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, ..., $\binom{n}{r} - 1$, 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. 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
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,

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

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

$$0 \leq d_1 < d_2 < \dots < d_r \leq n-1 \tag{8.3}$$

be the binary digits of B, having value 1.

Then the formula (8.1) takes the form

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

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,

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

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:*

1. *Repeat* {
2. $j = j - 1$;
3. $p = \binom{j}{k}$;
4. *if* $(temp \geq p)$
5. $temp = temp - p$;
6. $b_j = 1$;
7. *else* $b_j = 0$;
8. } *Until* $(b_j = 1)$;
9. *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 $B = b_{n-1}b_{n-2}\ldots b_0$ be a binary string,*

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

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

Step 2. *Let* $done = 0$

1. *Repeat* {
2. *Print* B
3. *Let* $NoOfZeros = 0, i = 0$ *and* $j = 1$.
4. *while* $(b_j = 1$ *or* $b_i = 0)$ *do* {
5. *if* $(b_i = 0)$ $NoOfZeros = NoOfZeros + 1$;
6. *if* $(j = n - 1)$ $done = 1$;
7. $i = j$;
8. $j = j + 1$
9. }
10. $b_j = 1$;
11. $j = i - NoOfZeros$;
12. *while*$(i \geq j)$ *do* {
13. $b_i = 0, i = i - 1$
14. }
15. *while*$(i \geq 0)$ *do* {
16. $b_i = 1, i = i - 1$
17. }
18. } *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 $B = b_{n-1}b_{n-2}\ldots b_0$, 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 $b_{j-2} \dots b_0$ 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.