Secret Sharing Schemes and its Applications

Thesis submitted to

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

under the Faculty of Technology

by

Deepika M P
Register No:4132

Under the guidance of
Dr. A. Sreekumar

Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.

March 2019

Secret Sharing Schemes and its Applications

Ph.D. thesis

Author:
Deepika M P
Research Scholar
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.
Email: deepika.it@adishankara.ac.in

Research Advisor:
Dr. A. Sreekumar
Professor
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.
Email: askcusat@gmail.com

Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.

March 2019

To My Dear Teachers

&

Loving Family

Dr. A. Sreekumar
Professor
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, India.

6^{th} March 2019

Certificate

Certified that the work presented in this thesis entitled "Secret Sharing Schemes and its Applications" is based on the authentic record of research carried out by Smt. Deepika M P under my guidance in the Department of Computer Applications, Cochin University of Science and Technology, Kochi-682 022 and has not been included in any other thesis submitted for the award of any degree.

A. Sreekumar
(Supervising Guide)

Phone : +91 484 2577602 +91 484 2556057 Email: askcusat@gmail.com

Dr. A. Sreekumar
Professor
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, India.

6^{th} March 2019

Certificate

Certified that the work presented in this thesis entitled "Secret Sharing Schemes and its Applications" submitted to Cochin University of Science and Technology by Smt. Deepika M P for the award of degree of Doctor of Philosophy under the Faculty of Technology, contains all the relevant corrections and modifications suggested by the audience during the pre-synopsis seminar and recommended by the Doctoral Committee.

A. Sreekumar
(Supervising Guide)

Phone : +91 484 2577602 +91 484 2556057 Email: askcusat@gmail.com

Declaration

I hereby declare that the work presented in this thesis entitled "Secret Sharing Schemes and its Applications" is based on the original research work carried out by me under the supervision and guidance of Dr. A. Sreekumar, Professor, Department of Computer Applications, Cochin University of Science and Technology, Kochi–682 022 and has not been included in any other thesis submitted previously for the award of any degree.

Deepika M P

Kochi– 682 022
6^{th} March 2019

Acknowledgment

The success and final outcome of my thesis required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my thesis. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I owe my deep gratitude to my supervisor Dr. A. Sreekumar, who introduced me to the world of 'Visual Cryptography'. I consider myself very fortunate to have him as my advisor. I have learned a lot from him over the past few years. The foundation stones of my dream to get doctorate are laid by him only. During my M.Tech Thesis, he is the person who introduced me to Visual Cryptography. The time spent with him and the conversation we had really boosted my spirits and ignited the passion of doing something in Secret Sharing. Without his guidance and constant feedback my research work would not have been achievable.

I am greatly thankful to Dr. B. Kannan, Professor and Head, Department of Computer Applications, CUSAT, for the motivation, support and guidance. I would also like to thank him for being very patient with me and for having faith in me. Furthermore, I thank him for providing an excellent research atmosphere at the Research Lab.

I sincerely thank Dr. K. V. Pramod, Eminent Professor, Department of Computer Applications, CUSAT, for his insightful suggestions, fruitful discussions and critical remarks. His dedication and energy are infectious. His jovial and affectionate nature is memorable.

I also would like to acknowledge the two backbones of the department Dr. M. Jathavedan and Prof. S. Malathi for the constant encouragement and support.

I would like to express my gratitude to Dr. G. Santhosh Kumar, Professor and Head, Department of Computer Science, CUSAT, because he is the person who introduced me to Dr. A Sreekumar during my

M.Tech Thesis work and there after I got the opportunity to carry out my research under the supervision of Dr. A sreekumar. I greatly appreciate the support received from him.

I am also thankful to all other faculty members, office staff, librarian and non teaching staff of Department of Computer Applications, who helped me during various stages of my research.

It has been a delight working in Research Lab of Department of Computer Applications. Thanks to the lively ambiance maintained by the past and present students of the lab. I enjoyed every bit of my life in this lab which have created many great researchers in the past. The lab staffs were very cooperative. They have extended their helping hand in many occasions.

Several other people from whom I draw lot of inspirations are Prof. S. G. Iyer, (Former Principal, Adi Shankara Institute of Engineering and Technology), Dr. Dorairangaswamy(Prinicpal, Adi Shankara Institute of Engineering and Technology, Prof. R. Rajaram, Dr. Abraham Varghese, Dr. Murali Parameswaran and all my CSE and IT family of Adi Shankara Institute of Engineering and Technology. I am very grateful to all of you.

I also acknowledge the tremendous support of my family during this endeavor. My parents M. P. Neelakandan and M. P. Radha, who have remarkable influence on shaping my career and believing in me and encouraging me to follow my dreams.

And finally to Pradeep, who has been by my side throughout my work, living every single minute of it, and without whom, I would not have had the courage to embark on this journey in the first place. And to my little stars Hari and Shiva for being such good cute super heros, and making it possible for me to complete what I started.

Above all, I thank the supreme power who created the universe and gave the mankind the supreme knowledge.

Deepika M P

Preface

Nowadays all of us are connected to a public network. In this situation instead of owner's machine, the data are usually stored on service provider's servers. The problem with this scenario is security. Anyone can steal important data of anyone or even other organizations. The traditional way to protect secret information is to use conventional encryption mechanisms. But it is not sure that the encrypted information will not be corrupted or the secret key will not be lost. We can't imagine the situation like that; when the encrypted information is corrupted or when the secret key is lost. Here the reliability of the information get compromised. Means, there is only security, no reliability. This problem is addressed by *secret sharing* and finds the solution for both security and reliability. Here the valuable data is distributed and stored at several places instead of keeping in a single place. When it is needed the secret can be constructed from the shares. The original motivation of secret sharing was to safeguard cryptographic keys from loss. The loss of a cryptographic key is equivalent to data loss as we cannot retrieve the original data back with out the encryption key. It is desirable to create backup copies of important keys but greater the number of copies made greater the risk. Secret sharing provides an efficient solution to this

problem by protecting important information being lost, modified, destroyed or getting into wrong hands.

The idea of *secret sharing scheme* is to start with a secret and divide it into pieces called *shares* or *shadows*, which are distributed amongst users such that the pooled shares of designated subsets of users allow reconstruction of the original secret.

A particularly interesting class of secret sharing schemes is *threshold scheme* for which the designated sets consist of all set of t or more participants. Such schemes are called t out of n *threshold schemes* or simply (t, n) schemes, where n is the total number of participants. Another class in which any authorized subset of participants can collate and access the secret data are called *generalized secret sharing* schemes.

This dissertation deals with the development of secret sharing schemes and its applications. As we all know there are some good schemes available for secret sharing from different authors. The main motivation of this dissertation are the below listed questions.

1. Does there exist any scheme for the secret sharing that can be used in both the situations (that is, for secret sharing among the participants and secret encryption) and provide the same level of security or better security when compared with already existing and using secret sharing/secret encryptions schemes?

2. Does there exist any scheme that can be used for the secret, regardless of its form, like digit, text, image etc.?

3. Does there exist any common cheater identification mechanism that can be used in combination with any secret sharing schemes?

4. Does there exist any scheme in which the shares of the key can be used for decrypting the secret information, instead of the key that is used in encryption mechanism?

So we have developed main three schemes for secret sharing. That can be used irrespective of the form of the secret. That means we can use the scheme on text data as well as image data. Among the three scheme one of the scheme can be used in both the situation, that is as secret sharing among the participants and as the secret encryption mechanism. As we mentioned at the beginning the original motivation of secret sharing was to safeguard cryptographic keys from loss. We have mentioned one application of the developed secret sharing scheme in broadcast encryption. In broad cast encryption the distribution of the key is one of the vital part. Here we have proposed a method to distribute the shares of the key, instead of the master key, among the communicating parties. And when we talk about security there will be attack. Here in secret sharing there is a chance to cheat in the phase of reconstruction of the secret. The cheater Identification and prevention are another research topic in this area. So we have considered a cheater identification mechanism that can be attached with any existing secret sharing schemes. The schemes which are developed are listed below.

1. Secret Sharing Scheme using Gray Code and XOR operation.
2. Visual Cryptography using Gray code and XOR operation.
3. Visual Cryptography using CRT and POB number system.
4. Visual Cryptography using Polynomial Interpolation.
5. Cheater Identification using SHA algorithm.
6. Key Distribution in Broadcast Encryption using Polynomial Interpolation.

In brief, our work in this thesis has made significant advancement in the state-of-the-art research on secret sharing irrespective of the form of the secret.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Secret Sharing Schemes

Computers and the networks of computers, with the intention of sharing, storing, and processing the information, have become less expensive and used widespread during these days. Hence the information in digital form is very common now and handling secret information has been a prominence issue. Sometime the secret is thought to be secure in a single hand and at other times it is thought to be secure when shared in many hands. In this dissertation the second case is considered, the secret sharing among the participants who involved in the communication. Along with the schemes for the secret sharing (including visual secret sharing), its applications are also considered for the study.

Secret sharing(also called **secret splitting**)refers to methods for distributing a secret amongst a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number, of possibly different types, of shares are combined together; individual shares are of no use on their own. A secret-sharing scheme is a method by which a dealer distributes shares to parties such

that only authorized subsets of parties can reconstruct the secret and unauthorized subsets of parties get no information whatsoever about the secret.

Secret sharing schemes are ideal for storing information that is highly sensitive and highly important. Examples include: encryption keys, missile launch codes, and numbered bank accounts. Each of these pieces of information must be kept highly confidential, as their exposure could be disastrous; however, it is also critical that they should not be lost. Traditional methods for encryption are ill-suited for simultaneously achieving high levels of confidentiality and reliability. This is because when storing the encryption key, one must choose between keeping a single copy of the key in one location for maximum secrecy, or keeping multiple copies of the key in different locations for greater reliability. Increasing reliability of the key by storing multiple copies lowers confidentiality by creating additional attack vectors; there are more opportunities for a copy to fall into the wrong hands. Secret sharing schemes address this problem, and allow arbitrarily high levels of confidentiality and reliability to be achieved. Secret sharing schemes are important in cloud computing environments as well. Thus a key can be distributed over many servers by a threshold secret sharing mechanism. The key is then reconstructed when needed. Secret sharing has also been suggested for sensor networks where the links are liable to be tapped by sending the data in shares which makes the task of the eavesdropper harder. The security in such environments can be made greater by continuous changing of the way the shares are constructed.

Secret sharing schemes are important tools in cryptography. Initially the secret sharing schemes are developed as a method to safe guard the cryptographic keys. Later it has found useful applications in several cryptographic protocols. One of the most important research area where the secret sharing schemes have found useful applications is secret image

sharing. Secure storage and transmission of confidential images like medical images can be done without using encryptions by secret sharing technique. Secure key distribution, implementation of effective access control mechanism, threshold encryption decryption, threshold signature generation, broadcast encryption are all major areas where the secret sharing schemes are used as the basic techniques or basic building blocks.

In the secret sharing scheme model there will be one dealer and n players. The dealer gives a share of the secret to the players, but only when specific conditions are fulfilled will the players be able to reconstruct the secret from their shares. The dealer accomplishes this by giving each player a share in such a way that any predefined group of t (for threshold) or more players can together reconstruct the secret but no group of fewer than t players can. Such a system is called a (t, n)-threshold scheme (sometimes it is written as an (n, t)-threshold scheme). A secure secret sharing scheme distributes shares so that anyone with fewer than t shares has no extra information about the secret than someone with 0 shares.

Consider for example the secret sharing scheme in which the secret phrase "$password$ "is divided into the shares "$pa------$ ", "$--ss----$ ", "$----wo--$ "and "$------rd$ ". A person with 0 shares knows only that the password consists of eight letters. He would have to guess the password from $26^8 = 208$ billion possible combinations. A person with one share, however, would have to guess only the six letters, from $26^6 =$ 308 million combinations, and so on as more persons collude. Consequently this system is not a "secure" secret sharing scheme, because a player with fewer than t secret-shares is able to reduce the problem of obtaining the inner secret without first needing to obtain all of the necessary shares.

In contrast, consider the secret sharing scheme where $\mathbf{X}$ is the secret to be shared, P_i are public encryption keys and Q_i their corresponding private keys. Each player J is provided with $P_1(P_2(...(P_N(X), Q_j)))$. In this scheme, any player with private key 1 can remove the outer layer of encryption; a

player with keys 1 and 2 can remove the first and second layer, and so on. A player with fewer than N keys can never fully reach the secret X without first needing to decrypt a public-key-encrypted blob for which he does not have the corresponding private key - a problem that is currently believed to be computationally infeasible. Additionally we can see that any user with all N private keys is able to decrypt all of the outer layers to obtain X, the secret, and consequently this system is a secure secret distribution system. In short there are two kinds of secret sharing schemes; one is secure secret sharing scheme and the other one is unsecured secret sharing scheme.

In this chapter we start with the motivation behind this work and the problem definition. The history of basic secret sharing schemes are reviewed then, which helps in understanding the core concept and developments in this area of study. We then emphasize on major contributions of the dissertation followed by the list of publications as part of this work are noted. Lastly, we describe the chapter wise organization of this dissertation.

1.1 Motivation

In[Liu68], Liu considered the following combinatorial problem :

Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened, if and only if, six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry? If we consider any five scientists together, there is a special lock, which they cannot open. Consider particular scientist, he must have the keys of those locks which cannot be opened by any five scientists from among the other ten scientists.

Among eleven scientists, five scientists can be selected in ${}^{11}\mathbf{C}_5 = 462$ ways and among ten scientists, five scientists can be selected in ${}^{10}\mathbf{C}_5 = 252$ways. So, the minimal solution uses 462 locks and 252 keys per

scientist. These numbers are clearly impractical, and they become exponentially worse when the number of scientists increases. Moreover, the secret documents are always as a single entity and are not being involved in the method. Since the secret is always in one piece, the level of security is low to that extent. The security in this case is solely depending on the locks and the keys. However, the cabinet with the document as a whole is at great risk. It is not hard to show that the minimal solution uses 462 locks and 252 keys per scientist. These numbers are clearly impractical, and that become exponentially worse when the number of scientists increases.

The motivation behind the development of secret sharing scheme is to share a secret among n participants in such a way that t or more of them (where $t \leq n$) can join together to retrieve the secret. The concept of secret sharing was independently introduced by Shamir [Sha79] and Blakley [Bla79] in 1979.

In [Sha79] Shamir has generalized the problem to one in which the secret is some data D (example, the safe combination) and the goal is to divide D into n pieces $D_1, D_2, \ldots, D_n$ in such a way that:

- Knowledge of any k or more D_i pieces makes D easily computable.
- Knowledge of any $k-1$ or fewer D_i pieces leaves D completely undetermined.

Such scheme is called a (k, n) threshold scheme. It is shown in *Figure* 1.1.

After this, lots of studies were carried out in this area. One of the most important research is visual secret sharing schemes. It is also known as Visual cryptography. Visual cryptography is paradigm of cryptography which allows visual information (e.g. images, printed text and handwritten notes) to be encrypted in such a way that its decryption can

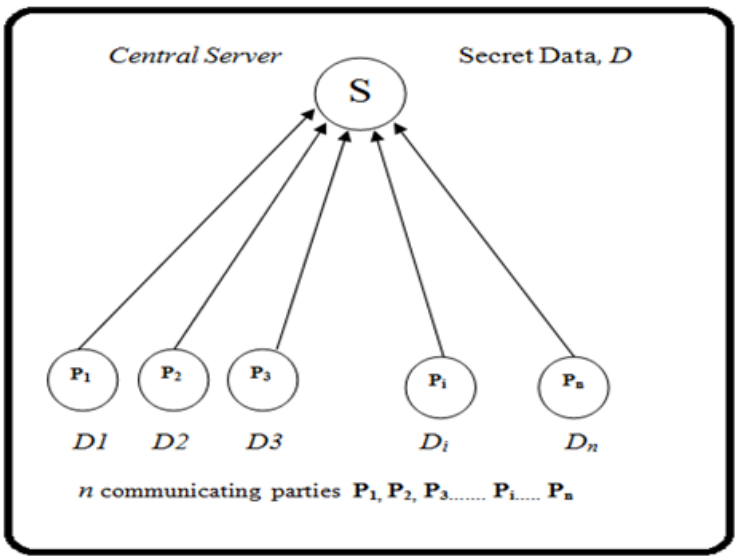


Figure 1.1: Distribution of secret data D among multiple parties

be done by the human eye, without the aid of computers. It avoids the need of complex mathematical computations during decryption and the secret image can be reconstructed using stacking (OR operation). There are diverse visual cryptography schemes based on the factors such as pixel expansion, contrast, security, meaningless or meaningful shares, type of secret image (either binary or color) and the number of secret images encrypted (single or multiple secret) etc. The following are the diverse visual cryptography schemes:

1. Traditional Visual Cryptography
2. Extended Visual Cryptography
3. Halftone Visual Cryptography
4. Recursive Threshold Visual Cryptography Scheme

5. Random Grids based Visual Cryptography
6. Color Visual Cryptography Schemes
7. Probabilistic Visual Cryptography
8. Region Incrementing Visual Cryptography
9. Progressive Visual Cryptography
10. Segment based Visual Cryptography Scheme
11. Cheating Immune Visual Cryptography Schemes
12. Size Invariant Visual Cryptography
13. User-friendly Visual Secret sharing scheme
14. Dynamic Visual Cryptography
15. OR and XOR Visual Cryptography

In visual cryptography(VC), all participants who hold shares are assumed to be honest, that is, they will not present false or fake shares during the phase of recovering the secret image. Thus, the image shown on the stacking of shares is considered as the real secret image. As we know where security is enforcing, there will be some way to exploit the system. So in this case the dishonest participant, also known as *cheater*, presents some fake shares such that the stacking of fake and genuine shares together reveals a fake image, there by (s)he may reconstruct the original one and can use the same for any illegal purpose.

So by considering all, related to the area secret sharing, in this dissertation we have considered the following questions:

1. Does there exist any scheme for the secret sharing that can be used in both the situations (that is, for secret sharing among the participants and secret encryption) and provide the same level of security or better security when compared with already existing and using secret sharing/secret encryptions schemes?

2. Does there exist any scheme that can be used for the secret, regardless of its form, like digit, text, image etc.?

3. Does there exist any common cheater identification mechanism that can be used in combination with any secret sharing schemes.?

4. Can we use the shares of the key for decrypting the secret information, instead of the key that is used in encryption mechanism.?

1.2 History of Secret Sharing

According to Time Magazine, May 4, 1992, control of nuclear weapons in Russia involves a two-out-of-three mechanism. In order to launch a nuclear missile, the cooperation of at least two parties out of three is needed. The three parties involved are the president, the Defense Minister, and the Defense Ministry. A similar situation can occur in a bank, where there is a vault which must be opened every day. The bank employs a number of senior tellers, who are trusted enough to participate in the opening of the vault, but not trusted to the tent that they themselves own the combination to the vault. We would like to design a system where for example any two of the senior tellers together can open the vault, but no individual alone can do so.

The above problems can be solved by means of a secret sharing scheme. Secret sharing schemes were independently introduced by Shamir [Sha79] and Blakley [Bla79] in 1979. Threshold secret sharing scheme is the most

popular scheme since 1979. The main schemes in the threshold schemes are discussed below.

1.2.1 Shamir 's Scheme

Secret sharing was invented independently by Adi Shamir [Sha79] and George Blakley [Bla79] in 1979. In [Sha79] Adi Shamir generalized the problem of eleven scientists who are working on a secret projects and proposed a software solution. The particular problem is described in [Liu68] by Liu. The problem is:

-Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present, what is the smallest number or locks needed? What is the smallest number of keys to the locks each scientist must carry? The solution is:

-462 locks and 252 keys per scientist.

These numbers are clearly impractical, and it becomes worst when the number of scientist increases.

In [Sha79] the author shows how to divide data D into n pieces in such a way that D is easily reconstructible from any k pieces, but even complete knowledge of $k-1$ pieces reveals absolutely no information about D.His goal is to divide D into n pieces $D_1, D_2, \ldots, D_n$ in such a way that:

- Knowledge of any k or more D_i pieces makes D easily computable.
- Knowledge of any $k-1$ or fewer D_i pieces leaves D completely undetermined(in the sense that all its possible values are equally likely).

Such a scheme is called a (k, n) threshold scheme.

Shamir's scheme is based on polynomial's interpolation.

Polynomial 's Interpolation

Given k points in the 2 dimensional plane $(x_1, y_1) \ldots (x_k, y_k)$ with distinct x_i's, there is one and only one polynomial $p(x)$ of degree $k-1$ such that $p(x_i) = y_i$ for all i. Without loss of generality, we can assume that the data D is a (can be interpreted as) number. To divide it into pieces D_i, we pick a random $k-1$ degree polynomial $p(x) = a_0 + a_1 x + ... + a_k x^{k-1}$ in which $a_0 = D$ and evaluate:

$D_1 = p(1), \ldots, D_i = p(i), \ldots, D_n = p(n)$

Given any subset of these D_i values (together with their identifying indices), we can find the coefficients of $p(x)$ by interpolation, and then evaluate $D = p(0)$. Knowledge of just $k-1$ of these values, on the other hand, does not suffice in order to calculate D.

That is, it takes k points to define a polynomial of degree $k-1$. The method is to create a polynomial of degree $k-1$ with the secret as the first coefficient and the remaining coefficients picked at random. Next find n points on the curve and give one to each of the participants. When at least k out of the n players reveals their points, there is sufficient information to fit a $(k-1)$ th degree polynomial to them, the first coefficient being the secret. Some of the useful properties of this (k, n) threshold scheme (when compared to the mechanical locks and keys solutions) are:

- The size of each piece does not exceed the size of the original data.
- When k is kept fixed, D_i pieces can be dynamically added or deleted (e.g., when executives join or leave the company) without affecting the other D_i pieces. (A piece is deleted only when a leaving executive makes it completely inaccessible, even to himself.)
- It is easy to change the D_i pieces without changing the original data D–all we need is a new polynomial $p(x)$ with the same free term. A frequent change of this type can greatly enhance security since the

pieces exposed by security breaches cannot be accumulated unless all of them are values of the same edition of the $p(x)$ polynomial.

- By using tuples of polynomial values as D_i pieces, we can get a hierarchical scheme in which the number of pieces needed to determine D depends on their importance. For example, if we give the company's president three values of $p(x)$, each vice-president two values of $p(x)$, and each executive one value of $p(x)$, then a $(3, n)$ threshold scheme enables checks to be signed either by any three executives, or by any two executives one of whom is a vice-president, or by the president alone.

Such a scheme is called a (k, n) threshold scheme. Efficient threshold schemes can be very helpful in the management of cryptographic keys. In order to protect data we can encrypt it, but in order to protect the encryption key we need a different method (further encryptions change the problem rather than solve it). The most secure key management scheme keeps the key in a single, well-guarded location (a computer, a human brain, or a safe). This scheme is highly unreliable since a single misfortune (a computer breakdown, sudden death, or sabotage) can make the information inaccessible. An obvious solution is to store multiple copies of the key at different locations, but this increases the danger of security breaches (computer penetration betrayal, or human errors). By using a (k, n) threshold scheme with $n = 2k - 1$ we get a very robust key management scheme. We can recover the original key even when $\lceil n/2 \rceil = k - 1$ of the n pieces are destroyed, but our opponents cannot reconstruct the key even when security breaches expose $\lceil n/2 \rceil = k - 1$ of the remaining k pieces. Threshold schemes are ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate. Ideally we would like the cooperation to be based on mutual consent, but the veto power this mechanism gives to

each member can paralyze the activities of the group. By properly choosing the parameters k and n, we can give any sufficiently large majority the authority to take some action while giving any sufficiently large minority the power to block it.

1.2.2 Blakley 's Scheme

It is based on hyper plane geometry. It uses principles of geometry to share the secret. It is a threshold secret sharing scheme. In threshold secret sharing scheme, the secret S is divided amongst a group of participants in such a way that for a specified t(where $1 < t < n$):

- Knowledge of any t or more shares make s computable.
- Knowledge of any $t - 1$ or fewer shares leaves S completely undetermined

To implement a (t, n) threshold scheme, each of the n users is given a hyper-plane equation in a t dimensional space over a finite field such that each hyper plane passes through a certain point. The intersection point of the hyper planes is the secret. When t users come together, they can solve the system of equations to find the secret. The secret is a point in a t dimensional space and n shares are affine hyper planes that pass through this point. An affine hyper plane in a t-dimensional space with coordinates in a field F can be described by a linear equation of the following form:

$c_1x_1 + c_2x_2 + \cdots + c_tx_t = y$

Blakley's secret sharing scheme can be represented as a linear system $Cx \pmod p = y$. where the matrix C and the vector y are obtained from the hyper plane equations. Reconstruction of original secret is simply finding the solution of a linear system of equations. The intersection point is obtained by finding the inter-section of any t of these hyper planes. The

secret can be any of the coordinates of the intersection point or any function of the coordinates.

Blakley's scheme is less space-efficient than Shamir's; while Shamir's shares are each only as large as the original secret, Blakley's shares are t times larger, where t is the threshold number of players. Blakley's scheme can be tightened by adding restrictions on which planes are usable as shares. The resulting scheme is equivalent to Shamir's polynomial system.

1.2.3 Li Bai's Scheme

Li Bai developed a threshold secret sharing based upon the invariance property of matrix projection. The scheme is divided in two phases:

- Construction of Secret Shares from Secret Matrix S
- Secret Reconstruction

The first phase, the construction of secret shares from secret matrix S is as follows:

1. Construct a random $m \times k$ matrix A of rank k where $m > 2k - 3$
2. Choose n linearly independent $k \times 1$ random vectors x_i
3. Calculate share $v_i = (A \times x_i) \pmod{p}$ for $1 \leq i \leq n$, where p is a prime number.
4. Compute $\$ = (A(A'A) - 1A') \bmod p$.
5. Solve $R = (S - \$) \pmod{p}$.
6. Destroy matrix A, x_i, $\$$, S.
7. Distribute n shares v_i to n participants and make matrix R publicly known.

The second phase, the secret reconstruction is as follows:

1. Collect k shares from any k participants, say the shares are $v_1, v_2, \ldots, v_k$ and construct a matrix $B = v_1 v_2 \ldots v_k$.
2. Calculate the projection matrix $\$ = (B(B'B) - 1B') \pmod{p}$.
3. Compute the secret $S = (\$ + R \pmod{p})$.

1.3 Thesis Contribution

The major contribution of this dissertation is in the development of secret sharing schemes and also exploring the use of it in typical application areas. This section is devoted to mention various contributions made by us in both the area of secret sharing and visual cryptography.

- We considered the previous research articles and schemes related to secret sharing and visual cryptography for the study and developed a secret sharing scheme using Gray Code and XOR operation that can be applicable for both data sharing and image sharing. Gray Code is also known as reflected binary code (RBC). It is termed after Franky Gray, who was a physicist and researcher at bell lab. It is a binary numeral system often used in electronics, but with many applications in mathematics. In Gray Code the two successive values differ in only one bit. The scheme which is developed using Gray Code and XOR operation, the secret is shared using the concept Gray Code and the secret is reconstructed using the XOR operation. The use of the scheme in the area of cryptography or secret(text, image) encryption is also explored.
- A specially designed number system called POB (Permutation Ordered Binary) system developed by Sreekumar et al [SS09] is

studied. We used the POB number system in visual secret sharing along with Chinese Remainder Theorem.

- Polynomial Interpolation is studied and we used Newton's polynomial interpolation along with Mod operator in sharing visual secrets. The only requirement of this scheme is, the image should be in PNG format.

- Major contribution of the dissertation is in the application of the secret sharing scheme using Lagrange polynomial interpolation in the area broadcast encryption. Here we have used the secret sharing scheme to share the master key and distributing the shares of the master key instead of sharing the key as such among the users.

- we have also considered a cheater identifications scheme in visual secret sharing schemes using Secure Hash Algorithm, that can be used along with any of the already existing schemes. The only limitation in this case is, the image should be in PNG format.

1.4 List of Publications

As part of the research work various papers were presented and published in peer reviewed International Journals as well as in Conference proceedings.

They are listed below:

1. Deepika M P, Dr. A Sreekumar, "Key Distribution Scheme in Broadcast Encryption Using Polynomial Interpolation", International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 15475-15483 Research India Publications. http://www.ripublication.com.

2. Deepika M P, Dr. A Sreekumar, "Visual Cryptography Scheme Using Gray Code and XOR Operation", International journal of current engineering and scientific research(IJCESR), (ISSN 23938374) Print, (ISSN 2394-0697) online. Vol. 4 Issue 9.TRO Publication. September 2017.

3. Deepika M P, Dr. A Sreekumar, "Visual Secret Sharing using Newton Interpolation Polynomial and Mod operator with PNG Images", IEEE 4th International Conference on Innovation in Information, Embedded and Communication Systems (ICIIECS17). March 17 -18 2017. DOI:10.1109/ICIIECS.2017.8275917 .IEEE Xplore.

4. Deepika M P, Dr. A Sreekumar, "Secret sharing scheme using Gray code and XOR operation", 2017 Second IEEE International Conference on Electrical, Computer and Communication Technologies (IEEE ICECCT 2017).Feb 22-24 2017. DOI: 10.1109/ICECCT. 2017.8117932. IEEE Xplore.

5. Deepika M P, Dr. A Sreekumar, "A Novel secret sharing scheme using POB number system and CRT", International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 11, Number 3 (2016) pp 2049-2054 Research India Publications. http://www.ripublication.com.

6. Deepika M P, Dr. A Sreekumar, "Cheater identification in Visual secret sharing schemes using SHA Algorithm and Alpha channel", International Journal of Computer Applications Technology and Research, Volume 4 Issue 11, 838 - 845, 2015, ISSN:- 23198656.

1.5 Organization of Thesis

The work mainly aims to develop secret sharing schemes , by considering all the questions mentioned in the motivation section of this chapter. An important application of the scheme is also mentioned in this dissertation. We have also tried to give a good stuff of literature survey so that a fresh researcher will get motivated to this area.

The thesis is organized into 10 main chapters and summarized with chapter 11 as concluding chapter. In Chapter 1, we give a brief introduction to the area secret sharing, motivation behind this work, a quick survey on the topic secret sharing, and the thesis contribution and the organization of the dissertation. From this chapter the reader will get an introduction to the topic and clear picture on the history of secret sharing schemes. In Chapter 2, we present a new scheme for secret sharing using Gray code and XOR operation. Here the gray code is used to construct the shares and the XOR operation is used to reconstruct the secret. The chapter is concluded with the application of the scheme.

As we already mentioned, Visual cryptography is one of the most important research area in the field of secret sharing. In Chapter 3, we give a survey on various visual secret sharing ($Visual\ Cryptography$) schemes. Visual Cryptography Scheme (VCS) is a cryptographic scheme in which visual information (images, text, scanned docs.etc) are encrypted in such a way that decryption can be done with the human visual system (human eyes).

In Chapter 4, a new visual cryptography scheme is proposed. Here the base for this scheme is the secret sharing scheme which is explained in Chapter 2.

In Chapter 5 and 6 we present two strong schemes for visual secret sharing. Strong schemes means, security wise and information loss wise these schemes are sound. One scheme is based on the polynomial

interpolation concept and another scheme is based on the POB number system and Chinese Remainder Theorem. In Chapter 5, we present a new method for sharing visual information (visual cryptography), which is in Portable Network Graphics (PNG) image format, among honest communicating participants using polynomial interpolation and Mod operator.

