Secret Sharing Schemes using Visual Cryptography

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
A.Sreekumar

Under the guidance of

Dr. S. Babusundar

Department of Computer Applications
Cochin University of Science and Technology
Kochi 22, India

June 2009

CERTIFICATE

Certified that the work presented in this thesis entitled "**Secret Sharing Schemes Using Visual Cryptography**" is based on the bona fide research work done by **A. Sreekumar** under my guidance in the Department of Computer Applications, Cochin University of Science and Technology, Kochi – 22, and has not been included in any other thesis submitted previously for the award of any degree.

Kochi – 22
June 23, 2009

Dr. S. Babusundar
Professor in Computer Applications,
(Supervising Guide)
Department of Computer Applications,
Cochin University of Science and Technology

DECLARATION

I hereby declare that the present work entitled "**Secret Sharing Schemes Using Visual Cryptography**" is based on the original work done by me under the guidance of Dr. S. Babusundar, Department of Computer Applications, Cochin University of Science and Technology, Kochi – 22 and has not been included in any other thesis submitted previously for the award of any other degree.

Kochi – 22,
June 23, 2009

A. Sreekumar

ACKNOWLEDGEMENT

There are many people who deserve more than just an acknowledgement on this page. The following people deserve my heartfelt gratitude.

My supervising guide, Dr. S. Babusundar, Professor, Department of Computer Applications, Cochin University of Science and Technology, for his inspiring guidance and constant encouragement for his valuable suggestions, constant support, encouragement and always sharing his positive thinking during my candidature.

My thanks goes to my colleagues Dr. K. V. Pramod, Dr. B. Kannan and Smt. S. Malathi, for providing me with a stimulating research environment.

I also wish to thank G. Santhosh Kumar for wonderful collaborations. I thank Smt. Sindhu Menon and Smt. Padmakumary for their timely supports.

I would like to thank all staff members of the Department of Computer Applications, for helping me in one way or other.

I am extremely grateful to my family for the support rendered to me throughout my research work. Above all, I thank the Supreme Power, who has always showered countless blessings upon me.

Sreekumar, A.

Contents

Chapter 1

Secret Sharing Schemes

1.1 Introduction

Handling secret has been an issue of prominence from the time human beings started to live together. Important things and messages have been always there to be preserved and protected from possible misuse or loss. Some time 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. Some of the formulae of vital combinations of medicinal plants or roots or leaves, in Ayurveda were known to a single person in a family. When he becomes old enough, he would rather share the secret formula to a chosen person from the family, or from among his disciples. There were times when the person with the secret dies before he could share the secret. Probably, similar incidents might have made the genius of those era to think of sharing the secrets with more than one person so that in the event of death of the present custodian, there will be at least one other person who knows the secret.

Secret sharing in other forms were prevailing in the past, for other reasons also. Secrets were divided into number of pieces and given to the same number of people. To ensure unity among the participating people, the head of the family would share the information with respect to wealth among his children and insist that after his death, they all should join together to inherit the wealth.

To test the valor of the youth of a nation, a king, would hide treasure in some place in his kingdom and information about it would be placed in pieces at different places of varying grades of difficulty to reach. Only the brave and the intelligent would reach the treasure.

Military and defense secrets have been the subject matter for secret sharing in the past as well as in the modern days. Secret sharing is a very hot area of research in Computer Science in the recent past. Digital media has replaced almost all forms of communication and information preservation and processing. Security in digital media has been a matter of serious concern. This has resulted in the development of encryption and cryptography. Uniform secret sharing schemes form a part of this large study.

A Secret sharing scheme is a method of dividing a secret information into two or more pieces, with or without modifications, and retrieving the information by combining all or predefined sub collection of pieces.

The pieces of information are called **shares** and the process responsible for the division is called **dealer**. A predefined subcollection of shares which contains the whole secret in some form is called an **allowed coalition**. The process responsible for the recovery of the secret information from an allowed coalition is called a **combiner**.

A share contains, logically, a part of the information, but will be of no use. Thus no single share is of any threat to the confidentiality of the secret information. It is also envisaged that after the dealer process is over, the original information can be destroyed forever. This would mean that even the person responsible for the dealer process will not be a threat, thereafter. The secret information is recovered from any allowed coalition using the recovery process called combiner. The combiner would be able to recover the secret information, only if, all shares in the allowed coalition is present and not with any fewer number of shares. Thus, in an allowed coalition, each member share is equally important such that without anyone of them, the secret information cannot be accessed.

Allowed coalition is also referred in the literature by other names too, such as, **authentic collection**, **qualified collection** or **authorized set**. We, in our work, preferred to call the sub collection of shares as allowed coalition. The set of all allowed coalitions of participants is called the **access structure** and is usually denoted by Γ.

Secret Sharing is an important tool in Security and Cryptography. In many cases there is a single master key that provides the access to important secret information. Therefore, it would be desirable to keep the master key in a safe place to avoid accidental and malicious exposure. This scheme is unreliable: if master key is lost or destroyed, then all information accessed by the master key is no longer available. A possible solution would be that of storing copies of the key in different safe places or giving copies to trusted people. In such a case the system becomes more vulnerable to security breaches or betrayal [53], [30]. A better solution would be, breaking the master key into pieces in such a way that only the concurrence of certain predefined trusted people can recover it. This has proven to be an important tool in management of cryptographic keys and multi-party secure protocols (see for example [33]).

As a solution to this problem, Blakley [9] and Shamir [53] introduced $(k,\ n)$ threshold schemes. A $(k,\ n)$-threshold scheme allows a secret to be shared among n participants, in such a way that, any k of them can recover the secret, but $k-1$, or fewer, have absolutely no information on the secret.

Ito, Saito, and Nishizeki [36] described a more general method of secret sharing. An access structure is a specification of all subsets of participants who can recover the secret and it is said to be monotone if any set which contains a subset that can recover the secret, can itself recover the secret. Ito, Saito, and Nishizeki gave a methodology to realize secret sharing schemes for arbitrary monotone access structures.
Subsequently, Benaloh and Leichter [5] gave a simpler and more efficient way to realize such schemes.

An important issue in the implementation of secret sharing scheme is the size of the shares distributed to the participants, since the security of a system degrades as the amount of the information that must be kept secret increases. So the size of the shares given to the participants is a key point in the design of secret sharing schemes. Therefore, one of the main parameters in secret sharing is, the **average information rate** ρ, of the scheme, which is defined as the ratio between the average length (in bits) of the shares given to the participants and the length of the secret. Unfortunately, in all secret sharing schemes the size of the shares cannot be less than the size of the secret, and so the information rate cannot be less than one. Moreover, there are access structures, for which, any corresponding secret sharing scheme must give to some participant a share of size strictly bigger than the secret size. Secret sharing schemes with information rate equal to one are called **ideal**. A secret sharing scheme is called efficient if the total length of the n shares is polynomial in n.

1.2 Principle of secret splitting

The simplest sharing scheme splits a message between two people. Consider the case where Daniel has a message M, represented as an integer, that he would like to split between two people Alice, and Bob, in such a way that neither of them alone can reconstruct the message. A solution to the problem readily lends itself: Choose a random number r. Then r and $M - r$ are independently random. He gives $M - r$ to Alice and r to Bob as their shares. Each share by itself means nothing in relation to the message, but together, they carry the message M. To recover the message, Alice and Bob have to simply add their shares together.

Here is another method in which Daniel splits a message between Alice and Bob:

1. Daniel generates a random-bit string R, of the same length as the message, M.
2. Daniel XORs M with R to generate S.
   i.e., $M \oplus R = S$.
3. Daniel gives R to Alice and S to Bob.
   To reconstruct the message, Alice and Bob have only one step to do:

4. Alice and Bob XOR their pieces together to reconstruct the message:

$$R \oplus S = M.$$

This technique is absolutely secure. Each piece, by itself, is absolutely worthless. Essentially, Daniel is encrypting the message with a one-time pad and giving the cipher text to one person and the pad to the other person. The one-time pad, which is an unbreakable cryptosystem, was developed by Gilbert Vernam and Joseph Mauborgne in 1917. It has perfect security [42]. No amount of computing power can determine the message from one of the pieces.

Shares can be constructed in several alternative forms using a random number. For example, $M - \frac{r}{2}$ and $M + \frac{r}{2}$ or Mr and $\frac{M}{r}$. Depending on the choice of constructing shares, suitable combiner has to be created.

It is easy to extend this scheme to more people:

Now let us examine the case where we would like to split the secret among three people. Any suitable splitting and combining method can be evolved. For example, the method employed for splitting the secret into two shares can be extended with the help of two random numbers r and s. For example, consider $M - r - s$, r and s as the three shares. To reconstruct the message M, simply add the shares. Similarly, we can evolve splitting and combining methods for a secret to be distributed as n shares with the condition that only when all of them are combined together, the secret could be recovered.

Daniel divides up a message into $n (\geq 2)$ pieces:

1. Daniel generates $n-1$ random-bit strings $S_1, \ldots, S_{n-1}$ having the same length as the message, M
2. Daniel XORs M with $n-1$ random-bit strings to generate S_n:
   i.e., $\quad M \oplus S_1 \oplus \ldots \oplus S_{n-1} = S_n.$
3. Daniel distributes the $S_i, (i = 1, \ldots, n)$ to the n participants.
4. The n participants working together can reconstruct the message:
   $S_1 \oplus S_2 \oplus \ldots \oplus S_{n-1} \oplus S_n = M.$

Note: This protocol has a problem: If any of the pieces gets lost or is not available, the message cannot be reconstructed, since each piece is as critical to the message as every other piece.

1.3 History of Secret Sharing

In [43], Liu considered the following 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 specific lock, which they cannot open. Consider a 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 $\binom{11}{5} = 462$ ways, and among ten scientists, five scientists can be selected in $\binom{10}{5} = 252$ ways. (More details about one form of distribution of keys of the various locks to the scientists is included in Appendix 1.)

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 is 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.

1.3.1 Threshold scheme

In 1979 Shamir [53] and Blakley [9] introduced the concept of sharing of the secret message as a means and a method of making the message secure. Under this scheme, the message M is divided into n pieces $M_1, M_2, M_3, \ldots, M_n$, with or without transformation of the message, in such a way that, for a specified k, $(2 \leq k \leq n)$,

1. knowledge of any k or more pieces-M_i makes M computable;
2. knowledge of any $k - 1$ or fewer M_i pieces leaves M completely undetermined (in the sense that all its possible values are equally likely).

Such a scheme is called a $(k,\ n)$-threshold scheme. The parameter $k \leq n$ is called the threshold value.

1.3.2 The Shamir Secret Sharing Scheme

Let $k,\ n \in \mathbb{Z}, k \leq n$. We will describe the $(k,\ n)$ Secret Sharing Scheme by Shamir. It uses a prime number, p, which is greater than n and the set of possible secret. The scheme is based on the following lemma.

Lemma 1.1
Let $k \in \mathbb{Z}$. Also let $x_i, y_i \in \mathbb{Z}/_p\mathbb{Z}, 1 \leq i \leq k$, where the

x_i are pairwise distinct. Then there is a unique polynomial $b \in (\mathbb{Z}/_p\mathbb{Z})[X]$ of degree $\leq k-1$ with $b(x_i) = y_i$, $1 \leq i \leq k$.

Proof: The Lagrange interpolation formula yields the polynomial

$$b(X) = \sum_{i=1}^{k} y_i \prod_{j=1, j\neq i}^{k} \frac{(x_j - X)}{(x_j - x_i)} \tag{1.1}$$

It satisfies $b(x_i) = y_i$, $1 \leq i \leq k$. This shows that at least one polynomial exists with the asserted properties. Now we determine the number of such polynomials.

Let $b \in (\mathbb{Z}/_p\mathbb{Z})[X]$ be such a polynomial. Write

$$b(X) = \sum_{j=0}^{k-1} b_j X^j, \quad \textit{where}, \; b_j \in \mathbb{Z}/_p\mathbb{Z}, \; 0 \leq j \leq k-1.$$

From $b(x_i) = y_i$, $1 \leq i \leq k$, we obtain the linear system

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{k-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{k-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_k & x_k^2 & \ldots & x_k^{k-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{k-1} \end{bmatrix} \tag{1.2}$$

The coefficient matrix

$$U = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{k-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{k-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_k & x_k^2 & \ldots & x_k^{k-1} \end{bmatrix}$$

is *Vandermonde matrix.* Its determinant is

$$det\, U = \prod_{1 \leq i < j \leq k} (x_i - x_j).$$

Since the x_i are distinct by assumption, the determinant is non zero. So the rank of U is k. This implies that the kernel of the coefficient matrix (1.2) has rank 0, and the number of solutions of our linear system is $p^0 = 1$. Hence the uniqueness. Now we are able to describe the scheme.

1.3.3 System Design

The dealer chooses a prime number p, which is greater than n and the set of possible secret and nonzero distinct elements $x_i \in \mathbb{Z}/_p\mathbb{Z}$, $1 \leq i \leq n$. Those elements in $\mathbb{Z}/_p\mathbb{Z}$ can, for example, be represented by their least nonnegative representative.

The shares

Let $S \in \mathbb{Z}/_p\mathbb{Z}$ be the secret.

1. The dealer secretly at random chooses elements $b_j \in \mathbb{Z}/_p\mathbb{Z}$, $1 \leq i \leq k-1$ and constructs the polynomial
$$b(X) = \sum_{i=1}^{k-1} b_i x^i + S. \tag{1.3}$$
It is of degree $\leq k-1$.
2. The dealer computes the shares $y_i = b(x_i),\ 1 \leq i \leq n$.
3. The dealer distributes the share (x_i, y_i) to the i^{th} shareholder, $1 \leq i \leq n$.

So the secret is value $b(0)$ of the polynomial $b(X)$.

Reconstruction of the secret

Suppose that k shareholders collaborate. Without loss of generality assume that the shares are numbered, such that, $y_i = b(x_i)$, $1 \leq i \leq k$ with the polynomial b[X] from (1.3). Now we have

$$b(x) = \sum_{i=1}^{k} y_i \prod_{j=1, j \neq i}^{k} \frac{x_j - X}{x_j - x_i} \tag{1.4}$$

In fact this polynomial satisfies $b(x_i) = y_i$, $1 \leq i \leq k$ and by lemma 1.1 there is exactly one such polynomial of degree $\leq k-1$. Therefore, the shareholders can reconstruct the secret as

$$S = b(0) = \sum_{i=1}^{k} y_i \prod_{j=1, j \neq i}^{k} \frac{x_j}{x_j - x_i} \tag{1.5}$$

1.3.4 A method of solution

Now a secret is shared by computing points on a random polynomial in $(\mathbb{Z}/_p\mathbb{Z})[X]$. So first we must find a way of representing the "plaintext" secret as a set of class modulo p. This is not really part of secret sharing process; it is merely a way to prepare the secret so that it can be shared. To keep the things as simple as possible, we will assume that the "plaintext" secret contains only words written in uppercase letters. Thus the secret is ultimately a sequence of letters and blank spaces. The first step consists of replacing each letter of the secret by a number, using the following correspondence:

A	B	C	D	E	F	G	H	I	J	K	L	M
10	11	12	13	14	15	16	17	18	19	20	21	22

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
23	24	25	26	27	28	29	30	31	32	33	34	35

The blank space between words is replaced by 99. Having done that, we obtain a number, possibly a very large one, if the secret is large. However it is not a number we want, but rather classes modulo p. Therefore, we must break the numerical representation of the secret into a sequence of positive integers, each smaller than p. These are called the *blocks* of the secret.

For example, the numerical representation of the proverb
"A SMALL LEAK WILL SINK A GREAT SHIP" is

109928221021219921141020993218212199
2818232099109916271410299928171825

If we choose the prime $p = 9973$, the numerical representation of the proverb above must be broken into blocks smaller than 9973. One way to do this is as follows:

1099-2822-1021-2199-2114-1020-9932-1821-2199-
2818-2320-9910-9916-2714-1029-9928-1718-25

When secret is reconstructed, one obtains a sequence of blocks. The blocks are then joined together to give the numerical representation of the secret. It is only after replacing the numbers by letters, according to the table above, that one obtains the original secret.

Note that we have made each letter correspond to a *two-digit number* in order to avoid ambiguities. For example, if we had numbered the letters so that A corresponds to 1, B to 2, and so on, then we wouldn't be able to tell whether 12 stood for AB or for the letter L, which is the twelfth letter of the alphabet.

Of course, any convention that is unambiguous can be used instead of the one above. For example, one might prefer to use ASCII code, since the conversion of characters is automatically done by the computer.

Example 1.1

Let us return to the example we considered above. We choose $p = 9973$. *To construct a (3, 5)-threshold scheme, where any three of five people can reconstruct* S, *suppose the dealer chooses* $x_i = i, 1 \leq i \leq 5$. *Also assume that the randomly selected coefficients* b_2 *and* b_1 *are 1572 and 7583 respectively.*

Thus to share the first block of the secret, we must compute the polynomial,
$F(x) = 1572x^2 + 7583x + 1099 \pmod{9973}$ at each x_i. Thus the five shares of the first block are:

$$
\begin{aligned}
s_1 &= F(1) = 1572.1^2 + 7583.1 + 1099 \equiv 281 \pmod{9973}\\
s_2 &= F(2) = 1572.2^2 + 7583.2 + 1099 \equiv 2607 \pmod{9973}\\
s_3 &= F(3) = 1572.3^2 + 7583.3 + 1099 \equiv 8077 \pmod{9973}\\
s_4 &= F(4) = 1572.4^2 + 7583.4 + 1099 \equiv 6718 \pmod{9973}\\
s_5 &= F(5) = 1572.5^2 + 7583.5 + 1099 \equiv 8503 \pmod{9973}
\end{aligned}
$$

Sharing the whole secret, we have the following sequence of blocks:

$$
\begin{aligned}
s_1 &= \text{281-2004-203-1381-1296-202-9114-1003-1381-}\\
&\quad \text{2000-1502-9092-9098-1896-211-9110-900-9180.}\\
s_2 &= \text{2607-4330-2529-3707-3622-2528-1467-3329-3707-}\\
&\quad \text{4326-3828-1445-1451-4222-2537-1463-3226-1533.}\\
s_3 &= \text{8077-9800-7999-9177-9092-7998-6937-8799-9177-}\\
&\quad \text{9796-9298-6915-6921-9692-8007-6933-8696-7003.}\\
s_4 &= \text{6718-8441-6640-7818-7733-6639-5578-7440-7818-}\\
&\quad \text{8437-7939-5556-5562-8333-6648-5574-7337-5644.}\\
s_5 &= \text{8503-253-8425-9603-9518-8424-7363-9225-9603-}\\
&\quad \text{249-9724-7341-7347-145-8433-7359-9122-7429.}
\end{aligned}
$$

Let us see how a block of a secret can be reconstructed from the three shares. For example, the first block of S can be reconstructed from the first blocks of the shares s_2, s_3 and s_5 by using the formula (1.5):

$$
\begin{aligned}
b[0] &= \frac{2607.3.5}{1.3} + \frac{8077.2.5}{-1.2} + \frac{8503.2.3}{-3.-2} \pmod{9973}\\
&= 2607.5 + 8077.(-5) + 8503 \pmod{9973}\\
&= -18847 \pmod{9973}\\
&= 1099
\end{aligned}
$$

Similarly each block can be reconstructed.

It may be noted that, we are working with prime modulo p, in which, the numbers that appear in the denominators of formula (1.5), have inverses. We can use the Extended Euclidean Algorithm to find the inverse: $m^{-1} \pmod{p}$, where, $m \not\equiv 0 \pmod{p}$. The algorithm and an example are given as Appendix 2.

For example, suppose we want to construct the first block of the secret from s_1, s_2 and s_5. Here,

$$\begin{aligned}
b[0] &= \frac{281.2.5}{1.4} + \frac{2607.1.5}{-1.3} + \frac{8503.1.2}{-4.-3} \pmod{9973} \\
&= \frac{281.5}{2} + \frac{2607.5}{-3} - \frac{8503.1}{-6} \pmod{9973} \\
&= \frac{281.(15) - 2607.10 + 8503}{6} \pmod{9973} \\
&= \frac{-13352}{6} \pmod{9973} \\
&= -13352 * 8311 \pmod{9973} \\
&\qquad [\mathit{because}\, 6^{-1} \equiv 8311 \pmod{9973}) \\
&= -110968472 \pmod{9973} \\
&= 1099 \pmod{9973}
\end{aligned}$$

1.4 Concluding remarks

We have seen the development of the subject from the simple case of (2, 2) sharing to the general $(k,\ n)$ sharing. Some examples

are also given. The chapter also contains an algorithm for the
key allotment. We have included simple examples to highlight 2
the various aspects of the existing sharing schemes.

Chapter 2

Evolution of Secret Sharing Schemes

2.1 Introduction

In this chapter, we discuss the evolution of Secret Sharing Schemes. Some important advancements in this area are discussed and illustrated with suitable examples. The difficulties and limitations of the different schemes is also discussed.

In this section we recall some general notations used and basic definitions of secret sharing schemes.

Definition 2.1
A *secret sharing scheme* permits a secret to be shared among a set $\mathcal{P}$ of n participants in such a way that only qualified subsets of $\mathcal{P}$ can recover the secret, and any non-qualified subset has

absolutely no information on the secret. In other words, a non-qualified subset knows only that the secret is chosen from a prespecified set (which we assume is public knowledge), and they cannot compute any further information regarding the value of the secret.

Definition 2.2

An *access structure* Γ is the set of all subsets of $\mathcal{P}$ that can recover the secret.

Definition 2.3

The collection of subsets of participants that cannot reconstruct the secret is called *prohibited access structure* or simply *prohibited structure* and is usually denoted by Δ.

Definition 2.4

Let $\mathcal{P}$ be a set of participants and $2^{\mathcal{P}}$ denotes the collection of all subsets of $\mathcal{P}$. A *monotone access structure* Γ on $\mathcal{P}$ is a subset $\Gamma \subseteq 2^{\mathcal{P}}$, such that,

$$A \in \Gamma, A \subseteq B \subseteq \mathcal{P} \Rightarrow B \in \Gamma.$$

i.e, if an access structure is monotone, then, any superset of an authorized subset must be authorized.

Definition 2.5

Let $\mathcal{P}$ be a set of participants and let $\mathcal{A} \subseteq 2^{\mathcal{P}}$. The *closure of* $\mathcal{A}$*,* denoted by *cl(*$\mathcal{A}$*),* is the set

$$\text{cl}(\mathcal{A}\ ) = \{\, C \mid \exists\, B \in \mathcal{A} \text{ such that } B \subseteq C \subseteq \mathcal{P}\,\}.$$

That is, the closure of an access structure Γ is the smallest monotone access structure containing Γ.

For a monotone access structure Γ, we have, $\Gamma = cl(\Gamma)$. Suppose Γ is an access structure on $\mathcal{P}$. Then $B \in \Gamma$ is a *minimal* authorized subset, if $A \notin \Gamma$ whenever $A \subset B$. The set of *minimal* authorized subsets of Γ is denoted by Γ_{min} and is called the *basis* of Γ. Similarly, for a prohibited structure Δ on $\mathcal{P}$, $B \in \Delta$ is a *maximal* unauthorized subset, if $A \notin \Delta$ whenever $A \supset B$. It is easy to see that, for every monotone access structure, there is a corresponding set of maximal unauthorized access sets.

We can see that a monotone access structure Γ is completely characterized by the family of its minimal authorized subsets Γ_{min}, via, $\Gamma = cl(\Gamma_{min})$. Hence monotone access structures can be determined by the corresponding family of its minimal authorized subsets.

Obviously, it is hard to imagine a meaningful method of sharing a secret in which the access structure does not possess the monotone property. It is assumed that there is always at least one subset of participants who can reconstruct the secret, i.e., $\Gamma \neq \phi$, and also that every participant belongs to at least one minimal qualified subset.

For sets X and Y and for elements x and y, to avoid overburdening of the notations, we often write x for $\{x\}$, xy for $\{x, y\}$, and XY for $X \cup Y$.

Example 2.1

Let $\mathcal{P}$ be $P_1P_2P_3P_4$ and $\mathcal{A} = \{P_1P_2P_3,\ P_1P_2P_4,\ P_1P_3P_4,\ P_2P_3\}$. The subset $\mathcal{A}$ is not a monotone subset, for both P_2P_3 and $P_1P_2P_3 \in \mathcal{A}$, where one is a subset of other.

The *closure* of $\mathcal{A}$, $cl(\mathcal{A}\,) = \{P_1P_2P_3,\ P_1P_2P_4,\ P_1P_3P_4,\ P_1P_2P_3P_4,\ P_2P_3,\ P_2P_3P_4\,\}$ and the set of *minimal* subsets of $\mathcal{A}$ is, $\mathcal{A}_{min} = \{P_1P_2P_4,\ P_1P_3P_4,\ P_2P_3\}$.

Example 2.2

Consider the following monotone access structure on $\mathcal{P} = P_1P_2P_3P_4$:

$$\begin{aligned} \mathcal{A} \;=\; \{\, & P_1P_2,\ P_2P_3,\ P_3P_4,\ P_1P_4,\ P_1P_2P_3, \\ & P_1P_2P_4,\ P_1P_3P_4,\ P_2P_3P_4,\ P_1P_2P_3P_4\,\}. \end{aligned}$$

The set of *minimal* authorized subsets of $\mathcal{A}$ is given by $\mathcal{A}_{min} = \{P_1P_2,\ P_2P_3,\ P_3P_4,\ P_1P_4\}$ and the corresponding maximal unauthorized access sets are P_1P_3 and P_2P_4.

Definition 2.6

A Secret Sharing Scheme is called *ideal,* if the size of the shares is less than or equal to the size of the secret.

Definition 2.7

A Secret Sharing Scheme is called *perfect,* if, no information about the secret is obtained on pooling of shares of any unauthorized set of participants.

2.2 Evolution of the schemes

In the initial stages of work on secret sharing, Blakley [9] and Shamir [53] considered only schemes with a (k, n)-threshold access structure. Benaloh showed an interactive verifiable (k, n)-threshold secret sharing scheme which is zero knowledge [6]. In [61], D. R. Stinson and S. A. Vanstone introduced the anonymous threshold scheme. Informally, in an anonymous secret sharing scheme, the secret is reconstructed without the knowledge of, which participants hold which shares. In such schemes the computation of the secret can be carried out by giving the shares to a black box that does not know the identities of the participants holding those shares. The authors proved a lower bound on the size of the shares for anonymous threshold schemes and provided optimal schemes for certain classes of threshold structures by using a combinatorial characterization of optimal schemes. Further results can be found in [51] and in [26].

