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 $B = b_1 b_2 b_3 \ldots b_t$. 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 $B = b_1 b_2 \ldots b_t$, there exist solutions, $S_1, S_2, \ldots, S_n$, such that, $B = S_1 \oplus S_2 \oplus \ldots \oplus S_n$, where, $S_1, S_2, \ldots, S_n$ 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 $B_t = b_1 b_2 \dots b_t$ *of length* t.
Output : An even parity string $E_{t+1} = e_1 e_2 \dots e_{t+1}$
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* $E_{t+1} = e_1 e_2 \dots e_{t+1}$.

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 $E_{t+1} = e_1 e_2 \dots e_{t+1}$.
Output : Two blocks $S^{(1)}_{t+1} = s^{(1)}_1 s^{(1)}_2 \dots s^{(1)}_{t+1}$ *and*
$S^{(2)}_{t+1} = s^{(2)}_1 s^{(2)}_2 \dots s^{(2)}_{t+1}$ *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 $E_{t+1} = S_{t+1}^{(1)} \oplus S_{t+1}^{(2)}$, 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
2 1**01**0*1***010**1***0*10*1**0 (Here $*$ indicates null bits.) and by Step 3, S_1 is randomly set as

4 $$1110110001010010011101001001110$$

Finally by Step 4. of Algorithm 7.2,
6 $S_2 = S_1 \oplus B_{t+1} = 01110101000101010101100100111$
Recovery : Compute $S_1 \oplus S_2$ and get

8 $$B_t = 10011001010001110010001011010 01$$

Last bit is 1 and is deleted to get B : 10011 00101 00011 10010
10 00101 10100.

7.3 A (n, n) Construction

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

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

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

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 $B_t = b_1 b_2 \dots b_t$.

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

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

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, $S_2, S_3, \dots, S_{n-1}$. They are all random balanced strings. In Step 3, from the equation,

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

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.