4.5) Compute MSB (for activation function – Algorithm 5)

* $\Pi_{MSB}(\{P_0, P_1\}, P_2)$:

⊙ Input : P_0 and P_1 hold $\langle a \rangle_0^{L-1}$ and $\langle a \rangle_0^{L-1}$ respectively

⊙ Output: P_0 and P_1 get $\langle MSB(a) \rangle_0^L$ and $\langle MSB(a) \rangle_1^L$

⊙ Common Randomness: P_0, P_1 hold a random bit β and random shares of 0 over L, denoted by u_0 and u_1 respectively.

⊙ Steps:

1) P_2 picks $r \xleftarrow{\$} \mathbb{Z}_{L-1}$. Next, P_2 generates $\langle r \rangle_j^{L-1}$, $\{\langle r[i] \rangle_j^P\}_i$, $\langle r[0] \rangle_j^L$, for $j \in \{0,1\}$ and sends to P_j.

2) For $j \in \{0,1\}$, P_j computes $\langle y \rangle_j^{L-1} = 2\langle a \rangle_j^{L-1}$ and $\langle v \rangle_j^{L-1} = \langle y \rangle^{L-1} + \langle r \rangle_j^{L-1}$

3) P_0, P_1 reconstruct v by exchanging shares.

4) P_0, P_1, P_2 call $\Pi_{PC}(\{P_0, P_1\}, P_2)$ with P_j, $j \in \{0,1\}$ having input $(\{\langle r[i] \rangle_j^P\}_{i \in [L]}, v, \beta)$ and P_2 learns β'.

5) P_2 generates $\langle \beta' \rangle_j^L$ and sends to P_j for $j \in \{0,1\}$

6) For $j \in \{0,1\}$, P_j executes step 7-8

7) $\langle \gamma \rangle_j^L = \langle \beta' \rangle_j^L + j\beta - 2\beta \langle \beta' \rangle_j^L$

8) $\langle \delta \rangle_j^L = \langle r[0] \rangle_j^L + j\,v[0] - 2\,v[0] \langle r[0] \rangle_j^L$

9) P_0, P_1, P_2 call $\Pi_{MatMul}(\{P_0, P_1\}, P_2)$ with $P_j, j \in \{0,1\}$ having input $(\langle\gamma\rangle_j^L, \langle\delta\rangle_j^L)$ and P_j learns $\langle\theta\rangle_j^L$.

10) For $j \in \{0,1\}$, P_j outputs

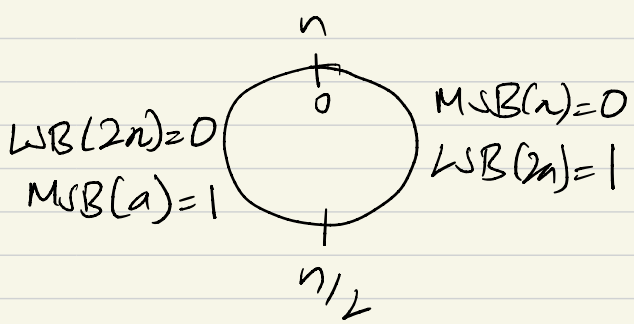
$$\langle\alpha\rangle_j^L = \langle\gamma\rangle_j^L + \langle\delta\rangle_j^L - 2\langle\theta\rangle_j^L + u_j$$

⊙ Explanation:

In the odd ring $\mathbb{Z}_{L-1}$, the MSB computation of a can be converted to $MSB(a) = LSB(y)$ where $y = 2a$.

In a group of order n, we have $MSB(a) = 1$ iff $a > n/2$ iff $a > 2a - n > 0$; if n is odd, then so is $(2a-n)$ and it follows that $MSB(a) = 1$ iff $LSB(2a) = 1$.

Let's take two elements in the group $a > n/2$ and $x < n/2$



Then, $MSB(x) = 0$

and, $2x > n \Rightarrow LSB(2x) = 0$ as $2x$ is even.

Hence, $LSB(2x) = MSB(x) = 0$

Again, $MSB(a) = 1$

and $2a > n \Rightarrow 2a \bmod n = 2a - n$ which is odd. Hence, $LSB(2a) = 1 = MSB(a)$

+ So, step 1 picks $x \in \mathbb{Z}_{L-1}$ and step 2 computes $y = 2a$ and $r = x + y$

+ Then, P_0, P_1 reconstruct r by exchanging shares.