Phillips and Phillips [49] considered a different model for anonymous secret sharing schemes. In their model, different participants are allowed to receive the same shares. They proved the interesting result that a strongly ideal scheme for an access structure Γ on n participants can be realized, if and only if, Γ is either a $(1, n)$-threshold structure, a (n, n)-threshold structure, or the closure of the edge set of a complete bipartite graph. Further results on this type of anonymous secret sharing schemes can be found in [16].

Non-anonymous secret sharing schemes for graph access structures have been extensively studied in several papers, such as [18] [19] [22] [15] [14] [59] [60].

Further works considered the problem of finding secret sharing schemes for more general access structures. D. R. Stinson [58] gives a comprehensive introduction to this topic.

Secret Sharing schemes based on Chinese Remainder Theorem was introduced by Mignotte [47]. Asmuth and Bloom [1] implemented a (k, n) threshold scheme based on Chinese Remainder Theorem in 1983.

A black-box secret sharing scheme for the threshold access structure is one which works over any finite Abelian group. G. Bertilsson and I. Ingemarsson [8] describes a construction method of practical secret sharing schemes using Linear Block Codes.

A more general approach has been considered by Karnin, Greene and Hellman [39], who invented the analysis (limited to threshold scheme) of secret sharing schemes when arbitrary probability distributions are involved.

Some other general techniques handling arbitrary access structures are given by Simmons, Jackson, and Martin [45] [56] and also suggested by Kothari [41].

In [17], Brickell introduced the *vector space construction* which provides secret sharing schemes for a wide family of access structures. In [58], Stinson proved that threshold schemes are vector space access structures.

During 1987 Ito, Saito, and Nishizeki [36] described a generalized method of secret sharing scheme whereby a secret can be divided among a set $\mathcal{P}$ of trustees such that any qualified subset of $\mathcal{P}$ can reconstruct the secret and unqualified subsets cannot. They have described a secret sharing scheme, for a generalized monotone access structure.

While in threshold schemes proposed by Blakley [9] and Shamir [53] and in the vector space schemes given by Brickell [17] the shares have the same size as the secret, in the schemes constructed by M. Ito, A. Saito, and T. Nishizeki [36] for general access structures, the shares are, in general, much larger than the secret.

An important issue in the implementation of secret sharing schemes is the size of shares, since the security of a system degrades as the amount of the information that must be kept secret increases. J. C. Benaloh and J. Leichter, proved that there exists an access structure (namely the path of length three) for which any secret sharing scheme must give to some participant a share which is from a domain larger than that of the secret.

Subsequently, Benaloh and Leichter [5] gave a simpler and more efficient way to realize such schemes. They also proved that no threshold scheme is sufficient to realize secret sharing on general monotone access structures. In support of their claim, they have shown that there is no threshold scheme such that the access structure $((A \vee B) \wedge (C \vee D))$ can be achieved. [see Example 2.3.]

In [6], Benaloh describes a homomorphism property that is present in many threshold schemes which allows shares of multiple secrets to be combined to form "composite shares" which are shares of a composition of the secrets. This property, makes the entity best suitable in implementing the cases in which, one requires high confidentiality, such as e-voting. While casting the vote, each voter will take the role of dealer, and the votes casted will be recorded in terms of shares given to each contesting candidate. Because of the homomorphism property, (i.e., $h(ab) = h(a).h(b)$,) one can combine shares, and compute the votes scored by each contesting candidate.

Capocelli, De Santis, Gargano and Vaccaro [22] proved that, there exist access structures for which the best achievable information rate (i.e., the ratio between the size of the secret and that of the largest share) is bounded away from 1. An ideal secret sharing scheme is a scheme in which the size of the shares given to each participant is equal to the size of the secret. Brickell and Davenport [18] showed a correspondence between ideal secret sharing schemes and matroids (see also [38]).The uniqueness of the associated matroid is established by Martin in [44]. Beimel and Chor [4] investigate the access structures for which an ideal scheme can be constructed for every possible size of the set of secrets.

The following are some "extended capabilities" of secret sharing schemes that have been studied.

- The idea of protecting against cheating by one or more participants is addressed in [46], [62], [50], [54], [20], [23]. The problem of identifying the cheater is solved by Tompa and Woll [62]. In a sense, it is an improvement on the works of Shamir [53]. A cheater might tamper with the content of a share and make the share unusable for combining, to retrieve the secret.
- Prepositioned schemes are studied in [55].
- Threshold schemes that permit disenrollment of participants are investigated and redistributing secret shares to new access structures has been considered in [10].
- Secret sharing schemes in which the dealer has the feature of being able (after a preprocessing stage) to activate a

particular access structure out of a given set and/or to allow the participants to reconstruct different secrets (in different time instants) by sending to all participants the same broadcast message have been analyzed in [13].

- Schemes for sharing several non-independent secrets simultaneously have been analyzed in [14].
- Schemes where different secrets are associated with different subsets of participants are considered in [37].
- The question of how to set up a secret sharing scheme in the absence of a trusted party is solved in [35].

De Santis, Desmedt, Frankel, and Yung [31] introduced the notion of threshold sharing for functions and they described how to share a key to a cryptographically secure function f in such a way that:

- Any k shareholders can collectively compute f.
- Even after taking part in the computation of f on some inputs, no set of up to $k-1$ shareholders can compute f on other inputs.

B. Chor and E. Kushilevitz [27] investigated secret sharing systems on infinite domain with finite access structures.

1994, Naor and Shamir [48] described a new $(k,\ n)$ visual cryptographic scheme using black and white images, where the dealer distributes a secret into n participants. In this scheme, a shared secret information (printed text, handwritten notes, pictures, etc.) can be revealed without any cryptographic computations. For example, in a $(k,\ n)$ visual cryptography scheme, a dealer encodes a secret into n shares and gives each participant a share, where each share is a transparency. The secret is visible if any k(or more) of participants stack their transparencies together (in an arbitrary order), but none can see the shared secret if fewer than k transparencies are stacked together. It is clear that the visual secret sharing scheme needs no computation in decryption. This property distinguishes the visual secret sharing schemes from ordinary secret sharing schemes. In [3], G. Ateniese, C. Blundo, A. D. Santis, and D. R Stinson gave a construction method to extend the $(k,\ n)$ visual cryptography scheme to a general access structure which is specified by qualified sets and forbidden sets. The qualified set is a subset of n participants that can decrypt the secret image while a forbidden set is a subset of participants that can gain no information of the secret image. A more detailed discussion about visual cryptographic scheme with examples are given in the first part of chapter 3.

Until the year 1997, although the transparencies could be stacked to recover the secret image without any computation, the revealed secret images (as in [2] [3] [32] [48]) were all black and white. In [63], Verheul and Van Tilborg used the concept of *arcs* to construct a colored visual cryptography scheme, where users could share colored secret images. The key concept for a c-colorful visual cryptography scheme is to transform one pixel to b 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. The color of one pixel depends on the interrelations between the stacked sub-pixels. For example, if we want to encrypt a pixel of color c_i, we color region i with color c_i on all sub-pixels. If all sub-pixels are colored in the same way, one sees color c_i, when looking at this pixel; otherwise one sees black.

A major disadvantage of this scheme is that the number of colors and the number of sub-pixels determine the resolution of the revealed secret image. If the number of colors is large, coloring the sub-pixels will become a very difficult task, even though we can use a special *image editing package* to color these sub-pixels. How to stack these transparencies correctly and precisely by human beings is also a difficult problem. Another problem is that when the number of sub-pixels is b, the loss in resolution from the original secret image to the revealed image becomes b.

In [34], Hwang proposed a new visual cryptography scheme which improved the visual effect of the shares (the shares in their scheme were significant images, while those in the previous scheme were meaningless images). Hwang's scheme is very useful when we need to manage a lot of transparencies; nevertheless, it can only be used in black and white images. For this reason, Chang, Tsai and Chen [24] proposed a new secret color image sharing scheme based on *modified visual cryptography*.

In that scheme, through a predefined Color Index Table (CIT) and a few computations they can decode the secret image precisely. Using the concept of modified visual cryptography, the recovered secret image has the same resolution as the original secret image in their scheme. However, the number of sub-pixels in their scheme is also proportional to the number of colors appearing in the secret image; i.e., the more colors the secret image has, the larger the shares will become. Another disadvantage is that additional space is needed to store the Color Index Table (CIT). In [25], Chang proposed a scheme wherein the size of the share is fixed and independent of the number of colors appearing in the secret image. Further, the pixel expansion was only 9, which was the least amongst the previously proposed methods. But this algorithm is applicable only for $(n,\ n)$ schemes. In paper [29], Tsai gives the concept of the sharing of the multiple secrets in the digital image.

2.3 General Secret Sharing Schemes

There are situations which require more complex access structures than the threshold ones. Shamir [53] discussed the case of sharing a secret between the executives of a company such that the secret can be recovered by any three executives, or by any executive and any vice-president, or by the president alone. This is an example of the so-called hierarchical secret sharing schemes. The Shamir's solution for this case is based on an ordinary $(3,\ n)-$ threshold secret sharing scheme. Thus, the president receives three shares, each vice-president receives two shares and, finally, every simple executive receives a single share.

The above idea leads to the so-called weighted (or multiple shares based) threshold secret sharing schemes. Benaloh and Leichter have proven in [5] that, there are access structures that cannot be realized using such schemes. We present next their example that proves this.

Example 2.3

Consider the access structure $\mathcal{A}$ defined by the formula $\mathcal{A}_{min} = \{AB,\ CD\}$, and assume that a threshold scheme is to be used to divide a secret value s among A, B, C, and D such that only those subsets of A, B, C, D which are in $\mathcal{A}$ can reconstruct s.

Let a, b, c, and d respectively denote the weight (number of shares) held by each of A, B, C, and D. Since A together with B

can compute the secret, it must be the case that $a + b \geq t$ *where* t *is the value of the threshold. Similarly, since* C *and* D *can together compute the secret, it is also true that* $c + d \geq t$*. Now assume without loss of generality that* $a \geq b$ *and* $c \geq d$*. (If this is not the case, the variables can be renamed.) Since* $a + b \geq t$ *and* $a \geq b, a+a \geq a+b \geq t$*. So* $a \geq t/2$*. Similarly,* $c \geq t/2$*. Therefore,* $a+c \geq t$*. Thus,* A *together with* C *can reconstruct the secret value* s*. This violates the assumption of the access structure.*

2.4 Applications

Most of the business organizations need to protect data from disclosure. As the world is more connected by computers, the hackers, power abusers have also increased, and most organizations are afraid to store data in a computer. So there is a need of a method to distribute the data at several places and destroy the original one. When a need of original data arises, it could be reconstructed from the distributed shares. Initially, when it was introduced, its goal was to present its customers a secure information storage media. Secret Sharing can provide confidentiality of the data base. For example, e-voting can be effectively implemented by secret sharing technique. It can ensure confidentiality. It aims to achieve the two somewhat divergent goals of data secrecy and data availability. If availability were the only goal, then simple duplication of the full data among n

places would prevent the loss of data upto $n-1$ places from erasing the secret. However, this would increase the threats also. Capturing any one place could disclose the secret to an adversary. If secrecy were the only goal, then solutions might include splitting the data into n pieces and storing each piece at each of the n places. This would require all n places accessible to get the secret. However, the destruction or alteration of any one piece would erase the distributed information. It ensures secrecy in the face of adversaries and yet achieves data integrity and availability with the cooperation of its shareholders. General concept of secret sharing is that, it doesn't want information to be centralized at one point. For example, in the preparation of plastic cards, such as ATM cards, it can provide good security. Presently, a vide range of its applications have been identified.

We present next the most important general secret sharing techniques.

2.5 Ito-Saito-Nishizeki Scheme

Ito, Saito, and Nishizeki [36] have introduced the so-called cumulative array technique for monotone access structures.

Definition 2.8

Let $\mathcal{A}$ be a monotone authorized access structure of size n and let $B_1, \ldots, B_m$ be the corresponding maximal unauthorized access sets. The *cumulative array f*or the access structure $\mathcal{A}$, denoted by $\mathcal{C}^{\mathcal{A}}$, is the $n \times m$ matrix, $(\mathcal{C}^{\mathcal{A}}_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}}$, where,

$$\mathcal{C}^{\mathcal{A}}_{i,j} = \begin{cases} 0, & \textit{if } i \in B_j \\ 1, & \textit{if } i \notin B_j \end{cases}$$

for all $1 \leq i \leq n$, and $1 \leq j \leq n$.

Let us consider now an arbitrary (m, m)-threshold secret sharing scheme with the secret S and the corresponding shares $s_1, \ldots, s_m$. In the $\mathcal{A}$-secret sharing scheme, the shares $I_1, \ldots, I_n$ corresponding to the secret S will be defined as

$$I_i = \{s_j | \mathcal{C}^{\mathcal{A}}_{i,j} = 1\},$$

for all $1 \leq i \leq n$.

Example 2.4

Let $n = 4$ and $\mathcal{A}_{min} = \{\{1,2\},\{3,4\}\}$. In this case, we obtain that $\overline{\mathcal{A}}_{max} = \{\{1,3\},\{1,4\},\{2,3\},\{2,4\}\}$ and $m = 4$.

The cumulative array for the access structure $\mathcal{A}$ is,

$$\mathcal{C}^{\mathcal{A}} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

In this case, $I_1 = \{s_3, s_4\}$, $I_2 = \{s_1, s_2\}$, $I_3 = \{s_2, s_4\}$ and $I_4 = \{s_1, s_3\}$, where s_1, s_2, s_3, s_4 are the shares of a (4, 4)-threshold secret sharing scheme with the secret S.

2.6 Benaloh-Leichter Scheme

Benaloh and Leichter [5] have represented the access structures using formulae. More exactly, for a monotone authorized access structure $\mathcal{A}$ of size n, they defined the set $\mathcal{F}_{\mathcal{A}}$ as the set of formulae on a set of variables $\{v_1, v_2, \ldots, v_n\}$ such that for every $\mathcal{F} \in \mathcal{F}_{\mathcal{A}}$, the interpretation of $\mathcal{F}$ with respect to an assignation of the variables is true if and only if the true variables correspond to a set $A \in \mathcal{A}$. They have remarked that such formulae can be used as templates for describing how a secret can be shared with respect to the given access structure. Because the formulae can be expressed using only $\wedge$ operators and $\vee$ operators, it is sufficient to indicate how to "split" the secret across these operators.

Thus, we can inductively define the shares of a secret S with respect to a formulae $\mathcal{F}$ as follows:

$$Shares(S,F) = \begin{cases} (S,i), & \text{if } F = v_i,\ 1 \le i \le n; \\ \bigcup_{i=1}^{k} Shares(S,F_i), & \text{if } F = F_1 \vee \cdots \vee F_k; \\ \bigcup_{i=1}^{k} Shares(s_i,F_i), & \text{if } F = F_1 \wedge \cdots \wedge F_k, \end{cases}$$

where, for the case $F = F_1 \wedge F_2 \wedge \cdots \wedge F_k$, we can use any (k,k)-threshold secret sharing scheme for deriving some shares $s_1, \ldots, s_k$ corresponding to the secret S and, finally, the shares as $I_i = \{s|(s,i) \in Shares(S,F)\}$, for all $1 \le i \le n$, where, F is an arbitrary formula in the set $\mathcal{F}_{\mathcal{A}}$.

Example 2.5

Let $n = 3$ and an authorized access structure $\mathcal{A}$ given by

$\mathcal{A}_{min} = \{\{1,2\},\{2,3\}\}$. For example, the formula $F = (v_1 \wedge v_2) \vee (v_2 \wedge v_3)$ is in the set $\mathcal{F}_{\mathcal{A}}$. In this case, Shares($S, F$), for some secret S, can be obtained as

$$\begin{aligned} Shares(S,F) &= Shares(S, v_1 \wedge v_2) \cup Shares(S, v_2 \wedge v_3) \\ &= Shares(s_1, v_1) \cup Shares(s_{2,1}, v_2) \cup \\ &\quad Shares(s_{2,2}, v_2) \cup Shares(s_3, v_3) \\ &= \{(s_1, 1),\ (s_{2,1}, 2),\ (s_{2,2}, 2),\ (s_3, 3)\}, \end{aligned}$$

where, $s_1, s_{2,1}$ and respectively, $s_{2,2}, s_3$ are shares of the secret S with respect to two arbitrary (2, 2)-threshold secret schemes. Thus, the shares corresponding to the secret S with respect to the access structure $\mathcal{A}$ are

$$I_1 = \{s_1\},\ I_2 = \{s_{2,1}, s_{2,2}\} \text{ and } I_3 = \{s_3\}.$$

Example 2.6

Consider the access structure $\Gamma_{min} = \{P_1P_2P_3,\ P_1P_4\}$.
Let the secret $s \in GF(2^r)$.

A secret sharing scheme for Γ_{min}can be realized in the following way:
Randomly choose $x,\ y \in GF(2^r)$.
Compute z such that $s = (x + y + z) \pmod{2^r}$.
Let $a_1 = x;\ a_2 = y;\ a_3 = z$ and $a_4 = y + z \pmod{2^r}$.

Example 2.7

Consider the access structure $\Gamma_{min} = \{P_1P_2P_3,\ P_1P_2P_4\}$.
Let $s \in GF(2^r)$.

A secret sharing scheme for Γ_{min} can be realized in the following way:
Randomly choose $x, y \in GF(2^r)$.
Compute z such that $s = (x + y + z) \pmod{2^r}$.
Let $a_1 = x; a_2 = y; a_3 = z$ and $a_4 = z$.

Example 2.8

Consider the access structure $\Gamma_{min} = \{P_1P_2P_4,\ P_1P_3P_4,\ P_2P_3\}$.
Let $s \in GF(2^r)$.

A secret sharing scheme for Γ_{min} can be realized in the following way:
Randomly choose $x, y \in GF(2^r)$.
Let $a_1 = x; a_2 = s + y; a_3 = s - y$ and $a_4 = y - x$.

Remark 2.1

A share I_i may contain many sub-shares, one sub-share for every minimal access set to which i belongs. Thus, an ordering of these sub-shares is required in order to select the correct sub-share corresponding to a certain access set in the reconstruction phase.

Remark 2.2

They also proposed using general $threshold_{k,m}$[1] *operators in order*

---

[1]For $m \geq 1,\ 1 \leq k \leq m, threshold_{k,m}$ denotes the formula

$$\bigvee_{1 \leq i_1 < i_2 < \ldots < i_k \leq i_k} \left(\bigwedge_{j=1}^{k} F_{i_j} \right).$$

Thus, $F_1 \vee F_2 \vee \ldots F_m = threshold_{1,m}(F_1, \ldots, F_m)$ and
$F_1 \wedge F_2 \wedge \ldots F_m = threshold_{m,m}(F_1, \ldots, F_m)$.

to construct smaller formulae, reducing in this way the size of the shares. In this case, the definition of $Shares(S, F)$ *can be extended for these operators as follows:*

$$Shares(S,F) = \cup_{i=1}^{m} Shares(s_i, F_i),$$

if $F = threshold_{k,m}(F_1, \ldots, F_m)$*, where* $s_1, \ldots, s_m$ *are the shares corresponding to the secret* S *with respect to an arbitrary* (k, m)*-threshold secret sharing scheme.*

Example 2.9

Let $n = 4$ *and a monotone authorized access structure* $\mathcal{A}$ *given by* $\mathcal{A}_{min} = \{\{2,3\},\{1,2,4\},\{1,3,4\}\}$*. For example, the formula* $F = (v_2 \wedge v_3) \vee (v_1 \wedge v_2 \wedge v_4) \vee (v_1 \wedge v_3 \wedge v_4)$ *is in the set* $\mathcal{F}_{\mathcal{A}}$*. Using the threshold operator, we can obtain a shorter formula, namely,* $(v_2 \wedge v_3) \vee threshold_{3,4}(v_1, v_2, v_3, v_4)$*.*

Example 2.10

Consider the access structure $\Gamma_{min} = \{P_1P_3P_4,\ P_1P_2,\ P_2P_3\}$*.*
Let $s \in GF(2^r)$*.*

A secret sharing scheme for Γ_{min} can be realized in the following way: Construct a (3,4) threshold scheme for the secret s and let $y_1, \ldots, y_4$ be the shares of this threshold scheme.
Let $a_1 = y_1;\ a_2 = y_2,\ y_4;\ a_3 = y_3$ and $a_4 = y_4$.

Example 2.11

Consider the access structure $\Gamma_{min} = \{P_1P_3P_4,\ P_1P_2,\ P_2P_3, P_2P_4\}$*.*
Let $s \in GF(2^r)$*.*

A secret sharing scheme for Γ_{min} can be realized in the following way:
Construct a (3, 5) threshold scheme for the secret s and let $y_1, \ldots, y_5$ be the shares of this threshold scheme.
Let $a_1 = y_1$; $a_2 = y_2, y_5$; $a_3 = y_3$ and $a_4 = y_4$.

Example 2.12
Consider the access structure $\Gamma_{min} = \{P_1P_2P_3,\ P_1P_2P_4,\ P_1P_3P_4\}$. *Let* $s \in GF(2^r)$.

A secret sharing scheme for Γ_{min} can be realized in the following way:
Randomly choose $x \in GF(2^r)$. Compute y such that $s = (x + y) \pmod{2^r}$. Construct a (2, 3) threshold scheme for the secret y and let $y_1, y_2 and y_3$ be the shares of this threshold scheme.
Let $a_1 = x; a_2 = y_1; a_3 = y_2$ and $a_4 = y_3$.

Example 2.13
Consider the access structure given by $\Gamma_{min} = \{P_1P_2, P_2P_3, P_3P_4, P_4P_5, P_5P_6, P_6P_7, P_7P_8, P_8P_1\}$. *Let* $s \in \{0, 1\}$.

Let the four distinct numbers $a, b, c, d \in B = \{0, 1, 2, 3\}$. Let $\mathcal{C}_0$ consists of all the 24 column matrices: $[\,\mathrm{a\,a\,b\,b\,c\,c\,d\,d}\,]$ and let $\mathcal{C}_1$ consists of all the 24 column matrices: $[\,\mathrm{a\,b\,b\,c\,c\,d\,d\,a}\,]$.
To share $s = 0$, the dealer randomly chooses one of the matrices in C_0, and to share $s = 1$, the dealer randomly chooses one of the matrices in C_1. The rows of chosen matrix defines shares given

to each one of the 8 participants.
Let $A = \{P_1P_2, P_3P_4, P_5P_6, P_7P_8\}$, and $B = \{P_2P_3, P_4P_5, P_6P_7, P_8P_1\}$. In this example, at the reconstruction stage, if $P_iP_j \in A$ and the value of the shares of P_i and that of P_j are equal or if $P_iP_j \in B$, and the value of the shares of P_i and that of P_j are not equal, the secret $s = 0$; otherwise secret $s = 1$.

2.7 Concluding remarks

In this chapter, the different research findings were analyzed and the efficiency aa well as the level of difficulty were brought out. Also discussed were, various examples to illustrate the secret sharing schemes in general.

Chapter 3

Visual Cryptography

3.1 Introduction

1994, Naor and Shamir [48] described a new $(k,\ n)$ visual cryptographic scheme using black and white images, where the dealer encodes a secret into n participants. In this scheme, a shared secret information (printed text, handwritten notes, pictures, etc.) can be revealed without any cryptographic computations. For example, in a $(k,\ n)$ visual cryptography scheme, a dealer encodes a secret into n shares and gives each participant a share, where each share is a transparency. The secret is visible if any k(or more) of participants stack their transparencies together, but none can see the shared secret if fewer than k transparencies are stacked together. By identifying that the result of stacking the transparencies are the same as Boolean-OR operation denoted by $\vee$ on the binary digits involved, it is possible to extend the Visual Cryptography schemes to any binary string. For example, the following scheme describes how one could implement Visual cryptography scheme for a single binary digit. In order to share a binary string, each binary digit in it could be shared independently, one after the other using the same scheme.

Example 3.1

Let the secret, $s, \in \{0, 1\}$. *The* $(2, 7)-$ *visual secret sharing problem can be solved as follows:*

$$\text{Let } A = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Let $\mathcal{C}_0$ *be the set of all the matrices obtained by permuting the columns of* A, *and* $\mathcal{C}_1$ *be the set of all the matrices obtained by permuting the columns of* B

To share a bit, $s = 0$ *or 1, the dealer randomly chooses one of the matrix* $\in \mathcal{C}_s$*. Each rows of chosen matrix defines shares to be given to each one of the 7 participants.*

A single share in either $\mathcal{C}_0$ *or* $\mathcal{C}_1$ *is a random choice of three 1s and four 0s, and so they are equally likely. So by having only one share, one cannot identify whether it is from* $\mathcal{C}_0$ *or from* $\mathcal{C}_1$*. On the other hand, if we combine (i.e., "OR") any two shares, we get a binary string of length 7, consists of all 0s, or four 1s and three 0s depending on whether the shares belong to* $\mathcal{C}_0$ *or* $\mathcal{C}_1$*. In this scheme, the size of one share is 7 bits. So a bit is expanded to 7 times.*

Since each binary digit in the secret is shared by choosing a matrix independently, there is no information to be gained by looking at any group of binary digits on a share, either. This demonstrates the security of the scheme.

Remark 3.1

For implementing the visual cryptographic scheme as above, one does not have to generate the entire collection of matrices such as $\mathcal{C}_0$ *and* $\mathcal{C}_1$*. One could simply generate two matrices* A *and* B *and store them. During the process of sharing individual bits, depending on the value of* s*, choose the matrix* A *or* B*, generate a random permutation,* μ*, of* $\{1,2,\ldots,m\}$*, where,* m *is the number of columns in it; and permute the rows of the chosen matrix with respect to* μ*. The rows of the resulting matrices may be regarded as shares, and be distributed to the various participants.*

3.2 Division of the pixel

In this section, we shall review the basic visual cryptography scheme proposed by Naor and Shamir. Here a secret black and white image is divided into two grey images. In order to share a secret black and white image, the concept of their scheme is to transform one pixel into two sub-pixels and divide each sub-pixel two color regions. The sub-pixels are half white and half black (can be called grey).

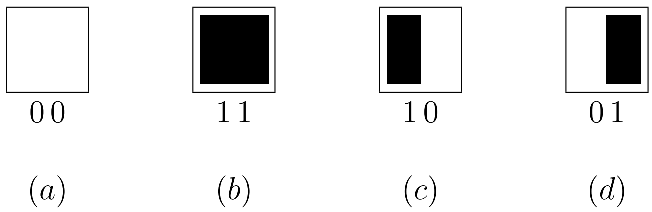

Figure 3.1: Different types of pixels along with the representation.
(a) White pixel (b) Black pixel
(c) LB pixel (d) RB pixel

For example, Figure 3.1 represents four different type of pixels. The first is a white pixel, the next is a black pixel, and the last two are grey pixels. Note that in the grey pixels, the black and white portions are different. Let us call these pixels as LB and RB pixels respectively. We represent a white pixel by 00, black by 11, LB-pixel by 10 and RB-pixel by 01. They can be thought of as modified version of pixels to be used in shares.

