clip

The image was passed through LCD1, and the key-bit download into LCD2 from the computer through a USB cable. Therefore, when the light passes through the two LCDs, the encryption processing is completed. When the light pass through the polarizer, it has say horizontal polarization, and when the light pass through the LCD1, the direction of polarization will turn 90 degree if the voltage of pixel is "on" (Figure 7-1). The second LCD which displays random key-bit will turn the light again by 90 degree. When the light arrives at the analyzer plane, some light may pass, while

other rays may not. The CCD then the available detects light. So on the analyzer plane the encrypted image is obtained.

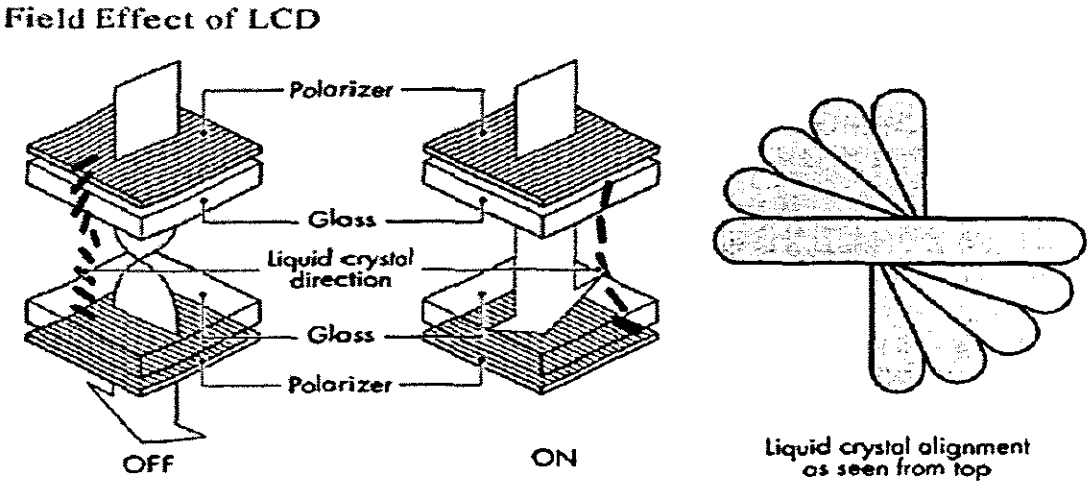


Figure 7-1 Cross-sectional view of a Typical Display

7.2 Transmitting Encryption Image over Wireless IP

The communication system chosen in this case was wireless IP. I choose the wireless PC card and accessing point of 3COM Company. Wireless LAN Manager allow one to create and store settings for the different wireless LAN scenarios that may be encountered, such as connection to a home or office wireless network, a wireless network at an airport or hotel, or an ad hoc (PC-to-PC) network. Particular features of this technology is shown:

- Monitor was status of the wireless LAN connection
- Configuring settings such as security and power management
- Creating, selecting, and distributing wireless LAN profiles
- Automatically set up power management settings for the wireless LAN adapter based on the active power source
- Perform diagnostics for troubleshooting
- Refresh the IP address

The encrypted image is downloaded into one computer through a USB cable. It is then transmitted to another computer over wireless IP LAN. This processing resembles the ground connected Internet LAN, which is commonly used to send some E-mail and data to other terminals. At the receiver point, the encrypted image is displayed onto LCD1, and using the original key-bit to XOR with the encrypted image the original image is obtained. This is shown in Figure 7-2.

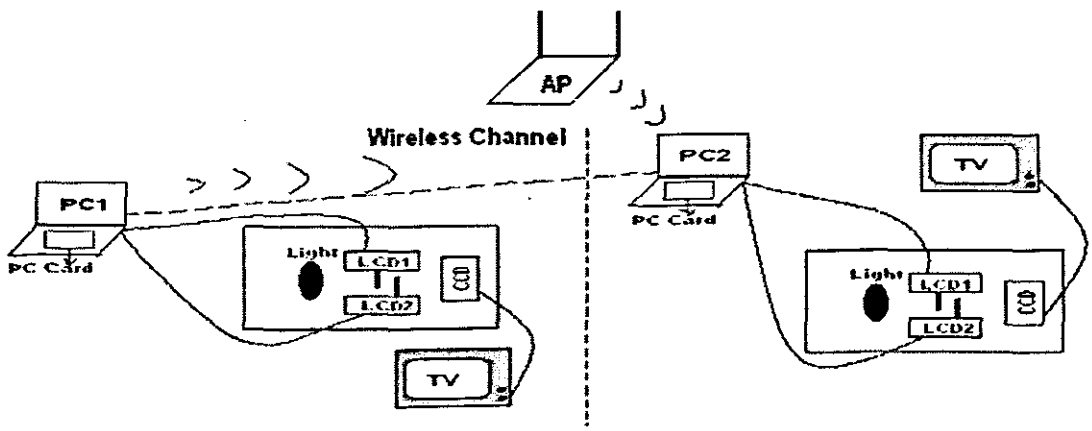


Figure 7-2 Wireless I/P Connected

7.2.1 General Description of the Wireless LAN

3Com Office Connect wireless products are built for small-office environments, but supplied in this case an affordable, reliable wireless network. The communication system was set up in the communication laboratory of Peninsula Technikon

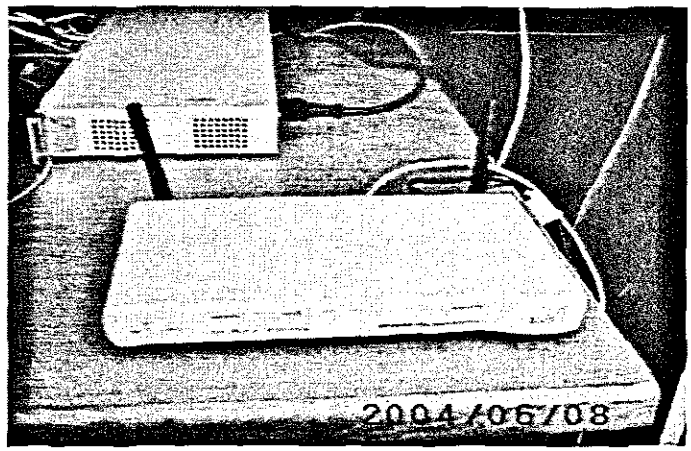


Figure 7-3 3COM Access Point

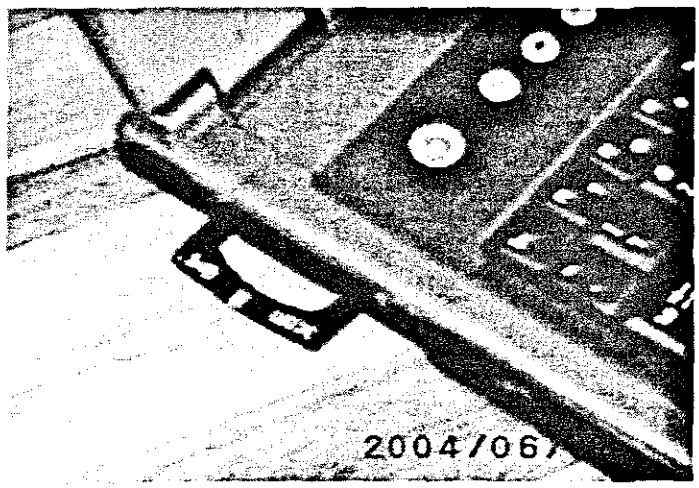


Figure 7-4 3COM PC Card for Laptop

The wireless LAN is multi standard and the Tri-mode 11a/b/g access points and clients provide universal wireless connectivity across all IEEE 802.11 networks. Additional products include 802.11g and 802.11b access points, gateways, and clients.

It allows speeds up to 54 Mbps. The access points supported up to 128 simultaneous users, while Cable/DSL Gateways allow up 128 wireless users or 253 wired users to securely share a broadband connection in this system.

The system also had Features like advanced 256-bit WPA encryption (Wi-Fi Protected Access) and 128-bit WEP shared encryption, which further ensured that wireless transmissions were private. In addition, Office Connect Wireless Cable/DSL Gateways helps keep wired transmissions secure with features state packet inspection firewall, VPN pass through and Hacker Pattern Detection. Dynamic Rate Shifting and Clear Channel Select allow you to take advantage of the highest connection speed and low traffic channels.

The 3Com 802.11b and 802.11g products were to be compatible, protecting current and future wireless expansion.

The wireless LAN is set up as follows:

The link was set up with the 3Com Connect Wireless 11g Access Point, 3Com Connect Wireless 11g PC Card, and 3Com Connect Wireless 11g USB Adapter. With this set-up it allowed accessing network resources, the Internet, and e-mail, all at speeds up to 54 Mbps at distances up to 100 meters (328 feet). These 11g products operate at almost five times the speed of existing 802.11b devices, so they are ideal for multimedia or other high-bandwidth applications.

The Wireless networks proved to be a viable alternative to wired network expansions and an attractive solution for this encryption application. Since there was no need to drill holes or run Ethernet cabling, was also cost-effective for temporary networks. The access point supported up to 128 simultaneous users.

7.3 Conclusion

After the computer simulation in Matlab, I designed and made the encode system. The main part of system was two LCD's, which took out from the Samsung digital cameras. The XOR algorithm is performed these two LCD's. And then the encrypted images are transmitted over wireless IP LAN, which is consisted of the PC cards and

wireless accessing point of Internet. Whole system can work, complete encryption and decryption, and set up wireless link between two terminals.