+ P_0, P_1, P_2 invoke Π_{PC} to compare $x > r$ and then get output β'.

+ Step 7 gets the result of comparison $\gamma = \beta' \oplus \beta$. Note that γ is also the wrap bit i.e $wrap(x, y, L-1) = (x > r) = \gamma$

If $x > r$, then the sum r wraps around and $\gamma = 1$

If $x \not> r$, then the sum doesn't wrap around, $\gamma = 0$.

+ Step 8 computes $\delta = x[0] \oplus r[0]$

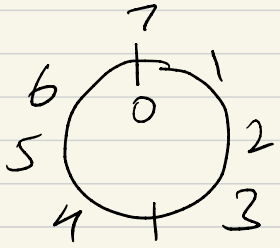
+ Step 9 privately computes $\gamma \cdot \beta$

+ Step 10 computes

$$LSB(y) = y[0] = \delta \oplus \gamma$$
$$= x[0] \oplus r[0] \oplus (x > r)$$

Let's examine the step 10 with $\mathbb{Z}_7$ group.

$\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$ - odd ring



Case 1: sum r=y+x doesn't wrap around $\Rightarrow Y=0$

$a = 1$ and $MSB(a) = 0$

$y = 2a = 2$ and $LSB(y) = 0$

So, we have $y = 2a$ and $LSB(y) = 0$ as y is always even

Now, two cases

i) x is even, $x = 4$

$r = x + y$ (even + even)

$= 4 + 2$

$= 6$ (even)

$\Rightarrow y[0] = x[0] \oplus r[0] \oplus Y$

$= 0$ (even) $\oplus$ 0 (even) $\oplus 0$

$= 0$ (even)

So, even = even $\oplus$ even

$0 = 0 \oplus 0$

ii) x is odd, $x = 3$

$r = x + y$ (odd + even)

$= 3 + 2$

$= 5$ (odd)

$\Rightarrow y[0] = x[0] \oplus r[0] \oplus Y$

$= 1$ (odd) $\oplus$ 1 (odd) $\oplus Y$

So, even = odd $\oplus$ odd $\Rightarrow 0 = 1 \oplus 1$

This means when $Y=0$, x and r both remain odd or even giving y as always even and hence $LSB(y) = 0$.

Case 2: sum r = x+n does wrap around ⇒ Y = 1

For this part, we have a = 2, y = 2n = 4

⇒ MSB(a) = LSB(y) = 0

Again, y is always even and hence LSB(y) = 0

Now, we have two cases

i) n is even, n = 6

$$r = x + y \quad (\text{even} + \text{even})$$
$$= 6 + 4$$
$$= 10 \bmod 7$$
$$= 3 \quad (\text{odd})$$

$$\Rightarrow y[0] = n[0] \oplus r[0] \oplus Y$$
$$= 0\,(\text{even}) \oplus 1\,(\text{odd}) \oplus 1$$
$$= 0 \quad (\text{even})$$

So, odd = even ⊕ odd ⊕ 1

ii) n is odd, n = 5

$$r = x + y \quad (\text{odd} + \text{even})$$
$$= 5 + 4$$
$$= 9 \bmod 7$$
$$= 2 \quad (\text{even})$$

$$\Rightarrow y[0] = n[0] \oplus r[0] \oplus Y$$
$$= 1\,(\text{odd}) \oplus 0\,(\text{even}) \oplus 1$$
$$= 0$$

So, even = odd ⊕ even ⊕ 1

Compared to case 1, where we had n and r both even or odd, so that n ⊕ r was always even, here we have Y = 1, which changes the parity of r.

But the change of parity of r is negated by ⊕ xoring with Y, as Y is 1 which made r change its parity in the first place.

This way y always remains even and $a = y[0] = 0$.

Note: we worked with example a=1 and a=2 where y didn't wrap around 7 but even if we take

a = 4 and a = 5 which are > 7/2

with MSB = 1, the value of

y = 8 mod 7 = 1 (odd) y = 10 mod 7 = 3 (odd)

So, in the case where y is odd we will have r = y + n + 0 (Y = 0)

= odd + (even or odd)

= odd or even

which will give back y always odd and if parity of r is changed when the sum wraps (Y = 1), xoring with Y = 1 conserves the parity of r, giving y always odd. Hence, if y is odd, LSB(y) = 1

⇒ MSB(a) = 1 where a > n/2.