3.3 Superposition of pixels

If we stack two LB pixels (or two RB pixels) we get nothing new, where as, if we stack an LB pixel and an RB pixel, we get a black pixel. This can be shown as in Figure 3.2. We can see that by the representation used for pixels, the superposition of two pixels can be thought of as if a binary "OR" operation.

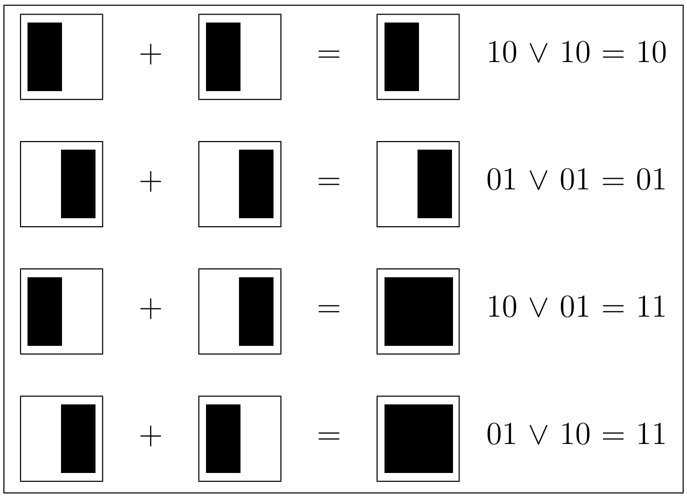

Figure 3.2: Superposition of two grey pixels.

3.4 Dealing of a B/W Image

3.4.1 Algorithm to share a pixel into two shares

The following algorithm specifies how to encode a single pixel into two shares:

Algorithm 3.1 (Share a single pixel into two shares)
Input: A pixel P, which is either Black or White
Output : Two sub-pixels s_1 and s_2.

Step 1. *Let $x \in \{H, T\}$ be the outcome of a coin toss*
if (P = white)
if (x = H) $r = 1$
else $r = 2$
else if (x = H) $r = 3$
else $r = 4$

Step 2. *Then the pixel P is encrypted as two sub-pixels in each of the two shares, as determined by the r^{th} row in the figure 3.3.*

Naor and Shamir devised the following scheme, illustrated in Figure 3.3 below.

Every pixel is encrypted using algorithm 3.1. Suppose we look at a pixel P in the first share. One of the two sub-pixels in P is black and the other is white. Moreover, each of the two possibilities "black-white" and "white-black" is equally likely to occur, independent of whether the corresponding pixel in the secret image is black or white. Thus the first share gives no clue as to whether the pixel is black or white. The same argument applies to the second share. Since all the pixels in the secret image were encrypted using independent random coin flips, there

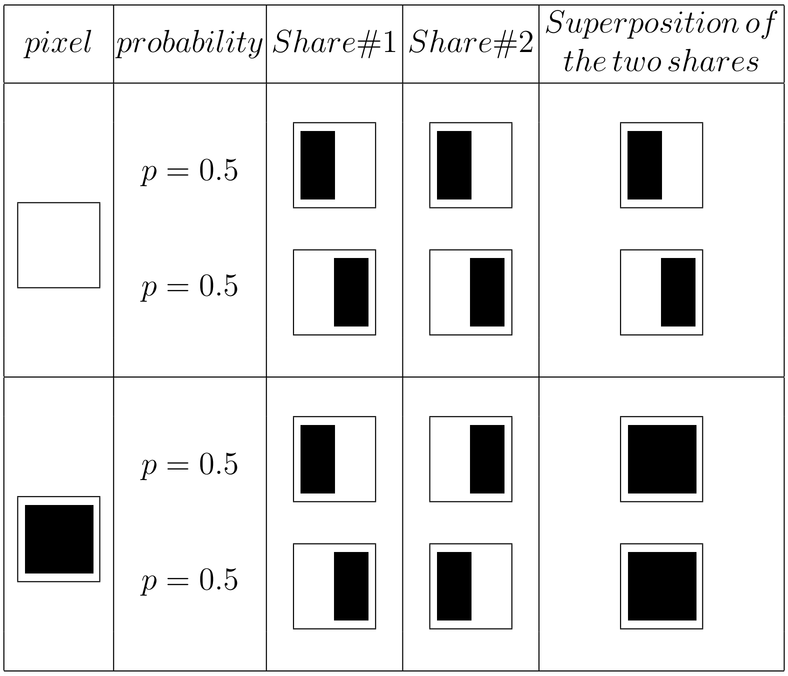

Figure 3.3: Superposition of two grey pixels.

is no information to be gained by looking at any group of pixels on a share, either. This demonstrates the security of the scheme.

Now let us consider what happens when we superimpose the two shares (here we refer to the last column of the figure 3.3. Consider one pixel P in the image. If P is black, we get two black sub-pixels when we superimpose the two shares; if P is white, we get one black sub-pixel and one white sub-pixel when we superimpose the two shares. Thus, we could say that the reconstructed pixel (consisting of two sub-pixels) has a grey level of 2, if P is black, and a grey level of 1, if P is white. There will be a 50% loss of contrast in the reconstructed image, but it should still be visible. In this case, each pixel is divided into two sub-pixels.

Definition 3.1
The ratio of the size of the share to the size of the secret is called the *blowing factor.*

Since the result of stacking of pixels can be completely determined by the binary "OR" operation, the visual cryptography scheme could also be implemented to any binary strings of 0s and 1s. This method could be extended to any number of participants. When more number of participants are involved, the pixels should be divided into more parts. For example, Noar and Shamir [48] described how to solve the $(2, n)$ visual secret sharing. We present next their solution.

3.4.2 Shamir's solutions for small k and n

$$\textit{Let } A = \begin{bmatrix} 1 & 0 & 0 \cdots 0 \\ 1 & 0 & 0 \cdots 0 \\ \cdots\cdots & & \\ 1 & 0 & 0 \cdots 0 \end{bmatrix} \textit{ and } B = \begin{bmatrix} 1 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 \cdots 0 \\ \cdots\cdots & & \\ 0 & 0 & 0 \cdots 1 \end{bmatrix}$$

The $(2, n)$ visual secret sharing problem can be solved by the following collections of $n \times n$ matrices:
$\mathcal{C}_0 = \{$all the matrices obtained by permuting the columns of $A\}$

and $\mathcal{C}_1 = \{$all the matrices obtained by permuting the columns of $B\}$

Any single share in either $\mathcal{C}_0$ or $\mathcal{C}_1$ is a random choice of one black and $n-1$ white sub-pixels. To share a pixel $P \in \{0,1\}$, randomly choose one of the matrix from $\mathcal{C}_P$. Then the pixel P is shared with the n participants, by giving each row of the chosen matrix to each participant. If we superimpose any two shares of a white pixel, will have one black and $n-1$ white sub-pixels, whereas any two shares of a black pixel, will have two black and $n-2$ white sub-pixels, which looks darker. So the shared secret bit is recovered. The visual difference between the two cases becomes clearer as we stack additional transparencies.

The blowing factor of this $(2,n)$ scheme is n. That is, the size of a share is n times larger than the size of the secret. It can be shown that the blowing factor can be made smaller. In example 3.2, we present a (2, 9) visual secret sharing, in which, the blowing factor is 6. In Chapter 5, we present a better scheme to achieve the same, in which the blowing factor is of $O(log_2 n)$.

Example 3.2

$$Let\ A = \begin{bmatrix} 1&1&0&0&0&0 \\ 1&1&0&0&0&0 \\ 1&1&0&0&0&0 \\ 1&1&0&0&0&0 \\ 1&1&0&0&0&0 \\ 1&1&0&0&0&0 \\ 1&1&0&0&0&0 \\ 1&1&0&0&0&0 \\ 1&1&0&0&0&0 \end{bmatrix}\ and\ B = \begin{bmatrix} 1&1&0&0&0&0 \\ 0&0&1&1&0&0 \\ 0&0&0&0&1&1 \\ 0&0&1&0&1&0 \\ 1&0&0&0&0&1 \\ 0&1&0&1&0&0 \\ 0&0&0&1&0&1 \\ 0&1&0&0&1&0 \\ 1&0&1&0&0&0 \end{bmatrix}$$

Let $\mathcal{C}_0$ be the set of all the matrices obtained by permuting the columns of A
and $\mathcal{C}_1$ be the set of all the matrices obtained by permuting the columns of B
In this example, one bit is expanded to six bits.

3.5 A general scheme for (k, k) Visual cryptography

We now describe a general construction which can solve any (k, k) visual secret sharing problem, having a blowing factor 2^{k-1}.

Let e_i be a column vector consisting of i 1s and $k - i$ 0s. The length of e_i is k, and so there are $\binom{k}{i}$ such vectors.
Let B_i be the exhaustive collection of all e_i's. B_i can be thought of as a matrix of order $k \times \binom{k}{i}$.

$$Let\ \ R = B_i^{(1)} \vee B_i^{(2)} \vee B_i^{(3)} \vee \ldots \vee B_i^{(r)},$$

where, $B_i^{(1)}, B_i^{(2)}, B_i^{(3)}, \ldots B_i^{(r)}$, are any r distinct rows from B_i. Let $n_0(R)$ and $n_1(R)$ denote the number of 0s and 1s, respectively, in R.

Consider a particular bit in R. It can be 0, if and only if, all the selected $B_i^{(j)}$'s have the corresponding bit 0. In other words, since any column contains exactly i 1s, the unselected $k-r$ rows collectively must have all the i 1s in the respective column. Hence $n_0(R) = \begin{pmatrix} k-r \\ i \end{pmatrix}$. Since the length of $R = \begin{pmatrix} k \\ i \end{pmatrix}$, the number of 1s in R is given by the following formula:

$$n_1(R) = \begin{pmatrix} k \\ i \end{pmatrix} - \begin{pmatrix} k-r \\ i \end{pmatrix}. \tag{3.1}$$

Lemma 3.1

Let k be a non negative integer. Then, if $k \neq 0$,

$$\sum_{\substack{i=0, \\ i \, is \, even}}^{k} \begin{pmatrix} k \\ i \end{pmatrix} = \sum_{\substack{i=0, \\ i \, is \, odd}}^{k} \begin{pmatrix} k \\ i \end{pmatrix} = 2^{k-1}, \tag{3.2}$$

and if $k = 0$,

$$\sum_{\substack{i=0, \\ i \, is \, even}}^{k} \begin{pmatrix} k \\ i \end{pmatrix} = 1, \quad and \quad \sum_{\substack{i=0, \\ i \, is \, odd}}^{k} \begin{pmatrix} k \\ i \end{pmatrix} = 0. \tag{3.3}$$

Proof: The case when $n = 0$, can be verified. So, consider the case when $n \neq 0$. From the equation

$$\sum_{i=0}^{k} (-1)^i . \begin{pmatrix} k \\ i \end{pmatrix} = (1-1)^k = 0 \tag{3.4}$$

separating the negative and nonnegative terms, we get first part of equation (3.2). Also we have,

$$2^k = (1+1)^k = \sum_{i=0}^{k} \binom{k}{i} . \tag{3.5}$$

So,

$$\sum_{\substack{i=0,\\ i\, is\, even}}^{k} \binom{k}{i} = \sum_{\substack{i=0,\\ i\, is\, odd}}^{k} \binom{k}{i} = 2^{k-1} \tag{3.6}$$

Let X denote the matrix obtained by concatenating B_i for all nonnegative even integer $i \leq k$, and let Y be the matrix obtained by concatenating B_i for all nonnegative odd integer $i \leq k$.

Now, the number of columns in the matrix X and that of Y are

$$\sum_{\substack{i=0,\\ i\, is\, even}}^{k} \binom{k}{i} , \quad and \quad \sum_{\substack{i=0,\\ i\, is\, odd}}^{k} \binom{k}{i} ,$$

respectively, and by lemma 3.1, both equal to 2^{k-1}.

So, both X and Y are the same order, $k \times 2^{k-1}$.

$$Let \;\; W = X^{(1)} \vee X^{(2)} \vee X^{(3)} \vee \ldots \vee X^{(r)}, \tag{3.7}$$

where, $X^{(1)}, X^{(2)}, X^{(3)}, \ldots X^{(r)}$, are any r distinct rows from X.

Then, by equation (3.1),

$$\begin{aligned}
n_1(W) &= \sum_{i \text{ is even}} \left\{ \binom{k}{i} - \binom{k-r}{i} \right\} \\
&= \sum_{i \text{ is even}} \binom{k}{i} - \sum_{i \text{ is even}} \binom{k-r}{i} \\
&= \begin{cases} 2^{k-1} - 2^{k-r-1}, & \text{if } r \neq k \\ 2^{k-1} - 1, & \text{if } r = k \end{cases} \\
&= \begin{cases} 2^{k-r-1}.(2^r - 1), & \text{if } r \neq k \\ 2^{k-1} - 1, & \text{if } r = k \end{cases} \qquad (3.8)
\end{aligned}$$

Similarly, if we take r distinct rows from Y, say, $Y^{(1)}, Y^{(2)}, Y^{(3)}, \ldots, Y^{(r)}$, and if we compute

$$Z = Y^{(1)} \vee Y^{(2)} \vee Y^{(3)} \vee \ldots \vee Y^{(r)}, \qquad (3.9)$$

then, the number of 1s in Z is given by,

$$\begin{aligned}
n_1(Z) &= \sum_{i \text{ is odd}} \left\{ \binom{k}{i} - \binom{k-r}{i} \right\} \\
&= \sum_{i \text{ is odd}} \binom{k}{i} - \sum_{i \text{ is odd}} \binom{k-r}{i} \\
&= \begin{cases} 2^{k-1} - 2^{k-r-1}, & \text{if } r \neq k \\ 2^{k-1}, & \text{if } r = k \end{cases} \\
&= \begin{cases} 2^{k-r-1}.(2^r - 1), & \text{if } r \neq k \\ 2^{k-1}, & \text{if } r = k \end{cases} \qquad (3.10)
\end{aligned}$$

Let $\mathcal{C}_0$ be the set of all the matrices obtained by permuting the columns of X Let $\mathcal{C}_1$ be the set of all the matrices obtained by permuting the columns of Y

Equation (3.8) and equation (3.10) tells that any $r(< k)$ shares of a secret bit from either $\mathcal{C}_0$ or $\mathcal{C}_1$ together has a random collection of $2^{k-r-1}.(2^r - 1)$ 1s. Consequently, the analysis of any $r(< k)$ shares makes it impossible to distinguish between $\mathcal{C}_0$ and $\mathcal{C}_1$. On the other hand, k shares from $\mathcal{C}_0$ results in a collection of single 0 along with $2^{k-1} - 1$ 1s, where as k shares from $\mathcal{C}_1$ results in a collection of all 1s(no 0s).

Example 3.3

Let $n = 4$. Consider the matrices X and Y obtained by concatenating $\{B_0, B_2, B_4\}$ and $\{B_1, B_3\}$ respectively.

$$So, \quad X = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$and \;\; Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be the set of all the matrices obtained by permuting the columns of X and Y respectively.
Any single row from $\mathcal{C}_0$ or $\mathcal{C}_1$, contains four 1s, any combined ($\vee$) pair of rows contains six 1s, any combined triplet of rows contains seven 1s, and any combined quadruple of rows contains seven or eight 1s depending on whether the rows were taken from $\mathcal{C}_0$ or $\mathcal{C}_1$.

In [48] Naor and Shamir also describes, how to extend a $(k,\ k)$ scheme to $(k,\ n)$ scheme for arbitrary $n > k$.

Various schemes have been discovered. But a generalized scheme is not invented so far.

3.6 Concluding remarks

In this chapter, we have seen how the Visual Cryptography schemes are distinguished from traditional secret sharing schemes. We have also seen some examples, to illustrate the benefits of Visual Cryptography.

Chapter 4

Modified Visual Cryptography

4.1 Introduction

We have seen that in the case of visual cryptography schemes, the result of stacking of transparencies, can be completely characterized by the boolean "OR" operation. We know that it favours 1s to 0s. i.e., If we "OR" two random bits, the result is more likely towards 1 than 0. When more random bits are involved, it will be more and more likely that the result is 1. So, when k increases, the distinguishing threshold for 0 bit and 1 bit will be at a higher level. So, it is natural that as k increases, the blowing factor also increases. This threshold will not effect the security of the system. Its purpose is only to distinguish the two bits from one another. So, if one could reduce the distinguishing threshold, the blowing factor may decrease. Since "XOR" does not favour either 0 or 1, it could be a better choice to "OR". This is the difference between traditional Visual Cryptography and Modified Visual Cryptography. This cannot be implemented in the case of images, where as for binary strings it can be done. It is easy to see that, in modified visual cryptography, the blowing factor will never increase, (if not decreased) compared with ordinary visual cryptography.

4.2 A Modified scheme for (k, k) Visual Cryptography

We now describe a general construction which can solve any $(k,\ k)$ modified visual secret sharing problem, having a blowing factor, one. Let B_i, X, and Y be the matrices defined in section 3.5. In Modified Visual Cryptography we perform $\oplus$ instead of $\vee$. So, let

$$R = B_i^{(1)} \oplus B_i^{(2)} \oplus B_i^{(3)} \oplus \ldots \oplus B_i^{(r)},$$

where, $B_i^{(1)}, B_i^{(2)}, B_i^{(3)}, \ldots B_i^{(r)}$, are any r distinct rows from B_i. We claim that,

$$n_1(R) = \sum_{\substack{j \\ j\ is\ odd}} \binom{r}{j} \binom{k-r}{i-j} \qquad (4.1)$$

Consider a particular bit in R. It can be 1, if and only if, there are an odd number of $B_i^{(j)}$'s having the corresponding bit 1.

Since any column contains exactly i 1s, the unselected $k-r$ rows collectively must have the remaining $(i-j)$ 1s. Since the rows are independent, this is possible in

$$\sum_{\substack{j=1 \\ j\ is\ odd}}^{r} \binom{r}{j} \binom{k-r}{i-j}$$

many places. Here, the range of j can be unrestricted, because $\binom{p}{q} = 0$, if $p < q$.

So, equation (4.1) is established.

$$Let\ \ W = X^{(1)} \oplus X^{(2)} \oplus X^{(3)} \oplus \ldots \oplus X^{(r)}, \tag{4.2}$$

where, $X^{(1)}, X^{(2)}, X^{(3)}, \ldots X^{(r)}$, are any r distinct rows from X. Then, by equation (4.1),

$$n_1(W) = \sum_{\substack{i \\ i\ is\ even}} \sum_{\substack{j \\ j\ is\ odd}} \binom{r}{j} \cdot \binom{k-r}{i-j} \tag{4.3}$$

Because the right side of this equation evaluates to a finite number, we can interchange the summation, and get,

$$n_1(W) = \sum_{\substack{j \\ j\ is\ odd}} \sum_{\substack{i \\ i\ is\ even}} \binom{r}{j} \cdot \binom{k-r}{i-j} \tag{4.4}$$

The inner $\sum$ runs on variable i, and so, $\binom{r}{j}$ is constant. So we get,

$$n_1(W) = \sum_{\substack{j \\ j\ is\ odd}} \left[\binom{r}{j} \cdot \sum_{\substack{i \\ i\ is\ even}} \binom{k-r}{i-j} \right] \tag{4.5}$$

Since i is even and j is odd, $i-j$ is odd, and so by a change of variable,

$$\begin{aligned}\sum_{\substack{i \\ i\ is\ even}} \binom{k-r}{i-j} &= \sum_{\substack{i \\ i\ is\ odd}} \binom{k-r}{i} \\ &= \begin{cases} 2^{k-r-1}, & \text{if } r \neq k \\ 0, & \text{if } r = k \end{cases} \qquad (4.6) \\ &\quad [by\ \ lemma\ 3.1,\end{aligned}$$

So,

$$n_1(W) = \begin{cases} 2^{k-r-1} \sum_{\substack{j \\ j\ is\ odd}} \binom{r}{j}, & \text{if } r \neq k \\ 0, & \text{if } r = k \end{cases} \qquad (4.7)$$

Again by lemma 3.1, being $r \neq 0, \sum_{\substack{j \\ j\ is\ odd}} \binom{r}{j} = 2^{r-1}$.
So, equation (4.7) becomes,

$$n_1(W) = \begin{cases} 2^{k-2}, & \text{if } r \neq k \\ 0, & \text{if } r = k \end{cases} \qquad (4.8)$$

Similarly, if we take r distinct rows from Y, say, $Y^{(1)}, Y^{(2)}, Y^{(3)}, \ldots, Y^{(r)}$, and if we compute

$$Z = Y^{(1)} \oplus Y^{(2)} \oplus Y^{(3)} \oplus \ldots \oplus Y^{(r)}, \qquad (4.9)$$

then, the number of 1s in Z is given by,

$$\begin{aligned} n_1(Z) &= \sum_{\substack{i \\ i\ is\ odd}} \sum_{\substack{j \\ j\ is\ odd}} \binom{r}{j} \cdot \binom{k-r}{i-j} \\ &= \sum_{\substack{j \\ j\ is\ odd}} \sum_{\substack{i \\ i\ is\ odd}} \binom{r}{j} \cdot \binom{k-r}{i-j} \end{aligned}$$

$$= \sum_{\substack{j \\ j \text{ is odd}}} \left[\binom{r}{j} \sum_{i \text{ is odd}} \cdot \binom{k-r}{i-j} \right] \tag{4.10}$$

Since both i and j are odd, $i - j$ is even, and so by a change of variable,

$$\begin{aligned} \sum_{\substack{i \\ i \text{ is odd}}} \binom{k-r}{i-j} &= \sum_{\substack{i \\ i \text{ is even}}} \binom{k-r}{i} \\ &= \begin{cases} 2^{k-r-1}, & \text{if } r \neq k \\ 1, & \text{if } r = k \end{cases} \qquad (4.11) \\ & \quad [by \;\; lemma \; 3.1, \end{aligned}$$

So, equation (4.10) becomes,

$$\begin{aligned} n_1(Z) &= \begin{cases} 2^{k-r-1} \sum_{\substack{j \\ j \text{ is odd}}} \binom{r}{j}, & \text{if } r \neq k \\ \sum_{j \text{ is odd}} \binom{r}{j}, & \text{if } r = k \end{cases} \\ &= \begin{cases} 2^{k-r-1}.2^{r-1} = 2^{k-2}, & \text{if } r \neq k \\ 2^{k-1}, & \text{if } r = k \end{cases} \qquad (4.12) \end{aligned}$$

Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be the set of all the matrices obtained by permuting the columns of X and Y, respectively.

Equation (4.8) and equation (4.12) tells that any $r(< k)$ shares of a secret bit from either $\mathcal{C}_0$ or $\mathcal{C}_1$ together has a random collection of 2^{k-2} 1s and 0s. Consequently, the analysis of $r(< k)$ shares makes it impossible to distinguish between $\mathcal{C}_0$ and $\mathcal{C}_1$. On the other hand, k shares from $\mathcal{C}_0$ results in a collection of only 0s, where as k shares from $\mathcal{C}_1$ results in a collection of only 1s.

4.2.1 Comparison of the schemes

While both the schemes are equally secure, in the former scheme, the result of combining $r(< k)$ shares (i.e., the number of 1s = $2^{k-r-1}.(2^r - 1)$,) varies on r, where as in latter one, it is a fixed value (i.e., 2^{k-2}). This phenomena does not enhance or reduce the security of the system. So, we suspect that the former scheme, has done some extra effort for unnecessarily distinguishing the number of shares combined, which is insignificant. So we strongly believe that the blowing factor could be reduced, by striking at a better modified visual cryptography scheme, than the corresponding one. When the secret is recovered by combining all the k shares, in the former, we have to search for the single 0 present, in case, the secret bit is 0. Where as in the latter one, because the result is either all zeros or all 1s, one can recover the secret bit just by looking at the first bit itself. So, though both are equally secure, the modified cryptographic scheme is at least more efficient in the combining process.

4.3 A simple Modified scheme for (k, k)

The following is a very simple algorithm to share a binary string in a (k, k) Modified Visual Cryptography scheme:

Algorithm 4.1 ((k, k) Modified Visual Cryptography construction)

Input: A secret binary bit $S \in \{0, 1\}$
Output : k bits $s_1, s_2, \ldots, s_k$

Step 1. *let* $y = 0$
  For $i = 1$ *to* $k - 1$ *do*
    Generate a random bit, say $x, \in \{0, 1\}$
    $s_i = x$
    $y = y \oplus x$
Step 2. $s_k = y \oplus S$
Step 3. *The shares are* $s_1, s_2, \ldots, s_k$

The algorithm 4.1 computes k shares of a single binary digit S. In Step 1, after setting a variable y is 0, it computes $k - 1$ shares, $s_i, 1 \leq i \leq k - 1$, which are nothing but random bits. Also note that, when the for loop in step 1 terminates, the value of y is $s_1 \oplus s_2 \oplus \ldots \oplus s_{k-1}$. In step 2., the last share, s_k is computed as, $s_k = y \oplus S = s_1 \oplus s_2 \oplus \ldots \oplus s_{k-1} \oplus S$. This implies that, $S = s_1 \oplus s_2 \oplus \ldots \oplus s_k$. All the $k - 1$ shares being random, and the secret S being unknown, s_k will also be random. So, there is no information to be gained by looking at r number of shares, for $r < k$. Each and every bit of the secret could be shared one after the other using the same algorithm. Since every bit is shared using random bits, looking at consecutive shares also gains no information. This proves the security of the scheme. The blowing factor of the scheme is 1.

4.4 Generalization of (3, 3) scheme

The following scheme generalizes the (3, 3) scheme described in the last chapter into a (3, n) scheme for an arbitrary $n > 3$. Let B be the black $n \times (n-2)$ matrix which contains only 1s, and let I be the identity $n \times n$ matrix which contains 1s on the diagonal and 0s elsewhere. Let BI denote the $n \times (2n-2)$ matrix obtained by concatenating B and I, and let $\overline{BI}$ be the Boolean complement of the matrix BI. Then $\mathcal{C}_0 = \{$all the matrices obtained by permuting the columns of $\overline{BI}\}$ $\mathcal{C}_1 = \{$all the matrices obtained by permuting the columns of $BI\}$ has the following properties: Any single share contains an arbitrary collection of $n-1$ black and $n-1$ white sub-pixels; any pair of shares have $n-2$ common black and two individual black sub-pixels; any stacked triplet of shares from $\mathcal{C}_0$ has n black sub-pixels, whereas any stacked triplet of shares from $\mathcal{C}_1$ has $n+1$ black sub-pixels, which looks darker.