Of course, this system has many shortcomings. Such as light source, detecting CCD etc. In future people use this system into the fact, I would improve the system to perfect in further.

CHAPTER 8

RESULTS

8.1 Experiment Results

A successful polarization encryption technique based on XOR logic was performed. Performing optical XOR operations with the random key bit stream using LCD units encrypts the original image. The resultant encrypted image, which is detected by a CCD camera, is displayed on computer. Then through the wireless LAN, the encrypted image is sent to other receiver. In the receiver, encrypted image is reconverted to binary image (original binary image). In this encoding system, each image is transmitted, after they are encrypted. If both of sender and receiver used the same random key-bit which did the original random key bit stream engender. On the receiver, they can decrypt the encrypted image with the same key-bit. Consequently, the security link is built for transmitting secret image between the long distant. Even if the adversary steal the encrypted image through the Internet, it is difficult for them to reconvert it without the same key-bit. On the other, if an encrypted image is used as a key in the security system for accessing to a restricted area, it can be decrypted by the proposed encryption method and then compared with the original reference image, which was stored it in the system. The adversary dose not know the encryption key bit stream, it will be difficult to know the original images used in the security system.

8.1.1 Review of The Technique

Performing optical XOR operations with the key bit stream that was generated by digital encryption algorithms encrypts the input image. The key bit stream generated

by stream cipher system is binary; a gray-level input image would be converted to a binary image. The image was projected the LCD 1. The LCD pixel has only two conditions, "On" or "Off". This feature is perfectly suitable for processing a binary number. The key bit stream was downloaded on the LCD 2. The optical XOR operations between the key bit stream and polarization encoding using the polarization characteristics of the LCD performed the binary of image. The results of the XOR operations, were detected by a CCD camera, and then converted to the encrypted image that was sent to on PC. But before decrypting the image, the encrypted image was sent through wireless I/P network. In chapter 7, the communication link IP over wireless and the PC card was discussed. In figure 8-1 the basic encryption system shown. The light is reflected from object, when the light cross the LCD, that path of light would be changed, so the XOR processing should be carried out. Namely, the real encryption image will be captured and transmitted.

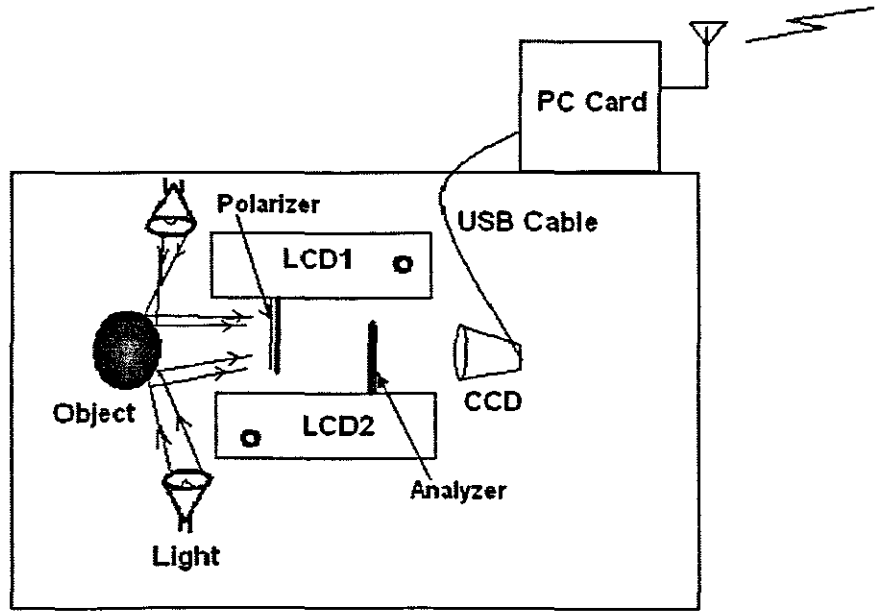
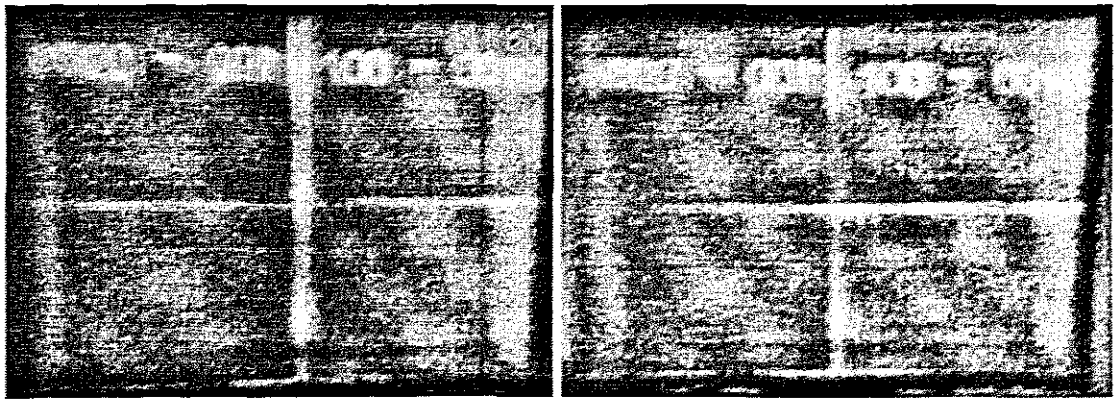


Figure 8-1 Sketch of encryption system

8.2 Focus Adjustment Problem

Several problems were observed in obtaining good result. The First of them was focus adjustment problem. When the two LCD units were aligned, it was difficult to keep

the alignment for a pixel-by-pixel match. The pixels of LCD were 640x256, so the center of focus was coordinate (320,256). Figure 8-2 shows details that the back LCD has landscape orientation displacement with the front LCD. So the center of focus of the back LCD was adjusted to (coordinate 309,262). Then the focus of two LCD units was a best as was possible aligned. This was the major obstacle is getting a good result with the low definition LCD's used in this encryption.



Focus Departure **Adjust Focus (320,256-309,262)**

Figure 8-2 Adjust Pair of LCD Units of Focus

8.3 Real Image Capture

The word "GOOD DAY!" and a picture were captured and displayed on the LCD1 and the key bit generated on the LCD2. A CCD was then used to detect the encrypted image. Below is shown several picture captured during process.