In Chapter 6 we introduce a scheme to share a secret among n participants, i.e. a n out of n secret sharing scheme, based on a new number system called Permutation Ordered Binary Number System ($POB\ number\ system$) and Chinese Remainder Theorem (CRT).

One of the applications that we have identified here for the scheme developed in Chapter 5 is in broadcast encryption. So the Chapter 7 gives a glimpse on the broadcast encryption schemes. And in Chapter 8, the key distribution scheme in broadcast encryption is described as the application of the secret sharing scheme that is developed in Chapter 5. In this scheme the master key (MK), which is used for the broadcast encryption is not shared with any of the user. Instead of that the shares of the master key is shared with the users involved in the communication.

Attacks/ cheating is the vital section to be considered while secret sharing. Protection of visual secret is the main concern in visual cryptography, not the protection and genuineness of the participants, who shares the secret. Chapter 9, mainly focus on the various cheater identification and prevention schemes that are proposed by various authors in the last decade.

Chapter 10, gives a description on a new cheater identification scheme using Secure Hash Algorithm. And in Chapter 11, a brief summary and future scope is mentioned.

Chapter 2

Secret Sharing Scheme Using Gray Code and XOR Operation

2.1 Introduction

The chapter discusses about a secret sharing scheme using Gray Code and XOR operation. The Gray Code is used to construct the shares and the XOR operation is used to reconstruct the secret. The main purpose of this scheme is secret sharing among communicating parties. The same scheme with some minor modification can be used as cryptographic algorithm as well.

As we know Code is a symbolic representation of discrete information. Codes are of different forms. Gray Code is one of the most important codes. It is a non-weighted code which belongs to a class of codes called minimum change codes. In this codes while traversing from one step to another step only one bit in the code group changes. In case of Gray Code two adjacent

code numbers differs from each other by only one bit. The Table 2.1 gives the clear idea about it.

Decimal Numbers	*Binary Code*	*Gray Code*
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Table 2.1: Binary and Gray Codes

Now let us concentrate on the table of Gray Code given, where we can find the difference of Binary Code from Gray Code while traversing through the table for their respective decimal numbers. Gray Code is a numerical code used in computing in which consecutive integers are represented by binary numbers differing in only one digit. The term Gray Code is often used to refer to a "reflected" code. The Reflected Binary Code (RBC), later known as Gray Code after the famous Frank Gray. Frank Gray was

a physicist and researcher at Bell Labs who made numerous innovations in television, both mechanical and electronic, and is remembered for the Gray Code.

Binary to Gray Code conversion is a very simple process. The Figure 2.1, shows the conversion process.There are several steps to do this types of conversions. Steps given below elaborate on the idea on this type of conversion.

1. The M.S.B. of the Gray Code will be exactly equal to the first bit of the given binary number.

2. Now the second bit of the code will be exclusive or(XOR) of the first and second bit of the given binary number, i.e. if both the bits are same the result will be 0 and if they are different the result will be 1.

3. The third bit of Gray Code will be equal to the exclusive or (XOR) of the second and third bit of the given binary number. Thus the Binary to Gray Code conversion goes on. One example given below can make the idea clear on this type of conversion.

Let Binary Code be $b_3b_2b_1b_0$. Then the respective Gray Code can be obtained as follows:

$g_3 = b_3$
$g_2 = b_3 \oplus b_2$
$g_1 = b_2 \oplus b_1$
$g_0 = b_1 \oplus b_0$
Example:
Binary Code: $b_3b_2b_1b_0$ = 1 1 1 0 then Gray Code: $g_3g_2g_1g_0$
$g_3 = b_3 = 1$
$g_2 = b_3 \oplus b_2 = 1 \oplus 1 = 0$
$g_1 = b_2 \oplus b_1 = 1 \oplus 1 = 0$
$g_0 = b_1 \oplus b_0 = 1 \oplus 0 = 1$

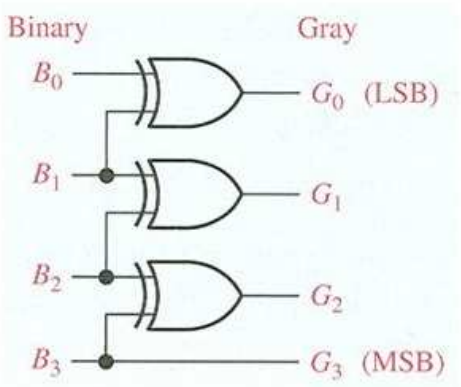


Figure 2.1: Binary to Gray Code Conversion

Final Binary Code: 1 0 0 1

2.2 Proposed Scheme: Gray Code-XOR Scheme

The scheme is based on the Gray Code and XOR operation. The shares of the secret are constructed using the Gray Code. The reconstruction of secret is simply by performing XOR operation on the shares. In the scheme the secret sharing phase constructs total 7 shares of the original secret. Among the seven shares, 3 shares fall in $Qualified\ Set$ and other four shares fall in $Forbidden\ Set$. Using this two sets of shares we can construct mainly three variant secret sharing schemes using the Gray Code and XOR operation. One is 7-out of-7 scheme, the second is 3-out of-3 scheme and the third is 3-out of-7 schemes. The next sections mainly focuses on this three variants of the Gray Code- XOR scheme.

2.2.1 7 out of 7 Scheme

In this section we give two main algorithm. One is for the share construction and the second one is for the reconstruction of secret. Algorithm 2.1, gives the share construction steps. As mentioned earlier, various shares can be constructed using the Gray Code conversion. And by performing XOR of all the shares the secret is reconstructed. The Algorithm 2.2 gives step wise process for the reconstruction of secret.

Definition 2.2.1. n-out of -n scheme: M is the message, then the message is divided or shared into n shares and for the reconstruction of M all the n shares are needed.

Definition 2.2.2. k-out of -n scheme: M is the message, then the message is divided or shared into n shares and for the reconstruction of M, k shares are needed.

The secret sharing process is shown in Figure 2.2. 5 bits blocks are the processing unit here and each share is constructed from the previous share. $M_1, M_2, \ldots, M_n$ are the n blocks of the secret, each having 5 bits in length. $G_1, G_2, \ldots, G_n$ are the n blocks of the shares, each having 5 bits in length. In each step the G_i is modified.

Algorithm 2.1: Share Construction

Input: The Secret , M.
Output: Seven Shares $S_1, S_2, S_3, S_4, S_5, S_6, S_7$.

1 Convert the Secret, M,into ASCII first then into binary form.
2 Consider the secret as 5 bits block and do the following steps.
`/* let` n `be the number of 5 bits block in the secret. */`
3 $i = 1, j = 1$.
4 **while** $i \leq n$ **do**
5 $M_i = \text{i}^{\text{th}}$ 5 bits block.
6 **while** $j \leq 7$ **do**
7 Convert 5 bits block, say M_i into the corresponding Gray value, say G.
8 Save $S_i = G$.
9 if i% 2 == 0
10 {
11 Save S_i in Qualified Set, say Q.
12 }
13 else
14 Save S_i in Forbidden Set, say F.
15 Update $M_i = G$.
16 **end**
17 **end**
18 Convert the shares S_1 through S_7 into the corresponding decimal values.

In the proposed system we are considering extended ASCII table. The Figure 2.3 and 2.4 shows Extended ASCII Table: Full list of ASCII characters, letters, symbols and signs.

Example 2.2.1. Suppose the secret information is; *me*?

The corresponding Decimal Value is 109 101 63.

The binary equivalent of the secret is 01101101 01100101 00111111

Divide it into 5 bits block(by adding padding bits at the left if required)

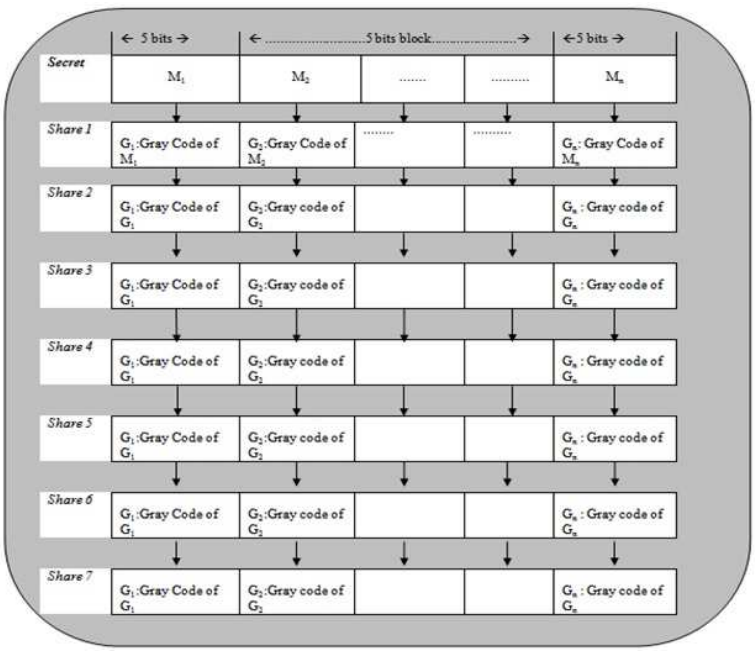


Figure 2.2: Secret Sharing Process-using Gray Code Construction

00110 11010 11001 01001 11111

The participant-shares are generated using Gray code of the secret. Initially the secret information is converted into the binary form and divided into 5 bits blocks. The first share is generated from the secret information. The first share is the gray code of the secret itself. The second share is generated using the first share. The second share is the gray code of the first share. Likewise all the 7 shares are generated. Table 2.2: shows the seven shares.

Here the share no:2,4 and 6 belongs to the Qualified Set and the rest four shares ; Share no:1,3,5 and 7 belongs to the Forbidden set. In 7 out of

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Figure 2.3: ASCII TABLE -PART I

7 secret sharing scheme all the shares are required to reconstruct the secret back.

The reconstruction of the secret is by the operation XOR. The Algorithm 2.2 gives the step wise procedure for the secret reconstruction. For the given example, the secret reconstruction is shown in Example 2.2.2.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	ø
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	

Figure 2.4: ASCII TABLE-PART II(Extended)

Algorithm 2.2: Secret Reconstruction

Input: Seven Shares $S_1, S_2, S_3, S_4, S_5, S_6, S_7$.
Output: The Secret , M

1 Convert the shares into binary form.
2 Consider the shares as 5 bits block and do the following steps
3 Perform block by block XOR on the shares.
4 Convert the result into corresponding decimal value and then to the ASCII text.
5 Save the result as the secret.

Share No	*Shares in Binary Form*	*Shares in Decimal*	*Final Shares*
1	00101 10111 10101 01101 10000	91 213 176	[õ°
2	00111 11100 11111 01011 11000	126 125 120	˜}x
3	00100 10010 10000 01110 10100	73 65 212	IAô
4	00110 11011 11000 01001 11110	109 225 62	má>
5	00101 10110 10100 01101 10001	91 81 177	[Q ±
6	00111 11101 11110 01011 11001	126 249 129	˜úy
7	00100 10011 10001 01110 10101	73 197 213	IÅõ

Table 2.2: Share Construction-7 out of 7 Scheme

Example 2.2.2. Consider the above example:
The shares are:
Share 1: [õ°
Decimal Value :91 213 176
Share 2: ˜}x
Decimal Value:126 125 120
Share 3: IAô
Decimal Value:73 65 212
Share 4:má>
Decimal Value:109 225 62
Share 5:[Q ±
Decimal Value:91 81 177
Share 6:˜úy
Decimal Value:126 249 129
Share 7: IÅõ
Decimal Value:73 197 213
Convert the shares into binary equivalent
01011011 11010101 10110000

01111110 01111101 01111000

01001001 01000001 11010100

01101101 11100001 00111110

01011011 01010001 10110001

01111110 11111001 01111001

01001001 11000101 11010101

As mentioned above in the Algorithm, the secret reconstruction from all seven shares; 7 out of 7 share scheme, all shares are just XORed. After XOR operation:

01101101 01100101 00111111

The Decimal Equivalent is

109 101 63

And finally the secret is *me*?.

2.2.2 3 out of 3 Scheme

In 3 out of 3 scheme, the share construction is given in the Algorithm 2.3.

It is a modified algorithm of Algorithm 2.1.

Algorithm 2.3: Share Construction

Input: The Secret , M.
Output: Three Shares S_1, S_2, S_3.

Convert the Secret, M, into ASCII first then into binary form.
Consider the secret as 5 bits block and do the following steps.
`/* let` n `be the number of 5 bits block in the secret. */`
$i = 1, j = 1, k = 1$
while $i \leq n$ **do**
    $M_i = \text{i}^{\text{th}}$ 5 bits block.
    while $j \leq 7$ **do**
        Convert 5 bits block, say M_i into the corresponding Gray value, say G.
        Save $S_i = G$.
        if i% 2==0
        {
        Save $S_k = S_i$.
        Update $k = k + 1$.
        }
        Update $j = j + 1$.
        Update M_i=S_i.
    end
    Update $i = i + 1$.
end
Convert the shares S_1 through S_3 into the corresponding decimal values and ASCII Text.

Example 2.2.3. Consider the same example :

Secret : *me*?

The 5 bits block after all the conversion process is

00110 11010 11001 01001 11111

The three shares according to the Algorithm 2.3 is given in the Table 2.3

Share No	*Shares in Binary Form*	*Shares in Decimal*	*Final Shares*
1	00111 11100 11111 01011 11000	126 125 120	˜}x
2	00110 11011 11000 01001 11110	109 225 62	má>
3	00111 11101 11110 01011 11001	126 249 129	˜úy

Table 2.3: Share Construction-3 out of 3 scheme

In 3 out of 3 secret sharing scheme all the three shares are required to reconstruct the secret back. The reconstruction of the secret is by the operation XOR. In Algorithm 2.2, the step wise procedure for the reconstruction of secret is given. Instead of 7 shares here 3 shares are XORed. For the given example, the secret reconstruction is shown in Example 2.2.4.

Example 2.2.4. consider the above example:
The shares are:
Share 1: ˜}x
Decimal Value:126 125 120
Share 2: má>
Decimal Value:109 225 62
Share 3:˜úy
Decimal Value:126 249 129
Convert the shares into binary equivalent
01111110 01111101 01111000
01101101 11100001 00111110
01111110 11111001 01111001

As mentioned above in the Algorithm, the secret reconstruction from all three shares; 3 out of 3 share scheme, all shares are just XORed.

After XOR operation:
01101101 01100101 00111111
The Decimal Equivalent is
109 101 63
And finally the secret is *me*?.

2.2.3 3 out of 7 Scheme

From the above two schemes, its clear that for reconstructing the secret back, we require only the shares from the Qualified Set, that means the share 2, share 4 and share 6. In 3 out of 7 Scheme, 7 shares are constructed as same as the 7 out of 7 Scheme and for reconstructing the secret the shares in the Qualified sets are XORed. It is actually the combination of the first two schemes.

This particular scheme is applicable in some special cases like, when communicating participants are with some special and different privileges. Consider an organization's top level structure having 7 employees E_1,E_2,E_3,E_4,E_5,E_6 and E_7. Among 7 employees, consider the first three employees E_1,E_2 and E_3, having some high and equal privileges and the rest 4 are having equal privileges which is lower than that of first three. In this particular situation 3 out of 7 scheme can be applied.

Distribute the shares in the Qualified Set among E_1,E_2 and E_3. And the shares in the Forbidden Set should be shared among E_4,E_5,E_6 and E_7. Doing this, there are two options to reconstruct the secret back, one is by colluding only the top level employees E_1,E_2 and E_3. And the second option is colluding all seven employees.

2.2.4 Security Analysis

The security analysis in this case shows that, if the algorithm is known then the reconstruction of the secret will be easy. By performing the Gray Code

to Binary Code conversion the secret may get exposed. By just performing the repeated Binary Code generation from one of the share, the secret information can be reconstructed. That means, there is no need of the collusion of all the shares. Before explaining with the example, let us see the theory behind the conversion of Gray Code to Binary Code. The Figure 2.5 shows the Gray Code to Binary Code conversion process.

1. The M.S.B. of the Gray Code will be exactly equal to the first bit of the given binary number.

2. Now the second bit of the code will be exclusive or(XOR) of the first and second bit of the given binary number, i.e. if both the bits are same the result will be 0 and if they are different the result will be 1.

3. The third bit of Gray Code will be equal to the exclusive or (XOR) of the second and third bit of the given binary number. Thus the Binary to Gray Code conversion goes on. One example given below can make your idea clear on this type of conversion.

Let Gray Code be $g_3g_2g_1g_0$. Then the respective Binary Code can be obtained as follows:

i.e.

$b_3 = g_3$

$b_2 = b_3 \oplus g_2$

$b_1 = b_2 \oplus g_1$

$b_0 = b_1 \oplus g_0$

Example:

Gray Code: $g_3g_2g_1g_0$ = 1 0 0 1 then Binary Code: $b_3b_2b_1b_0$

$b_3 = g_3 = 1$

$b_2 = b_3 \oplus g_2 = 1 \oplus 0 = 1$

$b_1 = b_2 \oplus g_1 = 1 \oplus 0 = 1$

$b_0 = b_1 \oplus g_0 = 1 \oplus 1 = 0$

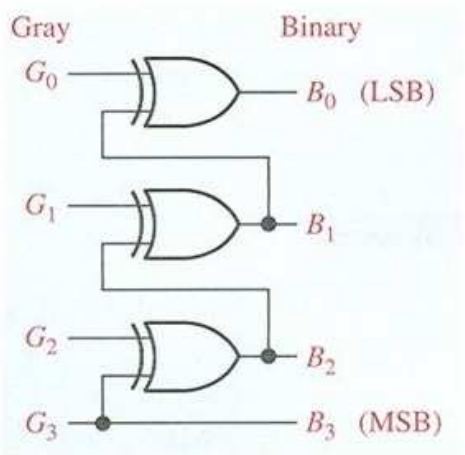


Figure 2.5: Gray Code to Binary Conversion

Final Binary Code: 1 1 1 0
For example consider one of the share(3rd share); IAô;
73 65 212 in decimal.
The binary equivalent of the share and it is:
01001001 01000001 11010100.
Perform the conversion process(Gray to Binary)
01111110 01111101 01111000.
And its text value is ˜}x
Repeat the step; at some stage the secret will be revealed.
The Binary Code of 01111110 01111101 01111000 is:

01011011 11010101 10110000.

Its text value is [õ°
The Binary Code of 01011011 11010101 10110000 is:

01001001 01000001 11010100.

Its text value is me?.
In this step (3rd step)the secret is reconstructed.

In order to avoid this situation, the blocks (5 bits blocks) are shuffled. The shuffling is not within the shares, its across the shares. And one thing to be noticed, the block wise shuffling should be performed.

2.2.5 Application

The secret sharing scheme discussed here can be used as an encryption scheme with a key. Here the key is nothing but the share number. The encryption scheme using the proposed system can be represented as bellow;

C=E(M,K)

M=D(C,K)

Where M is the secret message.

C is the cipher text.

E is the encryption algorithm, the Gray Code generation.

D is the decryption algorithm, the Binary Code generation.

K is the key.

The encryption and decryption are block processing. Here the block size is 5 bits.

Algorithm 2.4: Encryption

Input: The Secret , M and Key ,K.
Output: Ciphet Text C.

1 Convert the Secret(plain text),M, into binary form.
2 Divide the message into n 5 bits blocks.
`/* let` n `be the number of 5 bits block in the secret. */`
3 Read key; K.
`/* The number of digits in K should be less than or equal to` n. `*/`
4 Repeat the following two steps until all the blocks are processed.
5 Pick a block from the message and its corresponding digit from the key.
6 Covert the message block into the gray code, number of times the digit from the key is having.
7 Convert the resulting binary quantity into the decimal form ,say C.

Algorithm 2.5: Decryption

Input: The Cipher Text , C and Key ,K.
Output: The Secret Text S.

1 Convert the Cipher Text),C, into binary form.
2 Divide the cipher text,in binary form, into n 5 bits blocks.
3 Read key; K.
4 Repeat the following two steps until all the blocks are processed.
5 Pick a block from the cipher text and its corresponding digit from the key.
6 Covert the cipher block into the binary code, number of times the digit from the key is having.
7 Convert the resulting binary quantity into the decimal form ,say M.

Consider an example:

ENCRYPTION:

The message is; 255 212 234 199 121

The binary equivalent is:

11111111 11010100 11101010 11000111 01111001

Divide the message into 5 bits block:

11111 11111 01010 01110 10101 10001 11011 11001

Now select a key K. The conditions are:

(a) Digits in the key should be less than or equal to the number of blocks in the message.

(b) One more requirement is the digit that select for the key should be less than or equal to 7.

In short the key, K, should be an octal number having digits less than or equal number of blocks in the message.

Assume key K= 271. Here the number of digit in the key is less than the number of blocks in the message, so the digits in the key are repeated so that the total length is equal to that of the number of blocks (say n) in the plain text.

Now the key K= 27127127

Now perform the step 3; the message will become

Secret ,M 11111 11111 01010 01110 10101 10001 11011

Key, K 2 7 1 2 7 1 2

Cipher, C 11000 10101 01111 01101 11001 11001 11101

So the cipher text is:

11000101 01011110 11011100 11100111 10110001.

In decimal form 197 94 220 231 117.

DECRYPTION:

Suppose the cipher text is 197 94 220 231 117

The binary equivalent is 11000101 01011110 11011100 11100111 10110001

Divide the cipher text into 5 bits block:

11000 10101 01111 01101 11001 11001 11101 10001

Here the key is K= 271. The number of digit in the key is less than the number of blocks in the cipher, so the digits in the key are repeated so that the total length is equal to that of the number of blocks (say n) in the cipher text.

Now the key K= 27127127

Now perform the step 3; the cipher text will become

Cipher, C 11000 10101 01111 01101 11001 11001 11101 10001

Key, K 2 7 1 2 7 1 2 7

Secret, M 11111 11111 01010 01110 10101 10001 11011 11001

2.3 Concluding Remarks

In this chapter we have considered secret sharing scheme using Gray Code and XOR operation. The share size is the major concern in many cases, and in most of the cases the share size grows exponentially. Here the size of the share is same as the secret. It may noted that the scheme that is discussed here is also used as an encryption scheme. The encryption scheme is a symmetric cipher and a block cipher as well.

Chapter 3

Introduction to Visual Secret Sharing Schemes

3.1 Introduction

Image sharing is a subset of secret sharing because it acts as a special approach to the general secret sharing problem. The secret in this case are concealed images. Image sharing defines a scheme which is identical to that of general secret sharing. In (k, n) image sharing, the image that carries the secret is split up into n pieces (known as shares) and decryption is totally unsuccessful unless at least k pieces are collected and superimposed.

Visual Cryptography (VC) [NS95] was originally invented and pioneered by Moni Naor and Adi Shamir in 1994 at the Eurocrypt conference. Visual Cryptography is a new type of cryptographic scheme, which can be decoded concealed images without any cryptographic computation. As the name suggests, VC is related to human visual system. VC is a cryptographic technique which allows visual information (picture, text, etc) to be encrypted in such a way that the decryption can be performed by the human visual system. When shares are stacked the

human eye do the decryption. This allows anyone to use the system without any knowledge of cryptography and without performing any computations whatsoever. This is the main advantage of VC over the other popular conditionally secure cryptography schemes. The mechanism is very secure and very easy to implement.

Naor and Shamirs initial implementation assumes that the image or message is a collection of black and white pixels (means, binary images), each pixel is handled individually and it should be noted that the white pixel represents the transparent colour. One disadvantage of this is that the decryption process is lossy. The main area that suffers due to this lossy process is the contrast. One of the very important attribute in visual cryptography is the contrast of the recovered images, because it determines the clarity of the recovered secret by human visual system. The relative difference Hamming weight between the representation of white and black pixels signify the loss in contrast of the recovered secret. The newer schemes deal with gray scale and colour images which attempt to minimize the loss in contrast by using digital half toning.

The encryption problem is expressed as a k *out of* n secret sharing problem. For a given image or message, n shares are generated so that original image or message is visible if and only if any k of them are stacked together. The image remains hidden if fewer than k shares are stacked.

The important parameters of the scheme are:

1. m, the number of pixels in a share. This represents the loss in resolution from original image to the recovered one.

2. Λ, the relative difference in the weight between the combined shares that come from a white and black pixel in the original image, i.e., the loss in contrast.

3. γ, the size of the collection of C_0 and C_1. C_0 refers to the subpixel pattern in the shares for a white pixel, and C_1 refers to the subpixel pattern in the shares for a black pixel.

Consider 2 out of 2 Visual Cryptographic Scheme (commonly known as (2, 2)-VCS). The following matrices of the order 2×2 (2 matrices of the order 2×2, each for separately handling white and black pixel) are constructed.

$$S_0 = \begin{matrix} 1 & 0 \\ 1 & 0 \end{matrix}$$

$$S_1 = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

Construct C_0 and C_1 as follows:

$$C_0 = \left\{ \textit{all the matrices obtained by permuting columns of } \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \right\}$$

$$C_1 = \left\{ \textit{all the matrices obtained by permuting columns of } \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\}$$

$$C_0 = \left\{ \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \right\}$$

$$C_1 = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\}$$

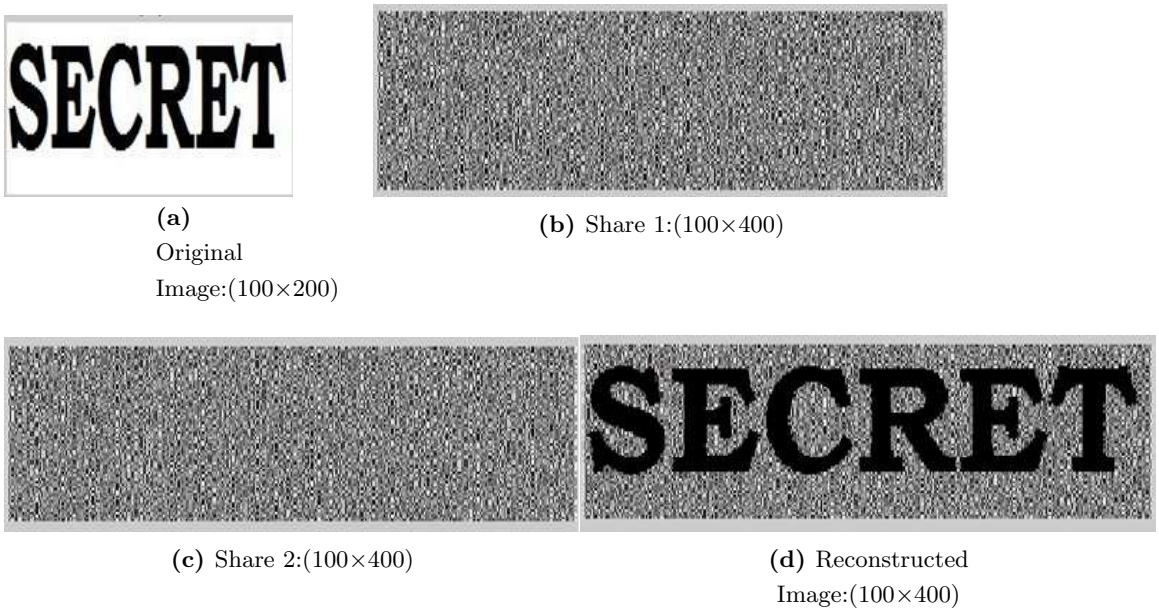
The shares can be generated in the following manner:

For each pixel of the original do:

Choose randomly one of the matrices in, C_0 or C_1 according to the color of the pixel, whether it is white or black respectively. i.e,

- C_0 if you want to share a white pixel.
- C_1 if you want to share a black pixel.

The chosen matrix defines the color of the 2 sub pixels in each of the 2 shares. Means, the first row is given to first share and second row is given to second share. Due to pixel expansion, each pixel from original image gets expanded into two sub pixels.



(a) Original Image:(100×200)

(b) Share 1:(100×400)

(c) Share 2:(100×400)

(d) Reconstructed Image:(100×400)

Figure 3.1: The result of (2,2)-VCS.

For reconstruction, the 2 separate shares are stacked together. Here individual shares will not reveal any information about the secret. Figure 3.1 shows an example for (2,2) VCS.

3.2 Size Invariant Visual Cryptography

The first paper to consider image size invariant VC was proposed by Ito et al.[IH98]. The traditional VCS employ pixel expansion. In pixel expansion, each share is m times the size of the secret image. Thus, it can lead to the difficulty in carrying these shares and consumption of more storage space. Ito's scheme removes the need for this pixel expansion. There are also some other studies which focus on the methods without pixel expansion [IH98] [YC05a] [YC05b] [YC06a] [YC06b]. [IH98] contain a scheme that removes

the need for the pixel expansion. As with traditional VC, $n \times m$ sets of matrices need to be defined for the scheme. Because this scheme uses no pixel expansion, m is always equal to one and n is based on the type of scheme being used for example a $(2, 3)$ scheme, $n = 3$.

In this scheme S_0 and S_1 is as follows:

$$S_0 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ \vdots & \dots & \dots & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$C_0 = \left\{ \textit{all the matrices obtained by permuting columns of } \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ \vdots & \dots & \dots & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \right\}$$

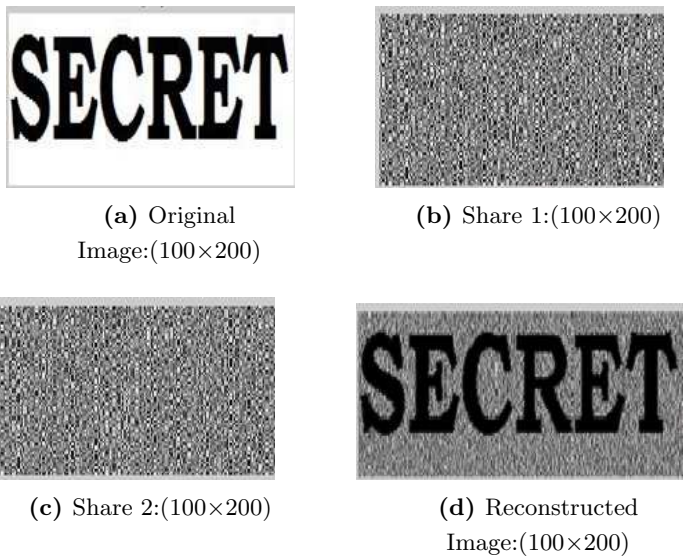
$$C_1 = \left\{ \textit{all the matrices obtained by permuting columns of } \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \right\}$$

To share a white pixel, one of the columns in C_0 is chosen and to share a black pixel, one of the columns in C_1 is chosen. The chosen column vector V, defines the colour of each pixel in the corresponding shares. The structure of this scheme is described by a boolean n- vector V as follows;

$V = \{v_1, v_2, \dots, v_n\}^{\mathrm{T}}$,

where v_i represents the colour of the pixel in the i^{th} shared image.

If $v_i = 1$ then the pixel is black, otherwise, if $v_i = 0$ then the pixel is white. To reconstruct the secret, traditional OR-ing is applied to the pixel



(a) Original Image:(100×200)

(b) Share 1:(100×200)

(c) Share 2:(100×200)

(d) Reconstructed Image:(100×200)

Figure 3.2: The result of size invariant (2,2)-VCS.

in V. An example based on the $(2, 2)$ scheme is shown in Figure 3.2. The size invariant scheme supports (k, n) and (n, n) threshold schemes.

