Escola Politècnica Superior **Universitat de Lleida**
Grau en Enginyeria Informàtica

Computational tools for problem solving

Lab list 8

Symmetric cryptography. The S-Box in AES.

The Advanced Encryption Standard is composed of several rounds where the bits of the key are XORed with those of the plaintext block, and the resulting bits are mixed together in basically three different procedures. The S-box in AES is one of these steps, and mathematically the S-box is basically a multiplicative inversion in the finite field $\mathbb{F}_{2^8}$ of 256 elements. This means one needs a way to convert the bytes of each block into elements of $\mathbb{F}_{2^8}$. The standard way to do so is to represent the byte as the coefficients of a polynomial of degree at most 7 over $\mathbb{F}_2$. For example the byte

$$00000010$$

is represented by the polynomial

$$0x^7 + 0x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 1x + 0 = x$$

and the byte

$$10101010$$

by the polynomial

$$1x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 0x^2 + 1x + 0 = x^7 + x^5 + x^3 + x.$$

In the AES finite field $\mathbb{F}_{2^8}$ the modulus chosen to operate elements is given by the pentanomial $m(x) = x^8 + x^4 + x^3 + x + 1 \in \mathbb{F}_2[x]$. The first step in AES S-box computes the inverse of the bytes represented as polynomials operated under this polynomial modulus $m(x)$. They can be computed using the Extended Euclidean Algorithm for polynomials very much like the computation of modular inverses in modular arithmetic $(\bmod\ m)$.

For example, since the modulus is equivalent to 0, then

$$m(x) = x^8 + x^4 + x^3 + x + 1 \equiv 0$$

and then

$$x(x^7 + x^3 + x^2 + 1) = 1 \pmod{m(x)}$$

hence the S-box sends 00000010 to 10001101.

After this field inversion, the second step in the S-box transforms the byte by a fixed transformation

$$\begin{pmatrix} s_0 \\ s_1 \\ s_1 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_1 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

However, in AES the actual computation of field inverses and matrix transformation is avoided, and S-box reduces to a battleship game in a lookup table that already performs the whole computation for you. This is done by dividing the byte into 2 parts of 4 bits each and then use a Hexadecimal digit to represent each part. For example the byte 0000 0010 is coded as $0\,2$ and 1100 1011 as *cb*.

Problem 1.

i) Find the output byte of the following input bytes in AES S-box as described above: $0\,2$, $c\,b$, $9\,a$.

ii) Show the actual computation of the inversions in *i*).

iii) Check your results using the AES S-box lookup table you will find by searching "Rijndael S-box" in the internet.