Figure 8-3 Original Image

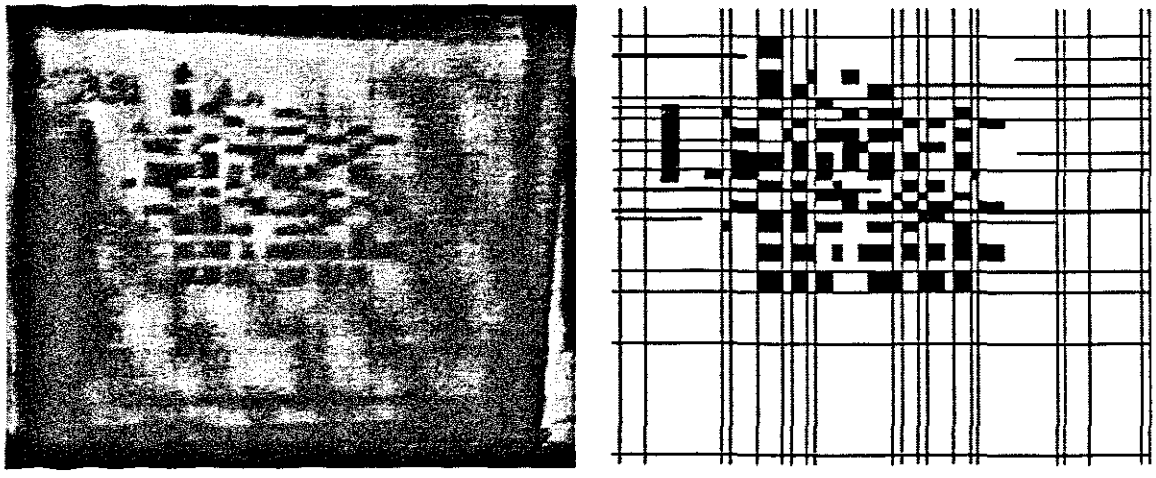


Figure 8-4 Random Key bit Stream

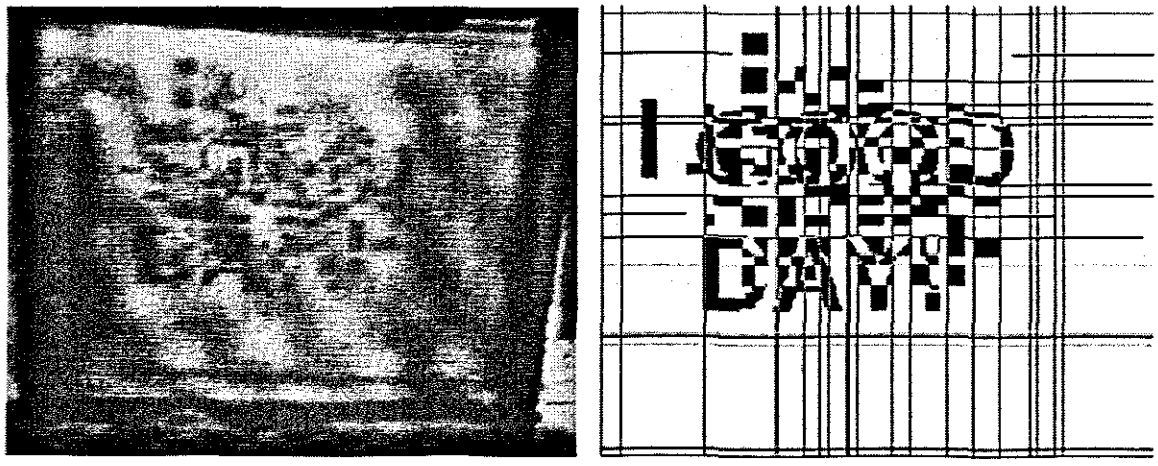


Figure 8-5 Encrypted Image

Figure 8-6 Decrypted Image

Figure 8-7 Original Image

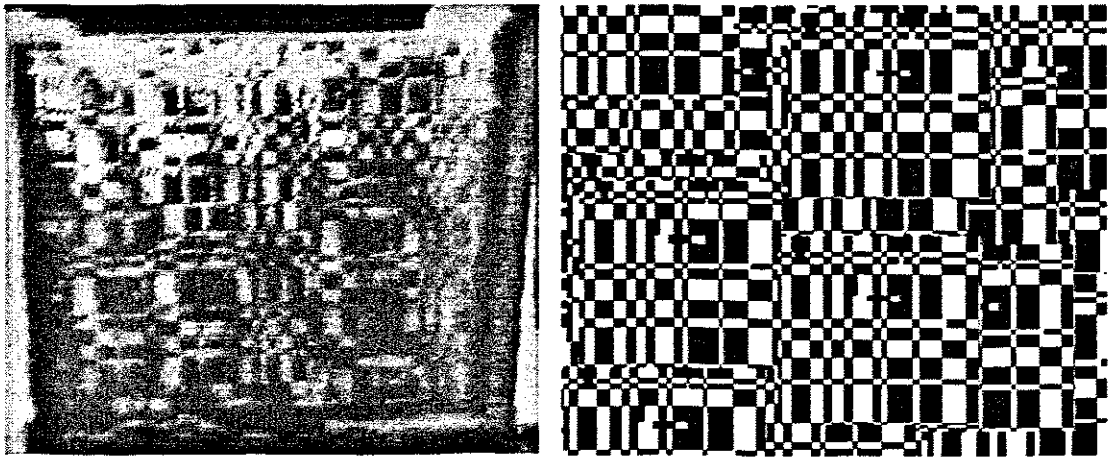


Figure 8-8 Random Key-bit

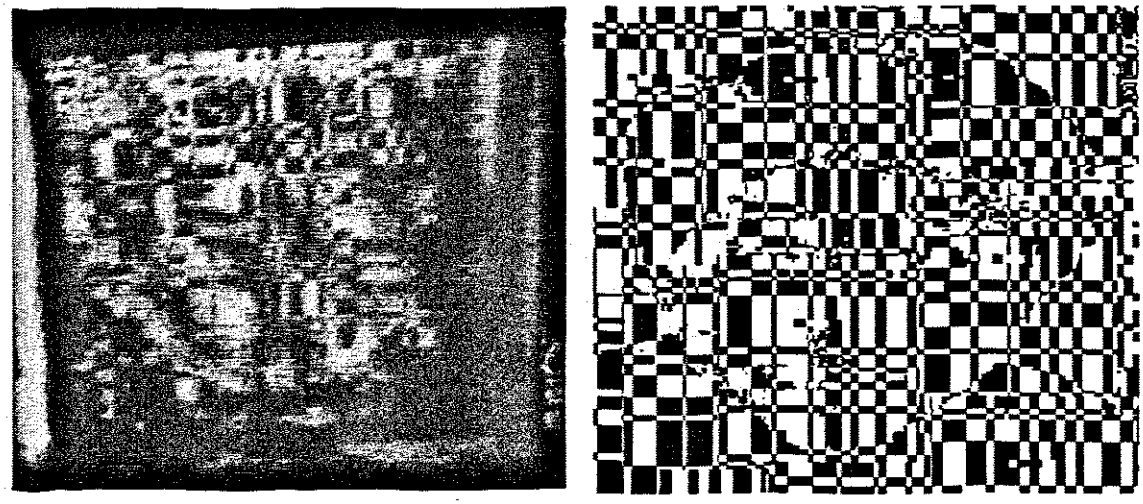


Figure 8-9 Encrypted Image

Figure 8-10 Decrypted Image

Figure 8-11 Comparing The Original and Decrypted Image

Software routines were written to capture the decrypted and original images in order to compare the difference. These routines are shown below. Routine 1 (Figure 8-12) displayed the decrypted image gray scale and routine 2 (Figure 8-13) shown the original image gray scale.

Information of Image (Figure 8-11)

Routine 1

```
info = imfinfo('Picture 96.jpg')
info =

        Filename: 'Picture 96.jpg'                    //Decryption image
     FileModDate: '24-Aug-2004 04:37:14'
        FileSize: 15872
          Format: 'jpg'
   FormatVersion: ''
           Width: 556
          Height: 311
        BitDepth: 24
       ColorType: 'truecolor'
  FormatSignature: ''

info = imfinfo('Picture 98.jpg')
info =
```

Routine 2

```
    Filename: 'Picture 98.jpg'                    //Original Image
  FileModDate: '24-Aug-2004 04:45:06'
    FileSize: 12792
      Format: 'jpg'
FormatVersion: ''
       Width: 556
      Height: 311
    BitDepth: 24
   ColorType: 'truecolor'
FormatSignature: ''

O = imread('Picture 98.jpg');
imhist(O)

D = imread('Picture 96.jpg');
figure, imhist(D)
```

imhist(I,n) displays a histogram with *n* bins for the intensity image *I* above a grayscale colorbar of length *n*. If you omit the argument, *imhist* uses a default value of *n = 256* if *I* is a grayscale image, or *n = 2* if *I* is a binary image.

