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

$$U_z = \begin{pmatrix} z \\ \lfloor \frac{z}{2} \rfloor \end{pmatrix} \tag{5.2}$$

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 $B = b_1b_2 \dots b_t$ *of length* t.
Output : n *blocks* $S_1, S_2, \dots, S_n$ *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 $A = a_1a_2 \dots a_t$ *and*
$B = b_1b_2 \dots b_t$ *of* t *blocks of* z *bits each.*
Output: The secret information $S = s_1s_2s_3 \dots s_t$.

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* $S = s_1s_2s_3 \dots s_t$.

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

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

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

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

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,

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

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

$$\frac{2^z}{z+1} \leq U_z \leq 2^{z-1} \tag{5.10}$$

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,

$$U_z \leq kB^z, \ for\ all\ z \geq n_0. \tag{5.11}$$

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 $4 \leq n \leq 9$, and in this case,
$\frac{6}{5}(\log_2 n) + 2 \geq 4.4 > z$.

If $z = 6$, then $11 \leq n \leq 20$, 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:

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

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 $\lfloor \frac{n}{2} \rfloor$ many pairs and one set left out. Let the secret be $B = B_1B_2B_3 \dots B_t$. 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* $S_1, S_2, \dots, S_n$*, 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* $S_1 S_2 \ldots S_{j-1} S_{j+1} \ldots S_n$, 2

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

Observe that S_j *is the missing share.* 4

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 1^{st}, 3^{rd}, 4^{th} and 5^{th} 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 2^{nd} 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.