4.5 Concluding remarks

Here, we have seen the difference between traditional Visual Cryptography and Modified Visual Cryptography. We have also proposed a very simple modified sharing scheme.

Chapter 5

Balanced Strings and Uniform Codes

5.1 Introduction

We have seen that in modified visual cryptography, the pixels are expanded by a factor, called the blowing factor. So if one needs to improve the efficiency, one has to reduce the blowing factor. In this chapter, we investigate solutions with small blowing factor.

For a (k, n) - modified visual cryptography scheme, all the possible collections of less than k shares for each of the binary bit should possess identical properties. Otherwise, some (may be partial) information is leaked out. So, we can use only alike shares, i.e., which have equal length, say z, (= blowing factor) and consists of same number of 1s (say r). So the number of possible shares are limited to $\binom{z}{r}$. This number is maximum when $r = \lfloor \frac{z}{2} \rfloor$ or $\lceil \frac{z}{2} \rceil$. By these choices of r, the shares are more or less balanced in the sense that it has almost same number of 1s and 0s. Let us define the things more precisely.

Definition 5.1
Let $n_0(w)$ and $n_1(w)$ denote the number of 0s and number of 1s in a binary string w. *We say that the string w is perfectly balanced,* if $n_1(w) = n_0(w)$.

Then, by our definition, no string of odd length is perfectly balanced. So we relax that condition, and introduce the concept balanced string.

Definition 5.2
A binary string w is considered as *balanced,* if $n_1(w) - n_0(w) =$ *0, (*or $\pm$ *1),* depending on whether the length of w is even or odd, as the case may be.

Definition 5.3
A balanced string is called a *Uniform Code,* if, and only if,

$$n_0(w) \leq n_1(w) \leq n_0(w) + 1. \tag{5.1}$$

For example, 011010, 0101101 are uniform codes, 1010001, 0101101 are balanced strings, where as 0100 is an unbalanced string. Irrespective of whether z is odd or even, a uniform code of length z consists of precisely $\lceil \frac{z}{2} \rceil$ many 1s and $r = \lfloor \frac{z}{2} \rfloor$ many 0s. Let U_z denote the number of uniform codes of length z. Then

$$U_z = \binom{z}{\lfloor \frac{z}{2} \rfloor} \tag{5.2}$$

We have investigated the suitability of uniform codes for secret sharing schemes, and seen that they are most suitable in modified visual cryptography.

In the next section, we present a secret sharing scheme with modified visual cryptography, in which, the 0s and 1s are expanded with uniform codes.

We can see that in a $(2,\ n)$ secret sharing scheme, each bit can be recovered by combining the corresponding modified version of the bits from any two out of the n shares, depending upon whether the shares are same or different. Let z be the length of modified version of a bit. These uniform codes (by applying a random column permutation) are the shares to be distributed to the n participants. So we have chosen z such that $n \leq U_z$. Because, we want to reduce the blowing factor, we choose the smallest integer z, such that $n \leq U_z$ where n is the number of participants.

This choice of z ensures the existence of enough distinct shares for distribution to the n participants.

It may be noted that our choice of z implies,

$$U_{z-1} < n \leq U_z, \tag{5.3}$$

otherwise z might not be the smallest integer with the said property. Since $n \geq 2$, (otherwise, no sharing at all), $U_z \geq 2$, and so $z \geq 2$. It can be proved that $z = O(\log n)$.

In fact, it can be shown that

$$z < \frac{6}{5}.(\log_2 n) + 2 \tag{5.4}$$

We consider two matrices, A and B, each of order $n \times z$. While rows in A are a random selection of identical Uniform codes, the rows in B consist of a random selection of distinct Uniform codes. The resulting structure can be described by an $n \times z$ Boolean matrix, $S = [s_{ij}]$, where $S_{ij} = 1$, if and only if, the j^{th} bit in the i^{th} share is 1.

A solution to the 2 out of n modified visual secret sharing scheme consists of two collections of $n \times z$ Boolean matrices $\mathcal{C}_0$ and $\mathcal{C}_1$. To share a bit of value 0, the dealer randomly chooses one of the matrices in $\mathcal{C}_0$, and to share a bit of value 1, the dealer randomly chooses one of the matrices in $\mathcal{C}_1$. The rows of the chosen matrix define the modified version of the bit to be given to the n participants.

Definition 5.4

The solution is considered *valid* if the following pair of conditions are met:

1. Any share of a secret bit from either $\mathcal{C}_0$ or $\mathcal{C}_1$ is indistinguishable in the sense that it contains a random selection of the same number of 1s and 0s.

2. The result of combining (means "OR" or $\oplus$, depends on whether it is traditional or modified Visual cryptography, as the case may be) any pair of shares of a secret bit from $\mathcal{C}_0$, must be distinguishable from that of $\mathcal{C}_1$.

Consequently, the analysis of a single share makes it impossible to distinguish between $\mathcal{C}_0$ and $\mathcal{C}_1$. At the same time, if two shares are available, one can reveal the secret.

5.2 An Efficient $(2, n)$- threshold scheme

Let B be an $n \times z$ matrix, in which each row represents a distinct uniform code, and A be an $n \times z$ matrix, in which each row is the same as the first row of B.

Then a $(2, n)-$ visual secret sharing problem can be solved by using the following collections of $n \times z$ matrices:
$\mathcal{C}_0$ = all the matrices obtained by permuting the columns of A
$\mathcal{C}_1$ = all the matrices obtained by permuting the columns of B
Any single share in either $\mathcal{C}_0$ or $\mathcal{C}_1$ is a random selection of $\lceil \frac{z}{2} \rceil$ 1s and $\lfloor \frac{z}{2} \rfloor$ 0s. Consequently, the analysis of a single share makes it impossible to distinguish between $\mathcal{C}_0$ and $\mathcal{C}_1$. However, combining two shares from $\mathcal{C}_0$ results in a binary string consisting of only 0s, where as two shares from $\mathcal{C}_1$ results in binary string which has one or more 1s.

The shares are constructed by using the Algorithm 5.1 described below:

Algorithm 5.1 ((2, n) uniform construction)
Input: A binary string $B = b_1 b_2 \ldots b_t$ *of length* t.
Output : n blocks $S_1, S_2, \ldots, S_n$ *of length* $t.z$

Step 1. *For* i = *1 to* n *do*
    Initialize each share S_i *to null.*
Step 2. *For* i = *1 to* t *do*
    if (b_t = *0) randomly select a matrix* C *from* $\mathcal{C}_0$.
    else randomly select a matrix C *from* $\mathcal{C}_1$.
  For j = *1 to* n *do*
    concatenate the j^{th} *row of* C *with* S_j.

It may be noted that each participant gets the same or different uniform codes depending on whether the respective bit is 0 or 1.

Algorithm 5.2 (To recover the secret information)
Input : Shares $A = a_1 a_2 \ldots a_t$ *and*
  $B = b_1 b_2 \ldots b_t$ *of* t *blocks of* z *bits each.*
Output: The secret information $S = s_1 s_2 s_3 \ldots s_t$.

Step 1. *For* i = *1 to* t *do*
    if ($a_i = b_i$) $s_i = 0$;
    else s_i = *1;*
Step 2. *The recovered secret* $S = s_1 s_2 s_3 \ldots s_t$.

Example 5.1

Let there be 10 *participants* 1, 2, …, 10 *and suppose the secret encoded in binary is 100110.*

The value of z, obtained from the inequality (5.3) is, $z = 5$ and the list of uniform codes of length 5 are shown in Table 5.1.

Table 5.1: The list of all the 10 uniform codes of length 5.

Sl. No.	Code	Sl. No.	Code
1.	00111	6.	10101
2.	01011	7.	10110
3.	01101	8.	11001
4.	01110	9.	11010
5.	10011	10.	11100

$$\textit{Let } A = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \textit{ and } B = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Let $\mathcal{C}_0 = \{$all the matrices obtained by permuting the columns of $A\}$ and $\mathcal{C}_1 =$ all the matrices obtained by permuting the columns of $B\}$

The shares computed for each participant are as shown in Table 5.2. Let us compare any two shares block-wise, for example,

Table 5.2: The shares computed for different participants.

Sl. No.	shares
1	01101 10110 11100 10101 01110 01011
2	01011 10110 11100 00111 11100 01011
3	00111 10110 11100 10011 11010 01011
4	01110 10110 11100 10110 10110 01011
5	11001 10110 11100 01101 01101 01011
6	10101 10110 11100 11001 01011 01011
7	11100 10110 11100 11100 00111 01011
8	10011 10110 11100 01011 11001 01011
9	11010 10110 11100 01110 10101 01011
10	10110 10110 11100 11010 10011 01011

3^{rd} and 5^{th} shares. We see that, the first blocks are different, the next two blocks are the same, subsequent two blocks are different, and the last blocks are same. So the first bit is 1, next two bits are 0s, and so on. The entire secret is 100110.

It may be seen that, if we just perform block bitwise-OR by using the two shares, we get the following bit sequence, 11111 10110 11100 11111 11111 01011 and each bit of the secret can be computed by counting the number of 1s in the successive blocks of 5 bits. If the number of 1s in a block is 3, the corresponding bit in the secret must be 0, and if more than 3, it must be 1.

5.3 An upper bound of the Blowing factor

Theorem 1

$$\frac{2^z}{z+1} \leq U_z \leq 2^{z-1}, \tag{5.5}$$

for all positive integers z.

Proof: This can be proved as follows:

First we prove that the recurrence relation satisfied by $U_z = \binom{z}{\lfloor \frac{z}{2} \rfloor}$ is,

$$U_z = \begin{cases} \left(\frac{2z}{z+1}\right) U_{z-1}, & \text{if } z \text{ is an odd number} \\ 2.\,U_{z-1}, & \text{if } z \text{ is an even number} \end{cases} \tag{5.6}$$

This can be done by taking the two cases separately as follows:

Case 1. z is an odd number, say, $z = 2m - 1$, where m is an integer

$$\begin{aligned} U_z &= \binom{2m-1}{m-1} \\ &= \frac{(2m-1)(2m-2)\ldots(m+1)}{1.2.\ldots.(m-1)} \\ &= \frac{(2m-1)}{m} . \frac{(2m-2)(2m-3)\ldots(m+1).m}{1.2.\ldots.(m-1)} \\ &= \left(\frac{2.z}{z+1}\right).U_{z-1} \end{aligned} \tag{5.7}$$

Case 2. z is an even number, say, $z = 2m$, where m is an integer

$$\begin{aligned} U_z &= \binom{2m}{m} \\ &= \frac{(2m)(2m-1)\ldots(m+1)}{1.2.\ldots.(m-1).m} \\ &= 2.\frac{(2m-1)(2m-2)\ldots(m+1)}{1.2.\ldots.(m-1)} \\ &= 2.\,U_{z-1} \qquad (5.8) \end{aligned}$$

So,

$$U_z = \begin{cases} \left(\frac{2z}{z+1}\right) U_{z-1}, & \text{if } z \text{ is an odd number} \\ 2.\,U_{z-1}, & \text{if } z \text{ is an even number} \end{cases}$$

Since $\left(\frac{2z}{z+1}\right) < 2$, whenever $z > 0$, equation (5.6) becomes,

$$2.\left(\frac{z}{z+1}\right) U_{z-1} \le U_z \le 2.U_{z-1} \qquad (5.9)$$

Applying the inequality (5.9) $(z-1)$ times, and using the fact that $U_1 = U_0 = 1$, we get,

$$\frac{2^z}{z+1} \le U_z \le 2^{z-1} \qquad (5.10)$$

Theorem 2

$U_z \notin O(B^z)$, for any $B < 2$.

Proof: If possible, assume that $U_z \in O(B^z)$, for some $B < 2$. Then $\exists k > 0$ and an n_0, such that,

$$U_z \le kB^z,\ for\ all\ z \ge n_0. \qquad (5.11)$$

Then by inequality (5.10), $\frac{2^z}{z+1} \leq kB^z$, for all $z \geq n_0$.
This implies that

$$\left(\frac{2}{B}\right)^z \leq k(z+1),\ for\ all\ z \geq n_0. \tag{5.12}$$

Since $\frac{2}{B} > 1$, inequality (5.12) is absurd, since, the left side is exponential and the right side is linear. Hence the theorem.

Theorem 3

$$\left(\frac{9}{5}\right)^{z-1} < \binom{z}{\lfloor \frac{z}{2} \rfloor}, \tag{5.13}$$

for all positive integers z, except $z = 3$ and 5.

Proof: It can be easily settled in the case of $z = 2, 4, 6,$ and 7 by comparing the respective values:

- when $z = 2$, $\left(\frac{9}{5}\right) < \binom{2}{1} = 2$,
- when $z = 4$, $\left(\frac{9}{5}\right)^3 = \frac{729}{125} < \binom{4}{2} = 6$,
- when $z = 6$, $\left(\frac{9}{5}\right)^5 = \frac{59049}{3125} < \binom{6}{3} = 20$,
- when $z = 7$, $\left(\frac{9}{5}\right)^6 = \frac{531441}{15625} < \binom{7}{3} = 35$.

If $z \geq 9$, we have,

$$\frac{9}{5} \leq \frac{2z}{z+1} \tag{5.14}$$

So, if $z \geq 8$, the recurrence relation (5.6) becomes,

$$\left(\frac{9}{5}\right) U_{z-1} \leq U_z \tag{5.15}$$

Applying the above inequality $(z-8)$ times, we get,

$$\left(\frac{9}{5}\right)^{z-7} U_7 \leq U_z \tag{5.16}$$

and hence we get, $\left(\frac{9}{5}\right)^{z-1} < U_z$, since $\left(\frac{9}{5}\right)^6 < U_7$.
So, $(\frac{9}{5})^{z-1} < U_z = \binom{z}{\lfloor \frac{z}{2} \rfloor}$, when z is any integer other than 3 and 5 and hence the theorem.

So, if we select z as per inequality (5.3), we have,

$$U_{z-1} < n \leq U_z, \tag{5.17}$$

and by Theorems 1, and 3, we get,

$$\left(\frac{9}{5}\right)^{(z-2)} < n \leq 2^{(z-1)}, \tag{5.18}$$

when $z-1$ is other than 3 or 5, i.e, when z is other than 4 or 6.

Taking logarithm, we get,

$$(z-2).\log_2\left(\frac{9}{5}\right) < \log_2 n \leq z-1.$$

Since $\frac{5}{6} < \log_2\left(\frac{9}{5}\right)$, we have,

$$\frac{5}{6}(z-2) < \log_2 n \leq z-1,$$

and hence,

$$z < \frac{6}{5}.(\log_2 n) + 2 \tag{5.19}$$

If $z = 4$, then $4 \le n \le 9$, and in this case, $\frac{6}{5}(\log_2 n) + 2 \ge 4.4 > z$.

If $z = 6$, then $11 \le n \le 20$, and in this case, $\frac{6}{5}(\log_2 n) + 2 > 6.15 > z$. So, equation (5.4) is established.

5.4 Concluding remarks

We have presented a secret sharing scheme, in which the size of a share is in the $O(\log_2 n)$ times the size of the original secret, where n is the number of participants. It may be noted that the the blowing factor of the scheme suggested by Shamir, is n.

Chapter 6

Scheme for $(n-1,n)$ threshold

6.1 Introduction

In this section, we present our method to construct an $(n-1,n)$ secret sharing scheme based on the modified visual cryptography. In this scheme, every bit is expanded to $\lceil \frac{n}{2} \rceil$ many bits.

6.2 A new scheme

Let the participants be $\{P_1, P_2, P_3, \ldots, P_n\}$. In this case, the access structure consists of all the $n-1$ participants, namely:

$$\Gamma = \bigcup_{i=1}^{n} P_1 P_2 \ldots P_{i-1} \widehat{P}_i P_{i+1} \ldots P_{n-1} P_n$$

Here the $\widehat{P_i}$ indicate the absence of the participants P_i in the set. The complete elements can be listed as follows:

$$\begin{array}{lllllllll}
1. & \widehat{P_1} & P_2 & P_3 & P_4 & \ldots & P_{n-2} & P_{n-1} & P_n \\
2. & P_1 & \widehat{P_2} & P_3 & P_4 & \ldots & P_{n-2} & P_{n-1} & P_n \\
3. & P_1 & P_2 & \widehat{P_3} & P_4 & \ldots & P_{n-2} & P_{n-1} & P_n \\
4. & P_1 & P_2 & P_3 & \widehat{P_4} & \ldots & P_{n-2} & P_{n-1} & P_n \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
n. & P_1 & P_2 & P_3 & P_4 & \ldots & P_{n-2} & P_{n-1} & \widehat{P_n}
\end{array}$$

We can see that the first two sets differ in P_1 and P_2; the next two sets differ in P_3 and P_4; and so on. If we combine these sets pairwise, if n is even, there are exactly $\frac{n}{2}$ pairs of sets and if n is odd, there are $\left\lfloor \frac{n}{2} \right\rfloor$ many pairs and one set left out. Let the secret be $B = B_1B_2B_3 \ldots B_t$. Our scheme will generate n shares for each bit B_i of the secret.

6.3 Algorithm for sharing one bit among n shares

The following Algorithm describes how to share a single bit b among n shares.

Algorithm 6.1 (Sharing one bit among n shares)
Input: A binary bit $b \in \{0, 1\}$
Output: The n *shares* $S_1, S_2, \ldots, S_n$*, where,*
each S_i *is of length* $\lceil \frac{n}{2} \rceil$ *bits.*

Step 1. *Let* $S_{i,j}$ *denote the* j^{th} *bit of* S_i

For $j = 1$ *to* $\lfloor \frac{n}{2} \rfloor$ *do*

$x = b$

For $i = 1$ *to* n *do*

if $(i \neq 2j - 1$ *AND* $i \neq 2j)$ $\{$

Generate a random number $r \in \{0, 1\}$

$S_{i,j} = r$

$x = x \oplus r$

$\}$

$S_{2j-1,j} = S_{2j,j} = x$

Step 2. *If (n is odd) then* $\{$ \\\\ *Here* $j = \lceil \frac{n}{2} \rceil$

$x = b$

For $i = 1$ *to* $n - 2$ *do*

Generate a random number $r \in \{0, 1\}$

$S_{i,j} = r$

$x = x \oplus r$

$S_{n-1,j} = x$

$\}$ \\\\ *Note that in this case,* $S_{n,j}$ *is unknown*

Step 3. *The shares are* $S_1, S_2, \ldots, S_n$

Algorithm 6.2 (Recover the shared secret bit b)

Input: $n - 1$ *shares* $S_1 S_2 \ldots S_{j-1} S_{j+1} \ldots S_n$,

each of length $\lceil \frac{n}{2} \rceil$ *bits*

Observe that S_j *is the missing share.*

Output: The shared secret bit b

Step 1. *Let* $c = \lceil \frac{j}{2} \rceil$ *and* $x = 0$

For $k = 1$ *to* n *do*

if $(k \neq j)$ $x = x \oplus S_{k,c}$

$b = x$

Step 2. *The shared secret bit is recovered as b*

Lemma 6.1

The above scheme is a $(n-1, n)$ *threshold secret sharing scheme, in which the size of a share is* $\lceil \frac{n}{2} \rceil$ *bits.*

Proof: It is easy to observe the following from Algorithm 6.1.

1. For each $j \in \{1, \ldots, \lfloor \frac{n}{2} \rfloor\}$, the Step 1. of the algorithm generates $n-2$ random bits and assigns one each to $S_{i,j}$ for $i \in \{1, \ldots, n\} \setminus \{2j-1, 2j\}$.

2. The final value of x computed in the inner for loop is
$x = b \oplus S_{1,j} \oplus \ldots \oplus S_{2j-2,j} \oplus S_{2j+1,j} \oplus \ldots \oplus S_{n,j}$

3. This value of x is assigned to $S_{2j-1,j}$ and $S_{2j,j}$.
So, $S_{1,j} \oplus \ldots \oplus S_{2j-1,j} \oplus S_{2j+1,j} \oplus \ldots \oplus S_{n,j} = b$
and $S_{1,j} \oplus \ldots \oplus S_{2j-2,j} \oplus S_{2j,j} \oplus \ldots \oplus S_{n,j} = b$

4. If n is odd, Step 2 of the algorithm generates $n-2$ random bits and assigns one each to $S_{i,j}$ for $i \in \{1, \ldots, n-2\}$.
The final value of x computed in the for loop is
$x = b \oplus S_{1,j} \oplus \ldots \oplus S_{n-2,j}$

5. This value of x is assigned to $S_{n-1,j}$.
So, $S_{1,j} \oplus \ldots \oplus S_{n-1,j} = b$

Algorithm 6.3 (Sharing a secret among n shares)

Input: A binary string $B = B_1B_2 \dots B_t$ *of length* t

Output : The n *shares* $S_1, S_2, \dots, S_n$*, where,*

each S_i *is of length* $\lceil \frac{n}{2} \rceil$ *times* t.

Step 1. *For* $i = 1$ *to* n *do*

Initialize S_i *to NULL*

Step 2. *For* $i = 1$ *to* t *do*

Compute the n *shares corresponding to* B_i

using Algorithm 6.1 and append to the

corresponding S_j*, for* $j = \{1, \dots, n\}$.

Algorithm 6.4 (Recover the shared secret)

Input: $n - 1$ *shares* $S_1S_2 \dots S_{j-1}S_{j+1} \dots S_n$,

each of length t *times* $\lceil \frac{n}{2} \rceil$

Observe that S_j *is the missing share.*

Output: The shared secret $B = B_1B_2 \dots B_t$

Step 1. *Let* $S_j^{(1)}, S_j^{(2)}, \dots S_j^{(t)}$ *be the consecutive bits of length* $\lceil \frac{n}{2} \rceil$ *in* S_j*, for* $j \in \{1, \dots, n\}$

For $i = 1$ *to* t *do*

Recover the secret bit B_i *by using Algorithm 6.2*

with input $S_j^{(i)}$*, for* $j \in \{1, \dots, n\}$

Step 2. *The shared secret is* $B = B_1B_2 \dots B_t$

Example 6.1

Let a (4, 5) threshold secret sharing scheme be constructed for the secret B *= 10111 10111 10111 (which corresponds to "www").*

Here n = 5, so each bit will be expanded to 3 bits. The random bits generated by the Algorithm 6.3, and assigned at various places in the shares are as follows: (the ∗ indicates NULL bit and - indicates an unknown bit)

Table 6.1: Random bits assigned in the shares by Algorithm 6.1.

S_1	∗10∗01∗10∗00∗10∗10∗01∗10∗10∗11∗10∗01∗01∗10∗00
S_2	∗10∗00∗10∗11∗01∗11∗10∗10∗00∗01∗01∗10∗10∗01∗11
S_3	1∗10∗10∗00∗01∗10∗10∗11∗01∗10∗11∗01∗10∗10∗01∗1
S_4	0∗∗1∗∗0∗∗0∗∗1∗∗0∗∗1∗∗0∗∗1∗∗0∗∗1∗∗0∗∗1∗∗0∗∗1∗∗
S_5	01–01–10–01–01–10–01–11–11–01–00–11–00–00–10–

The bit values at the NULL positions are evaluated and the final shares are as seen in Table 6.2.

Table 6.2: Final Shares computed by Algorithm 6.1.

S_1	010101010100110010101110010111110001001110000
S_2	010100010111101011110110000101101010010101011
S_3	101011010010111011001100111011100101001000101
S_4	000110011010111011100001110010100000101000101
S_5	01–01–10–01–01–10–01–11–11–01–00–11–00–00–10–

Suppose we want to reconstruct the secret from 1st, 3rd, 4th and 5th shares. If we compute $S_1 \oplus S_3 \oplus S_4 \oplus S_5$, we get, result as 10-01-11-11-10-11-01-10-10-10-11-01-10-11-100. Here 2nd share is missing. So every first bit in the block of 3 bits are selected as : 10111 10111 10111

Suppose we want to reconstruct the secret from 1^{st}, 2^{nd}, 3^{rd}, and 4^{th}. If we compute $S_1 \oplus S_2 \oplus S_3 \oplus S_4$, we get, result as

$$101100001011011001110101011011011110111011011$$

Here 5^{th} share is missing. So every third bit in the block of 3 bits are selected as : 10111 10111 10111

6.4 Concluding remarks

We have now presented an $(n-1, n)$-threshold secret sharing scheme, in which the size of a share is $\lceil \frac{n}{2} \rceil$ times the size of the secret.

Chapter 7

An Efficient Scheme - Using Balanced Strings

7.1 Introduction

In this chapter, we present our method to construct an (n, n) secret sharing scheme based on the modified visual cryptography. Assume that the secret is represented as a binary string $B = b_1 b_2 b_3 \dots b_t$. Our scheme will generate n shares after concatenating a single bit, b_{t+1} at the right end of the secret. The resulting structure of the share can be described as a $k \times t$ Boolean matrix $\mathcal{C} = [S_{ij}]$, where, $1 \leq i \leq n,\ 1 \leq j \leq (t+1)$ and $k \in O(2^n)$. The construction is considered valid if, for any Boolean string $B = b_1 b_2 \dots b_t$, there exist solutions, $S_1, S_2, \dots, S_n$, such that, $B = S_1 \oplus S_2 \oplus \dots \oplus S_n$, where, $S_1, S_2, \dots, S_n$ are rows in $\mathcal{C}$. In the proposed scheme, the rows of $\mathcal{C}$ consist of all the possible balanced strings of length t. By Theorem 2, the cardinality of the class of uniform codes and balanced strings are in $O(2^n)$. We can choose $\mathcal{C}$ as the set of all uniform code or balanced strings.

The proposed scheme is based on the following theorem related to even parity strings and balanced strings:

Theorem 4
Let T be an even parity binary string of length t. Then we can find two balanced strings A and B, such that $T = A \oplus B$.

Proof: We can assume, without loss of generality that, the leading $2m, (0 \leq m \leq \lfloor \frac{t}{2} \rfloor)$ digits of T are 1s and remaining $t - 2m (\geq 0)$ digits are 0s. Now, let $A = PQ$ be the binary string obtained by concatenating the strings P and Q, where, P is the perfectly balanced string consisting of exactly m 1s, followed by m 0s, and Q is the balanced string consisting of exactly $\lfloor \frac{t-2m}{2} \rfloor$ 1s and $\lceil \frac{t-2m}{2} \rceil$ 0s. Note that Q is perfectly balanced, only if t is an even number. Choose $B = \overline{P}Q$, where, $\overline{P}$ is the Boolean complement of P, so that $T = A \oplus B$. Since the complement of a perfectly balanced string is also a perfectly balanced string and concatenation of a perfectly balanced string and a balanced string is a balanced string, both A and B are balanced strings. Hence the theorem.