The pixel expansion involved in many schemes discussed so far leads on to a related topic within size invariant schemes, namely aspect ratio. Aspect ratio invariant secret sharing is presented by Yang and Chen [YC05a]. This aspect ratio invariant secret sharing scheme dramatically reduces the number of extra subpixels needed in constructing the secret. This results in smaller shares, closer to the size of the original secret while also maintaining the aspect ratio, thus avoiding distortion when reconstructing the secret. Alternatively this problem can be examined from the opposite end, trading overall share size and contrast. A size adjustment scheme is presented [YC05b] that allow the user to choose an appropriate share size that is practical for the current use of the share. If quality and contrast matters then the size of the share will increase,

where as the opposite can happen if these things are not overly important for a users particular application. Yang and Chen [YC06b] further progress this research by generalizing the aspect ratio invariant problem. To achieve the same relative position between two square blocks, to avoid distortion, the re sampling method in image scaling is used.

In 1996, Naor and Pinkas proposed an alternative VCS model for improving the contrast in [NP97]. In 1999, Blundo et al. [BDS03][ABSS01][ABSS96b] analyzed the contrast of the reconstructed image in k-out of-n VCS .Blundo et al. gave a complete characterization of 2-out of-n VCS having optimal contrast and minimum pixel expansion in terms of certain balanced incomplete block designs. Blundo et al.s research results are valuable for the researchers who are interested in the area of Visual Cryptography. The other research works done by different authors are found in[BWag][Cam00][DK04][CSFM05][YC06c][CCH$^+$07].

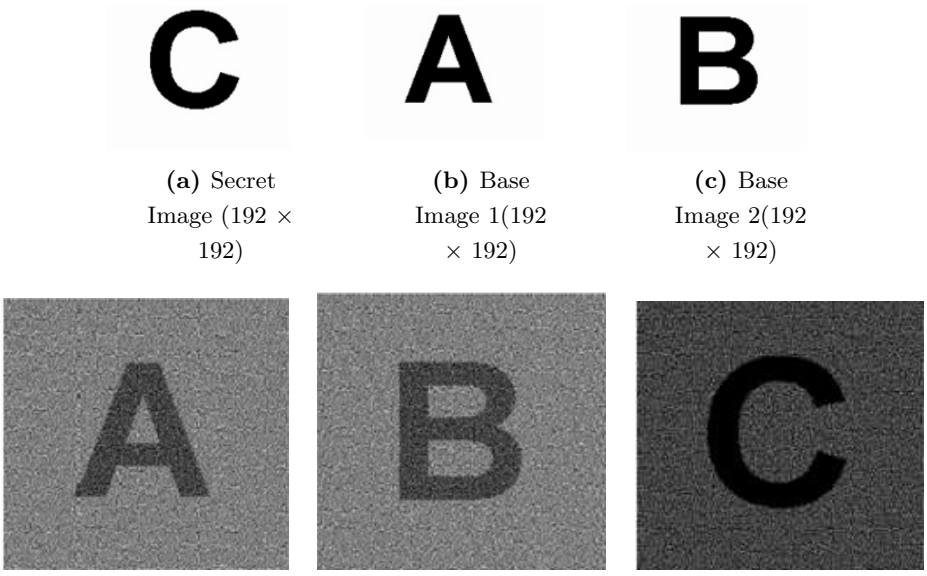
Viet and Kurosawa [DK04] proposed a VCS with reversing, in which the participants are also allowed to reverse their transparencies. But in this scheme there is a loss of resolution, since the number of pixels in the reconstructed image is greater than that in the original secret image. The concept of recursive hiding of secrets in visual cryptography was proposed by Gnanaguruparan and Kak [GK02]. This provides a method of hiding secrets recursively in the shares of threshold schemes, which permits an efficient utilization of data. In recursive hiding of secrets, several additional messages can be hidden in one of the shares of the original secret image. By using recursive threshold visual cryptography in network application, network load can be reduced. Visual cryptography schemes were also proposed to deal with gray-level images . The use of half toning techniques makes it possible that the ready made schemes designed for binary secret images can be directly applied to gray-level images [LT03b][Zha98]. The different gray-level visual cryptography schemes are studied by researchers. Applying visual cryptography techniques to color

images is a very important area of research because it allows the use of natural color images. Color images are also highly popular and have a wider range of uses when compared to other image types.

3.3 Extended Visual Cryptography

Extended VC takes the idea of visual cryptography further by creating shares which are meaningful to anyone who views them. This helps to alleviate suspicion that any encryption has taken place and also presents visually pleasing shares which incorporate all the previously mentioned features of VC. Extended visual cryptography schemes allow the construction of visual secret sharing schemes within which the shares are meaningful as opposed to having random noise on the shares. After the sets of shares are superimposed, the meaningful information disappears and the secret is recovered. This is the basis for the extended form of visual cryptography. Figure 3.3 shows an example of a (2, 2) EVCS. As can be seen from the figure, two meaningful shares are generated from the base images. During this share creation, the secret is encoded between each of the shares. After superimposing each share, the secret is completely recovered while the meaningful information on each share completely disappears[NY02].

In order to use this extended visual cryptography scheme, a general construction needs to be defined. Ateniese et al. [ABSS96a] have devised a mechanism by which we can generate the shares for the scheme. A stronger security model for EVCS is one in which the shares associated with a forbidden subset can be inspected by the user, meaning that the secret image will still remain totally hidden even if all n shares are previously known by the user. A symmetric approach to fully address a general (k, n) problem was also proposed [ABSS96a]. For each set of access structure, let $P = 1, 2, \ldots, n$ represent the set of elements called



(a) Secret Image (192 × 192)

(b) Base Image 1(192 × 192)

(c) Base Image 2(192 × 192)

(d) Extended Share 1 (384 × 384

(e) Extended Share 2 (384 × 384)

(f) Recovered Secret (384 × 384)

Figure 3.3: Extended Visual Cryptography

participants, and let $2P$ denote the set of all subsets of P. Let Qual/Forb be the collection of Qualified/Forbidden sets. The pair is called the access structure of the scheme. Any qualified set can recover the shares image by stacking its participants transparencies, while any forbidden set has no information on the shared image. In [ABSS96a]the authors propose a new technique to realize (k, n) VCS, which is better with respect to the pixel expansion than the one proposed by Naor and Shamir.

3.4 Colour Visual Cryptography

One of the most potentially useful types of visual cryptography scheme is color visual cryptography. The reason for this is that the majority of

people nowadays are more used to color images and interact with them more frequently. Natural color images can be used to share secrets; this provides a very helpful cover for unsuspicious hiding the fact that any encryption has taken place at all. However, some of these schemes do not work without a computer, which does defeat the main purpose of visual cryptography. Other color schemes do try to keep with the main ethos of instantaneous decryption without a computer.

Visual Cryptography schemes were applied to only black and white images till year 1997. Verheul and Van Tilborg proposed first color visual cryptography scheme [VHT97]. In this visual cryptography scheme one pixel is distributed into m sub pixels, and each sub pixel is divided into c color regions. In each sub pixel, there is exactly one color region colored, and all the other color regions are black. In 2000, Yang and Laih [NL00] proposed a different construction mechanism for the colored visual cryptography scheme. They argued that their method can be easily implemented and can get much better block length than Verheul and Van Tilborgs scheme.

F.Liu, C.K.Wu, X.J. Lin proposed a new approach for colored visual cryptography scheme [LWL08]. They proposed three different approaches for color image representation:

In first approach, colors in the secret image can be printed on the shares directly. It works similar to basic visual cryptography model. Limitations of this approach are large pixel expansion and quality of decoded image is degraded.

In second approach separate three color channels are used. Red, Green, Blue for additive model and Cyan, Magenta, Yellow for subtractive model. Then normal visual cryptography scheme for black and white images is applied to each of the color channels. This approach reduces the pixel expansion but quality of image gets degraded due to half toning process.

In third approach, binary representation of color of a pixel is used and secret image is encrypted at bit-level. This results in better quality of image.

A major common disadvantage of the above reviewed colored VCS is that the number of colors and the number of subpixels determine the resolution of the revealed secret image. If many colors are used, the subpixels require a large matrix to represent it. Also, the contrast of the revealed secret image will go down drastically. Consequently, how to correctly stack these shared transparencies and recognize the revealed secret image are the major issues. Hou, [Hou03] proposed a VCS for color images. His methods are based on halftone technique and color decomposition. Basic terminologies used in encrypting colored images via visual cryptographic method are discussed below.

1. Halftoning: This method uses the density of the net dots to simulate the gray level called, Halftone and transforms an image with gray level into a binary image before processing.

2. Color Decomposition: In this, every color on a color image can be decomposed into three primary colors:

   Cyan-Magenta-Yellow(C, M ,Y), if subtractive model is used

   Red-Blue-Green(R,G,B), if additive model is used.

   This method expands every pixel of a color secret image into a 2×2 block in the sharing images and keep two colored and two transparent pixels in the block.

3. Pixel expansion: Pixel expansion m refers to the number of subpixels in the generated shares that represents a pixel of the original input image. Smaller pixel expansion results in smaller size of the share. It represents the loss in resolution from the original picture to the shared one.

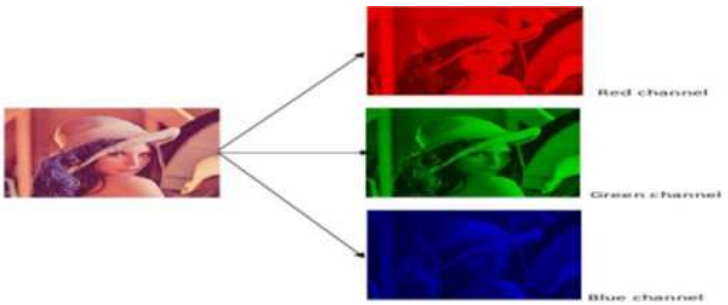


Figure 3.4: Color VCS

Recently, more and more applications of visual cryptography, such as authentication, human identification, copyright protection, watermarking, mobile ticket validation, visual signature checking etc. are introduced [FA04][HG06][HH05][NP97]. The print and scan application of VCS can be found in [DYK04]. In this application, scan the shares into a computer system and then digitally superimpose their corresponding shares. This would make possible secure verification of e-tickets or other documents. The developments and the research works done by other researchers in the different perspectives on visual cryptography, such as access structure, generation of shares and other aspects reported by different authors are discussed in [Hou03][HC06][KAL11][LT03a][Mac00][WH11].

3.5 Diverse Visual Cryptographic Schemes

Visual cryptography is paradigm of cryptography which allows visual information (e.g. images, printed text and handwritten notes) to be encrypted in such a way that its decryption can be done by the human eye, without the aid of computers. It avoids the need of complex mathematical computations during decryption and the secret image can be reconstructed using stacking (OR operation). There are diverse visual

cryptography schemes based on the factors such as pixel expansion, contrast, security, meaningless or meaningful shares, type of secret image (either binary or color) and the number of secret images encrypted (single or multiple secret) etc. The following are the diverse visual cryptography schemes:

1. Traditional Visual Cryptography
2. Extended Visual Cryptography
3. Color Visual Cryptography Schemes
4. Size Invariant Visual Cryptography
5. Recursive Threshold Visual Cryptography Scheme
6. Random Grids based Visual Cryptography
7. Halftone Visual Cryptography
8. Probabilistic Visual Cryptography
9. Region Incrementing Visual Cryptography
10. Progressive Visual Cryptography
11. Segment based Visual Cryptography Scheme
12. Cheating Immune Visual Cryptography Schemes
13. User-friendly Visual Secret sharing scheme
14. Dynamic Visual Cryptography
15. OR and XOR Visual Cryptography

Among these the Traditional Visual Cryptography, Size Invariant Visual Cryptography, Extended Visual Cryptography, and Color Visual Cryptography are already discussed in the previous section. The others are discussed below.

3.5.1 Recursive Threshold Visual Cryptography Scheme

A recursive [PK11][PK08][PKca] style of secret sharing takes into account a set of two shares which contain more than one secret. Recovering this secret requires rotation or shifting of the share to different locations on the corresponding share. In recursive hiding of secrets, the user encrypts additional information about smaller secrets in the shares of a larger secret without causing any expansion in the size of the latter, thereby increasing the efficiency of secret sharing. The idea here is to double the secret size at every step and so increases the information that every bit of share conveys to $(n-1)/n$ bit of secret i.e. almost 100 percentage.

3.5.2 Random Grids based Visual Cryptography

Random Grids(RG)[CT09][CT11][KK87][NY02] extends the solution to the secret sharing problem by implementing a collection of 2-D transparent and opaque pixels arranged randomly which reveals the secret to the Human Visual System(HVS) when being superimposed. Unlike other visual cryptography approaches, random grid does not need the basis matrices to encode the shares. Pixel expansion is disallowed which is therefore a great advantage of using Random Grids. Also, the sizes of secret image and the shares are identical to each other.

In general, a RG is defined as a transparency comprising a two-dimensional array of pixels, where each pixel can be fully transparent (white) or totally opaque (black), and the choice between the alternatives is made by a coin-ip procedure. Half of the pixels in a RG are white, and

the remaining pixels are black. In reported RG-based VC[CT11], the number of white pixels in a share is approximately half of the total number. Here, the concept of generalized RG is introduced, where the probability for a pixel in a share to be white becomes adjustable.

Figure 3.5: Random Grid
Based Visual Cryptography

3.5.3 Halftone Visual Cryptography

In a general halftone visual cryptography framework [ZAC] where a secret binary image is encrypted into high- quality halftone images or halftone shares. In particular, the proposed method applies the rich theory of blue noise halftoning to the construction mechanism used in conventional VC to generate halftone shares, while the security properties are still maintained. The same contrast is obtained over the whole decoded image. The halftone shares carry significant visual information to the viewers, such as landscapes, buildings, etc. According to the author, the visual quality obtained by the new method is significantly better than that attained by extended VC or any other available VC method known to date. Halftone VC is built upon the basis matrices and collections available in conventional VC.

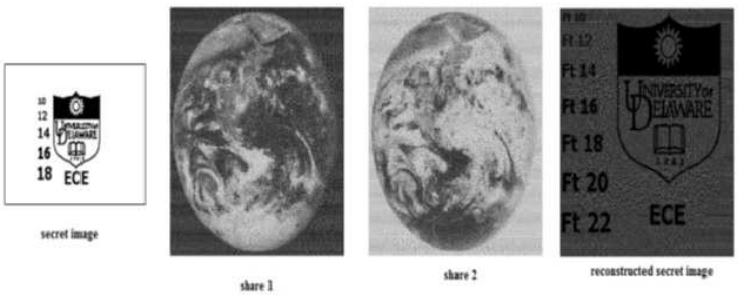


Figure 3.6: Halftone Visual Cryptography

3.5.4 Probabilistic Visual Cryptography

In [Yan04], new model called Probabilistic Visual Cryptography is introduced. In such a model the pixel expansion m is 1, means there is no pixel expansion. The reconstruction of the image however is probabilistic[AS92], meaning that a secret pixel will be correctly reconstructed only with a certain probability. However, while in the deterministic model the reconstruction of an approximation of the secret pixel is guaranteed. In Yang's probabilistic model the secret pixel is correctly reconstructed with some probability. Yang's aim is to provide schemes with no pixel expansion, which are obviously desirable. However the quality of the reconstructed pixel depends on how big the probabilities are of correctly reconstructing secret pixels.

3.5.5 Region Incrementing Visual Cryptography

In traditional visual cryptography scheme, one whole image is considered as a single secret and same encoding rule is applied for all pixels of one image. So it reveals either entire image or nothing. It may be the situation that different regions in one image can have different secrecy levels, so we

can not apply same encoding rule to all pixels. Ran-Zan Wang developed a scheme Region Incrementing Visual Cryptography for sharing visual secrets of multiple secrecy level in a single image [SJ12]. In this scheme, different regions are made of a single image, based on secrecy level and different encoding rules are applied to these regions.

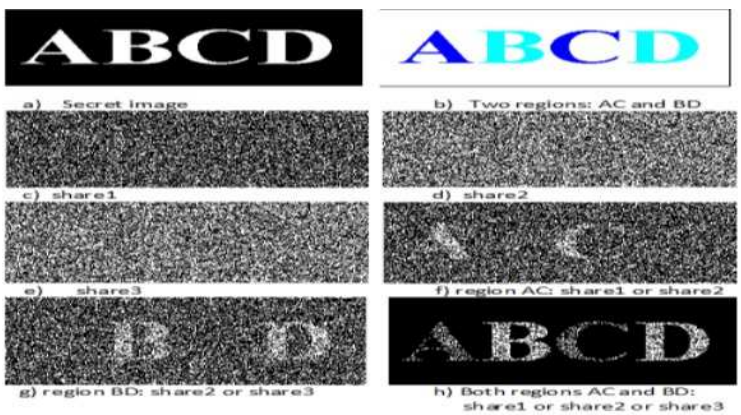


Figure 3.7: Region Incrementing Visual Cryptography

3.5.6 Progressive Visual Cryptography

Progressive Visual Cryptography(PVC) [JYK05] takes into consideration the premise of perfect secret recovery and high quality secret reconstruction. Many of the schemes do require computational effort in order to perfectly reconstruct the secret. A new sharing concept emerged known as Progressive Visual Cryptography which revealed the secret image progressively as more and more number of shares were stacked together.

The aforementioned VSS schemes are completely revealed the secret,

but cannot achieve progressive image sharing. In (k, n) visual secret sharing scheme, it is not possible to recover the secret image, though one less than k shares are available. This problem is solved in the progressive visual cryptography scheme developed by D. Jin, W. Q. Yan, and M. S. Kankanhalli [JYK05]. In progressive visual cryptography scheme, it is not necessary to have at least k shares out of n, as in (k, n) secret sharing scheme. If more than one share obtained, it starts recovering the secret image gradually. The quality of recovered image improves, as the number of shares received increases. In 2008, Fang [Fan08] combines the progressive VC-based VSS [JYK05] and the friendly VC-based VSS [TL03] methods to form a new one. Unfortunately, Fangs scheme still suffers the problem of pixel expansion up to four times. Even though Young-Chang Hou and Zen-Yu Quans PVC method [HQ11] generates noise-like shares, i.e. the generated shares are not meaningful, which are of more interest to hackers as they treat them as critical information in the transmission. Confidential images have no means to be secured when they are transmitted over the network. Young-Chang Hou and Zen-Yu Quans PVC method [HQ11] proposes a watermarking scheme which overcomes the drawbacks and also this paper deals with the color images instead of gray scale images.

3.5.7 Segment based Visual Cryptography Scheme

Traditional visual cryptography schemes were based on pixels in the input image. The limitation of pixel based visual cryptography scheme is loss in contrast of the reconstructed image, which is directly proportional to pixel expansion. Bernd Borchert proposed [Bor04] a new scheme which is not pixel-based but segment-based. It is useful to encrypt messages consisting of symbols represented by a segment display.

For example, the decimal digits 0, 1, . . . , 9 can be represented by seven-segment display. The advantage of the segment based encryption is that, it

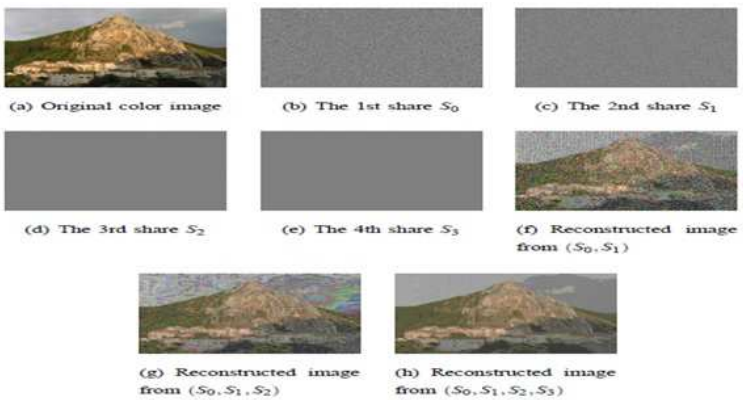


Figure 3.8: Progressive Visual Cryptography

may be easier to adjust the secret images and the symbols are potentially easier to realize for the human eye and it may be easier for a non expert human user of an encryption system to understand the working. The secret, usually in the form of digits is coded into seven segment display before encrypted. Two random share images will be generated during encryption. Decryption process involves the stacking of these two share images.

3.5.8 Cheating Immune Visual Cryptography Schemes

Prevention of cheating via authentication methods has been proposed [PS06]which focus on identification between two participants to help prevent any type of cheating taking place. Yang and Laih presented two types of cheating prevention; one type used an online trust authority to perform the verification between the participants. The second type involved changing the VC scheme whereby the stacking of two shares reveals a verification image; however this method requires the addition of extra pixels in the secret. Another cheating prevention scheme described by Horng et al., whereby if an attacker knows the exact distribution of

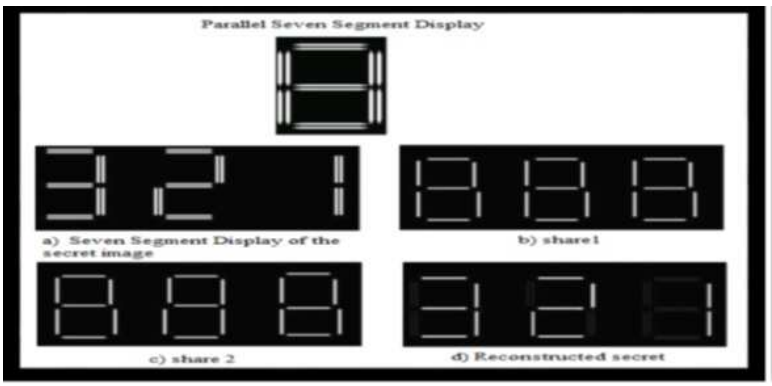


Figure 3.9: Segment Based Visual Cryptography

black and white pixels of each of the shares of honest participants then they will be able to successfully attack and cheat the scheme. Horng's method prevents the attacker from obtaining this distribution. Many Cheating Immune Visual Cryptography Schemes (CIVCS) have been proposed to address this problem. The CIVCS techniques can be classified as follows:

1. Make use of an online trusted authority who can verify the validity of the stacked shares.

2. Generate extra verification shares to verify the validity of the stacked shares.

3. Expand the pixel expansion of the scheme to embed extra authentication information.

4. Generate more than n shares to reduce the possibility that the cheaters can correctly guess the distribution of the victims' shares.

5. Make use of the genetic algorithm to encrypt homogeneous secret images.

3.5.9 User-friendly Visual Secret Sharing Scheme

This scheme is used to generate meaningful size invariant share images during encryption. Unfortunately, in the two previous schemes (i.e., extended VC or halftone VC) that generate meaningful contents, the size of the shares generated during encryption were at least four times larger than that of the original secret image. Chen and Tsao [CT11][TL03] proposed a novel random grid based visual secret sharing scheme that has been skillfully designed to produce meaningful (user-friendly) share images without pixel expansion. It explains a procedure with different light transmissions based on the share images and the logo image (cover image) used to make the shares user-friendly. To implement meaningfulness, this scheme adjusts the respective contrasts of some areas of the two generated random grids G_1 and G_2 based on the cover image. Figure 3.10 shows User-friendly Visual Secret Sharing Scheme.

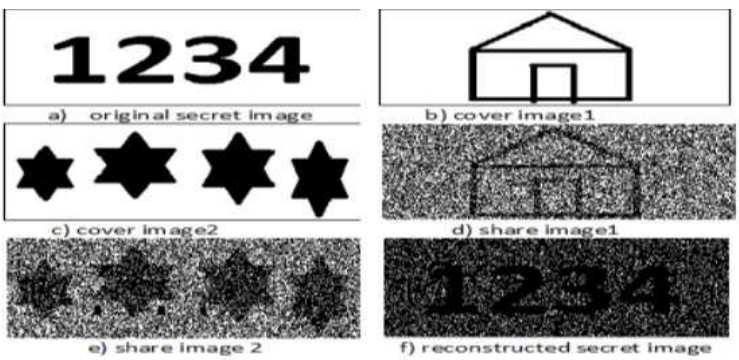


Figure 3.10: User-friendly Visual Secret Sharing Scheme

3.5.10 Dynamic Visual Cryptography

The core idea behind dynamic visual cryptography [WC98] is increasing the overall capacity of a visual cryptography scheme. That means, using a set of two or more shares, we can potentially hide two or more secrets. Multiple secret sharing is very useful when it comes to hiding more than one piece of information within a set of shares.

3.5.11 OR and XOR Visual Cryptography

A (k, n) Visual Cryptographic Scheme encodes a secret image into n shadow images (printed on transparencies) distributed among n participants. When any k participants superimpose their transparencies on an overhead projector (OR operation), the secret image can be visually revealed by a human visual system without computation. However, the monotone property of OR operation degrades the visual quality of reconstructed image for OR-based VCS (OVCS)[CW11]. Usually all the conventional visual cryptography schemes uses OR operation for stacking operations and so it is also called OR-based VCS. But it offers a poor visual quality image during decoding (stacking). XOR based VCS[THH$^+$05], which uses XOR operation, uses the properties of contrast and security advantage XOR-based VCS (XVCS).

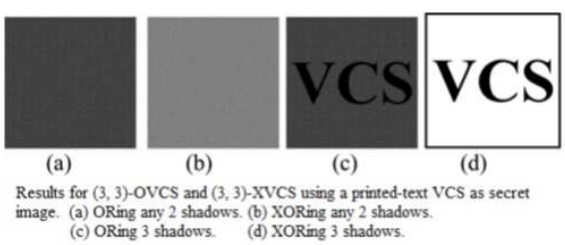


Figure 3.11: OR and XOR Visual Cryptography

3.6 Concluding Remarks

In this chapter we have considered some of the extended capabilities of secret sharing schemes, especially Visual Secret Sharing Scheme(called Visual Cryptography). We have done a survey on various Visual Cryptography Schemes, additional properties and also explored the schemes, which are efficient and easy to implement.

Chapter 4

Visual Cryptographic Scheme Using Gray Code and XOR Operation

4.1 Introduction

The chapter discusses about a new Visual Cryptographic Scheme(VCS) based on Gray Code and XOR operation. Here the shares are constructed using Gray Code and the secret is reconstructed by using the XOR operation. We have considered gray scale images here. The scheme is very simple and easy to implement as well. It doesn't have pixel expansion. The original secret image, shares and the reconstructed secret image are in same size. So the scheme is a lossless scheme.

Already we have discussed about the code concept. The Gray Code and Binary Code conversion process is discussed in Chapter 2. The Table 2.1 shows the idea clearly.

4.2 Proposed Scheme: VCS using Gray Code and XOR

The scheme is based on the Gray Code and XOR operation. In this method total 7 shares are generated from the secret image. We can construct two variant of VCS using this method, one is 7-out of-7 scheme, VCS (7, 7) and the second is 3-out of-3 scheme, VCS (3, 3). For this, two sets of shares are generated, Qualified Set of shares (Q-set) and Forbidden Set of shares (F-Set).The Q-set contain 3 shares among 7 shares and F-set contains 4 shares among 7 shares. From the secret image these two sets of shares are constructed using the Gray Code conversion. The reconstruction of secret image is simply by XORing the shares.

When compared to any other VCS scheme the advantage of this scheme is, 5 pixels are processed at a time instead of a single pixel. And one of the assumption in this case is, the number of columns in the image should be a multiple of 5. So here, a preprocessing phase to keep the image in the prescribed size is recommended here.

4.2.1 VCS(7,7)

The Algorithm 4.1 shows the share construction process in detail and Algorithm 4.2 shows the secret image reconstruction in detail.

Algorithm 4.1: Share Construction

Input: The Secret Image , IM, with dimension $r \times c$
Output: Seven Shares, $S_1, S_2, S_3, S_4, S_5, S_6, S_7$, each with dimension $r \times c$.

```
let i = 1,j = 1
while i ≤ r do
    Select 5 pixels at a time from the i^th raw of the secret image.
    Convert the 5 pixels into the corresponding binary value.
    Divide he binary value into 8 blocks each having 5 bits in length.
    let n = 8 (the number of 5 bits block)
    while i ≤ n do
        M_i = i^th 5 bits block.
        while j ≤ 7 do
            Convert 5 bits block, say M_i into the corresponding Gray value, say G
            Save S_i = G.
            if i% 2 == 0 {
            Save S_i in Qualified Set say Q.
            }
            else
            Save S_i in Forbidden Set say Q.
            Update M_i=G.
        end
    end
end
Convert the shares S_1 through S_7 into the corresponding decimal values.
```

Algorithm 4.2: Share Construction

Input: Seven Shares, $S_1, S_2, S_3, S_4, S_5, S_6, S_7$, each with dimension $r \times c$.

Output: The Secret Image , IM, with dimension $r \times c$.

```
1  Let i = 1, j = 1, k = 0
2  while i ≤ r do
3      while k ≤ r do
4          while j = 1 ≤ 7 do
5              Select the 5 pixel at a time from the ith raw of the jth
                share.
6              Convert each pixel value to its corresponding binary
                value.
7              Divide the binary value into 8 blocks each having 5 bits
                in length.
8              Save the Binary quantity in SS_j.
9              Update j = j + 1
10         end
11         Perform block by block XOR on the shares SS_1 through
            SS_7.
12         Convert the result into corresponding decimal value.
13         Save the result as the 5 pixel information of the secret
            image.
14         Update k = k + 5
15     end
16     Update i = i + 1
17 end
```

Example 4.2.1. Let us consider an example:

Suppose the first 5 pixels of the Secret Image is:

205, 163, 191, 240, 254

The participant shares are generated using Gray Code of the secret image. Initially the pixel values are converted into the binary form and divided into 5 bits blocks.

The binary equivalent of the pixel values are:
11001101 10100011 10111111 11110000 11111110
The 5 bit blocks are:
11001 10110 10001 11011 11111 11100 00111 11110

Apply the steps 4.1 to 4.10 of the Algorithm 1. The first 5 pixel of the first share is generated from the initial binary form of the pixel values. The first 5 pixel of the second share is generated from the first share. Likewise first 5 pixel values of the 7 shares are generated. Figure 4.1: shows the table having the seven shares related to the example that we have considered here.

Share No	*Shares in Binary Form*	*Shares in Decimal*
1	10101 11101 11001 10110 10000 10010 00100 10001	175 115 104 079 145
2	11111 10011 10101 11101 11000 11011 00110 11001	152 235 220 110 217 (Q-Set)
3	10000 11010 11111 10011 10100 10110 00101 10101	134 191 058 088 181
4	11000 10111 10000 11010 11110 11101 00111 11111	197 225 175 116 255 (Q-Set)
5	10100 11100 11000 10111 10001 10011 00100 10000	167 049 120 204 144
6	11110 10010 10100 11100 11001 11010 00110 11000	244 169 204 212 210 (Q-Set)
7	10001 11011 11110 10010 10101 10111 00101 10100	142 232 042 220 180

Figure 4.1: Showing Seven Shares

For reconstructing the secret image, the shares are collected and the 5 pixels from each share is considered at a time. As mentioned in the Algorithm 4.2, the binary values of the shares are divided into 5 bits blocks and finally the shares are XORed. It is shown in Figure 4.2. After XORing the resulting binary value is divided into 8 bits block and the corresponding decimal equivalent will be the first 5 pixels of the secret image.

