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 $K = k_1k_2 \dots k_8$ 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 $K = K_1K_2 \dots K_8$.
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 $[1 \ldots 9]$.

Step 2. *The input string K is extended to T*
by inserting one bit at position r.
Compute the binary string $T = T_1 T_2 \ldots T_9$

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

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:

$$\begin{aligned} A &: 101010100\ 100110100 \\ B &: 011001010\ 001010101 \\ Compute\ T = A \oplus B &= 110011110\ 101100001 \end{aligned}$$

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 $K = K_1K_2K_3 \dots K_8$.
Output : n shares $S_1, S_2, \dots, S_n$ *of length 7 bits.*

Step 1. *Let* $A_1, A_2, \dots A_n$ *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 $r = \left\lceil \frac{V(A_2)+1}{14} \right\rceil = \lceil \frac{102}{14} \rceil = 8$.

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.