- Recall that, in contrast to DES, the operations of AES have very simple (though somewhat advanced) mathematical descriptions.

  No mysteriously constructed S-boxes and P-boxes as in DES.

ByteSub (continued)

Each of the 16 bytes gets substituted as follows.

Note. The mathematical description below can be implemented in a **lookup table**: you can find this table in Table 5.1 of our book or, for instance, on wikipedia: https://en.wikipedia.org/wiki/Rijndael_S-box

- Interpret the input byte $(b_7 b_6 \ldots b_0)_2$ as the element $b_7 x^7 + \ldots + b_1 x + b_0$ of $\mathrm{GF}(2^8)$.
- Compute $s^{-1} = c_0 + c_1 x + \ldots + c_7 x^7$ (with 0^{-1} interpreted as 0).

  Important comment. This inversion is what makes AES highly nonlinear.

  If the ByteSub substitution was linear, then all of AES would be linear (because all other layers are linear; assuming we adjust the key schedule accordingly).
- Then the output bits $(d_7 d_6 \ldots d_1 d_0)_2$ are

$$\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

  Comment. The particular choice of matrix and vector has the effect that no ByteSub output equals the ByteSub input (or its complement).

Example 141. Invert $x^3 + 1$ in $\mathrm{GF}(2^8)$, constructed as in AES. [Example 138, again]

Solution. We use the extended Euclidean algorithm, and always reduce modulo 2:

$$\begin{aligned} \boxed{x^8 + x^4 + x^3 + x + 1} &\equiv (x^5 + x^2 + x + 1) \cdot \boxed{x^3 + 1} + x^2 \\ \boxed{x^3 + 1} &\equiv x \cdot \boxed{x^2} + 1 \end{aligned}$$

Backtracking through this, we find that Bézout's identity takes the form

$$\begin{aligned} 1 \equiv 1 \cdot \boxed{x^3 + 1} - x \cdot \boxed{x^2} &\equiv 1 \cdot \boxed{x^3 + 1} - x \cdot \left(\boxed{x^8 + x^4 + x^3 + x + 1} - (x^5 + x^2 + x + 1) \cdot \boxed{x^3 + 1} \right) \\ &\equiv (x^6 + x^3 + x^2 + x + 1) \cdot \boxed{x^3 + 1} + x \cdot \boxed{x^8 + x^4 + x^3 + x + 1}. \end{aligned}$$

Hence, $(x^3 + 1)^{-1} = x^6 + x^3 + x^2 + x + 1$ in $\mathrm{GF}(2^8)$.

Example 142.

(a) What happens to the byte $(0000\ 0101)_2$ during ByteSub?

(b) What happens to the byte $(0000\ 1001)_2$ during ByteSub?

Solution.

(a) $(0000\ 0101)_2$ represents the polynomial x^2+1.

By Example 138, its inverse is $(x^2+1)^{-1}=x^6+x^4+x$ in $\mathrm{GF}(2^8)$, which is $\boldsymbol{c}=(0101\ 0010)_2$.

$$\begin{bmatrix} 1&0&0&0&1&1&1&1\\ 1&1&0&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&1&0&0&0&1\\ 1&1&1&1&1&0&0&0\\ 0&1&1&1&1&1&0&0\\ 0&0&1&1&1&1&1&0\\ 0&0&0&1&1&1&1&1 \end{bmatrix} \underbrace{\begin{bmatrix} 0\\1\\0\\0\\1\\0\\1\\0 \end{bmatrix}}_{\boldsymbol{c}} + \begin{bmatrix} 1\\1\\0\\0\\0\\1\\1\\0 \end{bmatrix} = \begin{bmatrix} 0\\1\\1\\1\\1\\1\\0\\0 \end{bmatrix} + \begin{bmatrix} 1\\0\\0\\0\\1\\1\\1\\1 \end{bmatrix} + \begin{bmatrix} 1\\1\\1\\0\\0\\0\\1\\1 \end{bmatrix} + \begin{bmatrix} 1\\1\\0\\0\\0\\1\\1\\0 \end{bmatrix} = \begin{bmatrix} 1\\1\\0\\1\\0\\1\\1\\0 \end{bmatrix}$$

[This is just the usual matrix-vector product modulo 2. The highlighted columns are the ones which get added up during this matrix-vector product.]

Hence, the output of ByteSub is the byte $(0110\ 1011)_2$.

Check with lookup tables. Indeed, our computation matches $107=(0110\ 1011)_2$ in the lookup table in our book (row 0, column $(0101)_2=5$) or $(6\mathrm{B})_{16}=(0110\ 1011)_2$ on wikipedia (row $(0000)_2=(0)_{16}$, column $(0101)_2=(5)_{16}$).

(b) $(0000\ 1001)_2$ represents the polynomial x^3+1.

By Example 138 or 141, $(x^3+1)^{-1}=x^6+x^3+x^2+x+1$ in $\mathrm{GF}(2^8)$, which is $\boldsymbol{c}=(0100\ 1111)_2$.

$$\begin{bmatrix} 1&0&0&0&1&1&1&1\\ 1&1&0&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&1&0&0&0&1\\ 1&1&1&1&1&0&0&0\\ 0&1&1&1&1&1&0&0\\ 0&0&1&1&1&1&1&0\\ 0&0&0&1&1&1&1&1 \end{bmatrix} \underbrace{\begin{bmatrix} 1\\1\\1\\1\\0\\0\\1\\0 \end{bmatrix}}_{\boldsymbol{c}} + \begin{bmatrix} 1\\1\\0\\0\\0\\1\\1\\0 \end{bmatrix} = \begin{bmatrix} 1\\1\\1\\1\\1\\0\\0\\0 \end{bmatrix} + \begin{bmatrix} 0\\1\\1\\1\\1\\1\\0\\0 \end{bmatrix} + \begin{bmatrix} 0\\0\\1\\1\\1\\1\\1\\0 \end{bmatrix} + \begin{bmatrix} 0\\0\\0\\1\\1\\1\\1\\1 \end{bmatrix} + \begin{bmatrix} 1\\1\\1\\0\\0\\0\\1\\1 \end{bmatrix} + \begin{bmatrix} 1\\1\\0\\0\\0\\1\\1\\0 \end{bmatrix} = \begin{bmatrix} 1\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix}$$

Hence, the output of ByteSub is the byte $(0000\ 0001)_2$.

Check with lookup tables. Indeed, our computation matches the value 1 in the lookup table in our book (row 0, column $(1001)_2=9$) or $(01)_{16}$ on wikipedia (row $(0000)_2=(0)_{16}$, column $(1001)_2=(9)_{16}$).