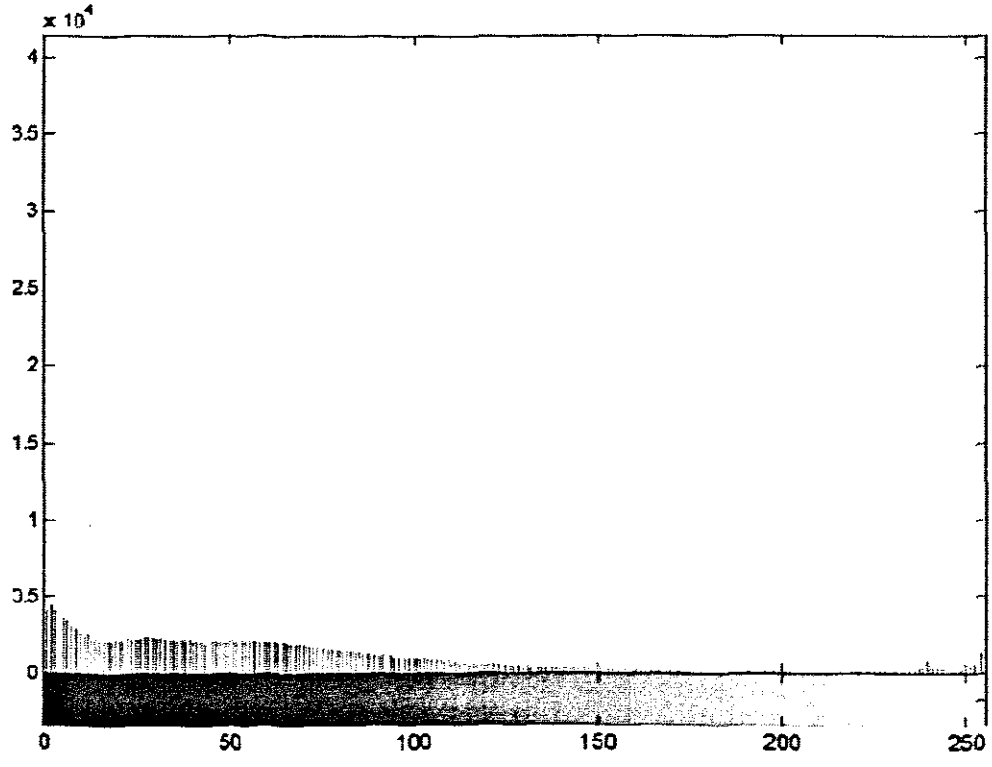


Figure 8-12 Histogram of Decryption Image Grayscale

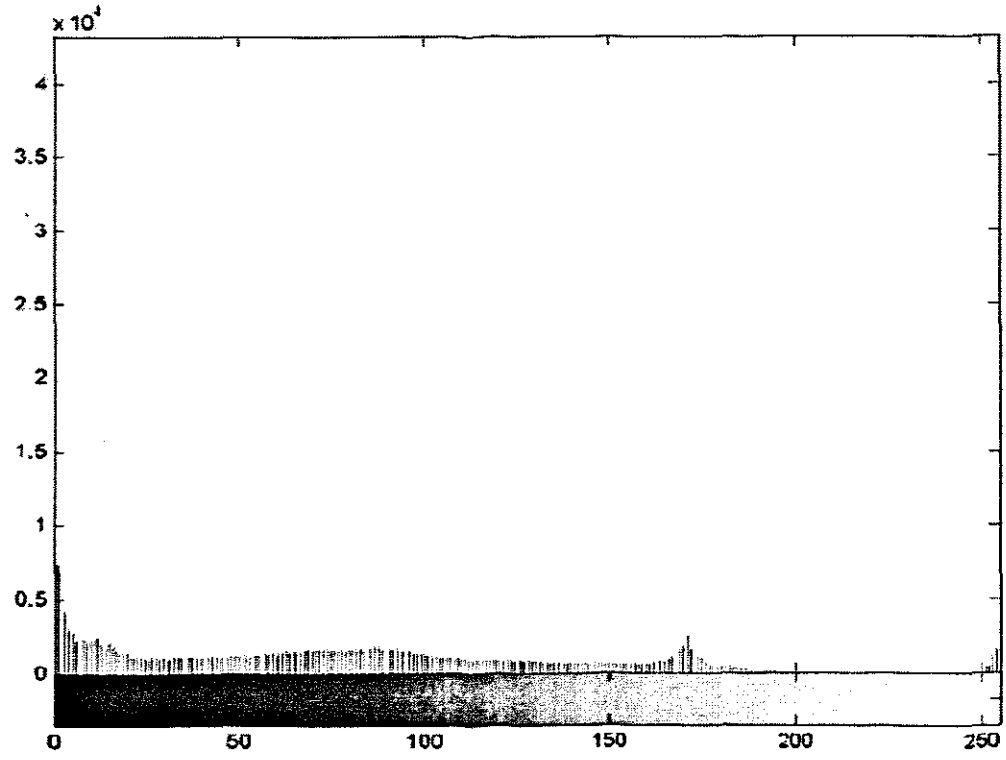


Figure 8-13 Histogram of Original Image Grayscale

Figure 8-12 and 8-13 are histogram of Figure 8-7 and Figure 8-10 image gray-level scale. Through comparing with these two images, the decryption image lost some gray-level information, but the basic tendency is similar with original image.

8.4 Application Encryption for Medicine Image

This optical encryption method could be applied for encrypting X-ray medicine images, which only contain black and white color. In the modern medicine, Internet video technique would be used to cooperate diagnosis between different doctors and different medical treatment institutions. On security consideration, real medicine images must be encrypted before they are transmitted. As well, the optical XOR encryption method is adapt to encrypt medicine images as fast, safety, and credible method. So long as both sides use the same key-bit to encrypt and decrypt image, then both of them can gain the same original image. The information of image is series binary data stream, through the XOR algorithm function with key bit, this

performing cannot increase the bit number, so cannot increase burden in the transmitting. When the encrypted images are decrypted, only the encryption image and key-bit are overlapped. Then detected polarizer light by a CCD, the decryption images were displayed on the monitor though a USB cable between the CCD and the monitor. In the telemedicine field, we can transmit encrypted X-ray images over IP wireless. The different doctors can diagnose for the patients wherever they are in the world, while their private information will be full security.

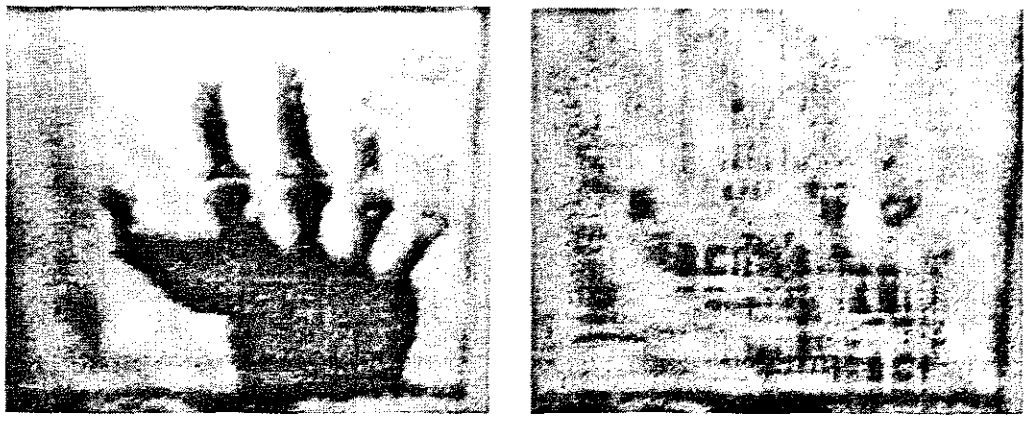


Figure 8-14 X-ray of Right Hand Bone and Decryption image in Receiver

8.5 Facts Contributing to Grayscale Image Loss

According to the result of the experiment, the decrypted image is not as good as original image. This project did not inclined image restoration. Many factors were influencing the reconverting process.

8.5.1 The CCD Noise

The optical XOR encryption technique could be performed with low cost LCD's and a CCD device. When photoconductive devices (CCD) detected the light, in the CCD they produce an electron, which form the voltage or current as output signal. Finally through A/D (analog to digital), the digital signal would be display on LCD or CRT

monitor. So the sensitive of CCD direct effect the output voltage. On the other hand, the A/D has mostly effected for output digital signal during A/D processing.

The CCD is made of the PN diode through the detecting the light to change the output voltage. The light controls the output analog voltage signal. Before the imaging, the analog signal would be converted to digital signal the A/D converter. I have described it in the Chapter 4.6. So many chips affect the output digital signal. [Stephan Baier (1993)]

The main limiting Factor of CCD Noise Sources:

- CCD-output stage KT/C-noise
- Semiconductor Noise—Shot, Flicker, White Noise
- Resistor/Thermal Noise
- ADC Quantization Noise
- Line Frequency, 50/60Hz

The noise floor sets the lower limit of the dynamic range in an image system. Different techniques are available to maximize the dynamic range and optimize for the input range of the A/D converter, but a thorough understanding about the noise sources is crucial. The main noise source, besides digital feedthrough, is called kT/C-noise of the FET reset switch caused by its channel resistance. This is discussed later.

Another limit is set by the quantization noise of the A/D converter. The rms quantization noise is expressed by the equation $q/\sqrt{12}$, with q being the bit size or LSB weight of the converter. For example, a 10-bit converter with a fullscale input range of 2V has a bit size of 2.0V/1024=1.953mV. Hence, the quantization noise is 564mVrms. Assuming a 0.1pF sense capacitor the detection limit would be at about 350 electrons due to the quantization noise.

The thermal noise is factor of the channel resistance (RON) of the FET switch (SW) used or the CCD camera. This noise is often termed as kT/C-noise. With a typical

value of 100 to 300 electrons (rms), this is the dominant limitation for the detection of small signals.

Flicker or *1/f* noise originates in the CCD MOSFET, and relates to the presence of traps associated with contamination and crystal defects in the semiconductor. Its magnitude is therefore process dependent. We used a 1.3m pixel CCD in the experiment.

Johnson noise or resistor noise is another factor. It is temperature dependent and equal to $\sqrt{(4KTRB)}$. White Noise (Johnson noise) has several origins. The noise of the load resistor (R_L).

White Noise: $e_{nw} = \sqrt{4KTBR_0}$

K = Boltzmann's constant = 1.38054 E -23

T = absolute temperature in Kelvin, (298°K = +25°C)

R = on resistance of switch in Ω

B = Noise Bandwidth in Hz

R_0 = output impedance of CCD output stage typical; $R_0 = 200\Omega to 20K\Omega$ 200Ω

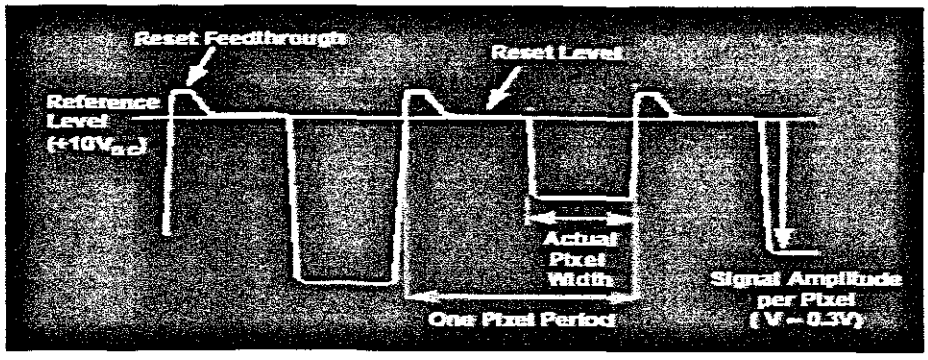


Figure 8-15 The CCD Output Signal

8.5.2 The Noise From LCD

In the system, the LCD is used as a monitor to display the original image, the random key-bit, and the decryption image. In the above the LCD imaging principle (Figure 4-3) was described The LCD is only used to change the direction of light phase under controlling voltage. If the voltage is not precisely controlled, which the liquid crystal cannot twist absolute 90 degree. It would reduce setup balance of black and white; importing noise into the imaging process. In addition the number of pixels of LCD directly effect the articulation of the image. The experimental setup in this project used a 1.6" color TFT LCD (640x512 pixels). Increasing the number of pixels would have provided a more detailed image.

8.5.3 Light Source Design

The light source is an important part in my system. The white LED was tried to supply light source, which were made of 63 LED's (9x7). The results were not satisfactory. The reason was that the light intensity was not average in the space, which caused the brightness to cause local flecks. A fluorescent lamp was tested and supplied soft white light. However the fluorescent lamp was working under 50HZ, so the frequency flash caused preponderance for noise when the CCD was detecting the light. At the same time, the light would be dispersed, and reflected, which also effected the articulation of the image.