Remark 7.1
Interchanging the number of 1s and 0s in Q, will lead to a decomposition of T in uniform codes. But decomposition in perfectly

balanced strings will be possible only if t is even. However, such a decomposition, in general, need not be unique. Also, once we find A, we can immediately obtain B, as $B = T \oplus A$.

It may be noted that, among the $2m$ 1s in T, exactly m 1s are in matched position with P, and the other m 1s are in matched position with Q. The matching can be made randomly. The bits in P and Q, corresponding to a 0 in T are same (either both 0 or both 1) and they can be assigned randomly, with ensuring that, $n_1(P) = n_1(Q) = \left\lfloor \frac{t}{2} \right\rfloor$.
Now we shall describe the construction details of a (2, 2)- secret sharing scheme and extend it to an (n, n)- scheme in the next section.

7.2 A (2, 2) Construction

Let $B = b_1 b_2 b_3 \ldots b_t$ be the secret information to be shared between two participants. We describe an efficient (2, 2) scheme by making use of the theorem 4. First of all, the necessary condition to use the theorem is that, the concerned string must be even parity. So, we extend the secret by appending a single bit at the right end. If we discard the appended last bit, we get precisely the secret. The length of the extended string is just one more than that of the secret. The Algorithm 7.1 extends the string and makes the resulting string an even parity.

Algorithm 7.1 (Append a single bit at the end)

Input: A binary string $B_t = b_1 b_2 \ldots b_t$ *of length* t.

Output : An even parity string $E_{t+1} = e_1 e_2 \ldots e_{t+1}$
of length $t + 1$*, such that* $e_i = b_i$*, for* $i \leq t$.

Step 1. *noOfOne = 0;*
    For i = 1 to t do
        $e_i = bi$;
        if ($b_i = 1$) *noOfOne = noOfOne + 1;*

Step 2. *if (noOfOne is odd)* $e_{t+1} = 1$;
    else $e_{t+1} = 0$;

Step 3. *The extended string is* $E_{t+1} = e_1 e_2 \ldots e_{t+1}$.

Now, using construction method in theorem 4, we split this extended string and obtain the two shares. The very simple algorithm 7.2, shown below, finds the decomposition of the extended string, as in theorem 4.

Algorithm 7.2 (Sharing an even parity binary string between two blocks)

Input: An even parity binary string $E_{t+1} = e_1 e_2 \ldots e_{t+1}$.

Output : Two blocks $S^{(1)}_{t+1} = s^{(1)}_1 s^{(1)}_2 \ldots s^{(1)}_{t+1}$ *and*
$S^{(2)}_{t+1} = s^{(2)}_1 s^{(2)}_2 \ldots s^{(2)}_{t+1}$ *of length* $t + 1$ *each.*

Step 1. *Set all bits of* $S^{(1)}_{t+1}$ *and* $S^{(2)}_{t+1}$ *null.*

Step 2. $noOfOne = 0$;
    For i = 1 to $(t + 1)$ *do*

if $(e_i = 1)$ *then*

$noOfOne = noOfOne + 1$;

if ($noOfOne$ *is odd*) $s_i^{(1)} = 1$;

else $s_i^{(1)} = 0$;

Step 3. *Randomly assign the rest null bits of* $S_{t+1}^{(1)}$ *to 0 or 1,such that* $n_1\left(S_{t+1}^{(1)}\right) = \left\lfloor \frac{t+1}{2} \right\rfloor$.

Step 4. *For* $i = 1$ *to* $t + 1$ *do*

$s_i^{(2)} = s_i^{(1)} \oplus e_i$.

The algorithm 7.3 shares any binary string between two shares, by using algorithm 7.1 and then algorithm 7.2.

Algorithm 7.3 (Sharing any binary string between two blocks)

Input: A binary string $B_t = b_1 b_2 \ldots b_t$.

Output : Two blocks $S_{t+1}^{(1)}$ *and* $S_{t+1}^{(2)}$ *each of length* $t + 1$

Step 1. *Let* $E_{t+1} = e_1 e_2 \ldots e_{t+1}$ *be the extended string obtained by Algorithm 7.1 with the input* B_t.

Step 2. *Obtain the shares* $S_{t+1}^{(1)}$ *and* $S_{t+1}^{(2)}$ *by Algorithm 7.2 with input* $E_{t+1.}$

Algorithm 7.4 (Recover the secret information)

Input : Two shares S_1 *and* S_2 *of 0s and 1s of length* $t + 1$

Output: The secret information $B_t = b_1 b_2 \ldots b_t$.

Step 1. $B_{t+1} = S_1 \oplus S_2$

Step 2. *The recovered secret is* $B = b_1 b_2 b_3 \ldots b_t$
(Note that b_{t+1} *is unwanted.)*

Recovery: From $E_{t+1} = S^{(1)}_{t+1} \oplus S^{(2)}_{t+1}$, it follows that, if we just discard last bit of E_{t+1} we get B_t. i.e, the recovery procedure is that, just $\oplus$ the two shares, we get the extended string, and discard the last appended bit we get the secret. Hence the following lemma:

Lemma 7.1

The Algorithm 7.3 described above is a (2, 2)- modified visual cryptography scheme, in which the size of the share is just one bit more than the size of secret. More over, all the shares are balanced strings.

Example 7.1

Let the secret B *be*

$$10011\,00101\,00011\,10010\,00101\,10100$$

(which corresponds to the word "secret").

Here length of the secret $t = 6 * 5 = 30$. By Step 1. of Algorithm 7.3, the extended secret is

$$B_{t+1} = 10011\,00101\,00011\,10010\,00101\,10100\,1.$$

By Step 1. of Algorithm 7.2, initialize S_1 and S_2 null.

In Step 2, S_1 is computed as 1∗∗01∗∗0∗1∗∗∗010∗∗1∗∗∗0∗10∗1∗∗0 (Here ∗ indicates null bits.) and by Step 3, S_1 is randomly set as

$$11101100010100100111010010011110$$

Finally by Step 4. of Algorithm 7.2, $S_2 = S_1 \oplus B_{t+1} = 01110101000101010101011001001111$

Recovery : Compute $S_1 \oplus S_2$ and get

$$B_t = 10011001010001110010001011010001$$

Last bit is 1 and is deleted to get B : 10011 00101 00011 10010 00101 10100.

7.3 A (n, n) Construction

We in this section develop a secret sharing scheme among n blocks.

Algorithm 7.5 (Sharing a secret among n blocks)
Input: A binary string $B_t = b_1 b_2 \ldots b_t$ of length t.
Output: n blocks $S_1, S_2, \ldots, S_n$ of length $t + 1$.

Step 1. $b_{t+1} = 0$;
Step 2. *Randomly assign n-2 blocks,*
$\{S_2, \ldots, S_{(n-1)}\}$*, with* $\lceil \frac{t+1}{2} \rceil$ *0s and* $\lfloor \frac{t+1}{2} \rfloor$ *1s.*
Step 3. *Compute* $K_{t+1} = B_{t+1} \oplus S_2 \oplus \ldots \oplus S_{(n-1)}$.

Step 4. *if ($K_{t+1}$is odd parity) then*

$$k_{t+1} = \overline{k_{t+1}}.$$

$$b_{t+1} = \overline{b_{t+1}}.$$

Step 5. *Compute S_1 and S_n by Algorithm 7.2, with input K_{t+1}, such that, $K_{t+1} = S_1 \oplus S_n$.*

Algorithm 7.6 (Recover the secret information)

Input : n shares S_1, S_2, ...,S_n of length $t + 1$

Output: The secret information $B_t = b_1 b_2 \ldots b_t$.

Step 1. *Compute the string $B_{t+1} = b_1 b_2 b_3 \ldots b_{t+1}$ such that $B_{t+1} = S_1 \oplus S_2 \oplus S_3 \oplus \ldots \oplus S_n$*

Step 2. *Discard the last bit of B_{t+1} and the recovered secret B_t is $b_1 b_2 b_3 \ldots b_t$*

Lemma 7.2

The Algorithm 7.5 described above, is an (n, n)- modified visual cryptography scheme, in which the size of the share is just one bit more than the size of secret. More over, all the shares are balanced strings.

Proof: It is clear that Step 1 of algorithm 7.5 appends a single bit at the end of the input string B_t and the extended string B_{t+1} is obtained. Note that the last bit appended is insignificant. In Step 2. it generates $n - 2$ shares, $S_2, S_3, \ldots, S_{n-1}$. They are all random balanced strings. In Step 3, from the equation,

$$K_{t+1} = B_{t+1} \oplus S_2 \oplus \ldots \oplus S_{(n-1)} \tag{7.1}$$

the following equation holds:

$$B_{t+1} = K_{t+1} \oplus S_2 \oplus \ldots \oplus S_{(n-1)} \tag{7.2}$$

In step 4, we ensure that K_{t+1} is even parity. If not, the last insignificant bit will be toggled to make it even parity. In this case, it also toggles the last bit of B_{t+1}, so that equation (7.2) is still valid. Finally, in step 5, share, K_{t+1}, between two shares $S_1 \oplus S_n$ by Algorithm 7.2 with input K_{t+1}. So, $B_{t+1} = S_1 \oplus S_2 \oplus \ldots \oplus S_{(n-1)} \oplus S_n$. Further more, each of the blocks $S_1, S_2, \ldots, S_n$ is a balanced string.

Example 7.2

For a (5, 5) threshold scheme, secret $B = 101101110$ *is taken.*

By step 1, the extended string, B_{t+1} of length 10 is,
10110111 00.
Randomly assign five 1s and five 0s to 3 rows $\{S_2, S_3, S_4\}$ in S. Therefore,

$$\begin{aligned} S_2 &= 1011000101, \\ S_3 &= 0101010110, \textit{ and} \\ S_4 &= 1100101010. \end{aligned}$$

Step 3. computes $K = 10011001\ 01$, and
in Step 5., 10011001 0 is split into

$$\begin{aligned} S_1 &= 1010110010, \textit{and} \\ S_5 &= 0011010110. \end{aligned}$$

All the 5 shares are as listed below:

$$\begin{aligned} S_1 &= 1010110010, \\ S_2 &= 1011000101, \\ S_3 &= 0101010110, \\ S_4 &= 1100101010, \textit{ and} \\ S_5 &= 0011010110. \end{aligned}$$

Recovery: Computes $S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_n$, and obtains

$$B_{t+1} = 10110111\ 01.$$

Deleting the last bit of B_{t+1}, we get the secret as

$$B_t = 10110111\ 0.$$

7.4 Security Analysis

In this section, we discuss the security of the proposed scheme. In order to show the security of the (2, 2) construction, suppose an illegal user gets one of the two shares. Lemma 7.3 shows that, guessing the secret correctly, is very difficult.

Lemma 7.3

With only one share, the probability of guessing the shared secret correctly in our construction is $\binom{t+1}{\lfloor \frac{t+1}{2} \rfloor}^{-1}$.

Proof: In our construction, it is easy to observe that each share contains $\lceil \frac{t+1}{2} \rceil$ 1s. There are $\binom{t+1}{\lfloor \frac{t+1}{2} \rfloor}$ many variations for a block, and the probability of guessing one block correctly is $\binom{t+1}{\lfloor \frac{t+1}{2} \rfloor}^{-1}$. Hence the probability of an illegal user, who has only one share, guessing the shared secret is $\binom{t+1}{\lfloor \frac{t+1}{2} \rfloor}^{-1}$.

In order to show the security of an (n, n) construction, suppose there are fewer than n participants cooperating to guess the shared secret. Lemma 7.4 shows that even though there are $n-1$ participants cooperating, the probability of guessing the shared secret correctly is still very low.

Lemma 7.4

The probability of guessing the shared secret correctly in our construction is $\binom{t+1}{\lfloor \frac{t+1}{2} \rfloor}^{-1}$, *if only* $n-1$ *shares are used to guess the share.*

Proof: The proof is similar to that of Lemma 7.3.

7.5 Concluding remarks

In this chapter, we have classified three types of balanced strings, and established a very strong theorem related to balanced string. As per the theorem, any string can be written as the ring sum $(\oplus)$ of two balanced strings. We have used this property and presented a secret sharing scheme, in which the size of a share is just one bit more than the size of the original secret.

Chapter 8

Permutation Ordered Binary Number System

8.1 Introduction

In the course of our research work we have formulated a new number system. This number system is found to be very useful and more efficient than the conventional number systems under use. We have used this number system in some of our newly introduced secret sharing schemes.

8.2 A new number system

We consider a general number system, called, Permutation Ordered Binary (POB) Number System with two non negative integral parameters, n and r, where $n \geq r$. The system is denoted by POB(n, r). In this number system, we represent all integers in the range $0, \ldots, \binom{n}{r} - 1$, as a binary string, say $B = b_{n-1}b_{n-2}\ldots b_0$, of length n, and having exactly r 1s.

Each digit of this number, say, b_j is associated with its position value, given by

$$b_j . \binom{j}{p_j} \ , \ \ where, \ \ p_j = \sum_{i=0}^{j} b_i \ ,$$

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

i.e.,

$$V(B) = \sum_{j=0}^{n-1} b_j . \binom{j}{p_j} \tag{8.1}$$

It can be proved that, since exactly $\binom{n}{r}$ such binary strings exist, each number will have a distinct representation. In order to emphasize that a binary string, $B = b_{n-1}b_{n-2}\ldots b_0$ is a POB-number, we denote the same by using the suffix 'p'. For example, 001110100_p is a POB(9, 4) number represented by 33. However, such a string, regarded as a binary number will have a decimal value of 116. We can arrange all those string in the ascending order, by considering this decimal value as in Table 8.1 . Indeed, Table 8.1 represents POB(9, 4) number system completely.

Table 8.1: List of POB(9,4) numbers

Sl.	POB Numbers	Binary		Sl.	POB Numbers	Binary
No.	1 2 3 4 5 6 7 8 9	value		No.	1 2 3 4 5 6 7 8 9	value
0	0 0 0 0 0 1 1 1 1	15		31	0 0 1 1 1 0 0 0 1	113
1	0 0 0 0 1 0 1 1 1	23		32	0 0 1 1 1 0 0 1 0	114
2	0 0 0 0 1 1 0 1 1	27		33	0 0 1 1 1 0 1 0 0	116
3	0 0 0 0 1 1 1 0 1	29		34	0 0 1 1 1 1 0 0 0	120
4	0 0 0 0 1 1 1 1 0	30		35	0 1 0 0 0 0 1 1 1	135
5	0 0 0 1 0 0 1 1 1	39		36	0 1 0 0 0 1 0 1 1	139
6	0 0 0 1 0 1 0 1 1	43		37	0 1 0 0 0 1 1 0 1	141
7	0 0 0 1 0 1 1 0 1	45		38	0 1 0 0 0 1 1 1 0	142
8	0 0 0 1 0 1 1 1 0	46		39	0 1 0 0 1 0 0 1 1	147
9	0 0 0 1 1 0 0 1 1	51		40	0 1 0 0 1 0 1 0 1	149
10	0 0 0 1 1 0 1 0 1	53		41	0 1 0 0 1 0 1 1 0	150
11	0 0 0 1 1 0 1 1 0	54		42	0 1 0 0 1 1 0 0 1	153
12	0 0 0 1 1 1 0 0 1	57		43	0 1 0 0 1 1 0 1 0	154
13	0 0 0 1 1 1 0 1 0	58		44	0 1 0 0 1 1 1 0 0	156
14	0 0 0 1 1 1 1 0 0	60		45	0 1 0 1 0 0 0 1 1	163
15	0 0 1 0 0 0 1 1 1	71		46	0 1 0 1 0 0 1 0 1	165
16	0 0 1 0 0 1 0 1 1	75		47	0 1 0 1 0 0 1 1 0	166
17	0 0 1 0 0 1 1 0 1	77		48	0 1 0 1 0 1 0 0 1	169
18	0 0 1 0 0 1 1 1 0	78		49	0 1 0 1 0 1 0 1 0	170
19	0 0 1 0 1 0 0 1 1	83		50	0 1 0 1 0 1 1 0 0	172
20	0 0 1 0 1 0 1 0 1	85		51	0 1 0 1 1 0 0 0 1	177
21	0 0 1 0 1 0 1 1 0	86		52	0 1 0 1 1 0 0 1 0	178
22	0 0 1 0 1 1 0 0 1	89		53	0 1 0 1 1 0 1 0 0	180
23	0 0 1 0 1 1 0 1 0	90		54	0 1 0 1 1 1 0 0 0	184
24	0 0 1 0 1 1 1 0 0	92		55	0 1 1 0 0 0 0 1 1	195
25	0 0 1 1 0 0 0 1 1	99		56	0 1 1 0 0 0 1 0 1	197
26	0 0 1 1 0 0 1 0 1	101		57	0 1 1 0 0 0 1 1 0	198
27	0 0 1 1 0 0 1 1 0	102		58	0 1 1 0 0 1 0 0 1	201
28	0 0 1 1 0 1 0 0 1	105		59	0 1 1 0 0 1 0 1 0	202
29	0 0 1 1 0 1 0 1 0	106		60	0 1 1 0 0 1 1 0 0	204
30	0 0 1 1 0 1 1 0 0	108		61	0 1 1 0 1 0 0 0 1	209

Table 8.1 Continues

Sl. No.	POB Numbers 1 2 3 4 5 6 7 8 9	Binary value		Sl. No.	POB Numbers 1 2 3 4 5 6 7 8 9	Binary value
62	0 1 1 0 1 0 0 1 0	210		94	1 0 1 0 0 1 0 1 0	330
63	0 1 1 0 1 0 1 0 0	212		95	1 0 1 0 0 1 1 0 0	332
64	0 1 1 0 1 1 0 0 0	216		96	1 0 1 0 1 0 0 0 1	337
65	0 1 1 1 0 0 0 0 1	225		97	1 0 1 0 1 0 0 1 0	338
66	0 1 1 1 0 0 0 1 0	226		98	1 0 1 0 1 0 1 0 0	340
67	0 1 1 1 0 0 1 0 0	228		99	1 0 1 0 1 1 0 0 0	344
68	0 1 1 1 0 1 0 0 0	232		100	1 0 1 1 0 0 0 0 1	353
69	0 1 1 1 1 0 0 0 0	240		101	1 0 1 1 0 0 0 1 0	354
70	1 0 0 0 0 0 1 1 1	263		102	1 0 1 1 0 0 1 0 0	356
71	1 0 0 0 0 1 0 1 1	267		103	1 0 1 1 0 1 0 0 0	360
72	1 0 0 0 0 1 1 0 1	269		104	1 0 1 1 1 0 0 0 0	368
73	1 0 0 0 0 1 1 1 0	270		105	1 1 0 0 0 0 0 1 1	387
74	1 0 0 0 1 0 0 1 1	275		106	1 1 0 0 0 0 1 0 1	389
75	1 0 0 0 1 0 1 0 1	277		107	1 1 0 0 0 0 1 1 0	390
76	1 0 0 0 1 0 1 1 0	278		108	1 1 0 0 0 1 0 0 1	393
77	1 0 0 0 1 1 0 0 1	281		109	1 1 0 0 0 1 0 1 0	394
78	1 0 0 0 1 1 0 1 0	282		110	1 1 0 0 0 1 1 0 0	396
79	1 0 0 0 1 1 1 0 0	284		111	1 1 0 0 1 0 0 0 1	401
80	1 0 0 1 0 0 0 1 1	291		112	1 1 0 0 1 0 0 1 0	402
81	1 0 0 1 0 0 1 0 1	293		113	1 1 0 0 1 0 1 0 0	404
82	1 0 0 1 0 0 1 1 0	294		114	1 1 0 0 1 1 0 0 0	408
83	1 0 0 1 0 1 0 0 1	297		115	1 1 0 1 0 0 0 0 1	417
84	1 0 0 1 0 1 0 1 0	298		116	1 1 0 1 0 0 0 1 0	418
85	1 0 0 1 0 1 1 0 0	300		117	1 1 0 1 0 0 1 0 0	420
86	1 0 0 1 1 0 0 0 1	305		118	1 1 0 1 0 1 0 0 0	424
87	1 0 0 1 1 0 0 1 0	306		119	1 1 0 1 1 0 0 0 0	432
88	1 0 0 1 1 0 1 0 0	308		120	1 1 1 0 0 0 0 0 1	449
89	1 0 0 1 1 1 0 0 0	312		121	1 1 1 0 0 0 0 1 0	450
90	1 0 1 0 0 0 0 1 1	323		122	1 1 1 0 0 0 1 0 0	452
91	1 0 1 0 0 0 1 0 1	325		123	1 1 1 0 0 1 0 0 0	456
92	1 0 1 0 0 0 1 1 0	326		124	1 1 1 0 1 0 0 0 0	464
93	1 0 1 0 0 1 0 0 1	329		125	1 1 1 1 0 0 0 0 0	480

8.3 POB-representation is unique

We prove that the POB-representation is unique in the sense that the binary correspondence of a POB-number is unique.

Theorem 5 (POB-representation is unique)
The value of a POB-number, $V(B)$ of $B = b_{n-1}b_{n-2}\dots b_0$ computed by the formula (8.1) given above, produces distinct values in the range $0, \cdots, \binom{n}{r} - 1$.

Proof: First, we prove that,

$$0 \leq V(B) \leq \binom{n}{r} - 1 \quad (8.2)$$

and then we prove that formula computes distinct values for distinct POB-numbers.

Let $b_{d_1}, b_{d_2}, \dots, b_{d_r}$, with

$$0 \leq d_1 < d_2 < \dots < d_r \leq n-1 \quad (8.3)$$

be the binary digits of B, having value 1.

Then the formula (8.1) takes the form

$$V(B) = \sum_{i=1}^{r} \binom{d_i}{i} \quad (8.4)$$

From the inequalities listed in (8.3), we get,

$$\begin{array}{ccccc} d_{r-1} & \leq & d_r & - & 1 \\ d_{r-2} & \leq & d_{r-1} & - & 1 \\ d_{r-3} & \leq & d_{r-2} & - & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_1 & \leq & d_2 & - & 1 \end{array}$$

Adding the first k inequalities listed above, we get,

$$d_{r-k} \leq d_r - k, \quad for\ k = 0, 1, \ldots, r-1 \tag{8.5}$$

Substituting $k = r - i$, inequality (8.5) becomes,

$$d_i \leq d_r - r + i, \quad for\ i = r, r-1, \ldots, 1 \tag{8.6}$$

Combining the inequalities (8.3) and (8.6), we get,

$$0 \leq d_i \leq d_r - r + i, \quad for\ i = 1, 2, \ldots, r \tag{8.7}$$

It may also be noted that

$$\binom{n}{0} = 1, \ whenever\ n \geq 0 \tag{8.8}$$

$$\binom{p}{i} = \binom{p-1}{i-1} + \binom{p-1}{i} \tag{8.9}$$

$$\binom{p}{i} \leq \binom{q}{i} \ whenever\ p \leq q \tag{8.10}$$

So, equation (8.4) becomes,

$$\begin{aligned} V(B) &= \sum_{i=1}^{r} \binom{d_i}{i} \\ &\leq \sum_{i=1}^{r} \binom{d_r - r + i}{i}, \qquad [\ by\ (8.7)\ \&\ (8.10) \\ &= \binom{d_r - r + 1}{0} + \sum_{i=1}^{r} \binom{d_r - r + i}{i} - 1 \\ &\qquad [\ by\ (8.7)\ \&\ (8.8) \\ &= \binom{d_r + 1}{r} - 1 \qquad (8.11) \\ &\qquad [\ by\ (8.9)\ applied\ r\ times \end{aligned}$$

i.e, if j is the highest integer with $b_j = 1$, then

$$V(B) \leq \binom{j+1}{r} - 1.$$

In other words, if $V(B) \leq \binom{j+1}{r} - 1$, then

$$b_{n-1} = b_{n-2} = \cdots = b_{j+1} = 0$$

and if $V(B) \geq \binom{j+1}{r}$, then at least one of

$$b_{n-1}, b_{n-2}, \cdots, b_{j+1} \neq 0.$$

Since $d_r \leq n-1$, we get, $V(B) \leq \binom{n}{r} - 1$.

As $V(B)$ is the sum of non-negative terms, we have,

$$0 \leq V(B) \leq \binom{n}{r} - 1.$$

So, the above formula will generate a maximum of $\binom{n}{r}$ values.

Now, let $X = x_{n-1}x_{n-2}\ldots x_0$ be any POB-number having r 1s, such that $X > B$ (by considering them as binary numbers). Being $X > B$, there is at least a digit x_l in X such $x_l \neq b_l$. Let l be the biggest suffix such that $x_l \neq b_l$.
Then, $x_{n-1}x_{n-2}\ldots x_{l+1} = b_{n-1}b_{n-2}\ldots b_{l+1}$, $x_l \neq b_l$ and $X > B$ implies $x_l = 1$ and $b_l = 0$. Now consider the strings $X_l = x_l x_{l-1}\ldots x_0$ and $B_l = b_l b_{l-1}\ldots b_0$. Both the strings X_l and B_l have equal number of 1s, say $k \leq r$ and hence can be regarded

as POB numbers(may be with different parameters).
Being X_l starts with 1, $V(X_l) \geq \binom{l}{k}$, and B_l starts with 0, $V(B_l) \leq \binom{l}{k} - 1$.

So, $V(X_l) > V(B_l)$ and thus, we get $V(X) > V(B)$.

i.e., if X and B are two distinct POB-numbers then $V(X) \neq V(B)$ and hence, the formula (8.1) generates exactly $\binom{n}{r}$ POB-values. Therefore the POB-representation is unique. Hence the theorem.

Moreover, V( ) preservers the natural order in binary number system.

8.4 POB-number and POB-value

In a practical situation, for any (n, r) threshold secret sharing system, it is required to find out the distribution of all of its keys. In all there will be $\binom{n}{r-1}$ keys, to be distributed among n participants. Which means, given a key, we should identify participants who should hold that particular key. In a sense, the key no. is the POB-value, and the allotment to participants is contained in the corresponding POB-number. Essentially, the position of 1s in the POB-number represents the participants holding the specific key. Therefore, the problem of allotment of keys to participants is equivalent to finding the POB-number corresponding to a POB-value. We have developed an algorithm for this problem.

For a given pair of parameters n and r with $r \leq n$, the algorithm takes three inputs: n, r and *value* with $0 \leq value \leq \binom{n}{r} - 1$ and produce POB-number corresponding to the *value*.

