If $r_0, \ldots, r_{127}$ is a sequence of 128 bits, we use $a_0, \ldots, a_{15}$ to denote the 15 bytes that correpsond to such sequence, such that

$$a_i := \{r_{8i}, r_{8i+1}, \ldots, r_{8i} + 7\}$$

Let $S \in \{0,1\}^{4\times 4}$ be the *state* matrix where each $a \in S$ is a byte. We use $S[i,j]$ or $S_{i,j}$ to denote an individual byte in S.

A *word* is a sequence of four bytes. Each column (and row) in S is a word.

If you studied discrete mathematics at any point, you probably recall that any word over $\{0,1\}^k$ corresponds (through a bijection) to a polynomial in $P[k]$. In particular, each byte a in the array S corresponds to a polynomial of degree 8. Since there are 2^8 such bytes, we can treat any entry in S as a polynomial over the Galois field $P[2^8]$. We should also recall that addition over the polynomial field generated by $\{0,1\}^k$ corresponds to the bit-to-bit sum modulo 2, which in turn corresponds to XOR.

We also recall that multiplication is achieved via modular reduction, meaning that if p, q are the polynomials being multiplied, then their product (within the field) is

$$p(x)q(x) \bmod x^8 + x^4 + x^3 + x + 1$$

1 SubBytes

The *SubBytes()* algorithm is an invertible and non-linear transformation of the state. A substitution table, S-box, is applied to each byte of the state.

Let b denote an input byte to SBOX and $c := 01100011$. The S-box is used to compute b' as follows:

(1) Define $\widetilde{b}$ as follows:

$$\widetilde{b} := \begin{cases} 00 & b = 00 \\ b^{-1} & b \neq 00 \end{cases}$$

(2) For each bit b_i in $\widetilde{b}$, compute

$$b'_i := \widetilde{b_i} + b_{(i+4)} + \widetilde{b_{i+5}} + \widetilde{b_{i+7}} + c_i$$

where the sum in the indices is done over $\bmod 8$.

2 ShiftRows

ShiftRows() shifts the bytes in the last three rows cyclically. The number of positions by which the bytes are shifted depends on the row index r:

$$s'_{r,c} := s_{r,(c+r) \mod 4} \quad \text{for } 0 \leq r < 4, 0 \leq c < 4$$