Example 4.2.2. Consider the above example. The 7 shares are:
*Share No*1 : 175 115 104 079 145
*Share No*2 : 152 235 220 110 217

$Share\ No3$: 134 191 058 088 181
$Share\ No4$: 197 225 175 116 255
$Share\ No5$: 167 049 120 204 144
$Share\ No6$: 244 169 204 212 210
$Share\ No7$: 142 232 042 220 180
The 5 bits block of the binary equivalent of the shares
10101 11101 11001 10110 10000 10010 00100 10001
11111 10011 10101 11101 11000 11011 00110 11001
10000 11010 11111 10011 10100 10110 00101 10101
11000 10111 10000 11010 11110 11101 00111 11111
10100 11100 11000 10111 10001 10011 00100 10000
11110 10010 10100 11100 11001 11010 00110 11000
10001 11011 11110 10010 10101 10111 00101 10100
The 8 bits block of the XORed value of these shares will be:
11001101 10100011 10111111 11110000 11111110
Then the 5 pixel values of the secret image will be:
205, 163, 191, 240, 254

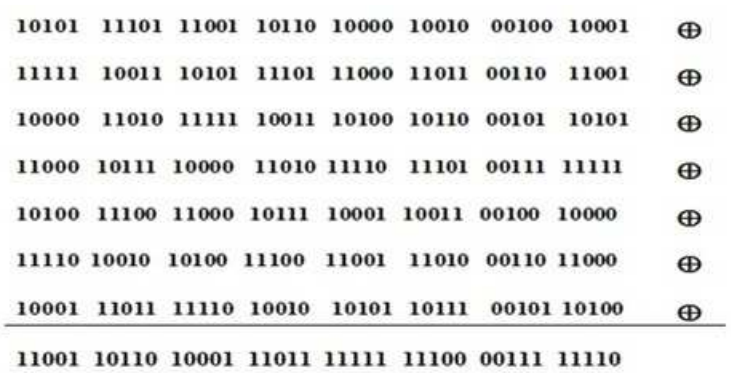


Figure 4.2: Example showing secret reconstruction from both Q-Set and F-Set-VCS(7,7)

4.2.2 VCS(3,3)

In VCS(3,3) scheme we are using only the shares in the Q-set. That means we are sharing the secret image into three shares (shares in the Q-set, mentioned in the previous algorithm) and for reconstructing the secret image back we are XORing the same three shares(Q-set Shares).

Consider the same example mentioned above.

The 5 Pixels in the secret image is:

205, 163, 191, 240, 254

The participant-shares are generated using Gray Code of the secret image. Initially the pixel values are converted into the binary form and divided into 5 bits blocks.

The binary equivalent of the pixel values are:

11001101 10100011 10111111 11110000 11111110

The 5 bit blocks are:

11001 10110 10001 11011 11111 11100 00111 11110

Figure 4.3: shows the table having the three shares(Q-set Shares) related to the example that we have considered here.

$ShareNo$	$SharesinBinaryForm$	$SharesinDecimal$
2	11111 10011 10101 11101 11000 11011 00110 11001	152 235 220 110 217 (Q-Set)
4	11000 10111 10000 11010 11110 11101 00111 11111	197 225 175 116 255 (Q-Set)
6	11110 10010 10100 11100 11001 11010 00110 11000	244 169 204 212 210 (Q-Set)

Figure 4.3: Showing Three Shares(Q-Set)

For reconstructing the secret image, all the three shares are XORed. It is shown in Figure 4.4.

11111	10011	10101	11101	11000	11011	00110	11001	⊕
11000	10111	10000	11010	11110	11101	00111	11111	⊕
11110	10010	10100	11100	11001	11010	00110	11000	⊕
11001	10110	10001	11011	11111	11100	00111	11110	

Figure 4.4: Example showing secret reconstruction from Q-Set-VCS(3,3)

4.3 Security Analysis

As mentioned in the Chapter 2, by performing the Gray Code to Binary Code conversion the secret may get exposed. By just performing the repeated Binary Code generation from one of the share image, the secret image can be reconstructed. That is there is no need of the collusion of all the shares. The section 2.2.3 contain the theory behind the conversion of Gray Code to Binary Code.

And as mentioned in the section, the block wise (5 bits blocks)shuffling across the shares will solve the problem.

4.4 Application

4.4.1 As Secret Sharing Scheme

Consider an organization's top level structure as shown in Figure 4.5: that is, the employees labeled A B and C is having high and equal privileges and D, E, F and G is having equal privileges which is lower than that of A, B and C. Then, if any secret information has to be shared among these employees we can use this scheme in two different ways. Two options to distribute the shares among these employees are listed below.

(a) Distribute the seven shares among them randomly. Here, for reconstructing the secret back, all seven employees should collude.

(b) Distribute the shares in the Q-Set among A, B and C. And the shares in the F-Set among D, E, F and G.

If we use the first option to distribute the shares among the employees, then the reconstruction is possible only by the collusion of all seven shares from all the employees. In some critical cases, we can use the second option to distribute the shares among high privileged employees (in this example employees labeled A, B, C). Here the reconstruction of the secret will be possible by collusion of shares in the Q-Set, which is distributed among the top level employees(Employees labeled A, B, C).

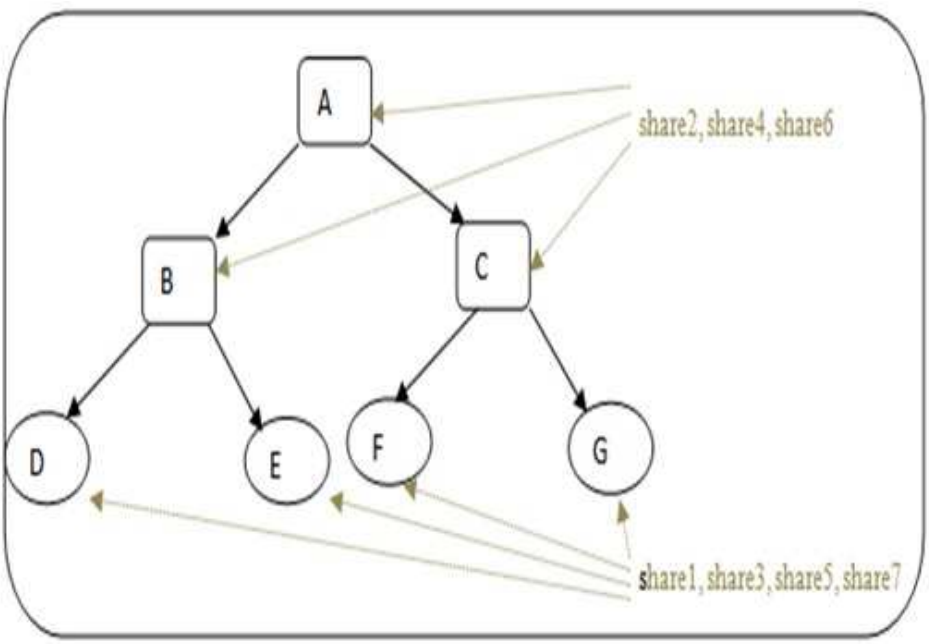


Figure 4.5: Sample Organization Structure

4.4.2 As Visual Data Encryption Scheme

The VCS scheme discussed in this chapter can be used as an encryption scheme for the images with a key. The encryption scheme using the proposed system can be represented as bellow:

CI=E(SI,K)

SI=D(CI,K)

Where SI is the secret image;

CI is the cipher image;

E is the encryption scheme; the Gray code generation

D is the decryption scheme; the Binary code generation

K is the key; Here the key, K, is a matrix of size 5×8, and the elements in the matrix should be between 1 and 7.

One of the most important point to mention in this scheme is, a matrix of pixels are considered from the secret image to process at a time. And the order of the matrix should be 5×5.

The encryption is block processing. Here the block size is 5 bits. The process is as follows:

1. Consider 5×5 pixels from the secret image (SI) at a time. {first 5 pixels from first 5 rows}
2. Convert the pixel values into binary form.
3. Divide the binary value into 5 bits blocks; {8 blocks will be there in one row.Total $5 \times 8 = 40$ blocks, each of length 5 bits}
4. Read key matrix of the order 5×8;say, K.
5. Repeat the following steps until all the blocks are processed.
    5.1. Pick a block and its corresponding digit from the key.
    5.2. Covert the block into the gray code, number of times the digit from the key is having.
6. Divide the resulting binary form into 8 bits block.
7. Convert the binary value into corresponding decimal value and save it as the 5×5 pixel values of the Cipher Image(CI).

8. Stop.

Like encryption, the Decryption is also blocking processing. The process is as follows;

1. Consider 5×5 pixels from the Cipher Image (CI) at a time.
2. Convert the pixel values into binary form.
3. Divide the binary value into 5 bits blocks.
4. Read key matrix of the order 5×8; K.
5. Repeat the following steps until all the blocks are processed.
   - 5.1. Pick a block and its corresponding digit from the key
   - 5.2. Covert the block into the binary code, number of times the digit from the key is having.
6. Divide the resulting binary form into 8 bits block.
7. Convert the binary value into corresponding decimal value and save it as the 5×5 pixel values of the Secret Image(SI).
8. Stop.

Encryption and decryption example is shown in Figure 4.6 and 4.7.

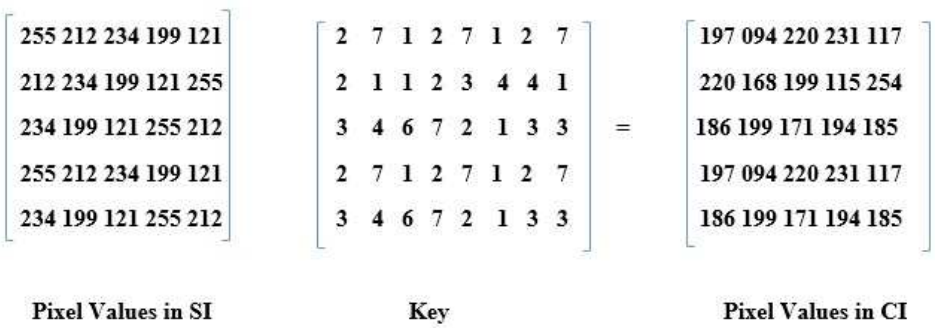


Figure 4.6: Encryption -Example

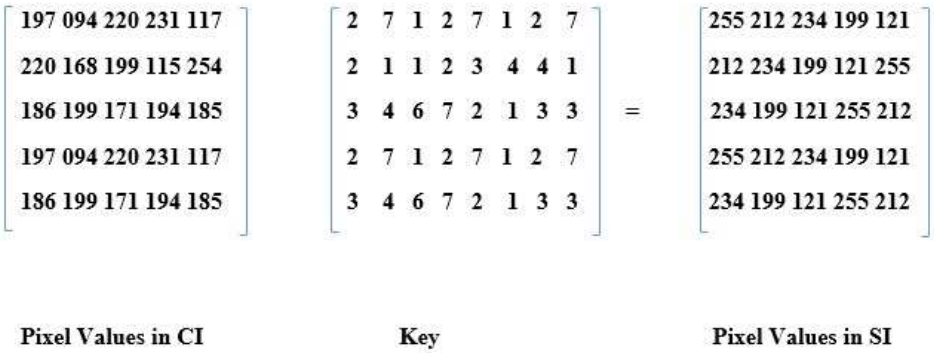


Figure 4.7: Decryption-Example

4.5 Concluding Remarks

As we have discussed here in this chapter the new scheme can be used as an encryption scheme along with a key, in addition to the secret sharing purpose. Here we have considered the gray scale images. This scheme can be extended to color images also. In the case of color images we have to perform the algorithm separately on three channels red, blue and green. The main advantage of the proposed scheme is, there is no pixel expansion, that means no information loss.

Chapter 5

Visual Secret Sharing using Newton Interpolation Polynomial and Mod Operator with PNG Images

5.1 Introduction

In this chapter we propose a method to share images, in Portable Network Graphics (PNG) format, with honest participants using Newton Interpolation Polynomial equations and Mod operator. The main features of the proposed scheme is, the involvement of a key, both in secret sharing phase as well as in the secret reconstruction phase and the key can be used to verify the honesty of the participants also. We consider gray scale images in PNG format for the proposed scheme. As first part, a brief note on Newton Polynomial Interpolation is mentioned and as second part an n-out of-n Visual Cryptography Scheme, $VCS(n, n)$, is proposed. Then to

enhance the security an additional factor, the key, is introduced and the enhanced versions of the scheme with one key, $VCS(n, n, k)$, and with n keys, $VSS(n, n, k_1\ to\ k_n)$, are also mentioned.

5.2 Polynomial Interpolation

In mathematics, Interpolation is the estimation of the value of a function of x, $(f(x))$, from certain known values of the function. If $x_0 < \cdots < x_n$ and $y_0 = f(x_0), \ldots, y_n = f(x_n)$ are known, and if $x_0 < x < x_n$, then the estimated value of $f(x)$ is said to be an interpolation. If $x < x_0$ or $x > x_n$, the estimated value of $f(x)$ is said to be an extrapolation.

If $x_0, \ldots, x_n$ are given, along with corresponding values $y_0, \ldots, y_n$, interpolation may be regarded as the determination of a function $y = f(x)$ whose graph passes through the $n + 1$ points, (x_i, y_i) for $i = 0, 1, \ldots, n$. There are infinitely many such functions, but the simplest is a Polynomial Interpolation function $y = p(x) = a_0 + a_1x + \cdots + a_nx^n$ with constant a_i's such that $p(x_i) = y_i$ for $i = 0, \ldots, n$. There is exactly one such interpolating polynomial of degree n or less.

Mainly two Polynomial Interpolation methods are there:

1. Newton Polynomial Interpolation.

2. Lagrange Polynomial Interpolation.

In this section we have given short notes on both the methods.

5.2.1 Note on Newton Polynomial Interpolation

Given a set of $k + 1$ data points,

$(x_0, y_0), \ldots, (x_k, y_k)$, where no two x_j are the same.

The interpolation polynomial in the Newton form is a linear combination of Newton basis polynomials:

$\mathrm{N}(x) = \sum_{j=0}^{k} t_j \; n_j \; (x)$

with the Newton basis polynomials defined as

$n_j(x) = \prod_{i=0}^{j-1} \; x\text{-}x_i$

for $j > 0$ and $n_0(x) = 1$

The coefficients are defined as

$t_j = [y_0, y_1, \ldots, y_j]$,where $[y_0, y_1, \ldots, y_j]$ is the notation for divided differences.

Thus the Newton polynomial can be written as

$$\mathrm{N}(x) = [y_0] + [y_0, y_1](x - x_0) + \cdots + [y_0, \ldots, y_k](x - x_0)(x - x_1) \ldots (x - x_{k-1})$$

5.2.2 How To Use Newton Interpolation Polynomial To Share And Reconstruct The Secret From A Set Of Point Pairs(X, Y)?

Let Secret, D=10.

Consider a 3- out of-3 secret sharing scheme, in which *three* shares are generated during secret sharing phase. Here a set having 3 point pairs (x, y) are generated and the y values are treated as the shares of the secret. And for the reconstruction of the secret all of the *three* shares are needed. The process is as follows:

We pick two random numbers as the coefficients. Let it be 5 and 2.

This gives us the polynomial $f(x) = 5x^2 + 2x + 10$.

Now find f(1), f(2), f(3).

f(1) = 17

f(2) = 34

f(3) = 61

So the 3 points generated from the polynomial are,(1, 17), (2, 34), (3, 61). And for the secret D=10, the constructed share values are:

Share 1 = 17

Share 2 = 34

Share 3 = 61

And for the reconstruction of the secret, the polynomial is interpolated over the range $1 \leq x \leq 3$. For that we have to construct the corresponding Newton's divided difference table. And it is shown in Table 5.1:

$x_0 = 1$	$y_0 = t_0 = 17$		
	$[y_0, y_1] = \frac{y_1 - y_0}{x_1 - x_0} = 17$	$t_1 = 17$	
$x_1 = 2$	$y_1 = 34$	$[y_0, y_1, y_2] = \frac{[y_1, y_2] - [y_0, y_1]}{x_2 - x_0}$	$t_2 = 5$
	$[y_1, y_2] = \frac{y_2 - y_1}{x_2 - x_1} = 27$		
$x_2 = 3$	$y_2 = 61$		

Table 5.1: Divided Difference Table :1

The interpolating polynomial is:

$$N(x) = [y_0] + [y_0, y_1](x - x_0) + [y_0, y_1, y_2](x - x_0)(x - x_1)$$

$$N(x) = 17 + 17(x - 1) + 5(x - 1)(x - 2)$$

To reconstruct the secret D, we need to take care of the constant part of Newtons polynomial. We can ignore the x-es:

So the secret, **D = 17+17*(-1) + 5*(-1)*(-2)= 10**

5.2.3 Note on Lagrange Polynomial Interpolation

The Lagrange Interpolating Polynomial is the polynomial $P(x)$ of degree $\leq (n-1)$ that passes through the points $(x_1, y_1 = f(x_1))$, $(x_2, y_2 = f(x_2))$,...,$(x_n, y_n = f(x_n))$ and is given by:

$$\mathrm{P}(x) = \sum_{j=1}^{n} P_j\ (x)$$

where

$$P_j(x) = y_j \prod_{k=1, k\neq j}^{n} \frac{x - x_k}{x_j - x_k}$$

Written explicitly,

$$P(x) = y_1 \frac{(x-x_2)(x-x_3)...(x-x_n)}{(x_1-x_2)(x_1-x_3)...(x_1-x_n)} + y_2 \frac{(x-x_2)(x-x_3)...(x-x_n)}{(x_2-x_1)(x_2-x_3)...(x_2-x_n)} + \ldots + y_n \frac{(x-x_2)(x-x_3)...(x-x_n)}{(x_n-x_1)(x_n-x_2)...(x_n-x_{n-1})}$$

Formula was first published by Waring (1779), rediscovered by Euler in 1783, and published by Lagrange in 1795.

Consider an example:

Interpolate $f(x) = x^3$ over the range $1 \leq x \leq 3$

The points are:

$(x_0, y_0) = (1, 1)$

$(x_1, y_1) = (2, 8)$

$(x_2, y_2) = (3, 27)$

The interpolating polynomial is:

$P(x) = 1\ \frac{(x-2)(x-3)}{(1-2)(1-3)} + 8\ \frac{(x-1)(x-3)}{(2-1)(2-3)} + 27\ \frac{(x-1)(x-2)}{(3-1)(3-2)}$

$P(x) = 6x^2 - 11x + 6$

5.2.4 How To Use Lagrange Interpolation Polynomial To Share And Reconstruct The Secret From A Set Of Point Pairs(X, Y)?

Let Secret, D=10.

Consider a 3- out of-3 secret sharing scheme, in which *three* shares are generated during secret sharing phase. Here a set having 3 point pairs (x, y) are generated and the y values are treated as the shares of the secret. And for the reconstruction of the secret all of the *three* shares are needed. The process is as follows:

We pick two random numbers as the coefficients. Let it be 5 and 2.

This gives us the polynomial $f(x) = 5x^2 + 2x + 10$.

Now find f(1), f(2), f(3).

f(1) = 17

f(2) = 34

f(3) = 61

So the 3 points generated from the polynomial are,(1, 17), (2, 34), (3, 61). And for the secret D = 10, the constructed share values are:

Share 1 = 17

Share 2 = 34

Share 3 = 61

To reconstruct the secret, D, Interpolate the polynomial over the range $1 \leq x \leq 3$ and the interpolating polynomial is

$P(x) = 17\ \frac{(x-2)(x-3)}{(1-2)(1-3)} + 34\ \frac{(x-1)(x-3)}{(2-1)(2-3)} + 61\ \frac{(x-1)(x-2)}{(3-1)(3-2)}$

From this Lagrange's polynomial we need to take care of the constant part only:

$l_0 = \frac{(x-2)}{(1-2)} \cdot \frac{(x-3)}{(1-3)} = \frac{(x-2)(x-3)}{2}$

$l_1 = \frac{(x-1)}{(2-1)} \cdot \frac{(x-3)}{(2-3)} = \frac{(x-1)(x-3)}{-1}$

$l_2 = \frac{(x-1)}{(3-1)} \cdot \frac{(x-2)}{(3-2)} = \frac{(x-1)(x-2)}{2}$

We can reconstruct the secret D, by ignoring the x-es, considering only the constant parts:

$D = 17.\frac{(-2)(-3)}{2} + 34.\frac{(-1)(-3)}{-1} + 61.\frac{(-1)(-2)}{2}.$

The secret, D=10

5.3 Proposed Scheme: Visual Secret Sharing Scheme using Polynomial Interpolation

Now, focus on the visual secret sharing scheme that uses polynomial interpolation. Here either Lagrange Interpolation or Newton Interpolation can be used. We have explained the method in this chapter using Newtons Interpolation method. It is labelled as $VSS(n, n)$ scheme. There are two algorithms discussed in this section. Algorithm 5.1 is all about the sharing of secret image. Algorithm 5.2 is all about reconstructing the secret image from the shares.

Definition 5.3.1. VSS(n,n): IMG is a the secret image, the image is divided or shared into n shares and for the reconstruction of the secret all of the n shares are needed.

Algorithm 5.1: Share Construction-Newton Polynomial Interpolation

Input: The Secret Image , IMG
Output: n shares of same size

Preprocesing : Convert the input image, IMG, into gray scale image, IMG_MODI. Assume that the image size is $N \times M$. where N is the number of rows and M is the number of columns in the original image.
[rows =number of rows in the image, cols= number of columns in the image]
$i = 0, j = 0$
while $i \leq rows$ **do**
  while $j \leq cols$ **do**
    Select the pixels values at (i, j) of IMG_MODI.
    Select n, the number of shares to be generated.
    Select $n-1$ random numbers as the coefficients say $C_1, C_2, \ldots, C_{n-2}, C_{n-1}$ of polynomial.
    Construct polynomial of the form;
    $f(x)$=$C_1x^{n-1} + C_2x^{n-2} + \cdots + C_{(n-2)}x^2 + C_{(n-1)}x^1 + S$
    Find $f(1), f(2), \ldots, f(n)$ values. Say $y_1, y_2, \ldots, y_n$
    $k = 1$
    while $k \leq rows$ **do**
      Assign A_k= $\lfloor y_k/256 \rfloor$.
      Modify y_k= $y_k \pmod{256}$.
      Save y_k as pixel values at (i, j) of the share k.
      Embed A_k in alpha channel at (i,j) of the share k.
    end
  end
end

Algorithm 5.2: Secret Reconstruction-Newton Polynomial Interpolation

Input: n shares of the same length (say $share_n$)
Output: the secret image/ secret visual data

[rows = number of rows in the image, cols= number of columns in the image]
$i = 0, j = 0, k = 1$
while $i \leq rows$ **do**
  while $j \leq cols$ **do**
    while $k \leq n$ **do**
      Select the pixels values at (i, j) of $share_k$ say S_k
      Select the value that is embedded in the alpha channel, say Ak
      Modify $S_k = 256 * A_k + S_k$
      Save $y_k = S_k$
    end
    Compute N$(x) = \sum_{j=0}^{k} t_j \; n_j \; (x)$
    where $n_j(x) = \prod_{i=0}^{j-1} x\text{-}x_i$
    t_j=$[y_0, y_1, \ldots, y_j]$
    where $[y_0, y_1, \ldots, y_j]$ is the notation for divided differences.
    Save S as the pixel values at (i, j) of the secret.
  end
end

Consider an example

Phase 1 : Share Generation

If the pixel value of the secret image is 255 and we construct 3 out of 3 visual cryptography scheme, then we have to construct a polynomial of degree 2. For that select 2 random coefficients, say 99 and 12. Suppose the constructed polynomial is as follows:

$f(x)$=$(99x^2 + 12x + 255) \pmod{256}$

Calculated the values f(1), f(2) and f(3).

The value f (1) = 366 (mod 256) = 110

The pixel value of the share 1 is 110.

The value to embed in the alpha channel is, say $A_1 = \lfloor 366/256 \rfloor = 1$

The value f (2) = 675 (mod 256) = 163

The pixel value of the share 2 is 163.

The value to embed in the alpha channel is, say $A_2 = \lfloor 675/256 \rfloor = 2$

The value f (3) = 1182 (mod 256) = 158

The pixel value of the share 3 is 158.

The value to embed in the alpha channel is, say $A_3 = \lfloor 1182/256 \rfloor = 4$

The pixel values and values embedded in alpha channel of shares are:

Share 1 = 110 value embedded in alpha channel = 1

Share 2 = 163 value embedded in alpha channel = 2

Share 3 = 158 value embedded in alpha channel = 4

Phase 2 : Secret Reconstruction

Consider the previously generated shares:

Share 1=110 value embedded in alpha channel = 1

Share 2=163 value embedded in alpha channel = 2

Share 3=158 value embedded in alpha channel = 4

Modify the value as following:

Share 1 = 256*1+110 = 366

Share 2 = 256*2+163 = 675

Share 3 = 256*4+158 = 1182

The corresponding Newton's divided difference table is shown in Table 5.2.

Compute S = 366+309(-1)+99(-1)(-2). Thus secret reconstructed.

S = 255.

$x_0 = 17$	$y_0 = t_0 = 366$		
	$[y_0, y_1] = \frac{y_1 - y_0}{x_1 - x_0} = 309$	$t_1 = 309$	
$x_1 = 18$	$y_1 = 675$	$[y_0, y_1, y_2] = \frac{[y_1, y_2] - [y_0, y_1]}{x_2 - x_0}$	$t_2 = 99$
	$[y_1, y_2] = \frac{y_2 - y_1}{x_2 - x_1} = 507$		
$x_2 = 19$	$y_2 = 1182$		

Table 5.2: Divided Difference Table :2

5.4 Enhancement on the Proposed System

A key factor can be introduced as an enhancement to the proposed scheme. In the enhanced version a secret key is used in the share construction phase. The same key is required in the secret reconstruction phase also. That means, only by colluding all the shares the secret cannot be reconstructed. The proper reconstruction of the secret requires the key element along with the shares. The algorithm for the enhanced visual secret sharing scheme, $VSS(n, n, k)$, that involves the key, k, is as mentioned in Algorithm 5.3(Share Construction) and Algorithm 5.4(Secret Reconstruction).

Definition 5.4.1. VSS(n,n,k): IMG is a the secret image and k is the key used for both encryption and decryption, the image is divided or shared into n shares using k and for the reconstruction all the n shares are needed along with k.

Algorithm 5.3: Share Construction $VSS(n, n, k)$

Input: The Secret Image , IMG
Output: n shares of same size

Preprocesing : Convert the input image, IMG, into gray scale image, IMG_MODI. Assume that the image size is $N \times M$. where N is the number of rows and M is the number of columns in the original image.
[rows =number of rows in the image, cols= number of columns in the image]
$i = 0, j = 0$
while $i \leq rows$ **do**
  while $j \leq cols$ **do**
    Select the pixels values at (i, j) of IMG_MODI.
    Select n, the number of shares to be generated.
    Select the key ,k. Select $n-1$ random numbers as the coefficients say $C_1, C_2, \ldots, C_{n-2}, C_{n-1}$ of polynomial.
    Construct polynomial of the form:
    $f(x) = C_1 x^{n-1} + C_2 x^{n-2} + \cdots + C_{n-2} x^2 + C_{n-)} x^1 + S$
    Find $f(k), f(k+1), \ldots, f((k+(n-1))$ values. Say $y_1, y_2, \ldots, y_n$
    $p = 1$
    while $p \leq n$ **do**
      Assign $A_p = \lfloor y_p / 256 \rfloor$.
      Modify $y_p = y_p \pmod{256}$.
      Save y_p as pixel values at (i, j) of the share p.
      Embed A_p in alpha channel at (i,j) of the share p.
    end
  end
end

Algorithm 5.4: Secret Reconstruction $VSS(n, n, k)$

Input: n shares of the same length (say $share_n$), the key k
Output: the secret image/ secret visual data

[rows = number of rows in the image, cols = number of columns in the image]
$i = 0, j = 0, k = 1$
while $i \leq rows$ **do**
  while $j \leq cols$ **do**
    while $k \leq n$ **do**
      Select the pixels values at (i, j) of $share_k$ say S_k
      Select the value that is embedded in the alpha channel, say Ak
      Modify $S_k = 256 * A_k + S_k$
      Save $y_k = S_k$
    end
    Compute N(x) $=\sum_{j=k}^{k+n-1} t_j \; n_j \; (x)$
    where $n_j(x) = \prod_{k \leq m \leq (k+(n-1)), m \neq j} \; x\text{-}x_m$
    $t_j = [y_0, y_1, \ldots, y_j]$
    where $[y_0, y_1, \ldots, y_j]$ is the notation for divided differences.
    Save S as the pixel values at (i, j) of the secret.
  end
end

Consider the previous example

Share generation:

If the pixel value of the secret image is 255 and we are constructing 3 out of 3 visual cryptography scheme, then we have to construct a polynomial of degree 2. For that select 2 random coefficients, say 99 and 12. Suppose the constructed polynomial is as follows:

$f(x) = (99x^2 + 12x + 255) \pmod{256}$

Select a key value, key = 17

Then generate f (key) through f(key+(3-1))

f(17) = (99*(17)2 +12*17+255) (mod 256) = 29070 (mod 256) = 142

f(18) = (99*(18)2+12*18+255) (mod 256) = 32547 (mod 256) = 35

f(19) = (99*(19)2+12*19+255) (mod 256) = 36222 (mod 256) = 126

So the pixel values in each share:

Share 1 : 142

Share 2 : 35

Share 3 : 126

The alpha channel entries in each share are:

Alpha channel entry in share 1; = $\lfloor 29070/256 \rfloor = 113$

Alpha channel entry in share 2; = $\lfloor 32547/256 \rfloor = 127$

Alpha channel entry in share 3; = $\lfloor 36222/256 \rfloor = 141$

Reconstruction:

Share 1 = 142 value embedded in alpha channel = 113

Share 1 = 35 value embedded in alpha channel = 127

Share 1 = 126 value embedded in alpha channel = 141

Modify the value as following:

Share 1 = 256*113+142 = 29070

Share 2 = 256*127+35 = 32547

Share 3 = 256*141+126 = 36222

The key is 17.

The secret is reconstructed from the Newton's divided difference table.The corresponding Newton's divided difference table is shown in Table 5.3.

$x_0 = 17$	$y_0 = t_0 = 29070$		
	$[y_0, y_1] = \frac{y_1 - y_0}{x_1 - x_0} = 3477$	$t_1 = 3477$	
$x_1 = 18$	$y_1 = 32547$	$[y_0, y_1, y_2] = \frac{[y_1, y_2] - [y_0, y_1]}{x_2 - x_0}$	$t_2 = 99$
	$[y_1, y_2] = \frac{y_2 - y_1}{x_2 - x_1} = 3675$		
$x_2 = 19$	$y_2 = 36222$		

Table 5.3: Divided Difference Table :3

Compute S = 29070+3477(-17)+99(-17)*(-18). Thus secret reconstructed.

S=255.

From the above example it is clear that the key enhances the security of the scheme. The $VSS(n, n, k)$ can be further extended into $VSS(n, n, k_1$ *to* $k_n)$. That is visual secret sharing n-out of-n scheme with n keys, each key for each participant. The n keys enhance the security of the scheme. The overview of the scheme is depicted in the Figure 5.1.