Algorithm 8.1 (Generate POB-number corresponding to a given POB-value)
In a POB(n, r) number system, if a POB-value, 'value' is given, the algorithm generates the binary digits of the corresponding POB-number: B, such that $value = V(B)$.
Input : Three numbers: n, r and value with $r \leq n$ and $0 \leq value \leq \binom{n}{r} - 1$.
Output: The POB-number $B = b_{n-1}b_{n-2}\ldots b_0$.

Step 1. *Let $j = n$ and $temp = value$.*
Step 2. *For k= r down to 1 do:*

```
        Repeat {
        j = j − 1;
        p = (j choose k);
        if (temp ≥ p)
            temp = temp − p;
            b_j = 1;
        else b_j = 0;
        } Until (b_j = 1);
Next k
```

Step 3. *if* $(j > 0)$

For $k = j - 1$ *down to 0 do:*

$b_k = 0;$

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

Lemma 8.1

Algorithm 8.1 generates the POB-number corresponding to the given POB-value.

Proof: At step 2, of the algorithm, a maximum of r b_js will be equal to 1. It may be observed that at any stage of the algorithm, $0 \leq temp$. Further, in any iteration of Step 2, for a k, at $j = k - 1$, $p = \binom{k-1}{k} = 0$ and so $temp \geq p$ (in line no. 4 of Step 2) and hence, b_j will be equal to 1, if not so for a higher value of j. Hence, it is clear that, after execution of Step 2, the binary string $B = b_{n-1}b_{n-2}\ldots b_0$ will have precisely r 1s and $n - j$ 0s. By Step 3, it will have r 1s and $n - r$ 0s.

It may also be noted that, in step 2 of the algorithm, the following two conditions hold good:

(i.) in line no. 1,

$$0 \leq temp \leq \binom{j}{k} - 1 \tag{8.12}$$

and (ii.) in line no. 9,

$$0 \leq temp \leq \binom{j}{k-1} - 1. \tag{8.13}$$

This can be proved as follows:

At the first time when the control reaches the line no. 1, in Step 2., we have, $temp = value, j = n, k = r$. So, inequality (8.12) trivially holds good as per the specification, $0 \leq value \leq \binom{n}{r} - 1$, mentioned in the input. In line no. 2, j is decremented by 1, so that in line no. 2, with new value of j, inequality (8.12) takes the form

$$0 \leq temp \leq \binom{j+1}{k} - 1 \tag{8.14}$$

In line no. 4, if $temp \leq p - 1$, where $p = \binom{j}{k}$, then b_j will be set to 0, and the Repeat $\cdots$ Until loop continues with none of the variables modified and control reaches line no. 1, so that inequality (8.12) holds good in this case.

On the other hand, if $temp \geq p$, then, $temp$ is decremented by a value of $p = \binom{j}{k}$, b_j will be set to 1, so that the Repeat $\cdots$ Until loop terminates and control reaches line no. 9. By using equation (8.9), the new value of $temp$ satisfies $0 \leq temp \leq \binom{j}{k-1} - 1$. i.e., inequality (8.13) holds good at line no 9.

In this case, value of k is decremented by 1, and if $k \geq 1$, the for loop continues and control reaches line no. 1, and inequality (8.13) becomes inequality (8.12) with the new value of k.

By principle of induction, the argument holds good for the new set of values of j, k and $temp$ so long as k reaches 1.

It may be noted that, when k reaches 1, in Step2, and for a j, when $b_j = 1$, at line no. 6 of Step 2, $temp \leq \binom{j}{k-1} - 1 = 0$. Since, $temp \geq 0, temp = 0$. In Step 3. we fills rest of b_js (if any), with 0. We have already ensured that there are exactly r number of b_js with 1s.

Whenever b_j is assigned 1, temp is diminished by p which is indeed $\binom{j}{k}$ and for the last j when b_j is assigned 1, in the algorithm, temp $= 0$. Thus POB-value of the B generated by the algorithm is *value* and the correctness of the algorithm is established.

If we want to compute all the POB-values sequentially, we could even have easier algorithm as follows:

Algorithm 8.2 (Generate all POB-numbers)
In a POB(n, r) number system, the algorithm prints all the POB Numbers sequentially.
Input : Positive integers n and r, with the condition $r \leq n$.
Output: All the POB-numbers in POB(n, r) number system.

Step 1. *Let $B = b_{n-1}b_{n-2}\dots b_0$ be a binary string,*

$$such that, b_i = \begin{cases} 1, & if\, 0 \leq i \leq r-1 \\ 0, & if\, r \leq i \leq n-1 \end{cases}$$

[B is the first POB-number in the POB(n, r) number system.]

Step 2. *Let* $done = 0$

```
Repeat {
   Print B
   Let NoOfZeros = 0, i = 0 and j = 1.
   while (b_j = 1 or b_i = 0) do {
      if (b_i = 0) NoOfZeros = NoOfZeros + 1;
      if (j = n − 1) done = 1;
      i = j;
      j = j + 1
   }
   b_j = 1;
   j = i − NoOfZeros;
   while(i ≥ j) do {
         b_i = 0, i = i − 1
   }
   while(i ≥ 0) do {
         b_i = 1, i = i − 1
   }
} Until (done = 1);
```

Given a POB-number B with POB-value $V(B)$, the algorithm 8.3, described below, will generate the successor of the POB-number, which corresponds to the value $V(B) + 1$. The algorithm may be used at the key distribution time for an easier and fast computation of the distribution of various keys.

In a POB(n, r) number system, given a POB-number $B = b_{n-1}b_{n-2} \dots b_0$, with POB-value $V(B)$, the following algorithm generates the binary digits of the POB-number, having POB-value $V(B) + 1$ and algorithm returns 1. If the input B is the last POB-number, the algorithm returns 0 as an indication that the output is not correct.

Algorithm 8.3 (Generate the next POB-number)
Input : An n digit POB-number $B = b_{n-1}b_{n-2}\dots b_0$.
Output: The POB-number corresponding to POB-value $= V(B)+1$, and return 1 or 0.

Step 1. *Search for the substring 01 in B from right end, i.e., find the max j, such that $b_j = 0$, $b_{j-1} = 1$*

Step 2. *If the search in Step 1 failed, return 0, as B contains no substring as 01, B is the maximum number that can be represented,*

Step 3. *Set $b_j = 1, b_{j-1} = 0$ and reverse the substring $b_{j-2}\dots b_0$ and return 1. The resulting string corresponds to $V(B)+1$.*

It can be seen that the algorithm 8.4 discussed below, generates the predecessor of POB-number, which corresponds to the value $V(B) - 1$

Algorithm 8.4 (Generate Predecessor POB-number)
Input : An n digit POB-number $B = b_{n-1}b_{n-2}\dots b_0$.
Output: The POB-number corresponding to POB-value $= V(B)-1$, and return 1 or 0.

Step 1. *Search for the substring 10 in B from right end, i.e., find the max* j*, such that* $b_j = 1$, $b_{j-1} = 0$

Step 2. *If the search in Step 1 failed, return 0, as B contains no substring as 10, and* $B = 0$*, the smallest number that can be represented.*

Step 3. *Set* $b_j = 0$, $b_{j-1} = 1$ *and reverse the substring* $b_{j-2} \ldots b_0$ *and return 1.*
The resulting string corresponds to $V(B) - 1$*.*

8.5 Illustrations

If $B = 001101010$, the next no. is 001101100;
If B = 000111100, the next no. is 001000111;
If B = 111100000, B is the largest number which can be represented, and so it returns zero. If $B = 101001100$, the predecessor no. is 101001010;
If B = 001000111, the predecessor no. is 000111100;
If B = 000001111, B is the smallest number which can be represented, and so it returns zero.

Remarks

Given two positive integral values n and r such that
$n \geq r$, there will be exactly $\binom{n}{r}$ members in POB(n, r). Using

Algorithm 8.1 and taking 0 . . . $\binom{n}{r}$ - 1 as POB-values, the corresponding POB-numbers can be generated and therefore the entire POB(n, r) system could be generated by the Algorithm 8.1.

8.6 Concluding remarks

We have generalized the concept of balanced string, and have introduced a new number system, called Permutation Ordered Binary Number System. We have proved that the POB-number representation is unique. Also, several algorithms to manipulate POB-number system are discussed. This number system has great potential in Secret Sharing.

Chapter 9

Improvement Scheme Using POB Numbers

9.1 Introduction

In this section we describe the construction details of a (2, 2) secret sharing scheme and in the next section, the construction details of an n out of n scheme for $n \geq 3$. The simplest version of the scheme assumes that the secret consists of a sequence of bytes and each byte is handled separately. The construction is based on the following theorem, which is a particular case (when $t = 9$) of the theorem 4, discussed in the last chapter.

Theorem 6

Let T be a binary string of even parity, having length 9. Then we can find two binary strings A and B each having exactly four 1s and five 0s such that $T = A \oplus B$.

9.2 A (2, 2) Construction

Let $K = k_1k_2 \ldots k_8$ be one byte of the secret information to be shared between two participants. In order to share the byte between two participants, we first extend the byte by inserting a bit at random position, $r, 1 \leq r \leq 9$. The inserted digit will be such that, the resulting extended string T is of even parity. This extended string T is split into two POB(9, 4) numbers, according to theorem 6, such that $T = A \oplus B$. The shares S_1 and S_2 are the values $V(A)$ and $V(B)$ represented by the POB-numbers A and B respectively. Note that $V(A)$ and $V(B)$ are 7 bits long.

9.2.1 Algorithm to Share one byte between two shares

The details of construction is described in the following Algorithm 9.1.

Algorithm 9.1 (Sharing a byte between two blocks)
Input: A binary string $K = K_1K_2 \ldots K_8$.
Output : Two blocks S_1 and S_2 of length 7 bits.

Step 1. *Let A and B are two 9 bits long integers.*
Set all the bits of A and B to null,
randomly select an integer r in [1...9].

Step 2. *The input string K is extended to T*
by inserting one bit at position r.
Compute the binary string $T = T_1T_2 \dots T_9$

$$\textit{where } T_i = \begin{cases} K_i, & \textit{if } i < r \\ K_{i-1}, & \textit{if } i > r \\ 0, & \textit{if } i = r \textit{ and } K \textit{ is even parity} \\ 1, & \textit{if } i = r \textit{ and } K \textit{ is odd parity} \end{cases}$$

Step 3. *noOfOne = 0;*
For i = 1 to 9 do
if (T_i = 1) then
$noOfOne = noOfOne + 1$;
if ($noOfOne$ is odd) $A_i = 1$;
else A_i = 0;

Step 4. *Randomly assign the rest null bits of A*
to 0 or 1, and let A consists of four 1s and five 0s.

Step 5. *let j = 0.*
For i = 1 to 9 do
$B_i = A_i \oplus T_i$

Step 6. *Let S_1 and S_2 be the POB-values corresponding*
to the POB-numbers A and B, respectively.

9.2.2 Algorithm to Recover the shared byte

Algorithm 9.2 (Recover the secret information)

Input : Two shares S_1 *and* S_2 *of length 7 bits each and the random integer* r*.*

Output: The secret information $K = K_1K_2K_3 \ldots K_8$*.*

Step 1. *Let* A *and* B *be the POB-numbers corresponding to* S_1 *and* S_2 *respectively.*

Step 2. *For* i *= 1 to* 8 *do*

if $(i \geq r)$ $j = i + 1$;

else $j = i$;

$K_i = A_j \oplus B_j$.

Step 3. *The recovered secret is* $K = K_1K_2K_3 \ldots K_8$

Lemma 9.1

The above scheme is a 2 out of 2 secret sharing scheme.

Proof: It may be observed that, in step 2 of Algorithm 9.1, the extended string T is of even parity. Since the length of T is 9, it can have a maximum of eight 1s. Let T contains $2m, (0 \leq m \leq 4)$ 1s. Then in Step 3, the $2m$ bits of A, corresponding to the 1s in T will be set to 1s and 0s equally. The Step 4 of Algorithm 9.1, ensures that A contains four 1s and five 0s. The string $B = A \oplus T$, computed in Step 5, also consists of four 1s and five 0s, as per Theorem 4. So the shares S_1 and S_2, which are POB-values of A and B, are each of 7 bits length. The condition

$B = A \oplus T$ in Step 5, implies $T = A \oplus B$, and if we drop out r^{th} bit of T, we get, K. Thus, the above scheme is a 2 out of 2 secret sharing scheme. Besides, each byte is shared by a seven bit string.

It may be seen that in algorithm 9.1, the size of shares is only 7 bits, while the size of the original secret message is 8 bits. The new scheme provides a gain of one bit per one byte of secret in its representation.

Example 9.1

Let us consider a secret of two bytes, say, K = *11011110 10100001*

Let the random numbers generated to share these two bytes be 4, and 3 respectively, so that the extended string T (inserted bits are underlined) is as follows:

Step 2. $110\underline{0}11110\ 10\underline{1}100001$.

The string A after step 3 and 4 are as follows:

Step 3. $10{*}\underline{*}1010{*}\ 1{*}\underline{0}1{*}{*}{*}{*}0$.

Step 4. 101010100 100110100

The string $B = A \oplus T$, computed in Step 5 is:

011001010 001010101.

The indices of these codes are 98, 88 and 59, 20.

The final shares are 1100010 1011000 and 0111011 0010100.

Recovery : The codes corresponding to the numbers are as follows:

$$A \quad : \quad 101010100\ 100110100$$
$$B \quad : \quad 011001010\ 001010101$$
$$\textit{Compute}\ \ T = A \oplus B \quad = \quad 110011110\ 101100001$$

Deleting the 4^{th} and 3^{rd} bits from the consecutive blocks of T, we get, the secret $K = 11011110\ 10100001$.

9.3 An (n, n) Construction

9.3.1 Algorithm to Share one byte between n shares

The details of construction is described in the following Algorithm 9.3.

Algorithm 9.3 (Sharing a secret among n blocks)
Input:A single byte string $K = K_1K_2K_3 \ldots K_8$.
Output : n shares $S_1, S_2, \ldots, S_n$ *of length 7 bits.*

Step 1. *Let* $A_1, A_2, \ldots A_n$ *be null strings of length 9 bits.*
Step 2. *Randomly assign n-2 POB(9,4)-numbers one for each of* $A_i, 2 \leq i \leq n-1$.
Let $r = \left\lceil \frac{V(A_2)+1}{14} \right\rceil$
Step 3. *The input string* K *is expanded to* T

by inserting one bit at position r.

Compute the binary string $T = T_1T_2 \dots T_9$

$$T_i = \begin{cases} K_i, & \text{if } i < r \\ K_{i-1}, & \text{if } i > r \\ 0, & \text{if } i = r \text{ and } K \text{ is even parity} \\ 1, & \text{if } i = r \text{ and } K \text{ is odd parity} \end{cases}$$

Step 4. *Let $W = T \oplus A_2 \oplus A_3 \oplus \dots \oplus A_{n-1}$*

Step 5. *Let $W = W_1W_2 \dots W_9$*

$noOfOne$ = 0;

For i = 1 to 9 do

if (W_i = 1) then

$noOfOne = noOfOne + 1$;

if ($noOfOne$ is odd) A_{1i} = 1;

else A_{1i} = 0;

Step 6. *Randomly assign the rest null bits of A_1 to 0 or 1,*

let A_1 consists of four 1s and five 0s.

Step 7. *Compute $A_n = W \oplus A_1$*

Step 8. *For i= 1 to n do*

$S_i = V(A_i)$.

Algorithm 9.4 (Recover the secret information) 2

Input : n shares S_1, S_2, ...,S_n of length 7 bits each.

Output: The secret information $K = K_1K_2K_3 \dots K_8$. 4

Step 1. *Let $A_1, A_2, \dots A_n$ be the POB-numbers corresponding to S_1, S_2, ...,S_n respectively and $r = \left\lceil \frac{S_2)+1}{14} \right\rceil$*

Compute $T = A_1 \oplus A_2 \oplus A_3 \oplus \ldots \oplus A_n$

Let $T = T_1 T_2 \ldots T_9$

Step 2. *For* i = *1 to* 8 *do*

if $(i \geq r)$ $j = i + 1$;

else $j = i$;

$K_i = T_j$.

Step 3. *The recovered secret is* $K = K_1 K_2 K_3 \ldots K_8$

Lemma 9.2

The above scheme is an n out of n secret sharing scheme.

Proof: In Step 2, of Algorithm 9.3, A_is are assigned as random POB(9, 4)-numbers, $V(A_2)$ is a random number in [0, ..., 125] and hence, $r = \left\lceil \frac{V(A_2)+1}{14} \right\rceil$, is uniformly at random number in [1,...,9]. It may be noted that after Step 3, the expanded string T is of even parity. It is clear that Step 4 of Algorithm 9.3, we have,

$$W = T \oplus A_2 \oplus A_3 \oplus \ldots \oplus A_{n-1}, \tag{9.1}$$

from which the following equation holds:

$$T = W \oplus A_2 \oplus A_3 \oplus \ldots \oplus A_{n-1} \tag{9.2}$$

Further more, since all the A_is are of even parity, W is also of even parity. The W is written as,

$$W = A_1 \oplus A_n, \tag{9.3}$$

by using Steps 5, 6, and 7, in the same way as what we have done in the case of Algorithm 9.1. Substituting equation (9.3) in equation (9.2), we get,

$$T = A_1 \oplus A_2 \oplus A_3 \oplus \ldots \oplus A_n \tag{9.4}$$

Finally, the shares, S_is, are POB-values corresponding to the POB-numbers A_is. In order to get the secret K, r^{th} bit of T is dropped out.

Example 9.2

For a (5, 5) threshold scheme, secret $K = 10110110$ *is taken.*

Randomly assign five 0s and four 1s to 3 rows $\{A_2, A_3, A_4\}$. Therefore,

$$\begin{aligned} A_2 &= 101100010, \\ A_3 &= 010101001, \textit{ and} \\ A_4 &= 110010100. \end{aligned}$$

Let the random number $r = \left\lceil \frac{V(A_2)+1}{14} \right\rceil = \left\lceil \frac{102}{14} \right\rceil = 8.$

The expanded string T as per step 3, of Algorithm 9.3 is $T = 101101110$

Step 4. Computes $W = 100110001$,
by Step 5., $A_1 = 1{*}{*}01{*}{*}{*}0$, and
by step 6., A_1 becomes $= 110010100$

By Step 7, $A_5 = 010100101$

The shares are the indices: 113, 101, 48, 113, 46. All the 5 shares are listed below:

$$\begin{aligned}
S_1 &= 1110001, \\
S_2 &= 1100101, \\
S_3 &= 0110000, \\
S_4 &= 1110001, \textit{ and} \\
S_5 &= 0101110.
\end{aligned}$$

Recovery: Compute $T = A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5$, and get 101101110. Deleting the 8^{th} bit, we get secret as $K = 10110110$.

9.4 Security Analysis

In the construction under the POB(9,4) number system there are a total of 126 shares corresponding to one byte of secret. The probability of a correct guess of a share is $\frac{1}{126}$ per byte of secret. This would mean that for a secret of m-bytes, the probability of correct guess of a share will be as low as $\left(\frac{1}{126}\right)^m$.

9.5 Concluding remarks

We have seen that, a 9 bit POB-number could be represented by a 7 bit binary number. By taking the benefit of this, we have proposed a secret sharing scheme. The algorithms for generating the shares and recovery of the secret are discussed. The proposed scheme is effective, where we have a gain of one bit for every 8 bits of information. The full potential of the newly introduced POB-number system is yet to be explored.

Chapter 10

Conclusions

We have given the theoretical background of Secret Sharing Schemes and the historical development of the subject. The evolution 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 disadvantages of the various forms secret sharing schemes are brought out. Several new schemes for both dealing and combining are included in the thesis. We have introduced a new number system, called, POB number system. Representation using POB number system has been presented. Algorithms for finding the POB number and POB value are given. We have also proved that the representation using POB number system is unique and is more efficient. Being a 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. 3 contains the list of publications of some of our research findings in this area.

We have improved many of the existing schemes and introduced a few new schemes. The introduction of POB number system and using it for some very efficient uniform secret sharing scheme is the most significant achievement of this research work.

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

APPENDIX 1

The Distribution of keys

Let us return to the example we considered in section 1.3. We denote the scientists by the letters: a, $b, \ldots, k$. As per our scheme, any 6 of the 11 scientists together should be able to open the cabinet using the keys in their possession. The scheme envisages the use of at least one key from each of the six scientists. There are in all 462 different locks and keys. The keys are numbered from 0 to 461. For each lock there must be exactly six keys as no five from among the 11 scientists could be able to open a particular lock. The allotment of each key to the scientists are denoted by 1s against their names in the column. For example key no.3 will be available with scientists - e, f, g, i, j and k. In other words, any permutation of six 1s and five 0s denote allotment of a specific key. Every such permutation can be considered as a unique 11 digit binary number having a specific decimal value. We have chosen to assign the key numbers in the ascending order of its decimal value. For example, key no.0 has 63 as decimal value, where as key no.35 has 343 as its value.

An algorithm for allocating the 462 keys is given in Table 10.1.

It may be noted that the numeric value corresponding to the distribution of keys of a specific lock can be easily computed as follows:

The key no. can be computed from the corresponding binary number in the table using the following formula:

$$keyno. = \sum_{j=0}^{10} b_j \binom{j}{p_j}$$

where

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

and $b_{10}b_9 \ldots b_0$ is the binary number. For example, the key no. corresponding to the binary number

$$\begin{aligned} 10110011010 &= \binom{10}{6} + \binom{8}{5} + \binom{7}{4} + \binom{4}{3} + \binom{3}{2} + \binom{1}{1} \\ &= 210 + 56 + 35 + 4 + 3 + 1 \\ &= 309. \end{aligned}$$

It may be noted that the table consists of all binary numbers of length 11 and having precisely 6 1s, arranged in the ascending order of its decimal value.

Table 10.1: The distribution of keys of various locks to the scientists.

Sl.	Scientists											Binary	Sl.	Scientists											Binary
No.	a	b	c	d	e	f	g	h	i	j	k	value	No.	a	b	c	d	e	f	g	h	i	j	k	value
0	0	0	0	0	0	1	1	1	1	1	1	63	33	0	0	1	0	0	1	1	1	1	1	0	318
1	0	0	0	0	1	0	1	1	1	1	1	95	34	0	0	1	0	1	0	0	1	1	1	1	335
2	0	0	0	0	1	1	0	1	1	1	1	111	35	0	0	1	0	1	0	1	0	1	1	1	343
3	0	0	0	0	1	1	1	0	1	1	1	119	36	0	0	1	0	1	0	1	1	0	1	1	347
4	0	0	0	0	1	1	1	1	0	1	1	123	37	0	0	1	0	1	0	1	1	1	0	1	349
5	0	0	0	0	1	1	1	1	1	0	1	125	38	0	0	1	0	1	0	1	1	1	1	0	350
6	0	0	0	0	1	1	1	1	1	1	0	126	39	0	0	1	0	1	1	0	0	1	1	1	359
7	0	0	0	1	0	0	1	1	1	1	1	159	40	0	0	1	0	1	1	0	1	0	1	1	363
8	0	0	0	1	0	1	0	1	1	1	1	175	41	0	0	1	0	1	1	0	1	1	0	1	365
9	0	0	0	1	0	1	1	0	1	1	1	183	42	0	0	1	0	1	1	0	1	1	1	0	366
10	0	0	0	1	0	1	1	1	0	1	1	187	43	0	0	1	0	1	1	1	0	0	1	1	371
11	0	0	0	1	0	1	1	1	1	0	1	189	44	0	0	1	0	1	1	1	0	1	0	1	373
12	0	0	0	1	0	1	1	1	1	1	0	190	45	0	0	1	0	1	1	1	0	1	1	0	374
13	0	0	0	1	1	0	0	1	1	1	1	207	46	0	0	1	0	1	1	1	1	0	0	1	377
14	0	0	0	1	1	0	1	0	1	1	1	215	47	0	0	1	0	1	1	1	1	0	1	0	378
15	0	0	0	1	1	0	1	1	0	1	1	219	48	0	0	1	0	1	1	1	1	1	0	0	380
16	0	0	0	1	1	0	1	1	1	0	1	221	49	0	0	1	1	0	0	0	1	1	1	1	399
17	0	0	0	1	1	0	1	1	1	1	0	222	50	0	0	1	1	0	0	1	0	1	1	1	407
18	0	0	0	1	1	1	0	0	1	1	1	231	51	0	0	1	1	0	0	1	1	0	1	1	411
19	0	0	0	1	1	1	0	1	0	1	1	235	52	0	0	1	1	0	0	1	1	1	0	1	413
20	0	0	0	1	1	1	0	1	1	0	1	237	53	0	0	1	1	0	0	1	1	1	1	0	414
21	0	0	0	1	1	1	0	1	1	1	0	238	54	0	0	1	1	0	1	0	0	1	1	1	423
22	0	0	0	1	1	1	1	0	0	1	1	243	55	0	0	1	1	0	1	0	1	0	1	1	427
23	0	0	0	1	1	1	1	0	1	0	1	245	56	0	0	1	1	0	1	0	1	1	0	1	429
24	0	0	0	1	1	1	1	0	1	1	0	246	57	0	0	1	1	0	1	0	1	1	1	0	430
25	0	0	0	1	1	1	1	1	0	0	1	249	58	0	0	1	1	0	1	1	0	0	1	1	435
26	0	0	0	1	1	1	1	1	0	1	0	250	59	0	0	1	1	0	1	1	0	1	0	1	437
27	0	0	0	1	1	1	1	1	1	0	0	252	60	0	0	1	1	0	1	1	0	1	1	0	438
28	0	0	1	0	0	0	1	1	1	1	1	287	61	0	0	1	1	0	1	1	1	0	0	1	441
29	0	0	1	0	0	1	0	1	1	1	1	303	62	0	0	1	1	0	1	1	1	0	1	0	442
30	0	0	1	0	0	1	1	0	1	1	1	311	63	0	0	1	1	0	1	1	1	1	0	0	444
31	0	0	1	0	0	1	1	1	0	1	1	315	64	0	0	1	1	1	0	0	0	1	1	1	455
32	0	0	1	0	0	1	1	1	1	0	1	317	65	0	0	1	1	1	0	0	1	0	1	1	459