8.5.4 Noise From Wireless Channel

The wireless IP technique was used to transmit the data, which obviously led to a decline in the signal power. In addition channel noise was added. There was not enough time to perform a thorough test on channel-induced noise and the specifications of the OFDM wireless system was all that was relied upon.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS

9.1 Introduction

The aim of the overall study was to develop an optical encryption processing system that was secure before entering the electronic communication system and to decrypt such an image after reception. The technique for achieving this has been based on using LCD technology to set up an encoding system and to connect up a wireless link to transmit encrypted images via a wireless LAN.

9.2 Problems Solved in the Dissertation

The problem solved in this dissertation was to adhere to the requirement to design and construct an optical encryption system based on the XOR technology and transmit the encrypted image through a wireless link to the receiver, whereupon it was decrypted again. This problem was divided into many sub-problems, which is described below:

9.2.1 Simulation XOR Image Encryption in Matlab

Although several methods, both traditional as well as optical were investigated in this research, this project was based on optical XOR encryption technology. The background and development optical XOR encryption techniques were done in a literature review. After studying XOR encryption logic theory and LCD technology, an analysis of the system security was performed and a model developed for simulation. Matlab software was written and used in order to experiment with various possibilities in the simulation of XOR encryption. Figure 9-1 shows a typical simulation result. An original image was converted to gray level and then encrypted using the software. After that the image was decrypted with a noticeable loss in

quality, although clearly definable edges were still visible after decryption. The encryption/decryption software worked well.

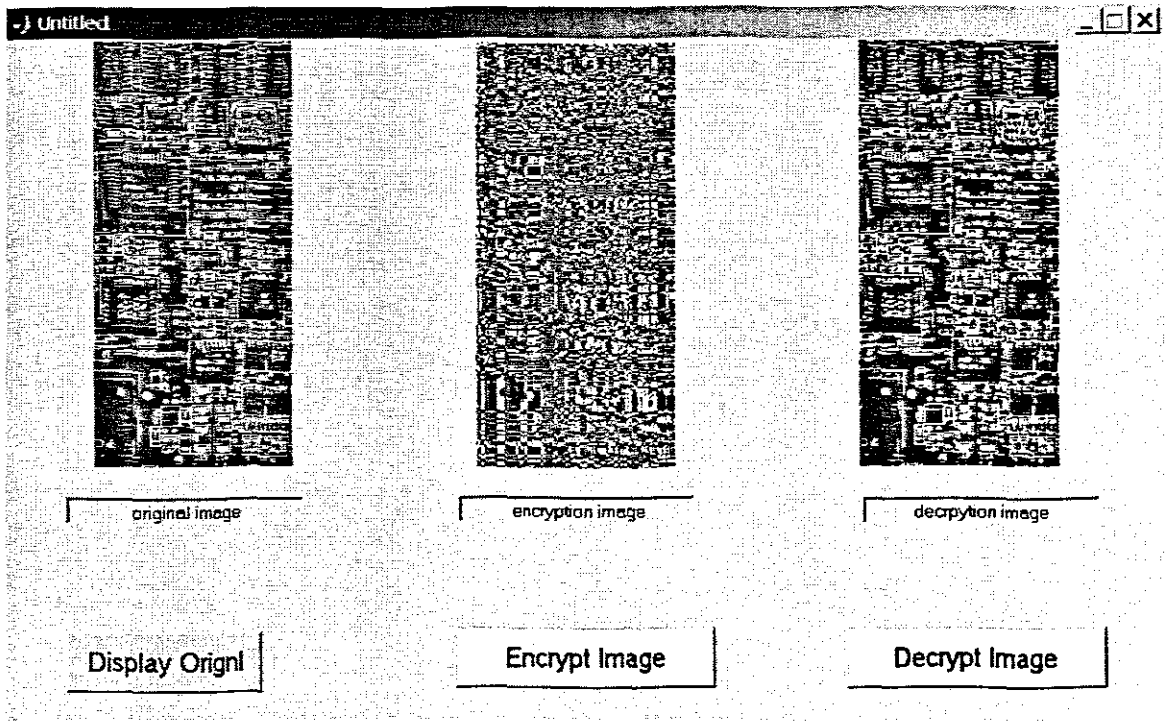


Figure 9-1 Simulating XOR In Matlab

9.2.2 XOR Encryption Using LCD's

After the simulation, the task was to design the XOR encryption system using LCD units. The LCD display theory was used extensively in the XOR encryption technique. On several occasions setbacks occurred such as breakages in the cable interfacing and controlling the LCD's. The simulation of the technique as well as theoretical analysis proved that it was viable to reliably use XOR encryption using two LCD's. The fact that light direction of polarization was influenced by transmission through LCD crystals made it possible to implement XOR logic using 2 LCD's on a pixel by pixel basis, which was sufficient for experimentation. A cross sectional view of a typical display is shown in figure 9-2 below.

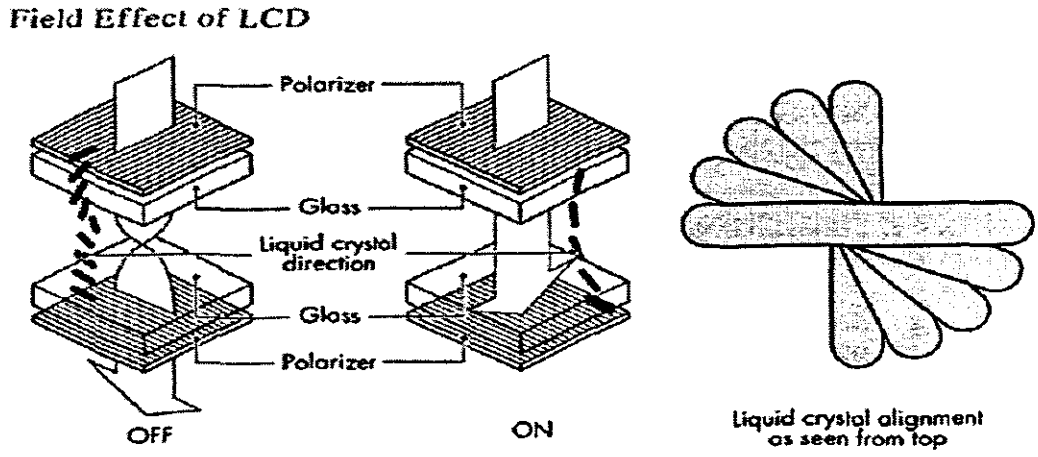


Figure 9-2 Cross-sectional View of a Typical Display

9.2.3 Setting up Wireless Link

Having obtained the encryption image it was necessary to transmit it to a remote receiver. A Wireless LAN link was set up with the IEEE 802.11 LAN. We used 3COM Company supplied equipment for the wireless LAN through the AP (Access Point) and PC Card (PCI). This wireless technique is one of popular wireless LAN solution method. So between the send and receiver, a wireless link was set up to transmit the encrypted image.

9.2.4 Light Source Design

The light source design, although not a major task in the beginning tuned out to be quite difficult. The light must be passed to the two LCD's detected by the CDD to perform XOR algorithm. Many different sources of light were tried to supply the light for the system including LED, incandescent lamp, and fluorescent lamps. I chose the fluorescent lamp as source. However, this experimentation is to testify veracious availability of optical XOR encryption. So the research on light sources will be in the future. Light source characterization is an important factor to improve the quality of detector result.

9.2.5 Performance of the Instrument

The final task required the testing of the device. All project systems were implemented as discussed. Via this experiment, the optical XOR encryption has been successfully implemented with a pair of LCD's and a CCD system. This optical encryption method was found suitable to encrypt clearly defined binary images, and could include medical images such as X-rays, which only have two gray-levels. The Wireless LAN unit was also set up and the images transmitted over this link between a sender and receiver. One could say that in the future a patient wherever they might be could be treated by different doctors, from different countries with full security of the patient record privacy. The overall experiment was successful but with much room for improvements. A better quality LCD and CCD would have produced far superior results.

9.3 Recommendations for Further Study

This encryption method has opened up many research avenues. Through this experimentation the optical XOR encryption has been proved implemental using low cost LCD and CCD technology. There are, however lots of room for improvement that would enhance the quality of the decrypted image. For example; a characterization and reduction of the channel noise needs investigation. Several gray level conversions instead of just two levels also need to be investigated. Perhaps even how to reconvert a color image. Techniques on how to reduce overall noise due to light source and CCD technology must also be performed.

REFERENCES AND BIBLIOGRAPHY

Eskicioglu, A. M., Town, J. and Delp, E. J.; 2003: *Security of Digital Entertainment Content from Creation to Consumption*, Signal Processing: Image Communication, Special Issue on Image Security. 18(4), pp. 237-262.

Wang R. K., Watson I. A., and Chatwin C.; 1996: Random phase encoding for optical security. Opt. Eng. **35,** 2464–2469.

Javidi B. and Ahouzi E.; 1998: Optical security system with Fourier plane encoding. Appl, Opt. **37,** 6247–6255.

Eskicioglu, A. M. and Delp, E. J.; 2001: *Overview of Multimedia Content Protection in Consumer Electronics Devices*, Signal Processing: Image Communication. 16(5), pp. 681-699.

