1 Multiplication Algorithm

Let $\mathbf{x}$ be a vector of dimension n, with Hamming weight close to $\sqrt{n}$ and $\mathbf{y}$ be any vector of dimension n. This document explain a multiplication algorithm that compute $\mathbf{x} \cdot \mathbf{y}$.

For that, we will exploit the fact of doing operations by blocks of 32 bits. We consider the multiplication of two vectors $\mathbf{x}$ and $\mathbf{y}$, using the matrix vector form. In fact we have that :

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x} \times \mathsf{rot}(\mathbf{y})^\top$$

We start by subdividing the matrix of size $n \times n$ into blocks of 32 bits, we have :

$$\mathsf{rot}(\mathbf{y})^\top = \begin{bmatrix} \boxed{y_0 \quad \cdots \quad y_{31}} & \boxed{y_{32} \quad \cdots \quad y_{63}} & \cdots & \boxed{\cdots y_{n-1}} \\ \boxed{y_{n-1} \cdots \quad y_{30}} & \boxed{y_{31} \quad \cdots \quad y_{62}} & \cdots & \boxed{\cdots y_{n-2}} \\ \vdots & \vdots & \vdots & \vdots \\ \boxed{y_2 \quad \cdots \quad y_{33}} & \boxed{y_{34} \quad \cdots \quad y_{65}} & \cdots & \boxed{\cdots y_1} \\ \boxed{y_1 \quad \cdots \quad y_{32}} & \boxed{y_{33} \quad \cdots \quad y_{64}} & \cdots & \boxed{\cdots y_0} \end{bmatrix}$$

Notice that in our case the parameter n is not multiple of 32, thus the last block in each line of the obtained matrix is of size less than 32. In fact, the last block is of size $\delta = n \bmod 32$. In order to obtain a block of 32 bits, we apply a padding of zeros 0. We can also do this by using the following mask, let $\mathsf{mask} = 2^{32} - 2^{\alpha}$ (where $\alpha = 32 - \delta$). We denote by $\mathbf{D}$, the new matrix.

$$\mathbf{D} = \begin{bmatrix} \boxed{y_0 \quad \cdots \quad y_{31}} & \boxed{y_{32} \quad \cdots \quad y_{63}} & \cdots & \boxed{\cdots y_{n-1} 0 \cdots 0} \\ \boxed{y_{n-1} \cdots \quad y_{30}} & \boxed{y_{31} \quad \cdots \quad y_{62}} & \cdots & \boxed{\cdots y_{n-2} 0 \cdots 0} \\ \vdots & \vdots & \vdots & \vdots \\ \boxed{y_2 \quad \cdots \quad y_{33}} & \boxed{y_{34} \quad \cdots \quad y_{65}} & \cdots & \boxed{\cdots y_1 \quad 0 \cdots 0} \\ \boxed{y_1 \quad \cdots \quad y_{32}} & \boxed{y_{33} \quad \cdots \quad y_{64}} & \cdots & \boxed{\cdots y_0 \quad 0 \cdots 0} \end{bmatrix}$$

Let $\mu = n - \delta$, we notice that the matrix $\mathbf{D}$ can be build using the following formula :

$$\mathbf{D}[i] = (2^{31} \times y_i,\ 2^{30} \times y_{(i+1) \bmod n}, \cdots,\ 2^0 \times y_{(i+31) \bmod n}) \text{ for } i \in [0, n-1] \qquad (1)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}[0] & \mathbf{D}[32] & \cdots & \mathbf{D}[\mu] \ \&\ \mathsf{mask} \\ \mathbf{D}[n-1] & \mathbf{D}[31] & \cdots & \mathbf{D}[\mu-1] \ \&\ \mathsf{mask} \\ \mathbf{D}[n-2] & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \mathbf{D}[0] & \cdots & \mathbf{D}[0] \ \&\ \mathsf{mask} \\ \vdots & \mathbf{D}[n-1] & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{D}[2] & \mathbf{D}[34] & \cdots & \mathbf{D}[\mu+2] \ \&\ \mathsf{mask} \\ \mathbf{D}[1] & \mathbf{D}[33] & \cdots & \mathbf{D}[\mu+1] \ \&\ \mathsf{mask} \end{bmatrix}$$

The idea of the algorithm is explained by the following toy example. Suppose that the Hamming weight of vector $\mathbf{x}$ is equal to 3, in particular

$$\mathbf{x} = (1, 1, 0, \cdots, 0, 1, 0)$$

We have that $\mathbf{x} \cdot \mathbf{y} = \mathbf{x} \times \mathsf{rot}(\mathbf{y})^\top = \mathbf{x} \times \mathbf{D}$, it's easy to see that

$$\mathbf{x} \times \mathbf{D} = \begin{bmatrix} \mathbf{D}[0]^\top \\ \mathbf{D}[32]^\top \\ \vdots \\ (\mathbf{D}[\mu] \ \& \ \mathsf{mask})^\top \end{bmatrix} \oplus \begin{bmatrix} \mathbf{D}[n-1]^\top \\ \mathbf{D}[31]^\top \\ \vdots \\ (\mathbf{D}[\mu - 1] \ \& \ \mathsf{mask})^\top \end{bmatrix} \oplus \begin{bmatrix} \mathbf{D}[2]^\top \\ \mathbf{D}[34]^\top \\ \vdots \\ (\mathbf{D}[\mu + 2] \ \& \ \mathsf{mask})^\top \end{bmatrix}$$

In fact the position of the 1s in the vector $\mathbf{x}$ indicates the rows of the matrix $\mathbf{D}$ that we need to xor to obtain the matrix vector product.

We give a brief description of this algorithm. The Algorithm 1, describes all the steps of the multiplication of two vectors $\mathbf{x}$ and $\mathbf{y}$. The Algorithm 2, is used to compute the value of $\mathbf{D}[i]$ for $i \in [0, n-1]$ using the vector $\mathbf{y}$.

Algorithm 1 Multiplication

Imput : a an array of size s that contains the support of the vector $\mathbf{x}$ and b an array of size $m = 1 + \lfloor n/32 \rfloor$ that contains the coordinates of the vector $\mathbf{y}$
Output : t an array of size m that contains the product $\mathbf{x} \cdot \mathbf{y}$
$\mathsf{mask} \longleftarrow 2^{32} - 2^{\alpha}$, with $\alpha = 32 - \delta$ and $\delta = n \bmod 32$
$\mathbf{D}[i]$, $i \in [0, n-1] \longleftarrow$ $\mathsf{ComputeTab}(b)$
for $0 \leq i < s$ **do**
    for $0 \leq j < m - 1$ **do**
        $val \longleftarrow (32 \times j - a[i]) \bmod n$
        $tmp[j] \longleftarrow \mathbf{D}[val]$
    end for
    $j \longleftarrow m - 1$
    $val \longleftarrow (32 \times j - a[i]) \bmod n$
    $tmp[j] \longleftarrow \mathbf{D}[val] \ \& \ \mathsf{mask}$
    for $0 \leq k < m$ **do**
        $t[k] \longleftarrow t[k] \oplus tmp[k]$
    end for
end for

Algorithm 2 ComputeTab

Imput : b an array of size $m = 1 + \lfloor n/32 \rfloor$ that is the coordinates of the vector $\mathbf{y}$
Output : $\mathbf{D}[i]$, $i \in [0, n-1]$
Compute $\mathbf{D}[i]$, $i \in [0, n-1]$ using the equation 1 and the array b.