Table 10.1 Continues

Sl.	Scientists											Binary
No.	a	b	c	d	e	f	g	h	i	j	k	value
66	0	0	1	1	1	0	0	1	1	0	1	461
67	0	0	1	1	1	0	0	1	1	1	0	462
68	0	0	1	1	1	0	1	0	0	1	1	467
69	0	0	1	1	1	0	1	0	1	0	1	469
70	0	0	1	1	1	0	1	0	1	1	0	470
71	0	0	1	1	1	0	1	1	0	0	1	473
72	0	0	1	1	1	0	1	1	0	1	0	474
73	0	0	1	1	1	0	1	1	1	0	0	476
74	0	0	1	1	1	1	0	0	0	1	1	483
75	0	0	1	1	1	1	0	0	1	0	1	485
76	0	0	1	1	1	1	0	0	1	1	0	486
77	0	0	1	1	1	1	0	1	0	0	1	489
78	0	0	1	1	1	1	0	1	0	1	0	490
79	0	0	1	1	1	1	0	1	1	0	0	492
80	0	0	1	1	1	1	1	0	0	0	1	497
81	0	0	1	1	1	1	1	0	0	1	0	498
82	0	0	1	1	1	1	1	0	1	0	0	500
83	0	0	1	1	1	1	1	1	0	0	0	504
84	0	1	0	0	0	0	1	1	1	1	1	543
85	0	1	0	0	0	1	0	1	1	1	1	559
86	0	1	0	0	0	1	1	0	1	1	1	567
87	0	1	0	0	0	1	1	1	0	1	1	571
88	0	1	0	0	0	1	1	1	1	0	1	573
89	0	1	0	0	0	1	1	1	1	1	0	574
90	0	1	0	0	1	0	0	1	1	1	1	591
91	0	1	0	0	1	0	1	0	1	1	1	599
92	0	1	0	0	1	0	1	1	0	1	1	603
93	0	1	0	0	1	0	1	1	1	0	1	605
94	0	1	0	0	1	0	1	1	1	1	0	606
95	0	1	0	0	1	1	0	0	1	1	1	615
96	0	1	0	0	1	1	0	1	0	1	1	619
97	0	1	0	0	1	1	0	1	1	0	1	621
98	0	1	0	0	1	1	0	1	1	1	0	622
99	0	1	0	0	1	1	1	0	0	1	1	627
100	0	1	0	0	1	1	1	0	1	0	1	629
101	0	1	0	0	1	1	1	0	1	1	0	630
102	0	1	0	0	1	1	1	1	0	0	1	633
103	0	1	0	0	1	1	1	1	0	1	0	634
104	0	1	0	0	1	1	1	1	1	0	0	636
105	0	1	0	1	0	0	0	1	1	1	1	655
106	0	1	0	1	0	0	1	0	1	1	1	663
107	0	1	0	1	0	0	1	1	0	1	1	667
108	0	1	0	1	0	0	1	1	1	0	1	669
109	0	1	0	1	0	0	1	1	1	1	0	670
110	0	1	0	1	0	1	0	0	1	1	1	679
111	0	1	0	1	0	1	0	1	0	1	1	683
112	0	1	0	1	0	1	0	1	1	0	1	685
113	0	1	0	1	0	1	0	1	1	1	0	686
114	0	1	0	1	0	1	1	0	0	1	1	691
115	0	1	0	1	0	1	1	0	1	0	1	693
116	0	1	0	1	0	1	1	0	1	1	0	694
117	0	1	0	1	0	1	1	1	0	0	1	697
118	0	1	0	1	0	1	1	1	0	1	0	698
119	0	1	0	1	0	1	1	1	1	0	0	700
120	0	1	0	1	1	0	0	0	1	1	1	711
121	0	1	0	1	1	0	0	1	0	1	1	715
122	0	1	0	1	1	0	0	1	1	0	1	717
123	0	1	0	1	1	0	0	1	1	1	0	718
124	0	1	0	1	1	0	1	0	0	1	1	723
125	0	1	0	1	1	0	1	0	1	0	1	725
126	0	1	0	1	1	0	1	0	1	1	0	726
127	0	1	0	1	1	0	1	1	0	0	1	729
128	0	1	0	1	1	0	1	1	0	1	0	730
129	0	1	0	1	1	0	1	1	1	0	0	732
130	0	1	0	1	1	1	0	0	0	1	1	739
131	0	1	0	1	1	1	0	0	1	0	1	741

Table 10.1 Continues

Sl. No.	a	b	c	d	e	f	g	h	i	j	k	Binary value
132	0	1	0	1	1	1	0	0	1	1	0	742
133	0	1	0	1	1	1	0	1	0	0	1	745
134	0	1	0	1	1	1	0	1	0	1	0	746
135	0	1	0	1	1	1	0	1	1	0	0	748
136	0	1	0	1	1	1	1	0	0	0	1	753
137	0	1	0	1	1	1	1	0	0	1	0	754
138	0	1	0	1	1	1	1	0	1	0	0	756
139	0	1	0	1	1	1	1	1	0	0	0	760
140	0	1	1	0	0	0	0	1	1	1	1	783
141	0	1	1	0	0	0	1	0	1	1	1	791
142	0	1	1	0	0	0	1	1	0	1	1	795
143	0	1	1	0	0	0	1	1	1	0	1	797
144	0	1	1	0	0	0	1	1	1	1	0	798
145	0	1	1	0	0	1	0	0	1	1	1	807
146	0	1	1	0	0	1	0	1	0	1	1	811
147	0	1	1	0	0	1	0	1	1	0	1	813
148	0	1	1	0	0	1	0	1	1	1	0	814
149	0	1	1	0	0	1	1	0	0	1	1	819
150	0	1	1	0	0	1	1	0	1	0	1	821
151	0	1	1	0	0	1	1	0	1	1	0	822
152	0	1	1	0	0	1	1	1	0	0	1	825
153	0	1	1	0	0	1	1	1	0	1	0	826
154	0	1	1	0	0	1	1	1	1	0	0	828
155	0	1	1	0	1	0	0	0	1	1	1	839
156	0	1	1	0	1	0	0	1	0	1	1	843
157	0	1	1	0	1	0	0	1	1	0	1	845
158	0	1	1	0	1	0	0	1	1	1	0	846
159	0	1	1	0	1	0	1	0	0	1	1	851
160	0	1	1	0	1	0	1	0	1	0	1	853
161	0	1	1	0	1	0	1	0	1	1	0	854
162	0	1	1	0	1	0	1	1	0	0	1	857
163	0	1	1	0	1	0	1	1	0	1	0	858
164	0	1	1	0	1	0	1	1	1	0	0	860
165	0	1	1	0	1	1	0	0	0	1	1	867
166	0	1	1	0	1	1	0	0	1	0	1	869
167	0	1	1	0	1	1	0	0	1	1	0	870
168	0	1	1	0	1	1	0	1	0	0	1	873
169	0	1	1	0	1	1	0	1	0	1	0	874
170	0	1	1	0	1	1	0	1	1	0	0	876
171	0	1	1	0	1	1	1	0	0	0	1	881
172	0	1	1	0	1	1	1	0	0	1	0	882
173	0	1	1	0	1	1	1	0	1	0	0	884
174	0	1	1	0	1	1	1	1	0	0	0	888
175	0	1	1	1	0	0	0	0	1	1	1	903
176	0	1	1	1	0	0	0	1	0	1	1	907
177	0	1	1	1	0	0	0	1	1	0	1	909
178	0	1	1	1	0	0	0	1	1	1	0	910
179	0	1	1	1	0	0	1	0	0	1	1	915
180	0	1	1	1	0	0	1	0	1	0	1	917
181	0	1	1	1	0	0	1	0	1	1	0	918
182	0	1	1	1	0	0	1	1	0	0	1	921
183	0	1	1	1	0	0	1	1	0	1	0	922
184	0	1	1	1	0	0	1	1	1	0	0	924
185	0	1	1	1	0	1	0	0	0	1	1	931
186	0	1	1	1	0	1	0	0	1	0	1	933
187	0	1	1	1	0	1	0	0	1	1	0	934
188	0	1	1	1	0	1	0	1	0	0	1	937
189	0	1	1	1	0	1	0	1	0	1	0	938
190	0	1	1	1	0	1	0	1	1	0	0	940
191	0	1	1	1	0	1	1	0	0	0	1	945
192	0	1	1	1	0	1	1	0	0	1	0	946
193	0	1	1	1	0	1	1	0	1	0	0	948
194	0	1	1	1	0	1	1	1	0	0	0	952
195	0	1	1	1	1	0	0	0	0	1	1	963
196	0	1	1	1	1	0	0	0	1	0	1	965
197	0	1	1	1	1	0	0	0	1	1	0	966

Table 10.1 Continues

Sl. No.	a	b	c	d	e	f	g	h	i	j	k	Binary value		Sl. No.	a	b	c	d	e	f	g	h	i	j	k	Binary value
199	0	1	1	1	1	0	0	1	0	1	0	970		231	1	0	0	1	0	0	0	1	1	1	1	1167
198	0	1	1	1	1	0	0	1	0	0	1	969		232	1	0	0	1	0	0	1	0	1	1	1	1175
200	0	1	1	1	1	0	0	1	1	0	0	972		233	1	0	0	1	0	0	1	1	0	1	1	1179
201	0	1	1	1	1	0	1	0	0	0	1	977		234	1	0	0	1	0	0	1	1	1	0	1	1181
202	0	1	1	1	1	0	1	0	0	1	0	978		235	1	0	0	1	0	0	1	1	1	1	0	1182
203	0	1	1	1	1	0	1	0	1	0	0	980		236	1	0	0	1	0	1	0	0	1	1	1	1191
204	0	1	1	1	1	0	1	1	0	0	0	984		237	1	0	0	1	0	1	0	1	0	1	1	1195
205	0	1	1	1	1	1	0	0	0	0	1	993		238	1	0	0	1	0	1	0	1	1	0	1	1197
206	0	1	1	1	1	1	0	0	0	1	0	994		239	1	0	0	1	0	1	0	1	1	1	0	1198
207	0	1	1	1	1	1	0	0	1	0	0	996		240	1	0	0	1	0	1	1	0	0	1	1	1203
208	0	1	1	1	1	1	0	1	0	0	0	1000		241	1	0	0	1	0	1	1	0	1	0	1	1205
209	0	1	1	1	1	1	1	0	0	0	0	1008		242	1	0	0	1	0	1	1	0	1	1	0	1206
210	1	0	0	0	0	0	1	1	1	1	1	1055		243	1	0	0	1	0	1	1	1	0	0	1	1209
211	1	0	0	0	0	1	0	1	1	1	1	1071		244	1	0	0	1	0	1	1	1	0	1	0	1210
212	1	0	0	0	0	1	1	0	1	1	1	1079		245	1	0	0	1	0	1	1	1	1	0	0	1212
213	1	0	0	0	0	1	1	1	0	1	1	1083		246	1	0	0	1	1	0	0	0	1	1	1	1223
214	1	0	0	0	0	1	1	1	1	0	1	1085		247	1	0	0	1	1	0	0	1	0	1	1	1227
215	1	0	0	0	0	1	1	1	1	1	0	1086		248	1	0	0	1	1	0	0	1	1	0	1	1229
216	1	0	0	0	1	0	0	1	1	1	1	1103		249	1	0	0	1	1	0	0	1	1	1	0	1230
217	1	0	0	0	1	0	1	0	1	1	1	1111		250	1	0	0	1	1	0	1	0	0	1	1	1235
218	1	0	0	0	1	0	1	1	0	1	1	1115		251	1	0	0	1	1	0	1	0	1	0	1	1237
219	1	0	0	0	1	0	1	1	1	0	1	1117		252	1	0	0	1	1	0	1	0	1	1	0	1238
220	1	0	0	0	1	0	1	1	1	1	0	1118		253	1	0	0	1	1	0	1	1	0	0	1	1241
221	1	0	0	0	1	1	0	0	1	1	1	1127		254	1	0	0	1	1	0	1	1	0	1	0	1242
222	1	0	0	0	1	1	0	1	0	1	1	1131		255	1	0	0	1	1	0	1	1	1	0	0	1244
223	1	0	0	0	1	1	0	1	1	0	1	1133		256	1	0	0	1	1	1	0	0	0	1	1	1251
224	1	0	0	0	1	1	0	1	1	1	0	1134		257	1	0	0	1	1	1	0	0	1	0	1	1253
225	1	0	0	0	1	1	1	0	0	1	1	1139		258	1	0	0	1	1	1	0	0	1	1	0	1254
226	1	0	0	0	1	1	1	0	1	0	1	1141		259	1	0	0	1	1	1	0	1	0	0	1	1257
227	1	0	0	0	1	1	1	0	1	1	0	1142		260	1	0	0	1	1	1	0	1	0	1	0	1258
228	1	0	0	0	1	1	1	1	0	0	1	1145		261	1	0	0	1	1	1	0	1	1	0	0	1260
229	1	0	0	0	1	1	1	1	0	1	0	1146		262	1	0	0	1	1	1	1	0	0	0	1	1265
230	1	0	0	0	1	1	1	1	1	0	0	1148		263	1	0	0	1	1	1	1	0	0	1	0	1266

Table 10.1 Continues

Sl.	Scientists											Binary	Sl.	Scientists											Binary
No.	a	b	c	d	e	f	g	h	i	j	k	value	No.	a	b	c	d	e	f	g	h	i	j	k	value
264	1	0	0	1	1	1	1	0	1	0	0	1268	297	1	0	1	0	1	1	1	0	0	0	1	1393
265	1	0	0	1	1	1	1	1	0	0	0	1272	298	1	0	1	0	1	1	1	0	0	1	0	1394
266	1	0	1	0	0	0	0	1	1	1	1	1295	299	1	0	1	0	1	1	1	0	1	0	0	1396
267	1	0	1	0	0	0	1	0	1	1	1	1303	300	1	0	1	0	1	1	1	1	0	0	0	1400
268	1	0	1	0	0	0	1	1	0	1	1	1307	301	1	0	1	1	0	0	0	0	1	1	1	1415
269	1	0	1	0	0	0	1	1	1	0	1	1309	302	1	0	1	1	0	0	0	1	0	1	1	1419
270	1	0	1	0	0	0	1	1	1	1	0	1310	303	1	0	1	1	0	0	0	1	1	0	1	1421
271	1	0	1	0	0	1	0	0	1	1	1	1319	304	1	0	1	1	0	0	0	1	1	1	0	1422
272	1	0	1	0	0	1	0	1	0	1	1	1323	305	1	0	1	1	0	0	1	0	0	1	1	1427
273	1	0	1	0	0	1	0	1	1	0	1	1325	306	1	0	1	1	0	0	1	0	1	0	1	1429
274	1	0	1	0	0	1	0	1	1	1	0	1326	307	1	0	1	1	0	0	1	0	1	1	0	1430
275	1	0	1	0	0	1	1	0	0	1	1	1331	308	1	0	1	1	0	0	1	1	0	0	1	1433
276	1	0	1	0	0	1	1	0	1	0	1	1333	309	1	0	1	1	0	0	1	1	0	1	0	1434
277	1	0	1	0	0	1	1	0	1	1	0	1334	310	1	0	1	1	0	0	1	1	1	0	0	1436
278	1	0	1	0	0	1	1	1	0	0	1	1337	311	1	0	1	1	0	1	0	0	0	1	1	1443
279	1	0	1	0	0	1	1	1	0	1	0	1338	312	1	0	1	1	0	1	0	0	1	0	1	1445
280	1	0	1	0	0	1	1	1	1	0	0	1340	313	1	0	1	1	0	1	0	0	1	1	0	1446
281	1	0	1	0	1	0	0	0	1	1	1	1351	314	1	0	1	1	0	1	0	1	0	0	1	1449
282	1	0	1	0	1	0	0	1	0	1	1	1355	315	1	0	1	1	0	1	0	1	0	1	0	1450
283	1	0	1	0	1	0	0	1	1	0	1	1357	316	1	0	1	1	0	1	0	1	1	0	0	1452
284	1	0	1	0	1	0	0	1	1	1	0	1358	317	1	0	1	1	0	1	1	0	0	0	1	1457
285	1	0	1	0	1	0	1	0	0	1	1	1363	318	1	0	1	1	0	1	1	0	0	1	0	1458
286	1	0	1	0	1	0	1	0	1	0	1	1365	319	1	0	1	1	0	1	1	0	1	0	0	1460
287	1	0	1	0	1	0	1	0	1	1	0	1366	320	1	0	1	1	0	1	1	1	0	0	0	1464
288	1	0	1	0	1	0	1	1	0	0	1	1369	321	1	0	1	1	1	0	0	0	0	1	1	1475
289	1	0	1	0	1	0	1	1	0	1	0	1370	322	1	0	1	1	1	0	0	0	1	0	1	1477
290	1	0	1	0	1	0	1	1	1	0	0	1372	323	1	0	1	1	1	0	0	0	1	1	0	1478
291	1	0	1	0	1	1	0	0	0	1	1	1379	324	1	0	1	1	1	0	0	1	0	0	1	1481
292	1	0	1	0	1	1	0	0	1	0	1	1381	325	1	0	1	1	1	0	0	1	0	1	0	1482
293	1	0	1	0	1	1	0	0	1	1	0	1382	326	1	0	1	1	1	0	0	1	1	0	0	1484
294	1	0	1	0	1	1	0	1	0	0	1	1385	327	1	0	1	1	1	0	1	0	0	0	1	1489
295	1	0	1	0	1	1	0	1	0	1	0	1386	328	1	0	1	1	1	0	1	0	0	1	0	1490
296	1	0	1	0	1	1	0	1	1	0	0	1388	329	1	0	1	1	1	0	1	0	1	0	0	1492

Table 10.1 Continues

Sl.	Scientists											Binary		Sl.	Scientists											Binary
No.	a	b	c	d	e	f	g	h	i	j	k	value		No.	a	b	c	d	e	f	g	h	i	j	k	value
330	1	0	1	1	1	0	1	1	0	0	0	1496		363	1	1	0	0	1	1	0	0	1	1	0	1638
331	1	0	1	1	1	1	0	0	0	0	1	1505		364	1	1	0	0	1	1	0	1	0	0	1	1641
332	1	0	1	1	1	1	0	0	0	1	0	1506		365	1	1	0	0	1	1	0	1	0	1	0	1642
333	1	0	1	1	1	1	0	0	1	0	0	1508		366	1	1	0	0	1	1	0	1	1	0	0	1644
334	1	0	1	1	1	1	0	1	0	0	0	1512		367	1	1	0	0	1	1	1	0	0	0	1	1649
335	1	0	1	1	1	1	1	0	0	0	0	1520		368	1	1	0	0	1	1	1	0	0	1	0	1650
336	1	1	0	0	0	0	0	1	1	1	1	1551		369	1	1	0	0	1	1	1	0	1	0	0	1652
337	1	1	0	0	0	0	1	0	1	1	1	1559		370	1	1	0	0	1	1	1	1	0	0	0	1656
338	1	1	0	0	0	0	1	1	0	1	1	1563		371	1	1	0	1	0	0	0	0	1	1	1	1671
339	1	1	0	0	0	0	1	1	1	0	1	1565		372	1	1	0	1	0	0	0	1	0	1	1	1675
340	1	1	0	0	0	0	1	1	1	1	0	1566		373	1	1	0	1	0	0	0	1	1	0	1	1677
341	1	1	0	0	0	1	0	0	1	1	1	1575		374	1	1	0	1	0	0	0	1	1	1	0	1678
342	1	1	0	0	0	1	0	1	0	1	1	1579		375	1	1	0	1	0	0	1	0	0	1	1	1683
343	1	1	0	0	0	1	0	1	1	0	1	1581		376	1	1	0	1	0	0	1	0	1	0	1	1685
344	1	1	0	0	0	1	0	1	1	1	0	1582		377	1	1	0	1	0	0	1	0	1	1	0	1686
345	1	1	0	0	0	1	1	0	0	1	1	1587		378	1	1	0	1	0	0	1	1	0	0	1	1689
346	1	1	0	0	0	1	1	0	1	0	1	1589		379	1	1	0	1	0	0	1	1	0	1	0	1690
347	1	1	0	0	0	1	1	0	1	1	0	1590		380	1	1	0	1	0	0	1	1	1	0	0	1692
348	1	1	0	0	0	1	1	1	0	0	1	1593		381	1	1	0	1	0	1	0	0	0	1	1	1699
349	1	1	0	0	0	1	1	1	0	1	0	1594		382	1	1	0	1	0	1	0	0	1	0	1	1701
350	1	1	0	0	0	1	1	1	1	0	0	1596		383	1	1	0	1	0	1	0	0	1	1	0	1702
351	1	1	0	0	1	0	0	0	1	1	1	1607		384	1	1	0	1	0	1	0	1	0	0	1	1705
352	1	1	0	0	1	0	0	1	0	1	1	1611		385	1	1	0	1	0	1	0	1	0	1	0	1706
353	1	1	0	0	1	0	0	1	1	0	1	1613		386	1	1	0	1	0	1	0	1	1	0	0	1708
354	1	1	0	0	1	0	0	1	1	1	0	1614		387	1	1	0	1	0	1	1	0	0	0	1	1713
355	1	1	0	0	1	0	1	0	0	1	1	1619		388	1	1	0	1	0	1	1	0	0	1	0	1714
356	1	1	0	0	1	0	1	0	1	0	1	1621		389	1	1	0	1	0	1	1	0	1	0	0	1716
357	1	1	0	0	1	0	1	0	1	1	0	1622		390	1	1	0	1	0	1	1	1	0	0	0	1720
358	1	1	0	0	1	0	1	1	0	0	1	1625		391	1	1	0	1	1	0	0	0	0	1	1	1731
359	1	1	0	0	1	0	1	1	0	1	0	1626		392	1	1	0	1	1	0	0	0	1	0	1	1733
360	1	1	0	0	1	0	1	1	1	0	0	1628		393	1	1	0	1	1	0	0	0	1	1	0	1734
361	1	1	0	0	1	1	0	0	0	1	1	1635		394	1	1	0	1	1	0	0	1	0	0	1	1737
362	1	1	0	0	1	1	0	0	1	0	1	1637		395	1	1	0	1	1	0	0	1	0	1	0	1738

Table 10.1 Continues

Sl.	Scientists											Binary		Sl.	Scientists											Binary
No.	a	b	c	d	e	f	g	h	i	j	k	value		No.	a	b	c	d	e	f	g	h	i	j	k	value
396	1	1	0	1	1	0	0	1	1	0	0	1740		429	1	1	1	0	1	0	0	1	0	0	1	1865
397	1	1	0	1	1	0	1	0	0	0	1	1745		430	1	1	1	0	1	0	0	1	0	1	0	1866
398	1	1	0	1	1	0	1	0	0	1	0	1746		431	1	1	1	0	1	0	0	1	1	0	0	1868
399	1	1	0	1	1	0	1	0	1	0	0	1748		432	1	1	1	0	1	0	1	0	0	0	1	1873
400	1	1	0	1	1	0	1	1	0	0	0	1752		433	1	1	1	0	1	0	1	0	0	1	0	1874
401	1	1	0	1	1	1	0	0	0	0	1	1761		434	1	1	1	0	1	0	1	0	1	0	0	1876
402	1	1	0	1	1	1	0	0	0	1	0	1762		435	1	1	1	0	1	0	1	1	0	0	0	1880
403	1	1	0	1	1	1	0	0	1	0	0	1764		436	1	1	1	0	1	1	0	0	0	0	1	1889
404	1	1	0	1	1	1	0	1	0	0	0	1768		437	1	1	1	0	1	1	0	0	0	1	0	1890
405	1	1	0	1	1	1	1	0	0	0	0	1776		438	1	1	1	0	1	1	0	0	1	0	0	1892
406	1	1	1	0	0	0	0	0	1	1	1	1799		439	1	1	1	0	1	1	0	1	0	0	0	1896
407	1	1	1	0	0	0	0	1	0	1	1	1803		440	1	1	1	0	1	1	1	0	0	0	0	1904
408	1	1	1	0	0	0	0	1	1	0	1	1805		441	1	1	1	1	0	0	0	0	0	1	1	1923
409	1	1	1	0	0	0	0	1	1	1	0	1806		442	1	1	1	1	0	0	0	0	1	0	1	1925
410	1	1	1	0	0	0	1	0	0	1	1	1811		443	1	1	1	1	0	0	0	0	1	1	0	1926
411	1	1	1	0	0	0	1	0	1	0	1	1813		444	1	1	1	1	0	0	0	1	0	0	1	1929
412	1	1	1	0	0	0	1	0	1	1	0	1814		445	1	1	1	1	0	0	0	1	0	1	0	1930
413	1	1	1	0	0	0	1	1	0	0	1	1817		446	1	1	1	1	0	0	0	1	1	0	0	1932
414	1	1	1	0	0	0	1	1	0	1	0	1818		447	1	1	1	1	0	0	1	0	0	0	1	1937
415	1	1	1	0	0	0	1	1	1	0	0	1820		448	1	1	1	1	0	0	1	0	0	1	0	1938
416	1	1	1	0	0	1	0	0	0	1	1	1827		449	1	1	1	1	0	0	1	0	1	0	0	1940
417	1	1	1	0	0	1	0	0	1	0	1	1829		450	1	1	1	1	0	0	1	1	0	0	0	1944
418	1	1	1	0	0	1	0	0	1	1	0	1830		451	1	1	1	1	0	1	0	0	0	0	1	1953
419	1	1	1	0	0	1	0	1	0	0	1	1833		452	1	1	1	1	0	1	0	0	0	1	0	1954
420	1	1	1	0	0	1	0	1	0	1	0	1834		453	1	1	1	1	0	1	0	0	1	0	0	1956
421	1	1	1	0	0	1	0	1	1	0	0	1836		454	1	1	1	1	0	1	0	1	0	0	0	1960
422	1	1	1	0	0	1	1	0	0	0	1	1841		455	1	1	1	1	0	1	1	0	0	0	0	1968
423	1	1	1	0	0	1	1	0	0	1	0	1842		456	1	1	1	1	1	0	0	0	0	0	1	1985
424	1	1	1	0	0	1	1	0	1	0	0	1844		457	1	1	1	1	1	0	0	0	0	1	0	1986
425	1	1	1	0	0	1	1	1	0	0	0	1848		458	1	1	1	1	1	0	0	0	1	0	0	1988
426	1	1	1	0	1	0	0	0	0	1	1	1859		459	1	1	1	1	1	0	0	1	0	0	0	1992
427	1	1	1	0	1	0	0	0	1	0	1	1861		460	1	1	1	1	1	0	1	0	0	0	0	2000
428	1	1	1	0	1	0	0	0	1	1	0	1862		461	1	1	1	1	1	1	0	0	0	0	0	2016

APPENDIX 2

The Extended Euclidean Algorithm

Suppose a and b are positive integers and d be their greatest common divisor. We know that the g.c.d can be written as a linear combination of the numbers. S, there exists integers x and y, such that,

$$ax + by = d \tag{10.1}$$

It may be noted that, except in some trivial cases, x and y will be of opposite signs. If x and y satisfies equation (10.1), so is $(x+qb)$ and $(y-qa)$, for any integer q. So, one can always find integers x any y, with $x > 0$ and $y < 0$, which satisfies the equation (10.1).