Doerr, G. and Dugelay, J.-L.; 2003: *A Guide Tour of Video Watermarking*, Signal Processing: Image Communication. 18(4), pp. 262-382.

Wu, C.-P. and Kuo, C.-C. J.; 2001: *Efficient Multimedia Encryption via Entropy Codec Design*, Proceedings of SPIE Security and Watermarking of Multimedia Content III. San Jose, CA. Volume 4314.

Simmons, G. J.; 1990: *Prepositioned shared secret and/or shared control schemes*, Advances in Cryptology – EUROCRYPT '89 Proceedings. Springer-Verlag. Pp. 436-467.

Eskicioglu, A. M., Delp, E. J. and Eskicioglu, M. R.; 2003: *New Channels for Carrying Copyright and Usage Rights Data in Digital Multimedia Distribution*,

International Conference on Information Technology: Research and Education. Newark, NJ, August 10-13.

Eskicioglu, A. M. and Eskicioglu, M. R.; 2002: *Multicast Security Using Key Graphs and Secret Sharing*, Proceedings of the Joint International Conference on Wireless LANs and Home Networks (ICWLHN 2002) and Networking (ICN 2002). Atlanta, GA, pp. 228-241.

Leeper, D. G.; 2003: Wireless Data Technologies, John Wiley & Sons, Page 2-4. ISBN: 0-470-84949-5

Goudail F., Bollaro F., Javidi B., and Refregier P.; 1998: Influence of perturbation in a double phase-encoding system. J. Opt. Soc. Am. A 15, 2629-2638.

Refregier P. and Javidi B.; 1995: Optical image encryption using input plane and Fourier plane random encoding. Opt. Lett. 20, 767-769.

Rosen J., Javidi B.; 2001: Hiding images in Halftone Pictures. Appl. Opt. 40, 3346.

Goodman J.; 1996: Introduction to Fourier Optics. McGraw Hill, New York, 2nd ed.

Refregier P. and Javidi B.; 1995: Optical image encryption using input plane and Fourier plane random encoding. Opt. Let. 20, 767-769.

Javidi B., Sergent A., Zhang G., and Guibert L.; 1997: Fault tolerance properties of a double phase encoding encryption technique. Opt. Eng. 36(4), 992-998.

Javidi B., Zhang G., and Li J.; 1998: Encrypted optical memory using double-random phase encoding. Proc. SPIE 3384, 130-136.

Refregier P. and Javidi B.; 1995: Optical image encryption based on input plane and Fourier plane random encoding. Opt. Lett. 20(7), 767-769.

Javidi B.; 1999: Noise performance of double-phase encryption compared to XOR encryption. Opt. Eng. 38(1), 9-19.

Han J-W., Park C.S., Ryu D.H., and Kim E.S.; 1999: Optical image encryption based on XOR operations. Opt. Eng. 38(1), 47-54.

Khan A. and Nejib U.; 1987: Optical logic gates employing liquid crystal optical switches. Opt. Eng. 26(2), 270-273.

Schmalz M.; 1994: Optical and electro-optical architectures for the compression and encryption of discrete signals and imagery. Proc. SPIE 2238, 121-130.

Gilmore J.; 1998: *Cracking DES: Secrets of Encryption Research,* Wiretap Politics and Chip Design. New York.

Bellare M., Guerin R., and Rogaway P.; 1995: XOR: New methods for message authentication using finite pseudorandom functions. In Don Coppersmith, editor, Proc. CRYPTO 95, pages 15-28. Springer. Lecture Notes in Computer Science No. 963.

Sheng Zhen Jing Hua Displays Co., Ltd.; 2001: About LCD & LCM. http://www.china-lcd.com/.

Baier S.; 1993: CCD Imaging Systems. Phys. Today. Page 7.2-7.32.

Agi I. and Gong L.; 1996: *An Empirical Study of Secure MPEG Video Transmission.* Proceedings of the Internet Society Symposium on Network and Distributed System Security. San Diego, CA, February, pp. 137-144.

Qiao L. and Nahrstedt K.; 1998: *Comparison of MPEG Encryption Algorithms.* International Journal on Computer and Graphics, Special Issue on Data Security in Image Communication and Network. 22(3).

Rappaport T. S.; 1995: Wireless Communications: Principles and Practice. Prentice Hall, New Jersey.

IEEE 802.11; 1997: *LAN MAN Standards of the IEEE Computer Society.* Wireless LAN medium access control (MAC) and physical layer (PHY) specification. IEEE Standard 802.11, 1997 Edition.

Rappaport T. S.; 1995: Wireless Communications: Principles and Practice. Prentice Hall, New Jersey.

FELZER A.P.; 2004: 409 bandpass transmission and investigation 14 to quaternary phase shift keying. Part I.

Hsieh M.H. and Wei C.H.; 1999: A Low-Complexity Frame Synchronization and Frequency Offset Compesation Scheme for OFDM Systems over Fading Channels. In IEEE Transactions on Vehicular Technology, Vol. 48, No. 5. September 1999.

Nee R. and Prasad R.; 2000: OFDM for Wireless Multimedia Communication. Artech House Publishers.

Frank S.; 1998: Visual Cryptography Kit. University of Cambridge.

APPENDICES

Appendix 1. Software of Simulation Encryption And Decryption Image

This software is about encryption and decryption images in Matlab. The one is encryption and decryption still image. Simulating optical XOR in Matlab.

The processes: 1. Display image and convert to binary image, 2. Create random key-bits, the key-bit consist of 0,1, 3. Using XOR algorithm function encrypt binary image and random key-bit, 4. Operate decryption image and key-bit, 5. Get original image.

The second software is multi-frame encryption process. These processes simulate real time video of encryption. The process is similar with still images. Just separating every frame to encrypt with key-bit. In this example, five frames are used. You can get result in Matlab.

1. [NEWimageXOR.m] Simulate in Matlab

Software Function: 1. Convert image, 2. XOR encrypt image, 3. And XOR decrypt image.

```
RGB= imread('board.tif');     % 1. Display original image
imshow(RGB)
RGB1= imresize(RGB,1.5);
I= rgb2gray(RGB);
figure,imshow(I)
I1= imresize(I,1.5);
figure,imhist(I)              % 1.1 RGB convert to gray and display, and display gray-
level value
```

```
BW= im2bw(I,0.25);
figure,imshow(BW)
BW1= imresize(BW,1.5);

randn('state',0)          %2. Create key-bit, using random function
k= randn(648,306);
figure,imshow(k);         %2.2 Display key-bit image
k1= im2bw(k,0.25);
figure,imshow(k1);
k2= imresize(k1,1.5);

E= XOR(k1,BW);            %3. Encrypt image
figure,imshow(E);
E1= imresize(E,1.5);
figure,imhist(E)

D= XOR(k1,E);             %4. Decrypt image
figure,imshow(D);
D1= imresize(D,1.5);


figure,subplot(3,3,1);                    %5. Display all the images
subimage([0,611],[0,1295],RGB1);
title('Orignal Image(RGB)');
subplot(3,3,2);
subimage([0,611],[0,1295],I1);
title('Gray-level Image');
subplot(3,3,3);
subimage([0,611],[0,1295],BW1);
title('Binary Image');
subplot(3,3,4);
```

```
subimage([0,611],[0,1295],k2);
title('Key-bit(Binary matrix)');
subplot(3,3,5);
subimage([0,611],[0,1295],E1);
title('Encrypted Image');
subplot(3,3,6);
subimage([0,611],[0,1295],D1);
title('Decrypted Image');

display('Processing encryption image and decryption image!')
```