Definition 5.4.2. $VSS(n, n, k_1 to k_n)$**:** IMG is a the secret image and $k_1, k_2, \ldots, k_n$ are a set of n keys, then the image is divided or shared into n shares using n keys and for the reconstruction all the n shares and n keys are needed.

Consider the example

Share Generation: If the pixel value of the secret image is 255 and if we are constructing 3 out of 3 visual cryptography scheme, we have to construct a polynomial of degree 2. For that select 2 random coefficients say, 99 and 12. Suppose the constructed polynomial is as follows:

$f(x) = (99x^2 + 12x + 255) \pmod{256}$

If the secret keys assigned for the 3 participants are: 7, 11 and 13.

Then the share constructions will be as follows:

Share 1:

f(7) = (99*(7)2 +12*(7)+255) (mod 256) = 5190 (mod 256) = 70.

The pixel value in share 1 is 70.

The value embedded in the alpha channel is = $\lfloor 5190/256 \rfloor = 20$

Share 2:

f(11) = (99*(11)2 +12*(11)+255) (mod 256) = 12366 (mod 256) = 78.

The pixel value in share 2 is 78.

The value embedded in the alpha channel is = $\lfloor 12366/256 \rfloor = 48$

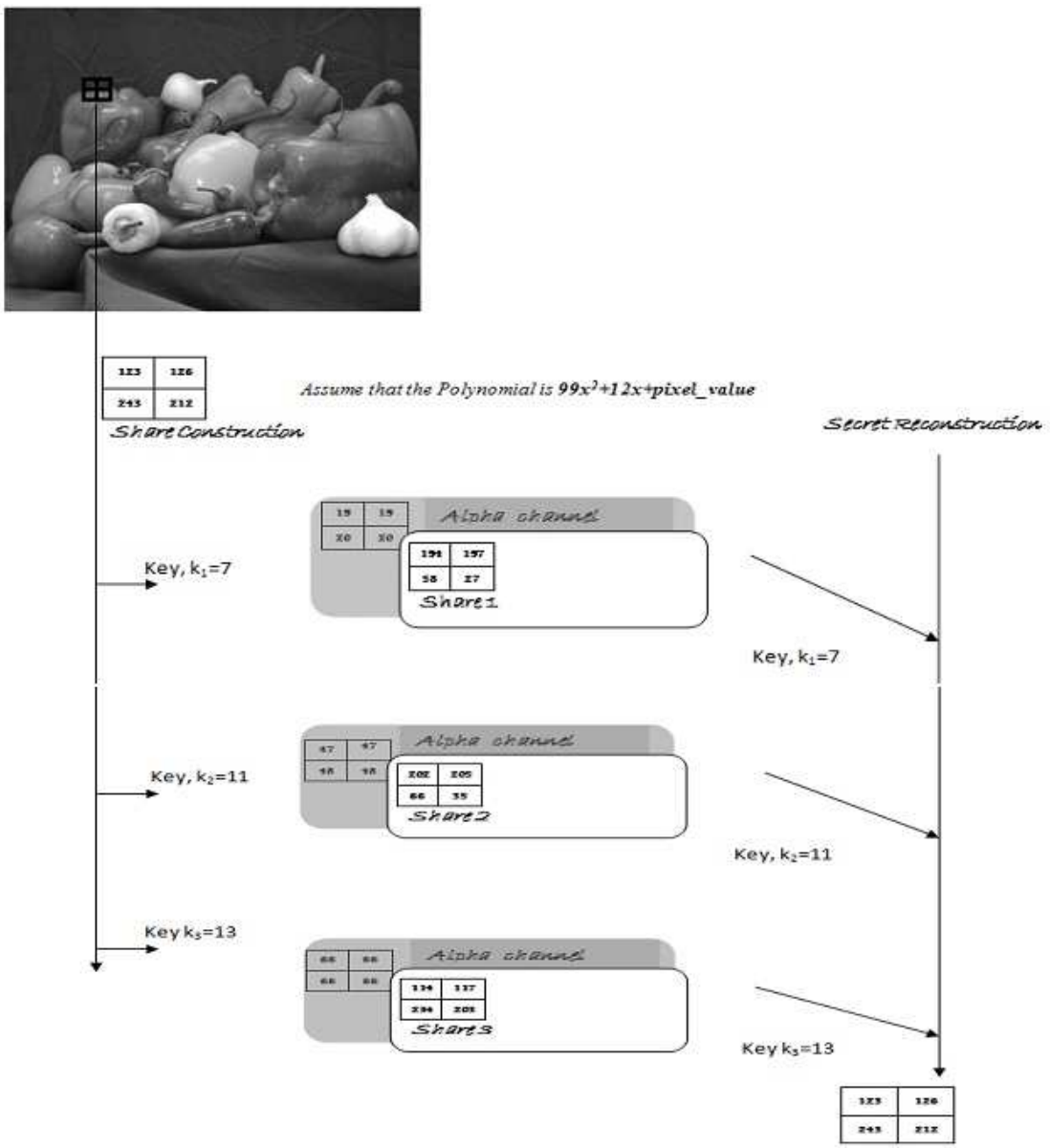


Figure 5.1: Visual Secret Sharing Scheme Using Newton Polynomial Interpolation

Share 3:

f(13) = (99*(13)2 +12*(13)+255) (mod 256) = 17142 (mod 256) = 246.

The pixel value in share 3 is 246.

The value embedded in the alpha channel is = $\lfloor 17142/256 \rfloor = 66$

The pixel values and values embedded in alpha channel of shares are:

Share 1 = 70 value embedded in alpha channel = 20

Share 2 = 78 value embedded in alpha channel = 48

Share 3 = 246 value embedded in alpha channel = 66

Secret reconstruction:

Secret can only be reconstructed if the shares along with the correct key are given.Initially modify the share values as follows:

Share 1 = 256*20+70 = 5190 , the key $k_1 = 7$

Share 2 = 256*48+78 = 12366, the key $k_2 = 11$

Share 3 = 256*66+246 = 17142, the key $k_3 = 13$

The corresponding Newton's divided difference table is shown in Table 5.4.

$x_0 = 7$	$y_0 = t_0 = 5190$		
	$[y_0, y_1] = \frac{y_1 - y_0}{x_1 - x_0} = 1794$	$t_1 = 1794$	
$x_1 = 11$	$y_1 = 12366$	$[y_0, y_1, y_2] = \frac{[y_1, y_2] - [y_0, y_1]}{x_2 - x_0}$	$t_2 = 99$
	$[y_1, y_2] = \frac{y_2 - y_1}{x_2 - x_1} = 2388$		
$x_2 = 13$	$y_2 = 171421$		

Table 5.4: Divided Difference Table :4

Then the secret, S is = 5190+1794(-7)+99(-7)*(-13). Thus secret reconstructed.

S=255

5.5 Concluding Remarks

We have proposed a visual secret sharing scheme using Polynomial Interpolation and Modular operation with PNG image that uses keys to enhance the security. The method mainly focuses on gray images. The important feature in this method is the involvement of the keys. The reconstruction of the secret requires not only the n shares but also the keys. Here the key can be used to authenticate the shares. The main advantage of this scheme is, there is no pixel expansion and thus no information loss. This method can be extended into the color images also.

Chapter 6

Visual Secret Sharing Using POB Number System and CRT

6.1 Introduction

In this chapter we introduce a new visual secret sharing scheme, to share the secret among n participants, i.e. a n-out-of-n secret sharing scheme, based on a new number system called Permutation Ordered Binary Number System (POB number system) and Chinese Remainder Theorem (CRT). This scheme is an efficient one with respect to security and performance. Even though the size of the shares is more than the size of the secret, the reconstructed secret will have the same size as the original secret. So the scheme is a loss less scheme. As first part a brief note on POB number system and CRT is mentioned and as second part a new n-out of-n VSS is proposed.

6.2 Theory Behind The Proposed Scheme

6.2.1 Permutation Ordered Binary Number System

POB, Permutation Ordered Binary, number system is a general number system with two nonnegative integral parameters, n and r, where $n \geq r$ developed. It is developed by A. Sreekumar et. al [SS09] as part of his research work. This number system is found to be very useful and more efficient than the conventional number system under use. In [SS09] they have used POB number system in a newly introduced secret sharing scheme.

The system is denoted by **POB(n,r)**. In this number system, it is possible to represent all integers in the range 0, ..., $\binom{n}{x} - 1$ as binary string, say B= b_{n-1} b_{n-2}.... b_0, of length n, and having exactly r 1s. Here B is called the POB number of that particular integer. Each digit of this number say, b_j, is associated with its position value given by

$b_j \times \binom{j}{p_j}$, where

$p_j = \sum_{i=0}^{j} b_i$

and the value represented by the POB number B, denoted by $V(B)$, will be the sum of the position values of all of the digits. That is:

$V(B) = \sum_{j=0}^{n-1} b_i \binom{j}{p_j}$

For example, 111001000 is a $POB(9, 4)$ number with value of 123. In this example;

$p_3 = 1$

$p_6 = 2$

$p_7 = 3$

$p_8 = 4$
$V(111001000)$ =1× $\binom{3}{1}$ + 1× $\binom{6}{2}$ + 1× $\binom{7}{3}$ + 1× $\binom{8}{4}$
=1 × 3+ 1 × 15+ 1 × 35 +1 × 70
=123

6.2.2 The Algorithm For Finding POB Numbers

For a given pair of parameters n and r with $r \leq n$, the algorithm takes three inputs: n, r, and val with $0 \leq val \leq \binom{n}{r} - 1$, or in a $POB(n, r)$ number system, if a POB value, say val is given, then the below mentioned algorithm generates $POB(n, r)$ number say B, such that $V(B) = val$.

Algorithm 6.1, Generate POB- number corresponding to a given POB value.

Algorithm 6.1: POB number corresponding to POB value

Input: Three numbers n,r, and val with $r \leq n$ and $0 \leq val \leq \binom{n}{r} - 1$
. **Output:** The POB number B= b_{n-1} b_{n-2}.... b_0
Let $j = n$ and $temp = val$
For k= r down to 1 do the following steps
(a) Repeat {
(b) i = j-1
(c) p = $\binom{j}{k}$
(d) if ($temp \geq p$) then (i) temp = temp-p ; (ii) $b_j = 1$
(g) else $b_j = 0$
(h) until ($b_j = 1$)
If ($j \geq 0$)
for k = j-1 down to 0 do the following
(a) $b_k = 0$

Remark: B= b_{n-1} b_{n-2}.... b_0 is the POB-number.

Lemma 1: Algorithm 1 generates the POB-number (n, r) corresponding to the given POB-value.

Table 6.1 shows the POB (9, 4) Number System.

Table 6.1: POB(9,4)Number System

$POBValue[V(b)]$	$POBNumber[B]$	$DecimalEquivalent$
0	0 0 0 0 0 1 1 1 1	15
1	0 0 0 0 1 0 1 1 1	23
2	0 0 0 0 1 1 0 1 1	27
3	0 0 0 0 1 1 1 0 1	29
4	0 0 0 0 1 1 1 1 0	30
5	0 0 0 1 0 0 1 1 1	39
6	0 0 0 1 0 1 0 1 1	43
7	0 0 0 1 0 1 1 0 1	35
8	0 0 0 1 0 1 1 1 0	36
9	0 0 0 1 1 0 0 1 1	51
10	0 0 0 1 1 0 1 0 1	53
11	0 0 0 1 1 0 1 1 0	54
12	0 0 0 1 1 1 0 0 1	57
13	0 0 0 1 1 1 0 1 0	58
14	0 0 0 1 1 1 1 0 0	60
15	0 0 1 0 0 0 1 1 1	71
16	0 0 1 0 0 1 0 1 1	75
17	0 0 1 0 0 1 1 0 1	77
18	0 0 1 0 0 1 1 1 0	78
19	0 0 1 0 1 0 0 1 1	83
20	0 0 1 0 1 0 1 0 1	85
21	0 0 1 0 1 0 1 0 1	86
22	0 0 1 0 1 1 0 0 1	89
23	0 0 1 0 1 1 0 1 0	90
24	0 0 1 0 1 1 1 0 0	92

Table 6.1: POB(9,4)Number System

$POBValue[V(b)]$	$POBNumber[B]$	$DecimalEquivalent$
25	0 0 1 1 0 0 0 1 1	99
26	0 0 1 1 0 0 0 1 1	101
27	0 0 1 1 0 0 1 1 0	102
28	0 0 1 1 0 1 0 0 1	105
29	0 0 1 1 0 1 0 1 0	106
30	0 0 1 1 0 1 1 0 0	108
31	0 0 1 1 1 0 0 0 1	113
32	0 0 1 1 1 0 0 1 0	114
33	0 0 1 1 1 0 1 0 0	116
34	0 0 1 1 1 1 0 0 0	120
35	0 1 0 0 0 0 1 1 1	135
36	0 1 0 0 0 1 0 1 1	139
37	0 1 0 0 0 1 1 0 1	141
38	0 1 0 0 0 1 1 1 0	142
39	0 1 0 0 1 0 0 1 1	147
40	0 1 0 0 1 0 1 0 1	149
41	0 1 0 0 1 0 1 1 0	150
42	0 1 0 0 1 1 0 0 1	153
43	0 1 0 0 1 1 0 1 0	154
44	0 1 0 0 1 1 1 0 0	156
45	0 1 0 1 0 0 0 1 1	163
46	0 1 0 1 0 0 1 0 1	165
47	0 1 0 1 0 0 1 1 0	166
48	0 1 0 1 0 1 0 0 1	169
49	0 1 0 1 0 1 0 1 0	170

Table 6.1: POB(9,4)Number System

$POBValue[V(b)]$	$POBNumber[B]$	$DecimalEquivalent$
50	0 1 0 1 0 1 1 0 0	171
51	0 1 0 1 1 0 0 0 1	177
52	0 1 0 1 1 0 0 1 0	178
53	0 1 0 1 1 0 1 0 0	180
54	0 1 0 1 1 1 0 0 0	184
55	0 1 1 0 0 0 0 1 1	195
56	0 1 1 0 0 0 1 0 1	197
57	0 1 1 0 0 0 1 1 0	198
58	0 1 1 0 0 1 0 0 1	201
59	0 1 1 0 0 1 0 1 0	202
60	0 1 1 0 0 1 1 0 0	204
61	0 1 1 0 1 0 0 0 1	209
62	0 1 1 0 1 0 0 1 0	210
63	0 1 1 0 1 0 1 0 0	212
64	0 1 1 0 1 1 0 0 0	216
65	0 1 1 1 0 0 0 0 1	225
66	0 1 1 1 0 0 0 1 0	226
67	0 1 1 1 0 0 1 0 0	228
68	0 1 1 1 0 1 0 0 0	232
69	0 1 1 1 1 0 0 0 0	240
70	1 0 0 0 0 0 1 1 1	263
71	1 0 0 0 0 1 0 1 1	267
72	1 0 0 0 0 1 1 0 1	269
73	1 0 0 0 0 1 1 1 0	270
74	1 0 0 0 1 0 0 1 1	275

Table 6.1: POB(9,4)Number System

$POBValue[V(b)]$	$POBNumber[B]$	$DecimalEquivalent$
75	1 0 0 0 1 0 1 0 1	277
76	1 0 0 0 1 0 1 1 0	278
77	1 0 0 0 1 1 0 0 1	281
78	1 0 0 0 1 1 0 1 0	282
79	1 0 0 0 1 1 1 0 0	284
80	1 0 0 1 0 0 0 1 1	291
81	1 0 0 1 0 0 1 0 1	293
82	1 0 0 1 0 0 1 1 0	294
83	1 0 0 1 0 1 0 0 1	297
84	1 0 0 1 0 1 0 1 0	298
85	1 0 0 1 0 1 1 0 0	300
86	1 0 0 1 1 0 0 0 1	305
87	1 0 0 1 1 0 0 1 0	306
88	1 0 0 1 1 0 1 0 0	308
89	1 0 0 1 1 1 0 0 0	312
90	1 0 1 0 0 0 1 0 1	323
91	1 0 1 0 0 0 1 0 1	325
92	1 0 1 0 0 0 1 1 0	326
93	1 0 1 0 0 1 0 0 1	329
94	1 0 1 0 0 1 0 1 0	330
95	1 0 1 0 0 1 1 0 0	332
96	1 0 1 0 1 0 0 0 1	337
97	1 0 1 0 1 0 0 1 0	338
98	1 0 1 0 1 0 1 0 0	340
99	1 0 1 0 1 1 0 0 0	344

Table 6.1: POB(9,4)Number System

$POBValue[V(b)]$	$POBNumber[B]$	$DecimalEquivalent$
100	1 0 1 1 0 0 0 0 1	353
101	1 0 1 1 0 0 0 1 0	354
102	1 0 1 1 0 0 1 0 0	356
103	1 0 1 1 0 1 0 0 0	360
104	1 0 1 1 1 0 0 0 0	368
105	1 1 0 0 0 0 0 1 1	387
106	1 1 0 0 0 0 1 0 1	389
107	1 1 0 0 0 0 1 1 0	390
108	1 1 0 0 0 1 0 0 1	393
109	1 1 0 0 0 1 0 1 0	394
110	1 1 0 0 0 1 1 0 0	396
111	1 1 0 0 1 0 0 0 1	401
112	1 1 0 0 1 0 0 1 0	402
113	1 1 0 0 1 0 1 0 0	404
114	1 1 0 0 1 1 0 0 0	408
115	1 1 0 1 0 0 0 0 1	417
116	1 1 0 1 0 0 0 1 0	418
117	1 1 0 1 0 0 1 0 0	420
118	1 1 0 1 0 1 0 0 0	424
119	1 1 0 1 1 0 0 0 0	432
120	1 1 1 0 0 0 0 0 1	449
121	1 1 1 0 0 0 0 1 0	450
122	1 1 1 0 0 0 1 0 0	452
123	1 1 1 0 0 1 0 0 0	456
124	1 1 1 0 1 0 0 0 0	464
125	1 1 1 1 0 0 0 0 0	480

6.2.3 Chinese Remainder Theorem

Some basic facts and conclusions of the CRT are summarized in this section. This mathematical background knowledge is of elementary importance for the efficient realization of proposed secret sharing scheme.

Theorem 1:Chinese Remainder Theorem

Let the numbers n_1, n_2 n_k be positive integers which are relatively prime in pair, i.e. $gcd(n_i, n_j) = 1$ when $i \neq j$.

Furthermore, let

$n = n_1 \times n_2 \times \ldots n_k$

a_1,a_2 $\ldots$ a_k be integers.

Then the system of congruences

$x \equiv a_1 \pmod{n_1}$

$x \equiv a_2 \pmod{n_2}$

$\ldots$

$x \equiv a_n \pmod{n_n}$

has a simultaneous solution x to all of the congruences, and any two solutions are congruent to one another *modulo* n. Furthermore there exists exactly one solution x, between 0 and $n - 1$.

The unique solution x of the simultaneous congruences satisfying $0 \leq x \leq n$ can be calculated as

$$\mathrm{x} = \Big(\sum_{i=0}^{i=k} x_i \; r_i \; s_i \Big) \pmod{n}$$

$$= (a_1 \; r_1 \; s_1) + (a_2 \; r_2 \; s_2) + \ldots + (a_k \; r_k \; s_k) \pmod{n}$$

where $r_i = \frac{n}{n_i}$

and $s_i = r_i^{-1} \pmod{n_i}$; for $i = 1, 2, \ldots, k$

Solution for a system of linear congruence using CRT:

Use the Chinese Remainder Theorem to find solution for the following system of congruence;

$x \equiv 3 \pmod 4$

$x \equiv 2 \pmod 3$

$x \equiv 4 \pmod 5$

First of all, establish the basic notation.

In this problem we have

$k = 3$, $a_1 = 3$, $a_2 = 2$, $a_3 = 4$,

$n_1 = 4$, $n_2 = 3$, $n_3 = 5$,

Calculate $n = 4 \times 3 \times 5 = 60$.

$r_1 = \frac{n}{n_1} = \frac{(60)}{(4)} = 15$

$r_2 = \frac{n}{n_2} = \frac{(60)}{(3)} = 20$

$r_3 = \frac{n}{n_3} = \frac{(60)}{(5)} = 12$

Calculate

$S_i = r_i^{-1} \pmod{n_i}$

For that compute; $s_i \; r_i \equiv 1 \pmod{n_i}$

In this problem we need to solve:

$15s_1 \equiv 1 \pmod 4$

$20s_2 \equiv 1 \pmod 3$

$12s_3 \equiv 1 \pmod 5$

and from the calculation, we find that $s_1 = 3$,$s_2 = 2$,and $s_3 = 3$.

x=$(a_1 \; r_1 \; s_1)$+$(a_2 \; r_2 \; s_2)$+$(a_3 \; r_3 \; s_3) \pmod{60}$

By substituting, we obtain:

x=$3 \times 3 \times 15 + 2 \times 2 \times 20 + 4 \times 3 \times 12$

$= 359$

which reduces to $x \equiv 59 \pmod{60}$.

6.3 Proposed Scheme: Visual Secret Sharing Scheme using POB Number System and CRT.

In this section, first we describe the algorithm for the 2 out of 2 Visual Secret Sharing scheme ($(VSS(2,2))$ using POB and CRT), then in next section the algorithm for an n-out of-n scheme ($(VSS(2,2))$, where n is greater than or equal to 3; ($n \geq 3$) is described. The proposed scheme is a block cipher; that is each byte is handles separately, for that the scheme assumes that the secret consists of a sequence of bytes. In the case of visual secrets(images), each pixel is handled separately while the construction of shares.

POB number system is used to enhance the security level in the proposed visual secret sharing scheme. And it is found very useful and more efficient than the conventional number systems. We have also used Chinese Remainder theorem in our newly introduced visual secret sharing scheme.

6.3.1 VSS(2,2) using POB and CRT scheme:

Here we are considering binary images. The share construction process is as follows;

Let IM be the secret image that is to be shared among two participant.

Let $M = m_7\ m_6\ m_5\ m_4\ \ldots\ m_0$ be the first 8 pixel in the IM.

We first find out the decimal equivalent of the secret information $m_7\ m_6\ m_5\ m_4\ \ldots\ m_0$, say it is D. Now we generate a POB $(11, 5)$ number corresponding to D, say it is B. That means M of having 8 bits are now converted into B of having 11 bits. Here B is the *POB number* of D.

Now we convert the POB number B, into decimal equivalent, say D'

Now randomly select two primary numbers less than 462 i.e., primary numbers less than $\binom{11}{5}$; say that is p and q respectively.

Then we find two quantities;

X=D' mod p

Y=D' mod q

Generate two POB $(11, 5)$ numbers corresponding to X and Y

Say S_1 = POB $(11, 5)$ numbers corresponding to X

S_2 = POB $(11, 5)$ numbers corresponding to Y.

Here, S_1 and S_2 represent the first 11 bits of two shares respectively. The complete secret sharing process is depicted in the Algorithm 6.1.

While reconstructing the secret back, the two shares are considered.Let S_1 and S_2 represent the first 11 bits of two shares respectively. Compute the POB Value of both the shares S_1 and S_2, say X and Y respectively.

Now consider the same prime numbers, which is used during the share construction process. Say that is p and q respectively. Using these quantities generate a system of two simultaneous congruence as below;

$Z \equiv X \pmod{p}$

$Z \equiv Y \pmod{q}$

Solve the above system of congruent equation using the theorem CRT. Let the solution be Z. Finally find the binary equivalent of Z, say it is M.

Where M=$m_7 m_6 \ldots m_0$, is the reconstructed secret.

Algorithm 6.2: Share Construction in ((VSS 2,2) using POB and CRT)

Input: The secret information $M = m_7\ m_6\ m_5\ \ldots\ m_0$ having 8 bits long

Output: Two blocks S_1 and S_2 of length 11 bits

1 Global values: Two relative prime numbers p and q
2 Let X, Y, S_1 and S_2 are 11 bit long integers. Set all the bits of X, Y, S_1 and S_2 to null.
3 The input string M is converted into decimal equivalent value, say D.
4 Let B be the POB (11, 5) number with value of D. [use Algorithm 6.1]. $B = b_{10}\ b_9\ b_8\ b_7\ \ldots b_0$
5 Convert the POB number B, into decimal equivalent, say D'
6 Let X= $D' \pmod p$; Y= $D' \pmod q$
7 Let S_1 is the POB $(11, 5)$ number with value of X and S_2 be the POB $(11, 5)$ number with value of Y[use Algorithm 6.1].

Algorithm 6.3: Secret Reconstruction in ((VSS 2,2) using POB and CRT)

Input: Two shares S_1 and S_2 of length 11 bits each.

Output: The secret information M = $m_7\ m_6\ m_5\ \ldots m_0$ having 8 bits long

1 Global values: Two relative prime numbers p and q
2 Let X and Y be the POB values corresponding to S_1 and S_2;
3 $X = V(S_1)$; $Y = V(S_2)$
4 Generate the system of simultaneous congruence like following;
5 $Z \equiv X \pmod p$; $Z \equiv Y \pmod q$
6 Solve the above system of simultaneous congruence using CRT method.And find the unique solution for the system, let it be Z
7 Find the binary equivalent of the Z, say M. where M is the secret information.M = $m_7\ m_6\ m_5\ \ldots m_0$

In the Algorithm 6.2 and 6.3, the algorithms involved in the VSS(2,2) secret sharing schemes using POB and CRT, the size of shares are 11 bits

and the size of the secret information is 8 bits. That means two 11 bits shares are generated from a single byte original secret information. There is pixel expansion during the sharing phase. But in the reconstruction time the secret image is 8 bit as such. So here, no loss of information and it is efficient also.

Example

Let us consider a secret, M=11001011

Secret sharing

Let p and q (The global values, prime numbers) are 37 and 89 respectively.

The decimal equivalent of M is 203, Say D=203

The POB (11, 5) number with the value 203 is 01100010110.

Say B = 01100010110

Decimal equivalent of B, say $D' = 790$.

Compute X and Y as follows;

$X = 790 \pmod{37}$

$Y = 790 \pmod{89}$

i.e. X=13 and Y=78

Let S_1 is the POB $(11, 5)$ number with value of 13 and S_2 be the POB $(11, 5)$ number with value of 78

The two shares S_1 and S_2 are as follows:

S_1= 00001101101

S_2= 00101011001

Recovery

Let p and q (The global values, prime numbers) are 37 and 89 respectively.

Find the POB values corresponding to S_1 and S_2.

$V(S_1)$ = V (00001101101) =13

$V(S_2)$ = V (00101011001) =78

The system of simultaneous congruence, generated from these quantities, is:

$Z \equiv 13 \pmod{37}$

$Z \equiv 78 \pmod{89}$

Apply CRT,and the unique solution obtained is 203. Say Z= 203.

The POB (11, 5) number with the value 203 is 01100010110.

Say B = 01100010110.

The binary equivalent of 203 is:

M $=(203)_d = (11001011)_2$

6.3.2 VSS(n,n) using POB and CRT scheme:

Algorithm 6.4: Share Construction in ((VSS n,n) using POB and CRT)

Input: The secret information M = $m_7\ m_6\ m_5 \ldots m_0$ having 8 bits long

Output: n blocks S_1,$S_2 \ldots S_n$ of length 11 bits

1 Global values: n relative prime numbers p_1, p_2, ..., p_n

2 Let X_1, X_2, ..., X_n and S_1,S_2, ..., S_n are 11 bit long integers. Set all the bits to null.

3 The input string M is converted into decimal equivalent value, say D.

4 Let B be the POB (11, 5) number with value of D. [use Algorithm 6.1]. B= $b_10\ b_9\ b_8\ b_7 \ldots b_0$

5 Convert the POB number B, into decimal equivalent, say D'

6 Let $X_1 = D' \pmod{p_1}$; $X_2 = D' \pmod{p_2}$, ..., $X_n = D' \pmod{p_n}$

7 Let S_1 is the POB $(11, 5)$ number with value of X_1 and S_2 be the POB $(11, 5)$ number with value of X_2 and likewise S_n be the $(11, 5)$ number with value of X_n [use Algorithm 6.1].

Algorithm 6.5: (Secret Reconstruction in (VSS n,n) using POB and CRT)

Input: n shares, S_1,S_2 ... S_n ,of length 11 bits
Output: The secret information M = m_7 m_6 m_5 ... m_0 having 8 bits long

1 Global values: n relative prime numbers p_1, p_2, ..., p_n
2 Let X_1, X_2, ..., X_n be the POB values corresponding to S_1, S_2, ..., S_n X_1 =V (S_1); X_2 =V (S_2); ... X_n = V(S_n)
3 Generate the system of simultaneous congruence like following; $Z \equiv X_1 \pmod{p_1}$; $Z \equiv X_2 \pmod{p_2}$; ... $Z \equiv X_n \pmod{p_n}$
4 Solve the above system of simultaneous congruence using CRT method.And find the unique solution for the system, let it be Z
5 Find the Binary equivalent of the Z, say M. where M is the secret information.M = m_7 m_6 m_5 ... m_0

Example 2

Let us consider 3 out of 3 schemes

Let us consider a secret, M=11001011

Secret sharing

Let p_1, p_2, and p_3 (The global values, prime numbers) are 37, 89 and 113 respectively.

The decimal equivalent of M is 203.

Say D=203

The POB $(11, 5)$ number with the value 203 is 01100010110.

Say B = 01100010110

Decimal equivalent of B, say $D' = 790$.

Compute X_1, X_2 and X_3 as follows:

$X_1 = 790 \pmod{37}$

$X_2 = 790 \pmod{89}$

$X_3 = 790 \pmod{113}$

i.e. X_1 =13 , X_2=78 and X_3 =112

Let S_1 is the POB (11, 5) number with value of 13, S_2 be the POB (11, 5) number with value of 78 and S_3 be the POB (11, 5) number with value of 112.

The two shares S_1 and S_2 are as follows:

S_1= 00001101101

S_2= 00101011001

S_3= 00111000101

Recovery

Let p_1, p_2, and p_3 (The global values, prime numbers) are 37, 89 and 113 respectively.

Find the POB values corresponding to S_1 , S_2 and S_3

$V(S_1)$ = V (00001101101) =13

$V(S_2)$ = V (00101011001) =78

$V(S_3)$ = V (00111000101) =112

The system of simultaneous congruence, generated using these quantities, is:

$Z \equiv 13 \pmod{37}$

$Z \equiv 78 \pmod{89}$

$Z \equiv 112 \pmod{113}$

Apply CRT, and the unique solution obtained is 203.

Say Z=203.

The POB (11, 5) number with the value 203 is 01100010110

Say B = 01100010110

The binary equivalent of 203 is:

M= $(203)_d = (11001011)_2$

6.4 Performance and security Analysis:

The proposed scheme has pixel expansion in the share construction phase. From the original secret of size 8 bits, we are constructing 11 bits shares.

So the space required to store the shares will be large .But in secret reconstruction phase, the size of the reconstructed secret is same as the original secret. From 11 bits shares, 8 bits original secret is reconstructed. So we can say that the proposed system is a loss less scheme.

The proposed scheme is under POB (11, 5) number system. That means there is a total of 462 shares corresponding to one byte of secret. The probability of a correct guess of a share will be $\frac{1}{462}$ per byte of secret. If there are m bytes in the secret, then this would mean that the probability of correct guess of a share will be as low as $\left(\frac{1}{462}\right)$.

6.5 Concluding Remarks

In the proposed scheme we have used both POB number system and CRT method. And both have great potential in secret sharing. So the scheme provides an effective way to improve the security and it is a loss less scheme as well.