The *Extended Euclidean Algorithm* will calculate d, and also two integers x and y, such that $ax+by = d$ at the same time. This explains why the resulting procedure is known as the Extended Euclidean Algorithm. The version of the algorithm we present here is the creation of D. E. Knuth, author of the famous book *The Art of Computer Programming.* The Algorithm can be found in volume 2 of the series; (see Knuth [40]. section 4.5.2, algorithm X.)

Algorithm 10.1 (Extended Euclidean Algorithm)
Input : Two positive integers a and b.
Output: Three integers d, x, and y such that equation (10.1) holds good.

Step 1. *Initialize* $x = 0, y = 1$

$c = a, d = b$

Step 2. *Repeat*

$r = c \pmod d$

$q = (c - r)/d$

if ($r = 0$) *GO TO Step 3.*

$t = x$

$x = y - x * q$

$y = t$

$c = d$

$d = r$

Step 3. $y = (d - a * x)/b$

Step 4. *The numbers x and y satisfies*

$ax + by = d = G.C.D(a, b)$

If G.C.D$(a, b) = 1$, then $ax + by = d$ becomes $ax \equiv 1 \pmod b$ and $by \equiv 1 \pmod a$. So, $a^{-1} \equiv x \pmod b$, as well as $b^{-1} \equiv y \pmod a$. We can use the above algorithm to find out the inverse, whenever it exists.

Example 10.1
Let us find the inverse of $655 \pmod{1234}, 655^{-1} \pmod{1234}$

The following table shows the values of the variables r, q, and x at 3^{rd} line in each iteration of Step 2. 2
Step 3 evaluates $y = 341$, which is the inverse of $655 \pmod{1234}$.

Table 10.2: Illustration of Extended Euclidean Algorithm

Iteration Number	Remainders (r)	Quotients (q)	(x)
1	579	1	0
2	76	1	1
3	47	7	-1
4	29	1	8
5	18	1	-9
6	11	1	17
7	7	1	-26
8	4	1	43
9	3	1	-69
10	1	1	112
11	0	3	-181

4

APPENDIX 3

2 List of Research Papers

Published Papers :

1. Uniform Secret Sharing Schemes for (2, n) Threshold
   Using Visual Cryptography:
   International Journal of Information Processing,
   Volume 2, Number 4, 2008 pp 82- 87.

2. International Conference held at I.I.T., Kanpur. The paper is available in the web-site of the conference at pages: 33 to 37. The URL is

   http://www.security.iitk.ac.in/hack.in/2009/repository/proceedings_hack.in.pdf

Accepted Papers:

3. An Efficient Secret Sharing Scheme for n out of n scheme
   using POB-number system:
   Journal of Discrete Mathematical Sciences and Cryptography

4. An Effective Secret Sharing Scheme for n out of n scheme
   using modified Visual cryptography:
   Journal of Computer Science

Communicated papers:

5. An Efficient Secret Sharing Scheme for $(n-1, n)$ threshold
   using Visual cryptography:
   International Journal of Information Processing.

APPENDIX 4

SYNOPSIS of the Ph. D. thesis

Submitted by
A. Sreekumar, Research Scholar (Part-time),
Department of Computer Applications,
Cochin University of Science and Technology,

Under the guidance of
Professor, Dr. S. Babusundar

Topic: **CRYPTOGRAPHY**

Title: **Secret Sharing Schemes using Visual Cryptography**

1. Introduction

Handling secret has been an issue of prominence from the time human beings started to live together. Important things and messages have been always there to be preserved and protected from possible misuse or loss. Some time 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. Some of the formulae of vital combinations of medicinal plants or roots or leaves, in

Ayurveda were known to a single person in a family. When he becomes old enough, he would rather share the secret formula to a chosen person from the family, or from among his disciples. There were times when the person with the secret dies before he could share the secret. Probably, similar incidents might have made the genius of those era to think of sharing the secrets with more than one person so that in the event of death of the present custodian, there will be at least one other person who knows the secret.

Secret sharing in other forms were prevailing in the past, for other reasons also. Secrets were divided into number of pieces and given to the same number of people. To ensure unity among the participating people, the head of the family would share the information with respect to wealth among his children and insist that after his death, they all should join together to inherit the wealth.

To test the valor of the youth of a nation, a king, would hide treasure in some place in his kingdom and information about it would be placed in pieces at different places of varying grades of difficulty to reach. Only the brave and the intelligent would reach the treasure.

Military and defense secrets have been the subject matter for secret sharing in the past as well as in the modern days. Secret sharing is a very hot area of research in Computer Science in the recent past. Digital media has replaced almost all forms of communication and information preservation and processing. Security in digital media has been a matter of serious concern. This has resulted in the development of encryption and cryptography. Uniform secret sharing schemes form a part of this large study.

1.1 Definition: A Secret sharing scheme is a method of dividing a secret information into two or more pieces, with or without modifications, and retrieving the information by combining all or predefined sub collection of pieces.

The pieces of information are called **shares** and the process responsible for the division is called **dealer**. A predefined sub collection of shares which contains the whole secret in some form is called an **allowed coalition**. The process responsible for the recovery of the secret information from an allowed coalition is called a **combiner**.

A share contains, logically, a part of the information, but will be of no use. Thus no single share is of any threat to the confidentiality of the secret information. It is also envisaged that after the dealer process is over, the original information can be destroyed forever. This would mean that even the person responsible for the dealer process will not be a threat, thereafter. The secret information is recovered from any allowed coalition using the recovery process called combiner. The combiner would be able to recover the secret information, only if, all shares in the allowed coalition is present and not with any fewer number of shares. Thus, in an allowed coalition, each member share is equally important such that without anyone of them, the secret information cannot be accessed.

Allowed coalition is also referred in the literature by other names too, such as, **authentic collection**, **qualified collection** or **authorized set**. We, in our work, preferred to call the sub collection of shares as allowed coalition.

Secret Sharing is an important tool in Security and Cryptography. In many cases there is a single master key that provides the access to important secret information. Therefore, it would be desirable to keep the master key in a safe place to avoid accidental and malicious exposure. This scheme is unreliable: if master key is lost or destroyed, then all information accessed by the master key is no longer available. A possible solution would be that of storing copies of the key in different safe places or giving copies to trusted people. In such a case the system becomes more vulnerable to security breaches or betrayal [53], [30]. A better solution would be, breaking the master key into pieces in such a way that only the concurrence of certain predefined trusted people can recover it. This has proven to be an important tool in management of cryptographic keys and multi-party secure protocols (see for example [33]).

As a solution to this problem, Blakley [9] and Shamir [53] introduced $(k,\ n)$ threshold schemes. A $(k,\ n)$ threshold scheme allows a secret to be shared among n participants, in such a way that, any k of them can recover the secret, but $k-1$, or fewer, have absolutely no information on the secret.

Ito, Saito, and Nishizeki [36] described a more general method of secret sharing. An access structure is a specification of all subsets of participants who can recover the secret and it is said to be monotone if any set which contains a subset that can recover the secret, can itself recover the secret. Ito, Saito, and Nishizeki gave a methodology to realize secret sharing schemes for arbitrary monotone access structures. Subsequently, Benaloh and Leichter [5] gave a simpler and more efficient way to realize such schemes.

An important issue in the implementation of secret sharing scheme is the size of the shares distributed to the participants, since the security of a system degrades as the amount of the information that must be kept secret increases. So the size of the shares given to the participants is a key point in the design of secret sharing schemes. Therefore, one of the main parameters in secret sharing is, the **average information rate** ρ, of the scheme, which is defined as the ratio between the average length (in bits) of the shares given to the participants and the length of the secret. Unfortunately, in all secret sharing schemes the size of the shares cannot be less than the size of the secret, and so the information rate cannot be less than one. Moreover, there are access structures, for which, any corresponding secret sharing scheme must give to some participant a share of size strictly bigger than the secret size. Secret sharing schemes with information rate equal to one are called **ideal**. A secret sharing scheme is called efficient if the total length of the n shares is polynomial in n.

2. Model of secret sharing

A common model of secret sharing has two phases. In the initialization phase, a trusted entity - the dealer, divides the secret information into shares and distributes the shares by secure means. In the reconstruction phase one of the allowed coalition submit their shares to a combiner, who reconstructs the secret. It is assumed that the combiner is an algorithm which only performs the task of reconstructing the secret. Various Secret Sharing Schemes have been proposed since 1979. The following are some of the known schemes:

1. Blakley's scheme using projective spaces over finite fields GF(q)
2. Simmons' scheme in terms of affine spaces
3. Shamir's scheme based on polynomial interpolation over finite fields.

In most of the schemes, when a great number of participants are involved, the scheme will become impractical. In the traditional

Secret Sharing Schemes, a shared secret information cannot be revealed without any cryptographic computations.

2.1 Visual Cryptography There are various connections between combinatorial structures and secret sharing. For example, a (2, 3) threshold scheme can be implemented based on a small Latin square. In 1994, Naor and Shamir invented a new type of secret sharing scheme, called Visual Cryptography scheme [48]. In secret sharing schemes using Visual Cryptography, a shared secret information (printed text, handwritten notes, pictures, etc.) can be revealed without any cryptographic computations. For example, in a $(k,\ n)$ visual cryptography scheme, a dealer encodes a secret into n shares and gives each participant a share, where each share is a transparency. The secret is visible if any k (or more) of participants stack their transparencies together, but none can see the shared secret if fewer than k transparencies are stacked together.

3. Problem specification

Secret sharing is one of the cryptographic techniques providing security measures to protect information. Due to difficulty of finding a general solution, those problems have been studied in several particular cases, and several sharing schemes have been proposed. So this particular work focuses on a generalized scheme, for at least some values of k, which works with any number of participants.

4. Objective and scope of this Research

Most of the business organizations need to protect data from disclosure. As the world is more connected by computers, the hackers, power abusers have also increased, and most organizations are afraid to store data in a computer. So there is a need of a method to distribute the data at several places and destroy the original one. When a need of original data arises, it could be reconstructed from the distributed shares. The primitive objective of this research is to provide a solution to this problem.

5. Contribution of the Thesis

The research work provides a better mechanism for secure storage of information. The thesis work proceeds into three phases.

1. The first phase deals with studies and findings in the area of secret sharing.
2. The second phase of the work relates to investigating new structures suitable for specific applications.
3. The third phase deals with the mathematical proofs of the new findings.

6. Design of the scheme

In this research work, we considered a special type of codes, called Uniform Codes to propose sharing schemes. A string of 0s and 1s is called a uniform code, if the number of 1's is either equal to or one more than the number of 0's. For example, 011010 and 1101001 are uniform codes where as 001 and 110110 are not. It can be seen that, if the length of a binary string is w, then the number of codes having length w, and having t 1's is $\binom{\mathrm{w}}{\mathrm{t}}$. For a given w, this number is maximum when $t = \lfloor \frac{n}{2} \rfloor$, the integer part of $\frac{n}{2}$. So the maximum number of codes with a given length occurs when they are uniform. Four efficient threshold schemes are proposed based on Modified Visual Cryptography introduced in 2002. All the schemes are based on the uniform codes. The first scheme proposed is an efficient (2, n) threshold scheme. This scheme provides an efficient way to hide a secret information in different shares, in which the size of the shares is just in $O(log_2\ n)$ times the original secret size, where n is the number of participants. The second scheme is a (3, n) threshold scheme in which the size of the shares is just in $O(n)$ times the original secret size, where n is the number of participants. The third scheme is $(n-1,\ n)$ threshold scheme in which the size of the share is in $O(n/2)$. We have generalized the concept of Uniform code by relaxing the constraints, and introduced a new number system, called *Permutation Ordered Number System* (or POB-Number system). The system has two parameters. We have developed some algorithms for efficiently representing the usual numbers in the new system, and vice-versa. Finally we found that a certain class of binary strings can be decomposed in the class of balanced strings, and Uniform Codes. By using the POB-Number system, we can represent Uniform codes and balanced strings effectively. We exploit this property, and developed an efficient sharing algorithm in which the size of the share is less than the size of the secret. We have come across the following finding: Let w be an even parity string and $n_1(w)$ denotes the number of 1's in a binary string w of length t. Then w can be written as $w = S_1 \oplus S_2 \oplus \ldots \oplus S_n,$, where, S_i is a Uniform Code, for each $i = 1,\ 2, \ldots,\ n$. Here $\oplus$ is the usual bitwise XOR operation. We have developed all the algorithms and illustrated them with appropriate examples. This scheme is very efficient, as the size of the share is less than the size of the original secret, in which we have a gain of 1/8.

7. Content of the thesis

The thesis is presented in 10 chapters. We have taken care to provide a good account of the literature survey and the theoretical background of the topic of study. All the details of the development of the newly proposed algorithms and the proofs of the claim are also included. Some of the algorithms have been presented, either in full or in parts, in conferences and journals. An account of these publications are also included.

The first chapter deals with the introduction. It contains the sketch of the development and progress of the topic of study.

The Second chapter deals with history and literature survey.

The Third chapter deals with the visual cryptography and its examples.

The Fourth chapter deals with modified cryptography.

The next four chapters deal with the solutions proposed by us, which is our contribution to this area of study. The findings are presented in conferences and others are either published or accepted for publication in journals. One of our research paper is published in the International Journal of Information Processing, Volume 2, Number 4, 2008 pp 82-87.

Another two papers are accepted for publication, and will be published within one month. A fifth paper is communicated for publication. The result is awaited. The details are included in the thesis

As a good by-product of this research work, we have developed a new number system. It is named as *Permutation Oriented Binary Number System* (**POB-number system**). In an International Conference at I.I.T., Kanpur, we have presented this part of the research work. The paper was one among the eleven selected papers out of a total of 40 research papers, submitted, in the areas of Cryptography and Network Security. We are happy to say that, our paper was ranked fourth among the 10 papers presented there. The paper is available in the web-site of the conference at pages: 33 to 37. The url is

http://www.security.iitk.ac.in/hack.in/2009/repository/proceedings_hack.in.pdf

The Ninth chapter deals with the most important result we have achieved. We have developed an algorithm, in which the secret could be shared among n participants with a single allowed coalition such that the size of the share is less the size of the secret message. The final chapter deals with the probable direction of future research work in this area.

Bibliography

[1] *C. Asmuth and J. Bloom*: A Modular Approach to Key Safeguarding, IEEE Transactions on Information Theory, vol.IT-29, no.2, 1983, pp. 208-210.

[2] *G. Ateniese, C. Blundo, A.D. Santis, and D. Atinson*: Constructions and Bounds for Visual Cryptography, Proceedings 23rd International Colloquium on Automata, Languages, and Programming (ICALP '96), 1099, 1996, pp. 416-428.

[3] *G. Ateniese, C. Blundo, A. D. Santis, and D. R. Stinson*: Visual Cryptography for General Access Structures, Information and Computation, vol. 129, no.2, 1996, pp. 86-106.

[4] *A. Beimel and B. Chor*: Universally Ideal Secret Sharing Schemes, Lecture Notes in Computer Science vol. 740, 1993, pp. 185-197.

[5] *J. C. Benaloh and J. Leichter*: Generalized Secret Sharing and Monotone Functions, Proceedings of Crypto '88, Advances in Cryptology, Lecture Notes in Computer Science, vol. 403, S. Goldwasser, Ed., Springer-Verlag, Berlin, 1990, pp. 27-35.

[6] *J. Benaloh*: Secret Sharing Homomorphisms - Keeping Shares of a Secret Secret, In Advances in Cryptology - CRYPTO '86, A. M. Odlyzko, Ed. 1987, vol. 263 of Lecture Notes in Computer Science, pp. 251-260, Springer-Verlag.

[7] *E. Bertinoro*: Secure and Selective Dissemination of XML Documents, ACM Transactions on Information System Security, vol. 5, No. 3, 2002, pp 290-331.

[8] *M. Bertilsson, I. Ingemarsson*: A Construction of Practical Secret Sharing Schemes using Linear Block Codes. In Proceedings AUSCRYPT '92, Springer Lecture Notes in Computer Science, vol. 718, pp. 67-79, 1993.

[9] *G. R. Blakley*: Safeguarding Cryptographic Keys, Proceeding of AFIPS 1979 National Computer Conference, vol. 48, New York, NY, June 1979, pp. 313-317.

[10] *B. Blakley, G. R. Blakley, A. H. Chan and J. L. Massey*: Threshold Schemes with Disenrollment. Lecture Notes in Computer Science 740, 1993, pp. 546-554.

[11] *G. R. Blakley and C. Meadows*: Security of Ramp Schemes, Proceeding of Crypto '84, Advances in Cryptology, Lecture notes in Computer Science, vol 196, 1985, G. R. Blakley and D. Chaum, Eds., Springer Verlag, pp 411-431.

[12] *C. Blundo, A. De Santis and U. Vaccaro*: Efficient Sharing of Many Secrets, Proceeding of STACS'93, Lecture Notes in Computer Science vol. 665, 1993, Springer Verlag, pp. 692-703.

[13] *C. Blundo, A. Cresti, A. De Santis and U. Vaccaro*: Fully Dynamic Secret Sharing Schemes. Theoretical Computer Science, vol. 165, no. 2, 1996, pp. 407-440.

[14] *C. Blundo, A. De Santis, L. Gargano and U. Vaccaro*: On the Information Rate of Secret Sharing Schemes, Lecture Notes in Computer Science vol. 740, 1993, pp. 149-169.

[15] *C. Blundo, A. De Santis, D. R. Stinson, and U. Vaccaro*: Graph Decomposition and Secret Sharing Schemes, Journal Cryptology. no. 8 1995, pp. 39-64.

[16] *C. Blundo and D. R. Stinson*: Anonymous Secret Sharing Schemes, Discrete Applied Mathematics, 77, 1997, pp 13-28.

[17] *E. F. Brickell*: Some ideal secret sharing schemes, Journal of Combin. Math. and Commbin. Comput. no. 9, 1989, pp 105-113.

[18] *E. F. Brickell and D. M. Davenport*: On Classification of Ideal Secret Sharing Schemes, Journal of Cryptology, vol 4, No. 2 1991, pp 123-134.

[19] *E. F. Brickell and D. R. Stinson*: Some Improved Bounds on the Information Rate Of Perfect Secret Sharing Schemes. Journal Cryptology vol. 5 no. 3, 1992, pp. 153-166.

[20] *E. F. Brickell and D. R. Stinson*: The Detection Of Cheaters In Threshold Schemes, SI AM J. on Discrete Math. no. 4, 1991, pp. 502-510.

[21] *R. Brinkman, J. M. Doumen and W. Jonker*: Using secret sharing for searching in encrypted data, In Workshop on Secure Data Management in a Connected World (SDM), 30 Aug 2004, Toronto, Canada. pp. 18-27. Lecture Notes in Computer Science 3178. Springer-Verlag. ISSN 0302-9743 ISBN 3-540-22983-3.

[22] *R. M. Capocelli, A. De Santis, L. Gargano and U. Vaccaro*: On the size of shares for secret sharing schemes. Journal Cryptology no. 6, 1993, pp. 157-168.

[23] *M. Carpentieri, A. De Santis and U. Vaccaro*: Size of shares and probability of cheating in threshold schemes, Presented at EUROCRYPT '93.

[24] *C. Chang, C. Tsait and T. Chen*: A New Scheme for Sharing Secret Colour Images in Computer Network, Proceeding of International Conference on Parallel and Distributed Systems, July 2000, pp 21-27.

[25] *Chin-Chen Chang and Tai-Xing Yu*: Sharing Secret Gray Image in Multiple Images, National Chung Cheng University, Taiwan, 2002.

[26] *D. Chen and D. R. Stinson*: Recent Results on Combinatorial Constructions for Threshold Schemes, Australasian Journal of Combinatorics, vol 1, 1990, pp. 29-48.

[27] *B. Chor, E. Kushilevitz*: Secret Sharing Over Infinite Domains, Journal of Cryptology, Vol 6, 1993, pp. 87-96

[28] *B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan*: Private Information Retrieval, IN FOCS, 1995, pp 41-50.

[29] *Chwei-Shyong Tsai, Chin-Chen Chang and Tung-Shou Chen*: Sharing Multiple Secrets in Digital Images, The Journal of Systems and Software, vol 64, 2002, pp.163-170.

[30] *D. Denning*: Cryptography and Data Security, Addison-Wesley, Reading, MA, 1983.

[31] *Y. Desmedt, A. De Santis, Y. Frankel, and M. Yung*: How to Share a Function Securely. In: Proceedings STOC '94, ACM Press, 1994, pp. 22-33.

[32] *S. Droste*: New Results on Visual Cryptography, Advances in Cryptology-CRYPTO'96, Lecture Notes in Computer Science, vol. 1109, 1996, pp. 401-415.

[33] *O. Goldreich, S. Micali, and A. Wigderson*: How to Play Any Mental Game, Proceeding of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, pp. 218-229.

[34] *R. Hwang and C. Chang*: Some Secret Sharing Schemes and their Applications, Ph. D. dissertation of the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, 1998.

[35] *I. Ingemarsson and G. J. Simmons*: A Protocol to set up Shared Secret Schemes Without the Assistance of a Mutually Trusted Party. Lecture Notes in Computer Science vol. 473, 1991, pp. 266-282.

[36] *M. Ito, A. Saito, and T. Nishizeki*: Secret Sharing Schemes Realizing General Access Structure, Proceeding of IEEE Global Telecommunications Conference, Globecom 87, Tokyo, Japan, 1987, pp. 99-102, . Journal version: Multiple Assignment Scheme for Sharing Secret, Journal Cryptology, vol 6, no. 6, 1993, pp. 15-20.

[37] *W. A. Jackson and K M. Martin and C. M. O'Keefe*: On Sharing Many Secrets, Lecture Notes in Computer Science 917, Advances in Cryptology, Proceedings of Asiacrypt'94, Springer Verlag, 1994, pp. 42-54.

[38] *W.A. Jackson and K. M. Martin*: Combinatorial Models for Perfect Secret Sharing Schemes, J. Corn-bin. Math. Combin. Comput., Vol. 28, 1998 pp 249-265.

[39] *E. D. Karnin, J. W. Greene and M. E. Hellman*: On Secret Sharing Systems, IEEE Transactions on Information Theory, vol.IT-29, no. 1, Jan 1983, pp 35-41.

[40] *D. E. Knuth*: The Art of Computer Programming, Vol. 2, Seminumerical algorithms,2nd edition, Addison-Wesley Publishing Company, Reading, Massachusetts, 1981.

[41] *S. Kothari*: Generalized Linear Threshold Scheme, Proceedings Crypto '84, Santa Barbara, CA (Aug 1984), 231-241. Published as Advances in Cryptology, ed. by Blakley and D. Chaum in Lecture Notes in Computer Science, vol. 196, ed. by G. Goos and J. Hartmains. Springer-Verlag, New York 1985.

[42] *Lawrence C. Washington , Wade Trappe*: Introduction to Cryptography: With Coding Theory ,Prentice Hall PTR, Upper Saddle River, NJ, 2002.

[43] *C. L. Liu*: Introduction to Combinatorial Mathematics, McGraw-Hill, New York, 1968.

[44] *K. M. Martin*: Discrete Structures in the Theory of Secret Sharing. Ph. D. Thesis, University of London, 1991.

[45] *K. M. Martin*: New Secret Sharing Schemes fron Old, Journal of Combin. Math. and Combin. Comput. no. 14, 1993, pp 65-77.

[46] *R. J. McEliece and D. V. Sarwate.*: On Sharing Secrets and Reed-Solomon Codes. Commun. of the ACM no. 24, 1981, pp. 583-584.

[47] *M. Mignotte*: How to share a secret, Cryptography Proceedings of the Workshop on Cryptography, Burg Feuerstein, 1982, volume 149 of LNCS, pp 371-375 Springer-Verlag, 1983.

[48] *M. Naor and A. Shamir*: Visual Cryptography, Advances in cryptology- EUROCRYPT94, Lecture Notes in Computer Science, vol. 950, 1995, pp. 1-12.

[49] *S. J. Phillips and N. C. Phillips*: Strongly Ideal Secret Sharing Schemes, J. Cryptology, Vol. 5 (1992), pp. 185-191.

[50] *T. Rabin and M. Ben-Or*: Verifiable Secret Sharing and Multiparty Protocols with Honest Majority. Proceedings 21st ACM Symp. on Theory of Computing, 1989, pp. 73-85.

[51] *P. J. Schellenberg and D. R. Stinson*: Threshold Schemes from Combinatorial Designs, J. Combin. Math. Combin. Comput., vol. 5, 1989, pp. 143-160.

[52] *P. D. Seymour*: On Secret-Sharing Matroids, J. Combin. Theory B vol. 56, 1992, pp. 69-73.

[53] *A. Shamir*: How to Share a Secret, Communications of the ACM, vol. 22, no. 11, Nov. 1979, pp. 612-613.

[54] *G. J. Simmons*: Robust Shared Secret Schemes or "How to be Sure You Have the Right Answer even though You don't know the question", Congressus Numerantium, vol. 8, 1989, pp 215-248.

[55] *G. J. Simmons*: Shared Secret and/or Shared Control Schemes, Lecture Notes in Computer Science vol. 537, 1991, pp. 216-241.

[56] *G. J Simmons, W. Jacjson and K. Martin*: The Geometry of Shared Secret Schemes. Bulletin of ICA vol. 1, 1991, pp. 71-88.

[57] *Dawn Xiaodong Song, David Wagner and Adrian Perrig*: Practical techniques for searchs on encrypted data, In IEEE Symposium on Security and Privacy, pp 44-55, 2000. http://citeseer.nj.nec.com/song00practical.html.

[58] *D. R. Stinson*: An Explication of Secret Sharing Schemes, Designs, Codes and Cryptography, vol. 2, 1992, pp 357-390.

[59] *D. R. Stinson*: New General Lower Bounds on the Information Rate of Secret Sharing Schemes. Lecture Notes in Computer Science, vol. 740, 1993, pp. 170-184.

[60] *D. R. Stinson*: Decomposition Constructions for Secret Sharing Schemes. IEEE Transactions on Inform. Theory, (40) 1994, pp 118-125.

[61] *D. R. Stinson and S. A. Vanstone*: A combinatorial approach to threshold schemes. SIAMJ. on Discrete Mathematics, 1(2):230-236, 1988.

[62] *M. Tompa and H. Woll*: How to share a Secret with Cheaters, Journal of Cryptography, vol. 1, no.2, 1988,pp 133-138

[63] *E. Verheul and H.V. Tilborg*: Constructions and Propertics of k out of n Visual Secret Sharing Schemes, Designs, Codes and Cryptography, vol. 11 no.2, 1997, pp. 179-196.