2. [movieXOR.m] Simulate in Matlab

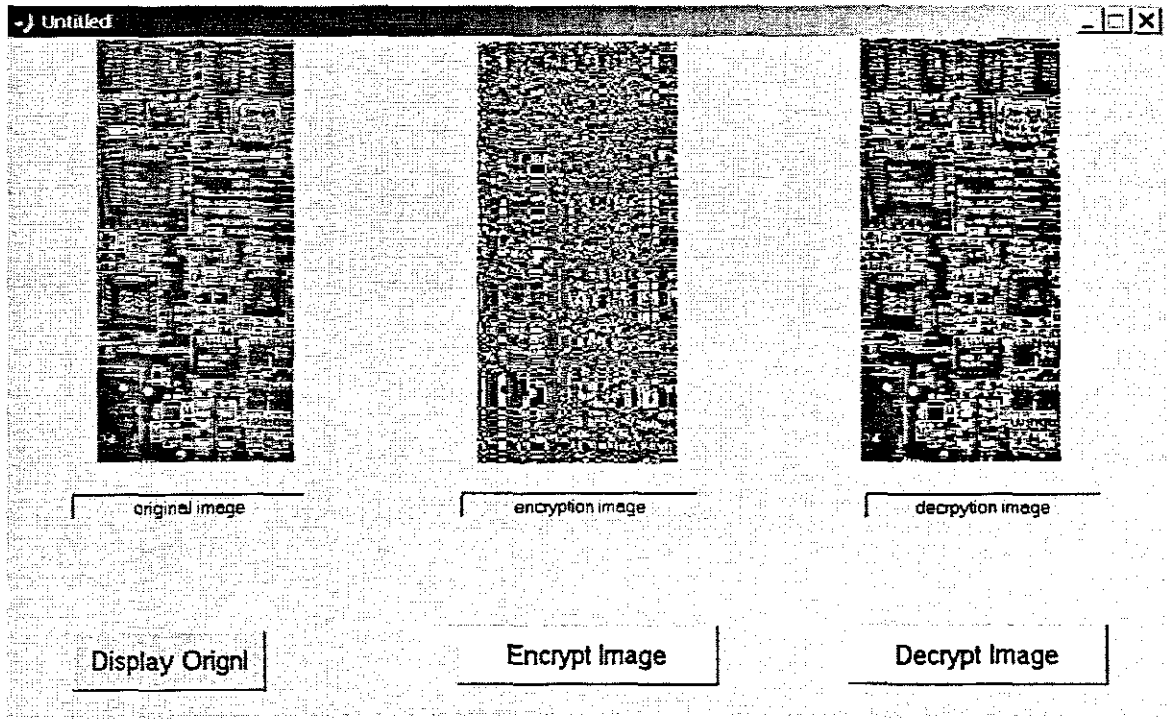
Function real-time Movie XOR

```
i0= imread('picture 19.jpg');  %1. Display color images (five frames)
i1= imread('picture 20.jpg');
i2= imread('picture 21.jpg');
i3= imread('picture 22.jpg');
i4= imread('picture 23.jpg');

g0= rgb2gray(i0);              %2. Convert to gray images
g1= rgb2gray(i1);
g2= rgb2gray(i2);
g3= rgb2gray(i3);
g4= rgb2gray(i4);

[j0,map]= gray2ind(g0,125);   %3. Convert to index images
[j1,map]= gray2ind(g1,125);
[j2,map]= gray2ind(g2,125);
[j3,map]= gray2ind(g3,125);
[j4,map]= gray2ind(g4,125);
```

```
t0= im2bw(i0,0.55);   %4. Convert to binary images
t1= im2bw(i1,0.55);
t2= im2bw(i2,0.55);
t3= im2bw(i3,0.55);
t4= im2bw(i4,0.55);

display('Right!')

M= cat(3,j0,j1,j2,j3,j4);        %5. Consist of five frames
s= size(M);

for i= 1:s(1,1)
   for j= 1:s(1,2)
      for k= 1:s(1,3)
         D(i,j,1,k)= M(i,j,k);
      end
   end
end

randn('state',0)                 %6. Create random ket-bits
m= randn(480,640);
m1= im2bw(m,0.08);

e0= XOR(t0,m1);                  %7. Encrypt every binary images and key-bits
e1= XOR(t1,m1);
e2= XOR(t2,m1);
e3= XOR(t3,m1);
e4= XOR(t4,m1);

d0= XOR(e0,m1);                  %8. Decrypt images
```

```
d1= XOR(e1,m1);
d2= XOR(e2,m1);
d3= XOR(e3,m1);
d4= XOR(e4,m1);

[a0,map1]= gray2ind(d0,125);     %9. Convert decryption images to index
[a1,map1]= gray2ind(d1,125);
[a2,map1]= gray2ind(d2,125);
[a3,map1]= gray2ind(d3,125);
[a4,map1]= gray2ind(d4,125);

M1= cat(3,a0,a1,a2,a3,a4);          %10. Consist of decryption images, and display
s1= size(M1);

for i9= 1:s1(1,1)
   for j9= 1:s1(1,2)
      for k9= 1:s1(1,3)
         D9(i9,j9,1,k9)= M1(i9,j9,k9);
      end
   end
end

figure,mov= immovie(D,map)
figure,imshow(e0)
figure,mov1= immovie(D9,map1)

display('OK!')
```

3. Control Windows



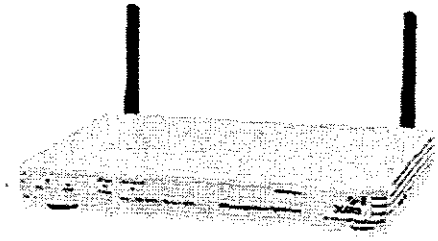

```
Function varargout = zyGui1(varargin)
% ZYGUI1 Application M-file for zyGui1.fig
%    FIG = ZYGUI1 launch zyGui1 GUI.
%    ZYGUI1('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 20-Nov-2003 07:30:12

if nargin == 0  % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % Use system color scheme for figure:
    set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));

    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);
```

```
    if nargout > 0
        varargout{1} = fig;
    end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK

    try
        if (nargout)
            [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
switchyard
        else
            feval(varargin{:}); % FEVAL switchyard
        end
    catch
        disp(lasterr);
    end

end


%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction prototypes to this file, and
%| sets objects' callback properties to call them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
```

```
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|
%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components in GUI using
%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA. A copy of the structure
%| is passed to each callback.  You can store additional information in
%| this structure at GUI startup, and you can change the structure
%| during callbacks.  Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.
% ------------------------------------------------------------
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)


disp('You are wolcome!');


 if get(handles.dis,'String') == 'Display'
   set (handles.dis,'String','Display Orignl');
```

```
else
   set (handles.dis,'String','Display');

    axes(handles.orig);
    imshow('board.tif');

end

function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)

disp('It is OK!');

if get(handles.endec,'String') == 'Encrypt Image'

    axes(handles.enc);
    imshow('zycode1.tif');

end

function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)

if get(handles.decry,'String') == 'Decrypt Image'

    axes(handles.dec);
    imshow('zydecode1.tif');

 end
```

Appendix 2. 3COM 11wbps Wireless LAN Accessing Point And PC Card



Product Specifications

• Users Supported: Up to 128 simultaneous users

• Standards Conformance: Wi-Fi 802.11b & WPA certified, IEEE 802.11b, IEEE 802.11g

• Data Rates: 802.11g: 54, 48, 36, 24, 18, 12, 9, 6 Mbps

802.11b: 11, 5.5, 2, 1 Mbps

• Frequency Band: 2.4 - 2.4835 GHz

• Operating Range: Indoor maximum: 100 meters (328 feet); Outdoor maximum: 457 meters (1,499 feet)

• Operating Channels: 5-7 (Israel); 10-13 (France, Jordan); 1-11 (U.S., Argentina, Brazil, Canada, Columbia, Mexico, Taiwan); 1-13 (elsewhere worldwide)

Receive Sensitivity: 802.11g

54 Mbps - 67.6 dBm

48 Mbps - 69.6 dBm

36 Mbps - 78.8 dBm

24 Mbps - 79.8 dBm

18 Mbps - 85.4 dBm

12 Mbps - 85.6 dBm

9 Mbps - 88.5 dBm

6 Mbps - 88.0 dBm

802.11b

11 Mbps - 82.8 dBm

5.5 Mbps - 78.8 dBm

2 Mbps - 89.9 dBm

1 Mbps - 89.9 dBm

• Wireless Medium: DSSS (Direct Sequence Spread Spectrum)

• Media Access Protocol: CSMA/CA

• Power: Supplied using external Office Connect power adapter: operating input voltage: 10-30V; operating frequency: 47-63Hz; maximum power consumption: 6.5W; maximum transmit power output: 17dBM

• Physical Ports: LAN: 1 10/100 Mbps Ethernet port

• Performance Features: Clear Channel Select, dynamic rate shifting, packet bursting

• Security: 256-bit WPA encryption 40-/64-bit and 128-bit WEP shared-key encryption

• Installation, Configuration & Management: Browser-based administration, wireless-installation and device-discovery wizards; pre-set defaults; save and restore configuration files

• LED Indicators: Power; LAN port status - Link, Speed and Activity; WLAN port status - Link, Activity; Alert/Diagnostics

• Regulatory Agency Approvals: Safety: UL 1950, EN 60950; CSA 22.7 #950, IEC 60950

Emissions: EN 55022 Class B, EN 55024, FCC Part 15, Class B, ICES-003 Class B, CNS 13438 Class A

Environmental: EN 60068 (IEC 68) UL Listed - CSA certified

• Environmental Operating Ranges: Operating temperature:

0° C to 40° C (32° to 105° F)

Non-operating temperature:

-40° to 70° C (-40° to 158° F)

Operating humidity:

0 to 90% non-condensing

• Dimensions & Weight: Height: 2.5 cm (1 in)

Width: 22 cm (8.7 in)

Depth: 13.5 cm (5.3 in)

Weight: 592 g (1.3 lb)

System Requirements

• Computer with an Ethernet 10BASE-T, 10/100, or 10/100/1000 interface configured for Internet communication

• Operating system that supports an Ethernet connection with an IP stack (Installation CD requires Windows 98, Windows ME, Windows XP, or Windows 2000)

• PCs and laptops require 802.11g or 802.11b adapter cards for wireless access

Package Contents

• Access point

• Power adapter

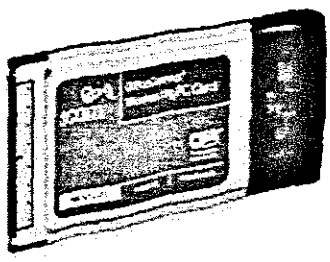
• Quick Start guide

• CD-ROM containing Access Point Discovery program and user guide

• One straight-through category 5 UTP patch cable

• Four rubber feet

3Com Connect Wireless 11g PC Card



Product Specifications

• Computer **Slot Type:** Type II or Type III 32-bit PC Card (3.3 V)

• Drivers **Supported:** NDIS 5: Me, 2000, 98 SE

NDIS 5.1: Windows XP

• Standards **Conformance:** Wi-Fi 802.11b & WPA certified, IEEE 802.11b, IEEE 802.11g