Chapter 7

Introduction to Broadcast Encryption Schemes

7.1 Introduction

In general, "to broadcast "(verb) is to cast or throw forth something in all directions at the same time. It is something like as shown in Figure 7.1.

Broadcast Encryption(BE) is a type of encryption scheme first proposed by Amos Fiat and Moni Naor in 1993. Their original goal was to prove that two devices, previously unknown to each other, can agree on a common master key for secure communications over a one-way communication path. Broadcast Encryption allows for devices that may not have even existed when a group of devices was first grouped together to join into this group and communicate securely. This chapter describes BE in general, a brief survey on the same, how a few different broadcast encryption schemes work and their merits and demerits.

Traditionally, secure transmission of information has been achieved through the use of public-key cryptography. For this system to work, communicating devices must know about each other and agree on

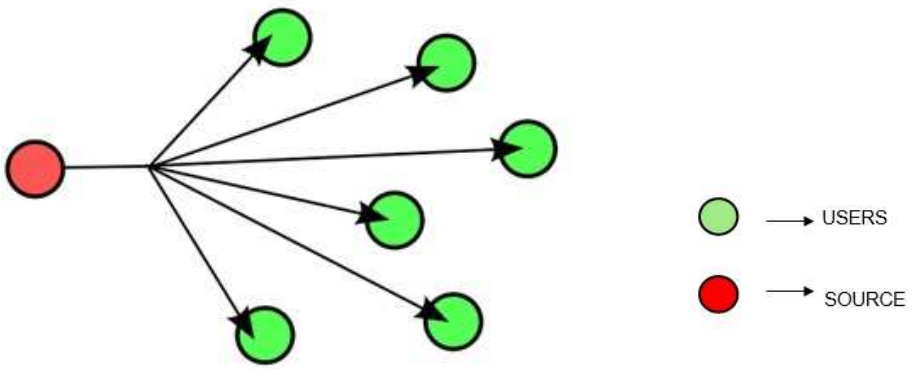


Figure 7.1: Broadcast Encryption

encryption keys (K) before transmission. Broadcast encryption seeks to solve the problem of two devices, previously unknown to each other, agreeing upon a common key. This can allow for new devices, even if they did not exist when the encrypted data was made, to be added to a group of acceptable devices. Since the same data is being sent to all devices, instead of a separately encrypted message for each, broadcast encryption must also ensure that only those devices in the privileged group will be able to decode the message. A. Fiat and M. Naor [FN94] first proposed the concept of broadcast encryption in 1993. In this scheme, sender allows to send a cipher text to some designated groups whose members of the group can decrypt it with his or her private key. However, nobody outside the group can decrypt the message. They considered a scenario where there is a center and a set of users. The center provides the users with prearranged keys when they join the system. At some point the center wishes to broadcast a message (e.g. a key to decipher a video clip) to a dynamically changing privileged subset of the users in such a way that non-members of the privileged class cannot learn the message. There are two solutions for this. One is give every user its own key and transmit an

individually encrypted message to every member of the privileged class. This requires a very long transmission (the number of members in the class times the length of the message). Another simple solution is to provide every possible subset of users with a key, i.e. give every user the keys corresponding to the subsets it belongs to. This requires every user to store a huge number of keys. A. Fiat and M. Naor provided solutions which are efficient in both measures, i.e. transmission length and storage at the users end.

Broadcast encryption is the cryptographic problem of delivering encrypted content (e.g. TV programs or data on DVDs) over a broadcast channel in such a way that only qualified users (e.g. subscribers who have paid their fees or DVD players conforming to a specification) can decrypt the content [LS98][CC89][FN94]. Several papers considered the problem of a center who wants to broadcast to a group [LLH89][SHD05][KSW03]. However, all these schemes are "one-time ", and the keys must be updated after every use. The main challenge arises from the requirement that the set of qualified users can change in each broadcast emission, and therefore revocation of individual users or user groups should be possible using broadcast transmissions, only, and without affecting any remaining users. So as efficient revocation is the primary objective of broadcast encryption, solutions are also referred to as revocation schemes[KSW03].

Broadcast encryption is widely used in the present day in many aspects, such as Voice over Internet Protocol(VoIP), TV subscription services over the Internet, communication among group members or from someone outside the group to the group members. This type of scheme also can be extended in networks like mobile multi shop networks, which each node in these networks has limitation in computing and storage resources.

In practice most BE systems are smart card-based. It has been well documented that pirate smart cards (also called pirate decoders) are

commonly built to allow non-paying customers to recover the content. Broadcast encryption schemes can be coupled with traceability schemes to offer some protection against piracy. If a scheme has x-traceability, then it is possible to identify at least one of the smart cards used to construct a given pirate card provided at most x cards are used in total. When a pirate card is discovered, the keys it contains are necessarily compromised and this must be taken into account when encrypting content.Earlier work in traceability does not deal with this; instead, the analysis stops with the tracing of smart cards (or, traitor users).

7.2 Related works

A. Fiat and M. Naor [FN94] first proposed the concept of broadcast encryption in 1993. Broadcast encryption is a cryptographic method for a center to efficiently broadcast digital contents to a large set of users so that only non revoked users can decrypt the contents. In broadcast encryption the center distributes to each user, u, the set, K_u, of keys called the user key set of u in the setup stage. Here the Authors assumed that the user keys are not updated afterwards, that is user keys are stateless. A session is a time interval during which only one encrypted message is broadcasted. The session key, SK, is the key used to decrypt the contents of the session. In order to broadcast a message, M, the center encrypts M using the session key SK and broadcasts the encrypted message together with a header which contains encryption of SK and the information for non revoked users to recover SK. In other words the center broadcasts:

$header$; E_{SK} (M)

Where E_{SK} is a symmetric encryption of M by SK. Then every non revoked user u computes $F(K_u, header) = SK$ and decrypts $E_{SK}(M)$ with SK where F is a predefined algorithm. But for any revoked user

u,$F(K_u, header)$ should not render SK. The length of the header, the computing time of F and the size of a user key are called the transmission overhead The main issue of broadcast encryption is to minimize the transmission overhead with practical computation cost and storage size.

J.A. Garay, J. Staddon and A. Wool [GSW00] proposed the notion of long-lived broadcast encryption schemes, here the purpose is to adapt to the presence of compromised keys and continue to broadcast securely to privileged sets of users. The basic approach is as follows. Initially, every user has a smart card with several decryption keys on it, and keys are shared by users according to a predefined scheme. When a pirate decoder is discovered, it is analyzed and the keys it contains are identified. Such keys are called "compromised ", and are not used henceforth. Similarly, when a user's contract runs out and he is to be excluded, the keys on his smart card are considered compromised. Over time, we may arrive at a state in which the number of compromised keys on some legitimate users smart card rises above the threshold at which secure communication is possible using the broadcast encryption scheme. In order to restore the ability to securely broadcast to such a user, the service provider replaces the user's old smart card with a new one containing a fresh set of keys.

As mentioned before, although it is not likely because of the large space of device keys, it is possible for all the keys to be compromised and for the encryption scheme to break. Garay, Staddon, and Wool proposed a way to extend the lifetime of a broadcast encryption scheme. They describe a system in which keys for devices are stored on smart cards. When a pirate decoder has been found, the keys associated with its smart card will be revoked. A user's keys can also be revoked if his subscription expires. When all the keys in an innocent device have been revoked, its smart card will have to be replaced with a new set of keys. Keys also need to be replaced if the contract for a given device has expired. Garay, Staddon, and Wool seek to minimize the number of smart cards that will need to

be replaced in a given period of time they define as an epoch. At the end of an epoch, the service provider must compute which users need to have smart cards replaced to continue secure communications. Thus, the cost of such a scheme becomes directly related to the cost of periodically replacing a number of smart cards in each epoch. For situations in which pirate decoders provide themselves and other unprivileged users access to content, traitor tracing schemes can be employed. Traitor-tracing schemes aim to make the construction of pirate decoders risky because once a compromised key is found, the smart card it came from can be revoked.

Halevy, Dani and Adi Shamir [HDS02]proposed the Layered Subset Difference (LSD) scheme. As we mentioned Broadcast Encryption schemes enable a center to broadcast encrypted programs so that only designated subsets of users can decrypt each program. The stateless variant of this problem provides each user with a fixed set of keys which is never updated. The best scheme published for this problem is the Subset Difference (SD) technique of Naor and Lotspiech [NNL01], in which each one of the n users is initially given $O(log^2(n))$ symmetric encryption keys. This allows the broadcaster to define at a later stage any subset of up to r users as "revoked ", and to make the program accessible only to their complement by sending $O(r)$ short messages before the encrypted program, and asking each user to perform an $O(log(n))$ computation. In "Layered Subset Difference"(LSD) technique, which achieves the same goal with $O(log^{1+\in}(n))$ keys, $O(r)$ messages, and $O(log(n))$ computation. This reduces the number of keys given to each user by almost a square root factor without affecting the other parameters. In addition, the author show how to use the same LSD keys in order to address any subset defined by a nested combination of inclusion and exclusion conditions with a number of messages which is proportional to the complexity of the description rather than to the size of the subset. The LSD scheme is truly practical, and makes it possible to broadcast an unlimited number of

programs to 256 million possible users by giving each new customer a smart card with one kilobyte of tamper-resistant memory. It is then possible to address any subset defined by t nested inclusion and exclusion conditions by sending less than $4t$ short messages, and the scheme remains secure even if all the other users form an adversarial coalition. This nesting inclusion and exclusion of subsets allows the following scenario.

Consider a football game being broadcast on a national level to a cable television company's subscribers. The television company allows all subscribers access to the broadcast, except for the local network where a blackout is in place. However, sports bars in the local viewing area with a special subscription are allowed to receive the broadcast, while any sports bar without the special subscription is still excluded. If the subscribers are grouped in a tree structure based on geography and subscription type this operation could easily be performed using the LSD method. If the leaf nodes are not grouped in a logical way, essentially the message will have to be encrypted using mostly leaf node keys and the number of messages broadcast will be on the order of the number of devices. This would be extremely impractical, so the grouping becomes very important.

Yevgeniy Dodis, Nelly Fazio [DF02] proposed Public Key Broadcast Encryption for Stateless Receivers. A broadcast encryption scheme allows the sender to securely distribute data to a dynamically changing set of users over an insecure channel. One of the most challenging settings for this problem is that of stateless receivers, where each user is given a fixed set of keys which cannot be updated through the lifetime of the system. Both of the above methods were originally designed to work in the centralized symmetric key setting, where only the trusted designer of the system can encrypt messages to users. On the other hand, in many applications it is desirable not to store the secret keys on-line, or to allow untrusted users to broadcast information. This leads to the question of building a public key broadcast encryption scheme for stateless receivers; in particular, of

extending the elegant SD/LSD methods to the public key setting. Naor et al. [NNL01] notice that the natural technique for doing so will result in an enormous public key and very large storage for every user. In fact, [NNL01] pose this question of reducing the public key size and users storage as the first open problem of their paper. They resolve this question in the affirmative, by demonstrating that an $O(1)$ size public key can be achieved for both of SD/LSD methods, in addition to the same (small) users storage and cipher text size as in the symmetric key setting.

Nam-Su Jho, Hwang [JHC+05] proposed one way chain based broadcast encryption schemes. A new broadcast encryption scheme based on the idea of "one key per each punctured interval ". It has been a general belief that at least one key per each revoked user(r) should be included in the overhead and hence r seems to be the lower bound of the transmission overhead in any broadcast encryption scheme with reasonable computation cost and storage size. In our scheme with punctured c intervals, however the transmission overhead is about:

$\frac{r}{p+1} + \frac{N-r}{c}$

which breaks the barrier of r. This scheme is very flexible with two parameters p and c. If a user device allows a large key storage like set-top boxes and mobile devices then we may take p as large possible to reduce the transmission overhead which is more expensive. If a user device has limited storage and computing power like smart cards and sensors, then we may set c as small as possible. Another remarkable feature of this scheme is that it does not have to preset the total number of users, any number of additional users can join at any time, which is not possible in tree based schemes.

The asymmetric group key agreement (ASGKA) which was introduced by Wu et al.[MWL06], the dynamic asymmetric group key agreement (DASGKA) which was introduced by Zhao et al[ZZT11]. In Wu et al. scheme, they propose an asymmetric group key agreement protocol based

on Aggregately Signature Based Broadcast (ASBB). An ASGKA protocol has the advantage over a symmetric group key agreement (GKA) protocol in that the ASGKA protocol can verify the sender of a message. Typically in an ASGKA protocol, it has two keys; one is a public group key, which is used as an encryption key for a message to a group and another is a private key, which a group member can use it individually as a decryption key, but in Wu et al.[MWL06]scheme which is based on ASBB, the encryption process is done by using a public group key and the decryption process is done by using a signature of a sender. This signature can be verified by using the public key of that sender. Their scheme does not require any controllers. As mentioned in Wu et al., their scheme does not improve in communication overhead for one-time group applications in which the members of the group are about fully dynamic as in ad hoc networks, because their scheme has heavy communication overhead in key establishment.

Norranut, Pipat[SH12] proposed Broadcast Encryption Based on Braid Groups Cryptography which is an alternative method in the public key cryptography and can reduce the computational cost. The concept of braid groups assists to avoid modular exponential operation in computation cost and the key tree helps in reducing the communication cost to constant round, so the computation cost and the communication cost can be minimized.

The Zhao et al.[ZZT11] scheme is constructed to fulfill the former scheme by introducing a dynamic asymmetric group key agreement. This scheme supports the environment in which users can join or leave the group efficiently without triggering a new key agreement protocol. There are two significant differences between the scheme. The first is that they obtain different decryption key. The decryption key for each member in the former scheme is different but in the later scheme is the same. The second is that the former scheme does not achieve dynamic joining and

leaving while the later does. The scheme is also an ASGKA protocol based on the braid groups based cryptography. The authors designed some protocols which support for the dynamic group broadcast such as join and leave protocols and get better efficiency. The scheme is made up of three algorithms; setup, encryption, and decryption. In the setup phase, when any user needs to join a group, he sends a join request message to a director. The director is one of the group members and everyone knows a public braid denoted as g. Each user can compute their own public keys p_{ki} from their private key k_i and the public braid g. The key tree is used to construct a public group key. The public group key p_k group can be computed individually from a user private key k_i and other public key according to a position of node in the tree. The concept of braid groups assists to avoid modular exponential operation in computation cost and the key tree helps in reducing the communication cost to constant round, so the computation cost and the communication cost can be minimized.

Broadcast encryption is an information fusion technique constructing an encrypted broadcast message by exploiting unique information of the users belonging to the receiver set. However, key management becomes an issue when new users join or existing users quit. The concept of identity-based cryptography introduced by Shamir [Sha84] overcomes the above mentioned issues. A.Muthulakshmi, R. Anitha[MA14] proposed Identity based broadcast encryption for multi-privileged groups using Chinese Reminder Theorem (CRT). Most group oriented applications require strict access control mechanisms to prevent unauthorized access to the group communication and hence protect the data. Access control is normally achieved by encrypting the group communication using a secret key shared by the privileged users of the group. This scheme is constructed using Chinese remainder theorem (CRT) and it achieves constant size cipher text when a message is broadcast to different users in a multi-privileged group. Identity-based broadcast encryption is tool for

communicating multiple copies of a single message to a selective group of users, identified by their identities in such a manner that others are unable to access the content. A multi-privileged group is a group of users where the users have different access privileges. This proposes an identity-based broadcast encryption scheme for multi-privileged groups that preserves the identities of the users which is developed using Chinese remainder theorem and bilinear pairing. It also ensures forward and backward secrecy with reference to user join and leave. Security of the scheme is proven under random oracle model.

M. Ak, K. Kaya, K. Onarliolu and A.A. Seluk [MKOS10] proposed an Efficient Broadcast Encryption with User Profiles. The proposed scheme is effective and can provide significant reductions in the transmission complexity of a BE system. The gains obtained by the proposed scheme turn out to be even more significant when a limited number of free riders can be tolerated in the system.

7.3 Concluding Remarks

In this chapter we tried to brief the Broadcast Encryption. We have done a survey on various schemes in Broadcast Encryption. Some of the schemes, in the area, are explored by considered the working, the merits and the demerits of the schemes.

Chapter 8

Key Distribution Scheme in Broadcast Encryption

8.1 Introduction

We have discussed about Broadcast Encryption(BE) in the Chapter 7 briefly. BE is a type of encryption scheme first proposed by Amos Fiat and Moni Naor in 1993. Their original goal was to prove that two devices, previously unknown to each other, can agree on a common key for secure communications over a one-way communication path. Broadcast encryption allows for devices that may not have even existed when a group of devices was first grouped together to join into this group and communicate securely. This chapter describes a new scheme for the key distribution/key management in broadcast encryption using polynomial Interpolation. Here the base is the secret sharing concept. Instead of distributing the key the shares of the key is distributed using secret sharing scheme. And the Lagrange Polynomial Interpolation is used to generate the shares of the key. A brief note on the Lagrange Interpolation

is mentioned in the Chapter 5. So before move on to the proposed method refer Chapter 5.

8.2 The proposed scheme:The key distribution in broadcast encryption using polynomial interpolation

Here we have considered a scenario where there is a Broadcast Center (BC) and a set of users. The BC provides the keys to the users when they join the system. Figure 8.1 shows the scenario, here U is the set of all users comes under the Broadcast Centre.

U=$\{$ u_1,u_2,u_3 , . . . ,u_n $\}$

Suppose we have some messages that are intended for a subgroup of users say T in U. And those messages should not be visible to other users in U. Figure 8.2 shows the present condition, here we want to securely send the message to the users in the group T. The possible solution is encrypting the messages with the key on which the BC and the users in T are agreed up on and broadcast the messages. In normal condition before the encryption the BC and the users of the group will agree up on a master key and BC use the master key for the further communication. If a new user is added into that group then the BC has to share the master key with the user too. It may require an additional communication between the BC and the users. And the worst condition is, when one of the user is removed from the group then the security will be compromised. Because the user is still have the master key, that is in use, with him. The solution is to change the master key. that means whenever a revocation occurs within a group some additional overhead will be there. Either change in the master key or additional message overhead will be there. The proposed method is useful for the key distribution in such scenario.

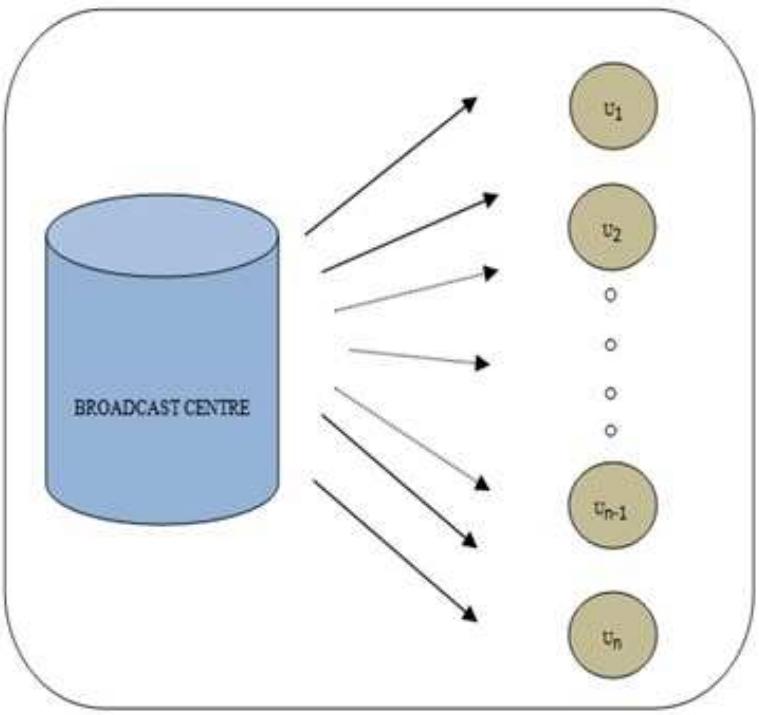


Figure 8.1: BC with Users

Suppose M is the message that is intended for the Group of users T. The Broadcast centre should encrypt the message using a key, say K and send to the users. We can represent the Scenario as follows:

C= E(M,K)

where M is the message.

K is the key.

E is the encryption scheme.

C is the encrypted message.

Now at the user's side the decryption should be possible. For decrypting the message, the key K should be available at the user side. In our proposed

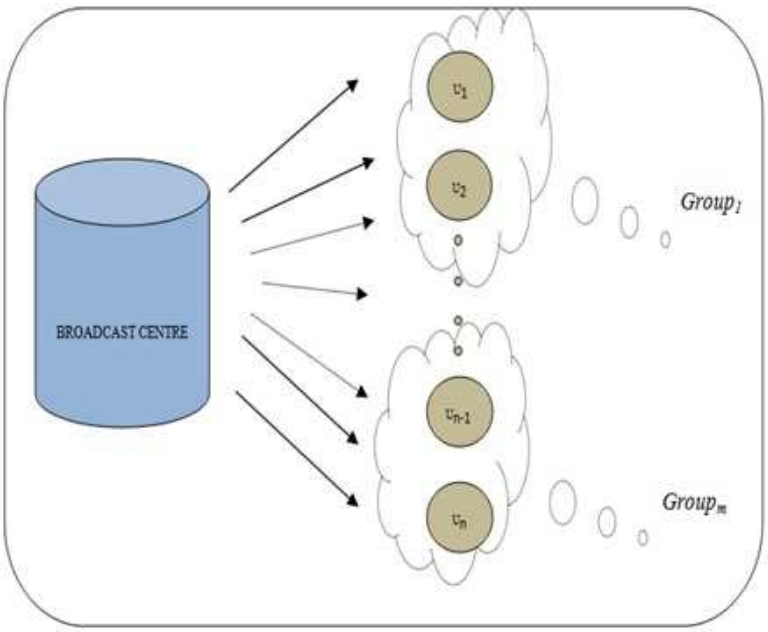


Figure 8.2: BC with Group of Users

method instead of sharing the original key, K, with the users of the group T, we are generation some share of the original key and distribute the shares to the users instead of the original key. And by using the share at the user side we can decrypt the original messages that are intended for the group T. Algorithm 8.1 shows the distribution of the shares of the key among the users in a particular group. Algorithm 8.2 describes the encryption process using the Key. Algorithm 8.3 describes the decryption of each user using the shares of the key.

Suppose M is the message that should be distributed securely among the users in the group T. The number of users in the group, T is n. Initially the BC has to select a key K and a constant z.

Algorithm 8.1: Distribution of the shares of the master key among the users

Input: K: the key

Output: k_1, k_2, ... k_n: the shares of key

1 Select z random numbers as the coefficients say $C_1, C_2 \ldots C_{z-1}, C_z$ of polynomial.
2 Construct polynomial of the form;
3 $f(x)$=$C_1x^z + C_2x^{z-1} + \cdots + C_{z-1}x^2 + C_zx^1 + M$
4 Select separate tokens for each users say, T_1, T_2, ..., T_n
5 Find $f(T_1), f(T_2), \ldots, f(T_n)$ values. Say $k_1, k_2, \ldots, k_n$
6 For each user select a random number say R_i
7 Each user find two more quantities
8 $k_{iR} = f(T_i + R_i)$, $k_{i2R} = f(T_i + 2R_i)$
9 Combine the key factors for each user i
10 $K_i = (k_i, k_{iR}, k_{i2R})$
11 For each user u_i distribute the K_i

Algorithm 8.2: Broadcast Encryption Process

Input: M : the message. K : the key

Output: C: the encrypted message for the users of group T

1 Encrypt the message M by using the key K
2 $C = E(M, K)$
3 Broadcast the encrypted message C to the group T.

Algorithm 8.3: Master Key Retrieval and Broadcast Decryption Process

Input: C: the encrypted message, K_i: the key share of user i
Output: M: the decrypted message

Assume that the user i from the Group T is decrypting the message using his own key share.
Accept the random number of the user i, say R_i
Accept the token of the user Ti
Represent the K_i as 3 pair values. Say $(x_1, k_1), (x_2, k_2), (x_3, k_3)$
we know $K_i = (k_i, k_{iR}, k_{i2R})$.
$x_1 = T_i$, $k_1 = k_i$.
$x_2 = T_i + 1 \times R_i$, $k_2 = k_{i(R)}$.
$x_3 = T_i + 2 \times R_i$,$k_3 = k_{i(2R)}$.
Compute the key K $= k_i l_i(x) + k_{(i(R))} l_{i(R)}(x) + k_{i(2R)} l_{i(2R)}(x)$, where;
$l_i(x) = \frac{x-k_2}{x_1-x_2} \times \frac{x-k_3}{x_1-x_3}$;
$l_{i(R)}(x) = \frac{x-k_1}{x_2-x_1} \times \frac{x-k_3}{x_2-x_3}$;
$l_{i(2R)})(x) = \frac{x-k_1}{x_3-x_1} \times \frac{x-k_2}{x_3-x_2}$;
Perform the decryption using the key K. M=(C,K)

In practical case, the main thing that we have to take care is the parameters with the BC and each user. It is shown in the Figure 8.3.

Consider an example:

Assumptions:

Suppose there are 8 users under a broadcast centre, BC and two groups under BC say U and V. Suppose the group U contain the 3 members and V contains 5 members. Let us consider the group U which is having 3 members.

Encryption phase:

Before broadcasting, the BC selects the secret key for the group U, say Ku .

Suppose the selected K_u is 255.

Broadcast Centre:	
M	Number of Groups under that broadcast centre.
User Information	The basic information related to the users.
Information of each group with the Broadcast Centre:	
N	Number of users in the group
$T_1\ T_2\ T_3 \ldots\ T_N$	The token number of each user in the group
K	The key used for the encryption
Information at each user:	
T_i	The token number of the user i
K_i	The key share of the user i $K_i = \{k_i, k_{iR}, k_{i2R}\}$ Where R is the random number selected by the user.

Figure 8.3: Information at Broadcast Centre

For constructing the polynomial, the degree of the polynomial (z) and a set of constants as coefficients $(C_1, C_2, \ldots, C_{z-1}, C_z)$ of the polynomial are to be selected.

Suppose

z = 3

C_1, C_2, C_3 = 19,13,83

So the polynomial is,

$f(x) = C_1x^3 + C_2x^2 + C_3x^1 + K$

$f(x) = 19x^3 + 13x^2 + 83x^1 + 255$

Now select tokens for the users, say T_1 T_2 and T_3

$T_1 = 15.$

$T_2 = 5.$

$T_3 = 3.$

Calculation of K_1:

Here the token $T_1 = 15$;

Then k_1 is calculated as follows;

$f(15) = 19 \times 15^3 + 13 \times 15^2 + 83 \times 15^1 + 255$

$k_1 = 68550$

Suppose the random number selected for the user is,

$R_1 = 2$

Find $f(15 + 2), f(15 + 2 \times 2)$

$f(17) = 19 \times 17^3 + 13 \times 17^2 + 83 \times 17^1 + 255$

= 104770

$f(19) = 19 \times 19^3 + 13 \times 19^2 + 83 \times 19^1 + 255$

= 136846

Then K_1 = (68550, 104770,136846)

Calculation of K_2:

Here the token $T_2 = 5$;

Then k_2 is calculated as follows;
$f(5) = 19 \times 5^3 + 13 \times 5^2 + 83 \times 5^1 + 255$
$k_2 = 3370$
Suppose the random number selected for the user is,
$R_2 = 1$
Find $f(5+1), f(5+2\times 1)$
$f(6) = 19 \times 6^3 + 13 \times 6^2 + 83 \times 6^1 + 255$
$= 5325$
$f(7) = 19 \times 7^3 + 13 \times 7^2 + 83 \times 7^1 + 255$
$= 7990$
Then $K_2 = (3370, 5325, 7990)$

Calculation of K_3:

Here token is $T_3 = 3$;
Then k_3 is calculated as follows;
$f(3) = 19 \times 3^3 + 13 \times 3^2 + 83 \times 3^1 + 255$
$k_3 = 1134$
Suppose the random number selected for the user is,
$R_3 = 3$
find $f(3+3), f(3+2\times 3)$
$f(6) = 19 \times 6^3 + 13 \times 6^2 + 83 \times 6^1 + 255$
$= 5325$
$f(9) = 19 \times 9^3 + 13 \times 9^2 + 83 \times 9^1 + 255$
$= 15906$
Then $K_3 = (1134,5325,15906)$.

Initially the BC selects the tokens for the user and generates the shares of the key for the users in the group. While encryption the message is encrypted by using the Key K. The full process is depicted in Figure 8. 4.

Decryption phase:

Consider the user 1, (u_1), who belongs to the group U.

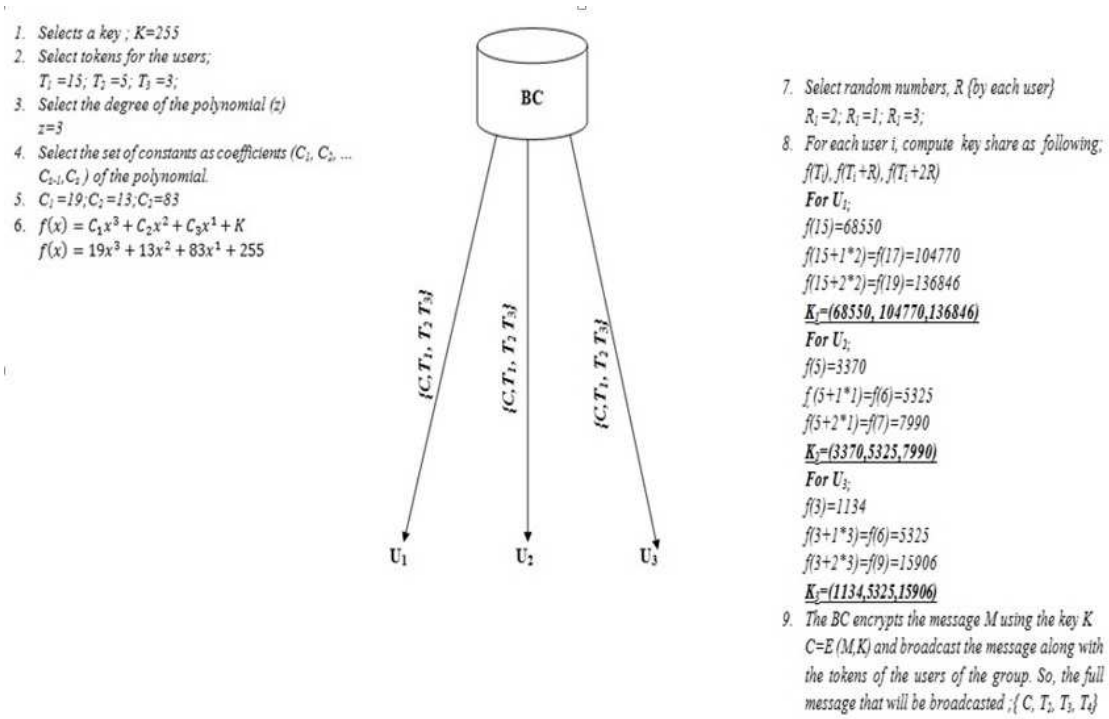


Figure 8.4:
Encryption
Phase

During the decryption phase each participant is verified by the corresponding token at the user interfaces itself, then the Key K is reconstructed by using the shares of key which is provided by the participant. As per the algorithm for key reconstruction calculation is as follows:

The user provides his Token (T_1 =15), random number (R_1=2), along with its key share, K_1 = (68550, 104770,136846).