• Data **Rates:** 54, 48, 36, 24, 18, 12, 9, 6 Mbps (802.11g)

11, 5.5, 2, 1 Mbps (802.11b)

• Frequency **Band:** 2.4 - 2.4835 GHz

• Operating **Range:** Indoor maximum: 100 meters (328 feet); Outdoor maximum: 457 meters (1,499 feet)

• Operating **Channels:** 5-7 (Israel); 10-13 (France, Jordan); 1-11 (U.S., Argentina, Brazil, Canada, Columbia, Mexico, Taiwan); 1-13 (elsewhere worldwide)

• Receive **Sensitivity:** 802.11g

54 Mbps - 67.6 dBm

48 Mbps - 69.6 dBm

36 Mbps - 78.8 dBm

24 Mbps - 79.8 dBm

18 Mbps - 85.4 dBm

12 Mbps - 85.6 dBm

9 Mbps - 88.5 dBm

6 Mbps - 88.0 dBm

802.11b

11 Mbps - 82.8 dBm

5.5 Mbps - 78.8 dBm

2 Mbps - 89.9 dBm

1 Mbps - 89.9 dBm

• Wireless **Medium:** DSSS (Direct Sequence Spread Spectrum)

• Media **Access Protocol:** CSMA/CA

• Reliability **Features:** Dynamic rate shifting

• Performance **Features:** Packet bursting, compression, concatenation, piggyback acknowledge, PRISM Nitro Directlink

• Security: 256-bit WPA encryption

40/64-bit and 128-bit WEP shared-key encryption

• Management: Wireless Card Manager, pre-set defaults

• LED **Indicators:** Link; Activity

• Regulatory/**Agency Approvals:** Safety: UL/CSA 60950, EN 60950

Emissions: FCC Part 15.247, RSS-210, EN 300 328-2, FCC Part 15 Subpart B,

(SAR) FCC OET Bulletin 65, RSS-102, prEN 50371

Environmental: EN 301 489-17

• Operating **Voltage:** 3.0V - 3.6V

• Maximum **Transmit Power Output:** 17dBM

• **Environmental Operating Ranges:** Temperature: 0° C to 50° C (32° to 122° F)

Humidity: 0 to 90% non-condensing

• **Dimensions:** Height: 11.3 cm (3.9 in)

Width: 5.4 cm (2.8 in)

Depth: 0.5 mm (0.2 in)

System Requirements

• Notebook PC with an available Type II or III 32-bit PC Card slot (3.3V)

• Notebook PC must be running Windows XP/Me/2000/98 SE

Appendix 3. Features of digital camera 101

The Digimax 101 is a product of Samsung Company. The digital camera has the following features.

- A fine resolution of 1.3 mega pixels
- 2 X digital zoom lens
- Movie clip recording function
- Date imprinting on a still image
- Multiple language support

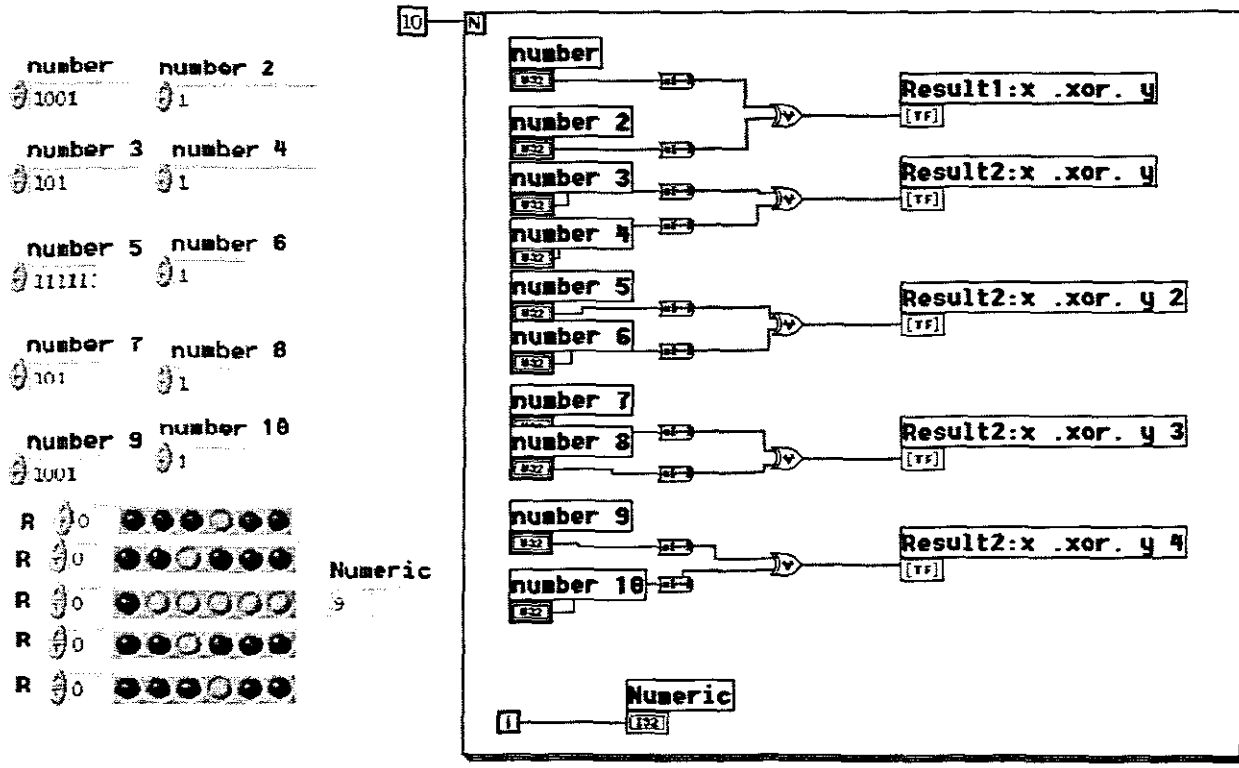
Appendix 4. Simulating XOR model in LabView

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where the flow of data determines execution.

In LabVIEW, you build a user interface by using a set of tools and objects. The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

You can purchase several add-on software toolsets for developing specialized applications. All the toolsets integrate seamlessly in LabVIEW. Refer to the National Instruments Web site at www.ni.com for more information about these toolsets.

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources and displays that information or moves it to other files or other computers. The below is a VI about XOR operation of two series binary number.



Appendix 5. Process of Setting up Optical Encryption System

A digital camera (The Samsung Digimax 101 1.3 Mega Pixels) was chose to perform optical XOR operation. This digital camera contains one 1.6" color TFT LCD (640x512 pixels). The two LCDs are used to perform optical XOR function.

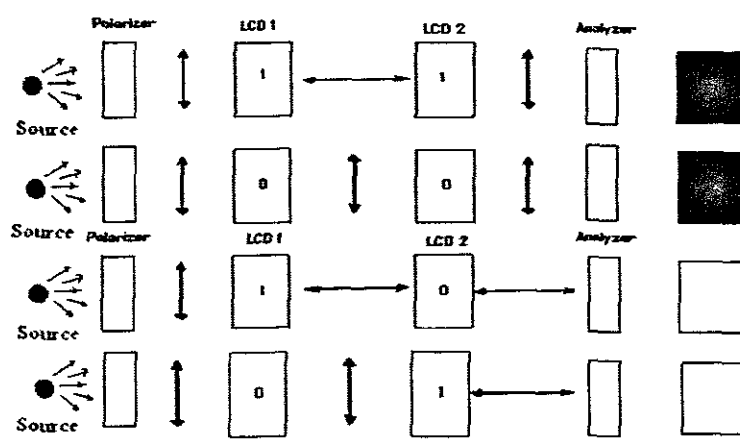


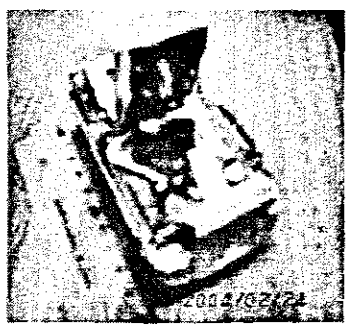
Figure 3-5 XOR optical logic operation using two LCDs

Figure 1 show the XOR operation. Two polarizers and two LCDs perform XOR function, where the polarizer, two LCDs, and analyzer are arranged in series.

Assembly Work:

1. Using screwdriver to open the back cover of camera. You can find 1.6" LCD connecting integrated circuit through the data wire, near by controlling plate, which control all kind of function of camera.

2. Levering LCD from the foundation, putting it uprightness with camera.



3. Taking out the background light using soldering iron, and the back polarizer of LCD 2 (remain the forward polarizer).

4. Performing the same processing to open another camera, but taking out the forward polarizer of LCD 1 (remain the back polarizer).

5. Putting the two cameras with face-to-face and parallel. As possible as fixing close the forward of LCD 1 with the back of LCD2.

6. Riveting the two cameras on the flat wood plate.

Performing XOR Function

7. Putting original binary image into the camera 1 through the USB cable, the image is displayed on the LCD 1.

8. Putting key bit stream into the camera 2 through the USB cable, the key bit is displayed on the LCD2.

9. Polarize light is detected by CCD and displaying on the computer, which is encryption image.

10. Saving the encryption image, and sending it through wireless link to Internet (receiver). On the receiver, you can reconvert original image after using the same key bit XOR the encrypted image. This is shown Figure 1-1).