The K_1 is represented as 3 pairs:

$(x_1, k_1), (x_2, k_2), (x_3, k_3)$

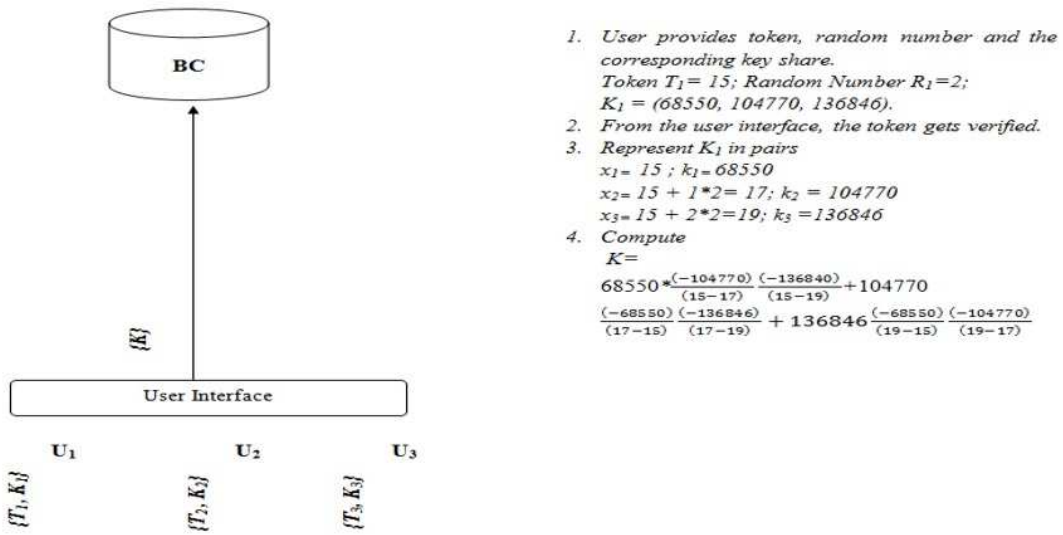


Figure 8.5:
Decryption
Phase

$x_1 = 15$; $k_1 = 68550$

$x_2 = 15 + 1 \times 2 = 17$; $k_2 = 104770$

$x_3 = 15 + 2 \times 2 = 19$; $k_3 = 136846$

Compute

$K = 68550 \times (\frac{(-104770)(-136840)}{(15-17)(15-19)}) + 104770 \times (\frac{(-68550)(-136846)}{(17-15)(17-19)}) + 136846 \times (\frac{(-68550)(-104770)}{(19-15)(19-17)})$

$K = 255$

Perform the decryption using the key K

M = (C, K)

The complete process is given in the Figure 8.5.

8.3 Concluding Remarks

The encryption keys play a vital role in almost all kind of encryption and decryption. Here we addressed a scheme in which the broadcast center does not share the key with any of the participating users/subscribers. The shares of the key are generated and distributed instead of the original key. The main advantage is that, the revocation of the user does not lead to the requirement of the new key for encrypting the secret and for broadcasting the same.

Chapter 9

Cheater Identification and Prevention in Visual Cryptography

9.1 Introduction

Protection of visual secret is the main concern in visual cryptography. The protection and genuineness of the participants, who shares the secret is also serious in this area. Since there is no restriction on the behaviour of the participants, any participant, called a cheater can reveal a forged share on purpose. Cheating identification and prevention in visual cryptography is a main thing to be considered in such situations. This chapter mainly focuses on the various cheater identification and prevention schemes that are proposed by various authors in the last decade.

9.2 Various Cheater Identification and Prevention Schemes

9.2.1 Horng et al.'s Cheating Activity and Prevention Scheme:

In 2006, Horng et al.[HCT06] proposed that cheating is possible in VC. In this scheme cheating is possible in (k, n) VC when k is smaller than n. The key point of cheating is how to predict and rearrange the positions of black and white subpixels in the victim's and cheater's share. The cheating activity of Horng et al.is that the $n-1$ cheaters collusively use their transparencies to know the secret and infer the victims transparencies Tv, thus they can generate a fake transparencies FTs to make the victim to accept the cheating image by stacking $FTs + Tv$. Here one of the most important question is ; **How Cheating works?**

Take $(2, 3)$ -VSS scheme as an example. A secret image is encoded into three distinct transparencies, denoted T_1, T_2 and T_3. Then, the three transparencies are respectively delivered to Alice, Bob, and Carol. Without lose of generality, Alice and Bob are assumed to be the collusive cheaters and Carol is the victim. In cheating, T_1 and T_2 to create forged transparency $T_2^{'}$ such that superimposing $T_2^{'}$ and T_3 will visually recover the cheating image. Precisely, by observing the following collections of 3×3 matrices which are used to generate transparencies, the cheaters can predict the actual structure of the victims transparency so as to create $T_2^{'}$

$$C^0 = \begin{matrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{matrix} \qquad C^1 = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

By observing the above matrices, two rows of above C^0 or C^1 matrix are

determined by the collusive cheaters. Therefore, the structure of each block in T_3 is exact the remaining row. For presenting a white pixel of cheating image, the block in T_2 is set to be the same structure of T_3. For presenting a black pixel of cheating image, the block in T_2 is set to be the different structure of T_3. Figure.9.2 shows the whole cheating process and Figure.9.1 shows the table, the cheaters create to change the decoded image. If the block in T_3 is [010], then T_2 is set to be [010] for a white pixel or it is set to be [001] for a black pixel. Formally, the cheaters can construct a Sub-Base Matrix (SBM) by T_1 and T_2, and then infer T_3.

	Pixel in Secret message	Block in Share S_A	Block in Share S_B	Block in Share S_C	Pixel in Cheating message	Block in Share S'_A	Block in Share S'_B
Case 1	white	[1 0 0]	[1 0 0]	[1 0 0]	white	[1 0 0]	[1 0 0]
Case 2	white	[1 0 0]	[1 0 0]	[1 0 0]	black	[0 1 0]	[0 0 1]
Case 3	black	[1 0 0]	[0 1 0]	[0 0 1]	white	[0 0 1]	[0 0 1]
Case 4	black	[1 0 0]	[0 1 0]	[0 0 1]	black	[1 0 0]	[0 1 0]

Figure 9.1: The concept of cheating in $(2, 3)$ scheme

In [HCT06]authors proposed two cheating prevention schemes:

1. **Authentication Based Cheating Prevention**
2. **2-out of (N+L) cheating prevention scheme**

The Authentication Based Cheating Prevention scheme solves the cheating problem by using verification shares to ensure from other participants are authentic and hence the recovered secrete image is authentic. However each participant is burden with a verification share.The scheme consists of shares S_i and verification shares V_i . Shares S_i are generated by any visual cryptographic scheme. Verification shares V_i, for i =1, 2, . . . ,n, generated by the verification shares generation process $f(.)$ and these shares are used to verify the correctness of the shares S_j, for j =1, 2, . . . ,n and $i \neq j$. Each participant P_i should

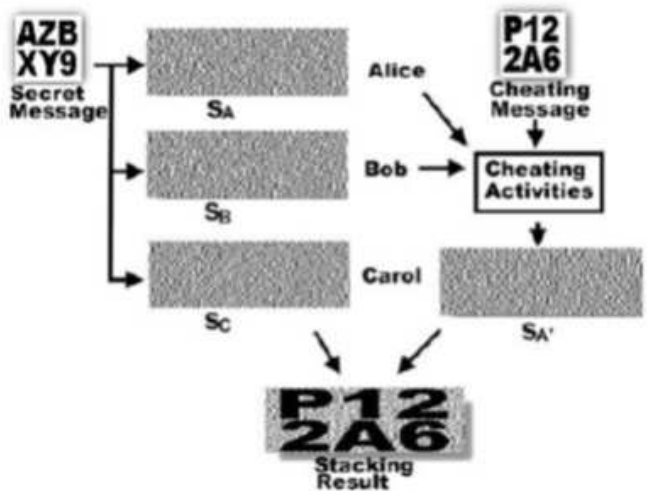


Figure 9.2: Cheating in visual cryptography

provide the dealer with a distinct verification logo L_i to be used for verifying the authenticity of other shares. All logos are confidential. The verification shares generation process is based on a 2-out-of-2 VC. Each verification share V_i is divided into $n-1$ regions, $R_{i,j}$ where $1 \leq j \leq n$, $j \neq i$ so that when stacking V_i and S_j the logo L_i appears in $R_{i,j}$.

Limitations:

- Each participant, however, was burdened with an extra verification share.
- VC requires total number of n^2 subpixels in all transparencies and this scheme requires total number of $2n^2$ subpixels.
- Possible to create a forged share without modifying any blocks within the victims region to pass the verification process when the number of n is becoming large.

The 2-out of (N+L) cheating prevention scheme is a simple prevention scheme, which is designed to make it harder for the cheaters to predict the structure of transparencies of the other participants. The method uses 2-out-of-(n+L) VC instead of 2-out-of-n, where $L \geq 1$. The dealer creates (n+L) shares but only delivers n shares to the n participants. The extra L shares are kept secret or destroyed by the dealer.

Let T be the transparency of a victim and let B be a block of T that corresponds to a black pixel of the secret image. Then the probability that cheaters can correctly guess the structure of B is $1/(1 + L)$.

Limitations:

- The extra L shares.
- Prevents black pixels of the secret image from cheating, but leave white pixels of it vulnerable.

9.2.2 Hu and Tzengs Cheating Activities:

In [HT07] the authors discussed about cheating problem in VC and extended VC. As we know in extended VCS the share are meaning full image.

There are two types of cheaters in the scenario, which is considered by the authors. One is a Malicious Participant (MP) who is also a legitimate participant, namely, $MP \in P$, and the other is a Malicious Outsider (MO) where $MO! \in P$. A cheating process against a VCS consists of the following two phases:

- **Fake share construction phase:** The cheater generates the fake shares.
- **Image reconstruction phase:** The fake image appears on the stacking of genuine shares and fake shares.

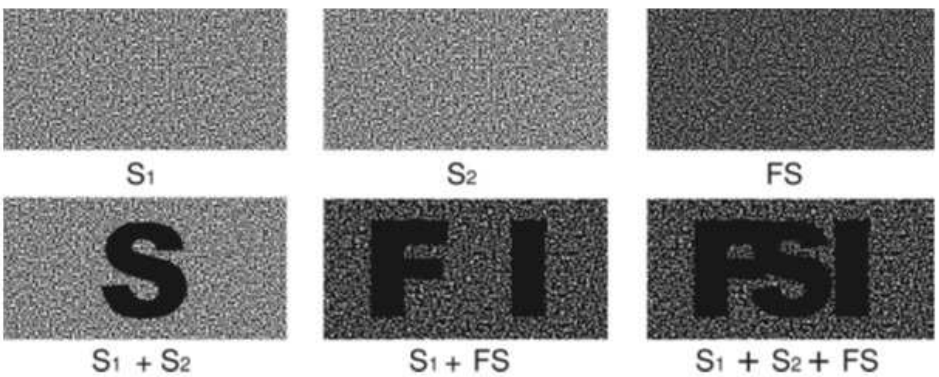


Figure 9.3: Example of cheating in a (2,2)-VCS

Figure 9.3, shows how to cheat participants in a (2,2)-VCS.

Hu and Tzeng [HT07] presented three kinds of cheating activities: CA-1, CA-2, and CA-3.

1. **Cheating method CA-1, initiated by an MP**
2. **Cheating method CA-2, initiated by an MO**
3. **Cheating method CA-3, against an EVCS**

Cheating Activities in Detail:

1. Algorithm for Cheating method CA-1, initiated by an MP:

Input: Share S_1. (We assume that the cheater is p_1)

Fake share construction phase:

Assume that each pixel of S_1 has x black and y white sub pixels. Then p_1 chooses a fake image and prepares r=m/$\lceil x \rceil$ -1, fake shares FS_1, FS_2,... FS_r as follows;

1) For each white pixel of the fake image, copy the corresponding sub pixels of the pixel in S_1 to each fake share.

2) For each black pixel of the fake image, randomly assign x black and white sub pixels to each fake share such that the pixel in the stacking of these shares and S_1 is perfect black.

Image reconstruction phase (the fake image):

Let Y = p_1,p_{i1}, p_{i2}, ... p_{iq} be a set of participants. If $Y! \in Q$, the stacking of genuine shares S_1,S_{i1},S_{i2} ... S_{iq} and fake shares FS_1,FS_2, ... FS_r shall reveal the fake shares.

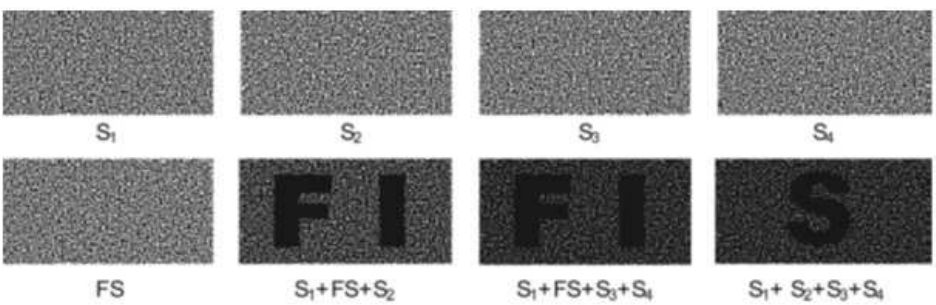


Figure 9.4: Example of cheating in a (4,4)-VCS by an MP

2. Algorithm for Cheating method CA-2, initiated by an MO:

Input: None

Fake share construction phase:

MO chooses a fake share and does the following:

1) Encode the fake image into two fake shares FS_1 and FS_2 with the optimal (2, 2) VCS.

2) Generate enough pairs of fake shares $FS_{1,i}$ and $FS_{2,i}$ with various sizes and sub pixel distributions, $1 \leq i \leq r$ for some r.

Image reconstruction phase (the fake image):

Let Y = p_1,p_{i1}, p_{i2}, ... p_{iq} be a set of participants and $Y! \in Q$. The stacking of S_1,S_{i1},S_{i2} ... S_{iq} and two fake shares $FS_{1,c}$,$FS_{2,c}$ shows the

fake image for some c, $1 \leq c \leq r$.

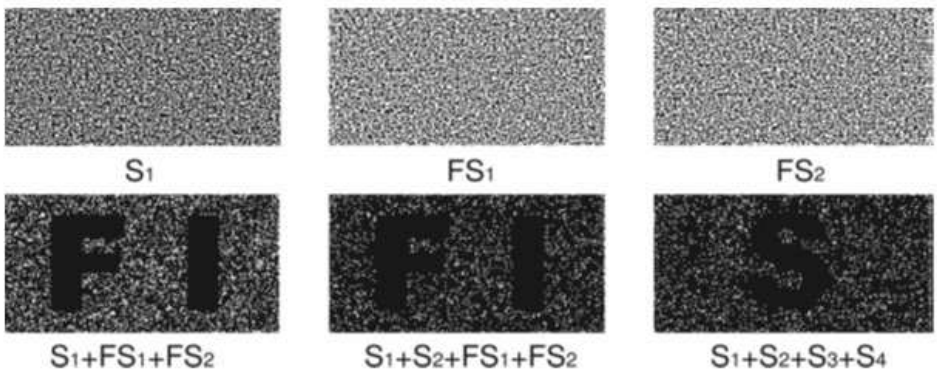


Figure 9.5: Example of cheating a (4,4)-VCS by an MO

3. Algorithm for Cheating method CA-3, against an EVCS:

Input: Share S_1. (We assume that the cheater is p_1)

Fake share construction phase:

p_1 chooses a fake image and does the following:

Create $S_1^{'}$, which is S_1 , but without the share image. The share image of S_1 is removed by changing d black sub pixels into white sub pixels in each black pixel, where d is the difference between the numbers of black sub pixels of a black and white pixel.

Create r=m/$\lceil x \rceil$ -1 temporary fake shares FS_1, $1 \leq i \leq r$, by using S_1 according to **CA-1.**

Randomly changed white sub pixels into black sub pixels of each pixel of the share image in $F_1^{'}$, $1 \leq i \leq r$.

Construct $F_1^{'}$ by randomly adding m black sub pixels (changing from white sub pixels) to each pixel in $F_1^{'}$, $1 \leq i \leq r$.

Image reconstruction phase (the fake image):

Same as in $CA - 1$

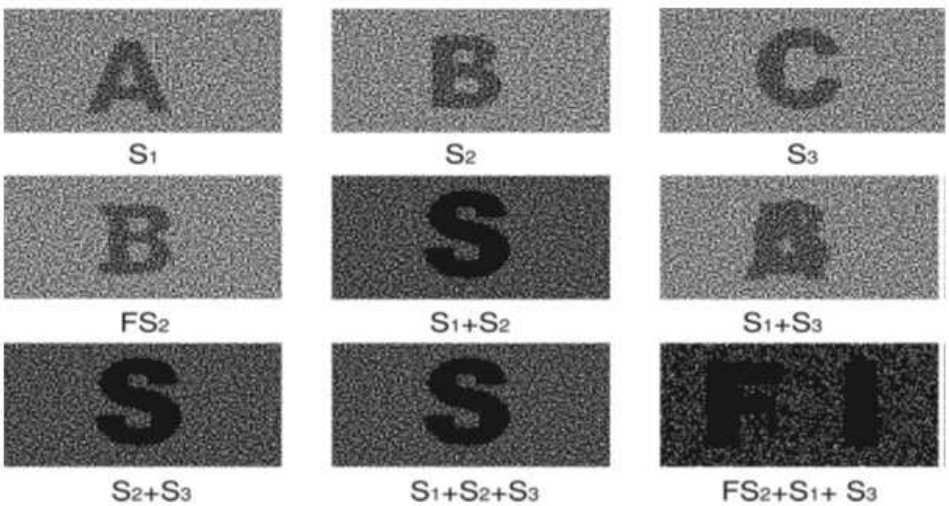


Figure 9.6: Example of cheating activity against EVCS

9.2.3 Du-Shiau Tsaia, Tzung-Her Chen, Gwoboa Horng (Homogenous)

In [TCH07], the authors proposed a new GA based Share Construction Method (GASCM). Figure 9.7 shows the ow chart for a typical binary GA.It starts with an initial set of random solutions, called "population". Each individual, named "chromosome", in the population is represented by a binary string. Through serial iterations, called "generations", the chromosomes evolve to minimize or maximize the fitness value by selection, crossover and mutation. After the algorithm converges, the best chromosome might be the optimum solution to the problem. Each stage of GA is further discussed as follows:

Chromosome creation: It encodes the solution to a chromosome and then initially generates random chromosomes to form the population.

Evaluation: Compute the fitness value of each chromosome. Before computing, the chromosome needs to be decoded to solution. By using the fitness value, the GA becomes a heuristic strategy that guides the search along with the best search directions.

Selection: During each iteration this process keeps the high fitness individuals, called *Gen_Good*, for mating and discards the other low fitness individuals, called *Gen_Bad*, for making room for offspring.

Crossover: A major genetic operator. It works on two chromosomes at a time and generates offspring by merging both chromosomes characteristics. Let variable *Pop_Size* denote the size of population. The crossover rate is defined to be

Crossover_Rate = Gen_Bad/Pop_Size

Mutation: Another genetic operator. It generates random changes in different chromosomes. A simple way is to change one or more genes. The important function of mutation is to avoid the case that the final solution is local optimal. The mutation rate is defined as the percentage of the total number of genes in the population.

Assumptions:

- **Assumption 1:** The appearance of each transparency is in a noise form and the brightness of each transparency is the same.
- **Assumption 2:** The secret image will be a password.

Definitions:

Definition 9.2.1. Two secret images are said to be homogeneous if they can be recognized as the same meaning.

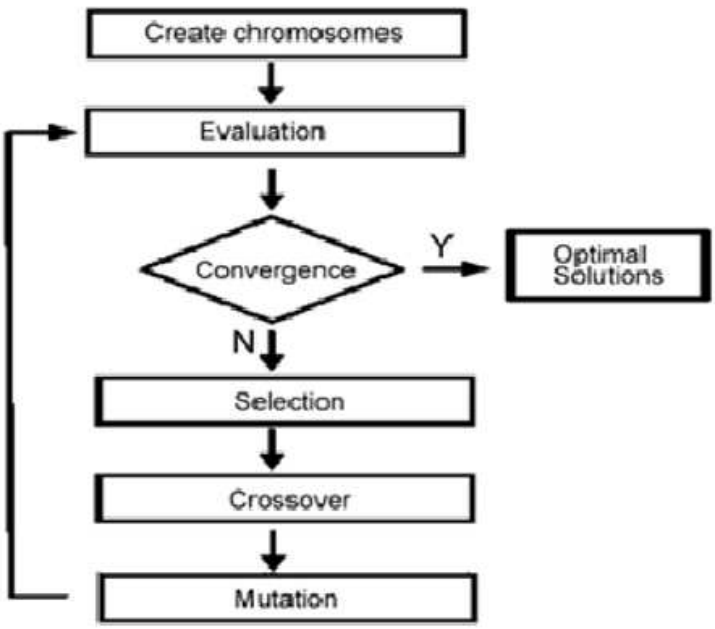


Figure 9.7: The flow chart of GA

Definition 9.2.2. A recovered secret image is acceptable if it is smooth in a sense that its boundary of black and white regions is clearly perceptible with > 0.1.

Definition 9.2.3. A recovered secret image is authentic if its visual meaning is acceptable and all corresponding secret images are homogeneous.

Focus on the cheating prevention scheme in 2-out-of-n VC. So far, we have tacitly assumed that the decoding of VC can be easily executed. This assumption is in order with respect to theoretical model. In general, however, It is well known that VC suffers from a graying effect and the

recovered image being much blurrier and darker than the original image. Furthermore, it is not easy to properly align two transparencies. Since the recovery of secret images depends on Human Visual System (HVS), the contrast of recovered secret images play an important role in guaranteeing that these images can be recognized as the actual secret image. Consequently, the contrast is defined as the $value > 0.1$.

Three phases:

Share construction, distribution and secret image recovery phase.

- **Share construction phase:**In the share construction phase, the dealer first generates six distinct homogeneous secret images. For simplicity, Figure 9.8 only shows three distinct secret images. Since all secret images can be recognized as the same meaning, three distinct secret images are homogeneous. Dealer generates four shares by means of the proposed GASCM.

  The algorithm for encoding is demonstrated as follows with parameters: a secret image X and $n = 4$.

  $CPSRS(X, n)$

  1. u $\leftarrow$ n!/2!(n 2)!
  2. use X to generate u distinct homogeneous secret images SI
  3. use $GASCM(SI, n)$ to generate n shares S
  4. returns S

- **Distribution phase :** In the distribution phase, the dealer individually distributes share s_1 for Alice, share s_2 for Bob, share s_3 for Carl and share s_4 for David, where $s_i \in S$.

- **Secret image recovery phase:** In the secret image recovery phase, when two transparencies are stacked together, the corresponding secret image can be visually recovered.

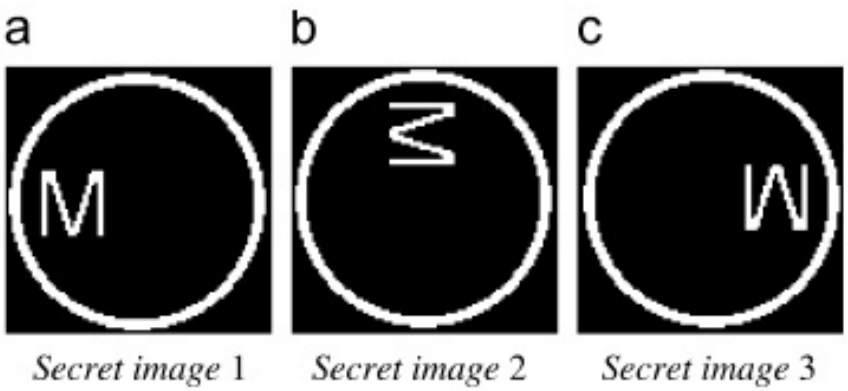


Figure 9.8: Three distinct secret images. (a) Secret image 1. (b) Secret image 2.(c) Secret image 3.

The GA used in the proposed scheme is a typical binary GA. For generating shares, each pixel in a secret image is encoded independently. Therefore, each pixel in the same spatial location of all distinct secret images is used to create a two-dimensional chromosome. The chromosome illustrated in Figure 9.9 is used to create initial population with random binary bits. The proposed scheme is more secure than 2-outof-(n + L) VC.

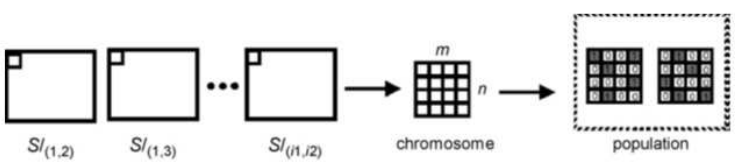


Figure 9.9: The chromosome

The main limitation that is addressed in this work is; Secret image consists of much more white pixels than black pixels, the proposed scheme

may not correctly reveal the secret. It only shows three distinct homogeneous secret images and corresponding decoded images. It is obvious that each one of three decoded images can be hardly recognized as the original secret image and low quality of decoded images.

9.2.4 De Prisco and De Santis's cheating activity:

In [PS10] De Prisco and De Santis proposed two Cheating Immune VSS schemes

1. Simple (2, n)-threshold and (n, n)-threshold schemes immune to deterministic cheating.

2. A better (2, n)-threshold scheme.

A (2, n)-threshold and a (n, n)- threshold scheme immune to deterministic cheating. The schemes are obtained by simply adding an extra column with all 0s to the base matrices of the schemes of Naor and Shamir. The scheme does not allow deterministic cheating. When the secret pixel is black, the cheaters cannot fool honest participants without some uncertainty

The (2, n)-threshold and a (n, n)- threshold schemes presented by authors are immune to deterministic cheating. However, they prevent deterministic cheating only when the secret pixel is black ($pr(1 \rightarrow 0) < 1$). When the secret pixel is white, the cheaters can deterministically fool a honest participant ($pr(0 \rightarrow 1) = 1$). This can be a problem if the secret image allows meaningful forging by only changing white pixels into black pixels.

The simple scheme has been showed some inherent weaknesses by the authors, as well as the white pixels are not protected without the complementary image. So they proposed a better scheme which is provably secure. They claimed this scheme for each black or white

block/pixel is cheating immune to deterministic cheating. In the better scheme, one pixel will be expanded to $2^n + n + 1$ sub pixels.

For example letter P could be changed into letter B and number 3 into number 8. So they provide a better (2, n)-threshold scheme immune to deterministic cheating. The scheme does not allow deterministic cheating for both black and white secret pixels (that is, $(pr(1 \rightarrow 0) < 1)$ and $pr(0 \rightarrow 1) < 1$). The scheme uses a bigger pixel expansion that makes it impossible for the cheaters to exactly figure out the share of the honest participant

9.2.5 Thasai ,Wang,Wu Scheme:

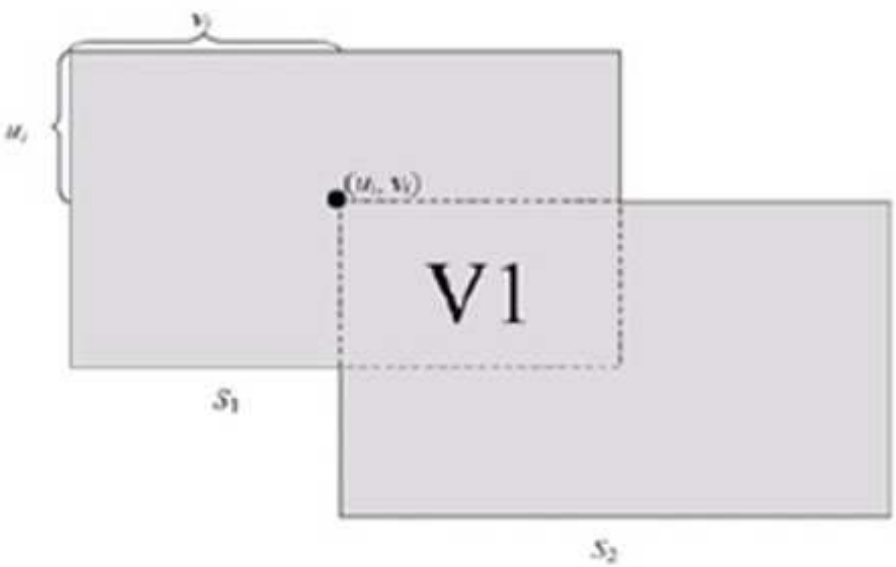


Figure 9.10: Conception of the scheme

In [TWWW11] the author discussed about a cheating prevention

scheme by referring the special position. Here the authors proposed a novel transformation for cheat prevention. The proposed scheme can transform the existing VC scheme into a cheating preventing VC scheme. Moreover they introduced the concept of a verification parameter to avoid the inconvenience of holding an extra share.

They first defined a verifiable parameter VP_i, which is composed of the coordinates of the width and height of the secret image. The verification parameter is only held by the participants. The participant cannot be promulgated by the participant until it is in the verification stage. In the proposed scheme they pursue two participants P_1 and P_2, who hold verification image VI_i, parameter VP_i and the secret share S_i. The content of the verification image of P_1 is VI_1 and the verification parameter of P_1 is $VP_1 = (u_1, v_1)$. During the verification stage, P_1 announces VP_1 to P_2. According to the verification parameter, VP_i, P_2 can stack S_2 on S_1 in a special position. If S_1 is a real share, the information VI_1 will appear on the stack result. If S_1 is a fake share, there will be no information on the stack result. The conception of this verification using a verification parameter is shown the Figure 9.10.

9.3 Concluding Remarks

In this Chapter we have considered Cheater Identification and Prevention Schemes , and explored its advantages and disadvantages. We have examined some schemes in cheating identification and prevention, and found that they are still improvable.

Chapter 10

Cheater Identification using SHA algorithm

10.1 Introduction

As we discussed in the previous chapter, the cheater identification is a vital section to be considered while secret sharing. Here in this chapter we discuss a novel method to identify cheater by adding some additional information with each share, called the authentication information, related to that particular share in alpha channel.The method can be used with any existing visual cryptographic scheme to identify the cheaters. We use Secure Hash Algorithm (SHA) for generating authentication information. There are a family of cryptographic hash functions and SHA is one of the popular hash functions. Hash functions appear in almost all the information security applications and are very useful also. It converts variable length numerical input value (the numerical equivalent of the actual secret information in most of the security applications) into a fixed length numerical value. The value returned as the result of hash function

is called *message digest* or *hash value* or *digital fingerprint* or *digest* or *checksum*. Figure 10.1 illustrate the hash function

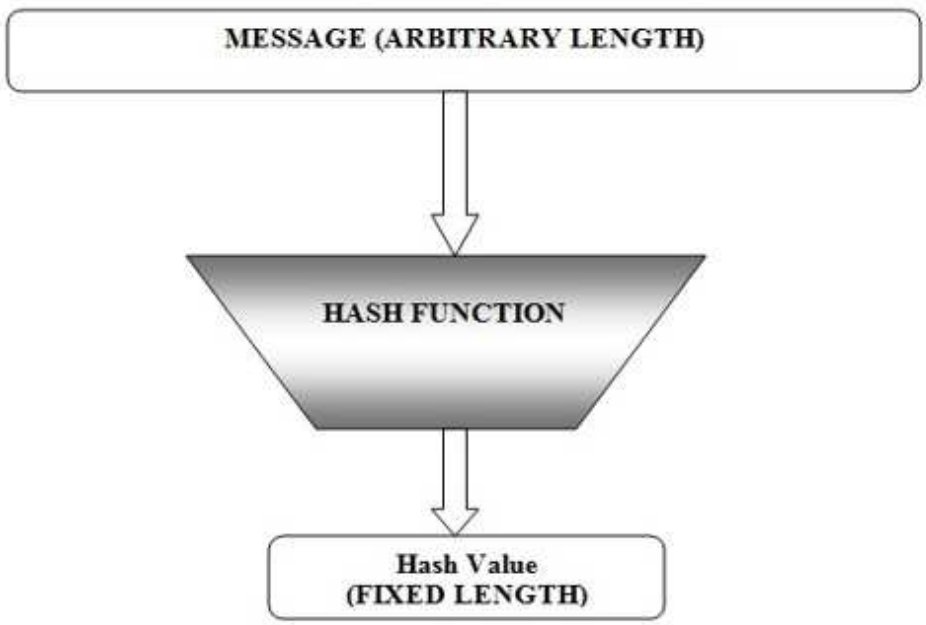


Figure 10.1: Hash Function

SHA is published by the National Institute of Standards and Technology(NIST). In this family of cryptographic hash function, the main standards are SHA-0, SHA-1, SHA-2 and SHA-3. The table in the Figure 10.2 gives the details of these standards.

There are two direct applications of hash function based on its cryptographic properties.

- Hash functions provide protection to password storage. Instead of storing password in clear, the hash values of the passwords are stored in the file.

YEAR	STANDARD	REMARKS
1993	SHA – 0	The original version is SHA-0.
		160-bit hash function.
		It had few weaknesses and did not become very popular.
1995	SHA - 1	It was designed to correct alleged weaknesses of SHA-0.
		It is the most widely used of the existing SHA hash functions.
		It is employed in several widely used applications and protocols including Secure Socket Layer (SSL) security.
		In 2005, a method was found for uncovering collisions for SHA-1 within practical time frame making long-term employability of SHA-1 doubtful.
2005	SHA-2	SHA-2 family has four further SHA variants, SHA-224, SHA-256, SHA-384, and SHA-512 depending up on number of bits in their hash value.
		No successful attacks have yet been reported on SHA-2 hash function.
		Its basic design follows design of SHA-1
2012	SHA-3	Efficient performance and good resistance for attacks.

Figure 10.2: Hash Standards

- Data integrity check is the most common application of the hash functions. This application provides assurance to the user about correctness of the data. The integrity check helps the user to detect any changes made to original file.

We are making use of this second application of hash function, which is data integrity check, to identify the cheater in visual cryptographic scheme.

10.2 Proposed Scheme: Cheater Identification using SHA

As we discussed, the proposed method can be used with any of the already existing visual cryptographic scheme. So after the phase of share

construction, an authentication signal is generated for each raw of the share using SHA-2 algorithm (Secure Hash Algorithm) and the generated signal is embedded in the alpha channel of the shares. In this particular method, SHA-512 is used from SHA-2 family. As we know, there are mainly two phases in the case of secret sharing methods;

(1) Secret sharing phase.

(2) Secret reconstruction phase.

We have designed two algorithms that describe the steps involved in these two phases respectively. Operation wise the main requirements in this case are;

- The image of the share should be Portable Network Graphics(PNG) format. Actually PNG images support alpha channel.
- The size of the secret image as well as shares should be $row \times 1024$. Where row represents the number of rows in the matrix form of the image. The number of columns in the matrix form of the image should be 1024 or its multiple.

In secret sharing phase mainly four sub phases are there:

- (a) Preprocessing of the secret image
- (b) Share construction
- (c) Authentication signal creation
- (d) Embedding the authentication signal in each share

In preprocessing phase, the secret image is resized to the prescribed size, $row \times 1024$ pixels, (where row represent the number of rows in the matrix form of the image. The number of columns in the matrix form of the image should be 1024 or it's multiple) because the application of SHA-512 algorithm will be simple, if the size of the secret image is $rows \times 1024$ pixels

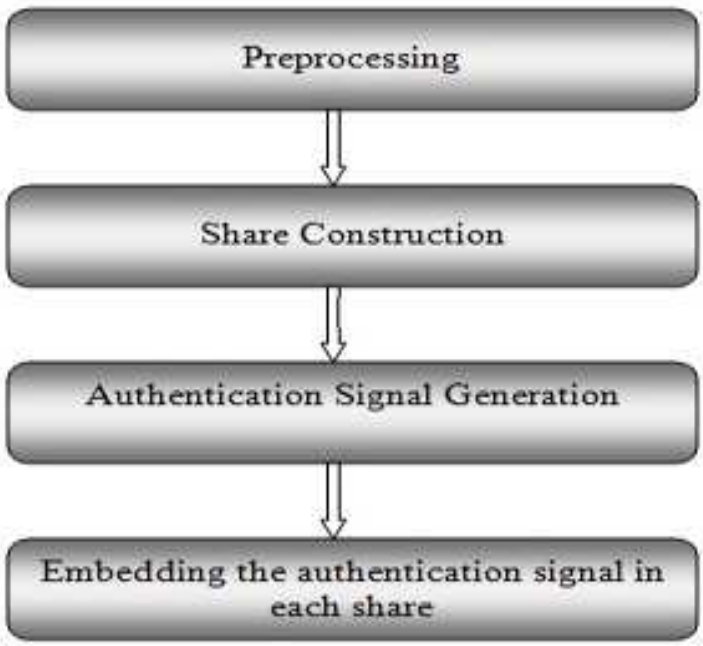


Figure 10.3: Phases in share construction

(In SHA-512, the size of the input value is 1024 bits). In the second phase, share construction, any size invariant secret sharing scheme can be used. One of the most important things that should be considered here, is the format of the shares. It should be in PNG format, because the PNG format provides the transparent channel called alpha channel with the image. In the next phase, by using SHA-512 algorithm generate the authentication signal as described in Algorithm 10.1. And finally embed the authentication signal in the alpha channel of each share and distribute the shares in to the participants who are involved in the communication.

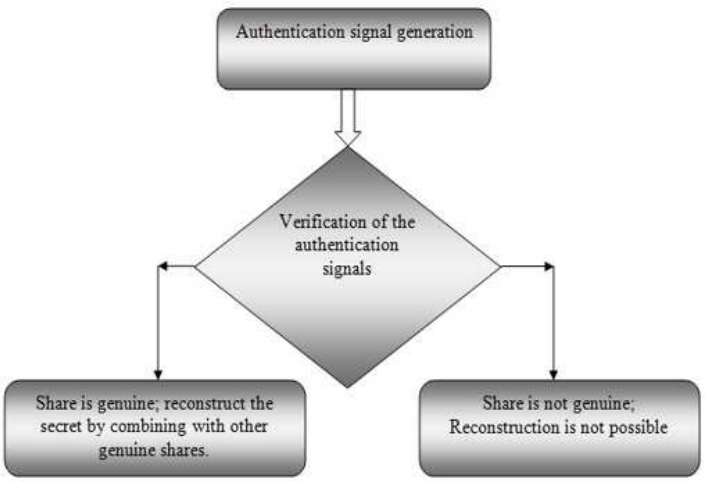


Figure 10.4: Checking the authenticity of share/participant in secret reconstruction phase

Algorithm 10.1: Construction of share with authentication signal

Input: The Share, S, having size $row \times 1024$
Output: Modified share,MS, with authentication signal embedded in the alpha channel having the size $row \times 1024$.

Take one row at a time from the input share, S.
$i = 0$
while $i \leq row$ **do**
    Apply SHA-512 on the bits(1024) of the i^{th} row of the share.
    Resulting 512 bits Hash Value is embedded as the i^th row of the alpha channel.
end
Return the modified share, MS.
Stop.

Algorithm 10.2: Checking the authenticity of share/partcipant in secret reconstruction

Input: Received Share, RS , of having the size $row \times 1024$
Output: Whether the RS is fake or not.

Take one row at a time from the received share, RS.
$i = 0$
while $i \leq row$ **do**
    Apply SHA-512 on the bits(1024) of the i^{th} row of the share.
    Compare the resulting 512 bits Hash Value with the authentication signal which is embedded in the i^th row of the alpha channel .
    If both 512 bits Hash Value are not matching then exit the loop by returning the share as the fake one, else continue the loop.

end
Return the share as the genuine one
Stop.

In Algorithm 10.1, the authentication signal of each share is generated and embedded in the alpha channel of the corresponding shares. It is described in the Figure 10.3. Algorithm 10.2, describes the process of checking the authenticity of share/participant in secret reconstruction phase. It is illustrated in Figure 10.4.

During secret reconstruction the authentication signal generation phase is again performed on the received shares (using the Algorithm 10.1). After that, the calculated authentication signal is compared with the one which is already with the received share (which is embedded in alpha channel of the share). If both the signals are same, then we can say that the share is genuine, that means, the corresponding participant is honest one. Otherwise we can say that the received share is a fake or the participant is a cheater. The basic idea behind this is, if the share is get modified then the authentication signal of the modified share will be different from the original. This is actually, the strength of the SHA-512

algorithm. In SHA-512, same hash value for two different messages will never happen. A single bit change in the message will change the hash value. Here we can note the significance of the PNG image format as well, the change in the pixels of the image will not change the information that is embedded in the alpha channel.

10.3 Security Analysis

The security of the proposed method directly depends on the strength of the SHA-512 algorithm. Suppose a single pixel value changed from 0 to 1 or 1 to 0 for any of the shares, then the number of bit positions that differ between the authentication signal generated in reconstruction phase and the authentication signal generated and embedded in the secret sharing phase is 253, almost the half the bit positions of the authentication signal (of 512 bits), indicating that SHA-512 has a good avalanche effect.

10.4 Concluding Remarks

In this chapter we have discussed about a novel method for cheater identification in the secret sharing schemes. The most important advantage of this method is, it is applicable for all the secret sharing schemes. And the security analysis shows that even if a single pixel is modified in any of the shares then the authentication signal generated using SHA will be different for that share, and that will lead to the identification of the cheater.

Chapter 11

Summary and Future Directions

In this chapter we summarize our contribution in this thesis, draw several useful inferences and suggest future scopes.

11.1 Brief Summary

The secret sharing area is really vast and the mathematical foundation is really fascinating. The area is an active area of research from 1979. Secret sharing schemes are ideal for storing information that is highly sensitive and highly important. Examples include: encryption keys, missile launch codes, and numbered bank accounts. Each of these pieces of information must be kept highly confidential, as their exposure could be disastrous, however, it is also critical that they should not be lost. Traditional methods for encryption are ill-suited for simultaneously achieving high levels of confidentiality and reliability.

We have done a detailed review of the secret sharing schemes and also visual cryptographic schemes, which comes under different categories like.

Secret sharing scheme:

1. Shamir 's Scheme
2. Blakley 's Scheme
3. Li Bai 's Scheme

Diverse visual cryptography schemes:

1. Traditional Visual Cryptography
2. Extended Visual Cryptography
3. Halftone Visual Cryptography
4. Recursive Threshold Visual Cryptography Scheme
5. Random Grids based Visual Cryptography
6. Color Visual Cryptography Schemes
7. Probabilistic Visual Cryptography
8. Region Incrementing Visual Cryptography
9. Progressive Visual Cryptography
10. Segment based Visual Cryptography Scheme
11. Cheating Immune Visual Cryptography Schemes
12. Size Invariant Visual Cryptography
13. User-friendly Visual Secret sharing Scheme
14. Dynamic Visual Cryptography

15. OR and XOR Visual Cryptography

This helped in the thorough understanding of the existing schemes and their advantages and disadvantages. Development of application specific schemes are our major objective. And simple and efficient schemes are developed using different number theoretic techniques.

The following are the summary of the major contributions:

- We considered the previous research articles and schemes related to secret sharing and visual cryptography for the study, and developed a secret sharing scheme using Gray Code and XOR operation that can be applicable for both data sharing and image sharing. Gray Code is also known as Reflected Binary Code (RBC). It is termed after Franky Gray, who was a physicist and researcher at bell lab. It is a binary numeral system often used in electronics, but with many applications in mathematics. In Gray Code the two successive values differ in only one bit. The scheme which is developed using Gray Code and XOR operation, the secret is shared using the concept Gray Code and the secret is reconstructed using the XOR operation. The use of the scheme in the area of cryptography or secret(text, image) encryption is also explored.

- A specially designed number system called POB (Permutation Ordered Binary) system developed by Sreekumar et al [SS09] is studied. We used the POB number system in Visual Secret Sharing along with Chinese Remainder Theorem.

- Polynomial Interpolation is studied and we used Newton's polynomial Interpolation along with Mod operator in sharing visual secrets. The only requirement of this scheme is, the image should be in PNG format.

- Major contribution of the dissertation is in the application of the secret sharing scheme using Lagrange Polynomial Interpolation in the area Broadcast Encryption. Here we have used the secret sharing scheme to share the master key and distributing the shares of the master key instead of sharing the master key as such among the users.
- We have also considered a cheater identifications scheme in visual secret sharing schemes using Secure Hash Algorithm, that can be used along with any of the already existing schemes. The only limitation in this case is, the image should be in PNG format.

11.2 Future Directions

We have given the theoretical background of secret sharing schemes and the historical development of the subject. The evaluation of the various schemes are accounted in the initial chapters. We have included a few examples to improve the readability of the thesis. We have tried to maintain the rigor of the treatment of the subject.

The limitations and advantages of the various forms of secret sharing schemes are brought out and several new schemes are included in the thesis. Being new system, there is much scope for further development in this area.

Our research findings are well appreciated by the research community in computer science. $Appendix - B$ contains the list of publications of some of our research findings in this area.

All the new schemes we have introduced, have potential for a lot of research activities in future. We propose to continue this work and explore the possibilities of applications these schemes in other areas also.

Appendix A

List of Notations

$A \in B$	A belongs to B.
X^n	$X \times X \times \cdots \times X$.
$a \mod b$	The remainder of the integer division of a by b.
$n!$	n factorial $1 \times 2 \times \cdots \times n$.
$a \equiv b(\mod m)$	a and b are congruent modulo m.
$a^{-1} \mod n$	Multiplicative inverse of a modulo m for some $a \in \mathbb{Z}_m^*$.
$\oplus$	The XOR operation.
$O(n)$	The time complexity of an algorithm.
$\binom{n}{r}$	Combinatorial symbol n choose r.
$E(P,k)$	Encryption based on key k.
$D(C,k)$	Decryption based on key k.
SHA	Secure Hash Algorithm.
VC	Visual Cryptography.
VCS	Visual Cryptographic Scheme.
VSS	Visual Secret Sharing.
RG	Random Grid.
PVC	Progressive Visual Cryptography.
$CIVCS$	Cheating Immune Visual Cryptography Scheme.

n	Number of participants in secret sharing.
$\mathcal{K}, k$	Key.
$\mathcal{M}$	Secret Messages.
P_i, U_i	The participant or user i.
$S_i, Share_i$	Share assigned to user i.

Appendix B

List of Publications Related to This Thesis and Achievement

Part of the work presented in this thesis has been published/communicated to journals

1. Deepika M P, Dr. A Sreekumar, "Key Distribution Scheme in Broadcast Encryption Using Polynomial Interpolation", International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 12, Number 24 (2017), pp. 15475-15483, Research India Publications, http://www . ripublication . com.

2. Deepika M P, Dr. A Sreekumar, "Visual Cryptography Scheme Using Gray Code And XOR Operation", International journal of current engineering and scientific research(IJCESR), (ISSN 23938374) Print, (ISSN 2394-0697) online, Vol. 4 Issue 9, TRO Publication, September 2017.

3. Deepika M P, Dr. A Sreekumar, "A Novel secret sharing scheme using POB number system and CRT", International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 11, Number 3 (2016), pp. 2049-2054, Research India Publications, http://www.ripublication.com.

4. Deepika M P, Dr. A Sreekumar, "Cheater identification in Visual secret sharing schemes using SHA Algorithm and Alpha channel", International Journal of Computer Applications Technology and Research, Volume 4 Issue 11, pp.838 - 845, 2015, ISSN 23198656.

Part of the work included in the thesis has been presented in various National/International conferences

1. Deepika M P, Dr. A Sreekumar, "Visual Secret Sharing using Newton Interpolation Polynomial and Mod operator with PNG Images", IEEE 4th International Conference on Innovation in Information, Embedded and Communication Systems (ICIIECS'17), March 17-18, 2017, DOI:10.1109/ICIIECS.2017.8275917, IEEE Xplore.

2. Deepika M P, Dr. A Sreekumar, "Secret sharing scheme using Gray code and XOR operation", Second IEEE International Conference on Electrical, Computer and Communication Technologies (IEEE ICECCT 2017), February 22-24, 2017, DOI: 10.1109/ICECCT. 2017.8117932, IEEE Xplore.

Notable achievements/contribution during the period of thesis work

1. Won the title "Young Scientist of the year 2017", by IOSRD(International Organization of Scientific Research and

Development), in IOSRD Annual awards-2017 held on 29^{th} and 30^{th} Dec 2017, at Chennai.

2. Won the title "Outstanding Woman in Engineering (Specialization - Information Technology)", by the Venus International Women Awards-VIWA 2018 held on 3^{rd} March 2018, at Chennai.

Bibliography

[ABSS96a] G. Ateniese, C. Blundo, A D Santis, and D. Stinson. Visual cryptography for general access structures. *Information and Computation*, 129(2):86–106, 1996.

[ABSS96b] G. Ateniese, C. Blundo, A D. Santis, and D. Stinson. Extended schemes for visual cryptography. *Theoretical Computer Science*, 250:1–16, June 1996.

[ABSS01] G. Ateniese, C. Blundo, A D. Santis, and D. Stinson. Extended capabilities for visual cryptography. *Theoretical Computer Science*, 250(1-2):143–161, 2001.

[AS92] N. Alon and J Spencer. The probabilistic method. *Wiley-Interscience*, 2, 1992.

[BDS03] C. Blundo, A D DArco, P. Santis, and D.R Stinson. Contrast optimal threshold visual cryptography schemes. *SIAM Journal on Discrete Mathematics*, 16(2):224–261, 2003.

[Bla79] G. Blakley. Safeguarding cryptographic keys. *In Proc. of AFIPS National Computer Conference*, 1979.

[Bor04] Bernd Borchert. Segment-based visual cryptography. pages 1–9, WSI-2007-04.

[BWag] Ingrid Biehl and Susanne Wetzel. Traceable visual cryptography. *In ICICS97: Proceedings of the First International Conference on Information and Communication Security*, London,UK,1997 Springer-Verlag.

[Cam00] Alistair Campbell. Traceable visual cryptography. *The designers Lexicom Chronicle Books, San Francisco, CA, USA*, 2000.

[CC89] G H Chiou and W T Chen. Secure broadcasting using the secure lock. *IEEE Trans. on Software Engineering*, 15:929–934, 1989.

[CCH+07] Yung Fu Chen, Yung Kuan Chan, Ching Chun Huang, Meng Hsiun Tsai, and Yen Ping Chu. A multiple-level visual secret sharing scheme without image size expansion. *Information Sciences*, 177(21):4696–4710, 2007.

[CSFM05] Stelvio Cimato, Alfredo De Santis, Anna Lisa Ferrara, and Barbara Masucci. Ideal contrast visual cryptography schemes with reversing. *Information Processing Letters*, 93(4):199–206, 2005.

[CT09] T H Chen and K H Tsao. Visual secret sharing by random grids revisited. *Pattern Recognition*, 42(9):2203–2217, 2009.

[CT11] T H. Chen and K H. Tsao. Threshold visual secret sharing by random grids. *Journal of Systems and Software.*, 84(7):1197–1208., 2011.

[CW11] T H Chen and C S Wu. Efficient multi-secret image sharing based on boolean operations. *Signal Processing*, 91(1):90–97, 2011.

[DF02] Yevgeniy Dodi and Nelly Fazio. Public key broadcast encryption for stateless receivers. *ACM Workshop on Digital Rights Management-Springer*, 2002.

[DK04] Quang Viet Duong and Koru Kurosawa. Almost ideal contrast visual cryptography with reversing. *In CT-RSA*, 2004.

[DYK04] J Duo, W Yan, and M. Kankanhalli. Visual cryptography for print and scan applications. *In Proc. IEEE Int. Symp. Circuits Syst.*, pages 572–575, 2004.

[FA04] M S. Fu and O C. Au. Joint visual cryptography and watermarking. *In Proc. IEEE Int. Conf. Multimedia ERpo*, pages 975–978, 2004.

[Fan08] W. P. Fang. Friendly progressive visual secret sharing. *Pattern Recognition*, 41:1410–1414, 2008.

[FN94] A. Fiat and M. Naor. Broadcast Encryption. *Advances in Cryptology - Crypto 93 Lecture Notes in Computer Science. Springer*, 773:480–491, 1994.

[GK02] Meenakshi Gnanaguruparan and Subhasn Kak. Recursive hiding of secrets in visual cryptography. *Cryptologia*, 26(1):68–76, 2002.

[GSW00] J A Garay, J Staddon, and A Wool. Long-lived broadcast encryption. *Advances in Cryptology CRYPTO2000 Lecture Notes in Computer Science*, 1880:333–352, 2000.

[HC06] Y T. Hsu and L W. Chang. A new construction algorithm of visual cryptography for gray level images. *in Proc. IEEE Int. Symp. Circuit Syst.*, pages 1430–1433, 2006.

[HCT06] G. Horng, T H Chen, and D S Tsai. Cheating in visual cryptography. *Des Codes and Cryptography*, 38(2):219–236, 2006.

[HDS02] Halevy, Dani, and Adi Shamir. The LSD broadcast encryption scheme. *Proceedings of the 22Nd Annual International Cryptology Conference on Advances in Cryptology*, pages 47–60, 2002.

[HG06] A. Houmansadr and S. Ghaemmaghami. A novel video watermarking method using visual cryptography. *In Proc. IEEE Int. Conf. Eng. Intell. Syst.*, pages 1–5, 2006.

[HH05] C S. Hsu and Y C. Hou. Copyright protection scheme for digital images using visual cryptography and sampling methods. *Opt. Eng.*, 44(7):077003–10, 2005.

[Hou03] Y C. Hou. Visual cryptography for color images. *Pattern Recognit.*, 36:1619–1629, 2003.

[HQ11] Young Chang Hou and Zen Yu Quan. Progressive visual cryptography with unexpected shares. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(11), November 2011.

[HT07] C M Hu and W G Tzeng. Cheating prevention in visual cryptography. *IEEE Trans.Image Process*, 16(1), 2007.

[IH98] Ryo Ito and Hidenori Kuwakdo Hatsukazu. Image size invariant visual cryptography. *IEICE Trans. Fundamentals*, pages 2172–2177, 1998.

[JHC+05] N S Jho, J Y Hwang, J H Cheon, M Kim, D H Lee, and E S Yoo. One-way chain based broadcast encryption scheme.

In Advances in Cryptology-Eurocrypt-Springer, 3494:559–574, 2005.

[JYK05] D Jin, W Q Yan, and M S Kanakanhalli. Progressive color visual cryptography. *Journal of Electronic Imaging*, 14, 2005.

[KAL11] InKoo Kang, Gonzalo R. Arce, and Heung Kyu Lee. Color extended visual cryptography using error diffusion. *IEEE Transactions on Image Processing*, 20(1):132–145, 2011.

[KK87] O. Kafri and E. Keren. Encryption of pictures and shapes by random grids. *Optics Letters*, 12(6):377–379, 1987.

[KSW03] Noam Kogan, Yuval Shavitt, and Avishai Wool. A practical revocation scheme for broadcast encryption using smart cards. *24th IEEE Symposium on Security and Privacy (Extended abstract)*, May 2003.

[Liu68] Chung Laung Liu. Introduction to combinatorial mathematics, McGraw-Hill, New York. 1968.

[LLH89] C. Laih, J. Lee, and L. Harn. A new threshold scheme and its application is designing the conference key distribution cryptosystem. *Information Processing Letters*, 32:95–99, 1989.

[LS98] Michael Luby and Jessica Staddon. Combinatorial bounds for broadcast encryption. *In: Nyberg K. (eds) Advances in Cryptology EUROCRYPT'98. EUROCRYPT 1998. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg*, 1403:512–526, 1998.

[LT03a] C C. Lin and W H. Tsai. Visual cryptography for gray-level images by dithering techniques. *Pattern Recognit. Lett.*, 24:349–358, 2003.

[LT03b] Chang Chou Lin and Wn Hsiang Tsai. Visual cryptography for gray-level images by dithering techniques. *Pattern Recognition Letters*, 24:349–358, 2003.

[LWL08] F. Liu, C K. Wu, and X J. Lin. Colour visual cryptography schemes. *IET Information Security*, 2(4):151–165, 2008.

[MA14] A. Muthulakshmi and R. Anitha. Identity-based broadcast encryption for multi-privileged groups using chinese remainder theorem. *Int. J. Information and Computer Security*, 6(3), 2014.

[Mac00] L A. MacPherson. Gray level visual cryptography for general access structrue. *M. Eng. thesis.Univ. Waterloo, Ontario, Canada*, 2000.

[MKOS10] Ak. Murat, K. Kaya, K. Onarlioglu, and A A Seluk. Efficient broadcast encryption with user profiles. *Information Sciences*, 180(6):1060–1072, 2010.

[MWL06] C. Ma, Y. Wu, and J. Li. Broadcast group-oriented encryption for group communication. *Communications,Circuits and Systems Proceedings*, pages 1623–1626, 2006.

[NL00] Ching Nung and Chi Sung Laih. New colored visual secret sharing schemes. *Design, codes and Cryptography*, 20(3):325–336, 2000.

[NNL01] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. *In: Kilian J. (eds) Advances in*

Cryptology CRYPTO 2001. CRYPTO 2001. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2139:41–62, 2001.

[NP97] M. Naor and B. Pinkas. Visual authentication and identification. *Adv. Cryptology*, 1294:322–336, 1997.

[NS95] M. Naor and A. Shamir. Visual cryptography. *In Proceedings of the Advances in Cryptology, Eurocrypt 94, in LNCS*, 950:1–12, 1995.

[NY02] M. Nakajima and Y. Yamaguchi. Extended visual cryptography for natural images. *J. WSCG*, 10(2):303–310., 2002.

[PK08] A. Parakh and S Kak. A recursive threshold visual cryptography scheme. *Cryptology ePrint Archive, Report 2008/535*, 2008.

[PK11] A. Parakh and S. Kak. Space efficient secret sharing for implicit data security. *Information Sciences*, 181:335–341, 2011.

[PKca] A. Parakh and S Kak. A tree based recursive information hiding scheme. *Proceedings of IEEE ICC 2010 Communication and Information System Security Symposium (ICC10 CISS)*, 2010,May 23-27,Cape Town, South Africa.

[PS06] R De Prisco and A De Santis. Cheating immune (2,n)-threshold visual secret sharing. *4116*, pages 216–228, 2006.

[PS10] R De Prisco and A De Santis. Cheating immune threshold visual secret sharing. *Comput. J.*, 53:1485–1496, 2010.

[SH12] Norranut Saguansakdiyotin and Pipat Hiranvanichakorn. Broadcast encryption based on braid groups. *IJCSNS International Journal of Computer Science and Network Security*, 12(2), February 2012.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 1979.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. *Proceedings of CRYPTO'84-Springer-Verlag*, 196:47–53, 1984.

[SHD05] C H Scott, Huang, and Ding Zhu Du. New constructions on broadcast encryption and key pre distribution schemes. *Proc. IEEE Computer and Communications Societies INFOCOM 2005 doi* : 10.1109/$INFCOM$.2005.1497919, pages 515–523, March 2005.

[SJ12] Shyong Jian Shyu and Hung Wei Jiang. Efficient construction for region incrementing visual cryptography. *IEEE Transactions on Circuits and Systems For Video Technology*, 22(5), May 2012.

[SS09] A Sreekumar and S B Sundar. An efficient secret sharing scheme for n out of n scheme using POB number system. *Hack.in*, pages 33–37, 2009.

[TCH07] D S Tsai, T H Chen, and G. Horng. A cheating prevention scheme for binary visual cryptography with homogeneous secret images. *Pattern Recognition*, 40(8):2356–2366, 2007.

[THH+05] P Tuyls, Hollmann, D L Henk, J H V Lint, and L. Tolhuizen. XOR based visual cryptography schemes. *Des Codes Crypt.*, 37(1):169–186: doi:10.1007/s10623–004–3816–4, 2005.

[TL03] C C Thien and J C Lin. An image-sharing method with user-friendly shadow images. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1161–1169, 2003.

[TWWW11] C S Tasai, H C Wang, H C Wu, and C H M Wang. Cheating immune threshold visual secret sharing. *International Journal Of Innovative Computing ,Information And Control*, 7(7(A)), July 2011.

[VHT97] Eric R Verheul, C A Henk, and Van Tilborg. Constructions and properties of k out of n visual secret sharing schemes. *Design, codes and Cryptography*, 11(2):179–196, 1997.

[WC98] C C Wu and L H Chen. Study on visual cryptography. *Maters thesis -Institute of Computer and information Science, National Chiao tung University, Taiwan,R.O.C*, 1998.

[WH11] Ran Zan Wang and Shuo Fang Hsu. Tagged visual cryptography. *IEEE Signal Processing Letters,*, 18(11), November 2011.

[Yan04] Ching Nung Yang. New visual secret sharing schemes using probabilistic method. *Pattern Recognition Letters*, 25(4):481–494, 2004.

[YC05a] Ching Nung Yang and Tse Shish Chen. Aspect ratio invariant visual secret sharing schemes with minimum pixel expansion. *Pattern Recognition Letters*, 26(2):193–206, 2005.

[YC05b] Ching Nung Yang and Tse Shish Chen. Size adjustable visual secret sharing schemes. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, 88-A(9):2471–2474, 2005.

[YC06a] Ching Nung Yang and Tse Shish Chen. New size-reduced visual secret sharing schemes with half reduction of shadow size. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences.*, 89-A(2):620–625, 2006.

[YC06b] Ching Nung Yang and Tse Shish Chen. Reduced shadow size in aspect ratio invariant visual secret sharing schemes using a square block wise operation. *Pattern Recognition*, 39(7):1300–1314, 2006.

[YC06c] Ching Nung Yang and Tse Shish Chen. Visual secret sharing scheme: Improving the contrast of a recovered image via different pixel expansions. *In Aurelio C.Campilho and Mohamed S.Kamel,editors, ICIAR(1) Lecture Notes in Computer Science ,Springer*, 4141:468–479, 2006.

[ZAC] Z. Zhou, G R. Arce, and G Di Crescenzo. Halftone visual cryptography. *IEEE TRANS. IMAGE PROCESS*, 15(8):2441.

[Zha98] Yuefeng Zhang. Space-filling curve ordered dither. *Computers and graphics*, 22(4):559–563, 1998.

[ZZT11] Xingwen Zhao, Fangguo Zhang, and Haibo Tian. Dynamic asymmetric group key agreement for ad hoc networks. *Ad Hoc Networks*, 9:928–939, 2011.