Finite Fields in RIJNDAEL

Operations on Bytes. Consider a byte $b = (b_7, b_6, \ldots, b_1, b_0)$ (an 8-bit vector) as a polynomial with coefficients in $\{0,1\}$:

$$b \mapsto b(x) = b_7x^7 + b_6x^6 + \cdots + b_1x + b_0 \ .$$

RIJNDAEL makes use of the following operations on bytes, interpreting them as polynomials:

(1) Addition: polynomial addition by taking XOR of coefficients.

$$\begin{array}{ccccccccccc} & b_7x^7 & + & b_6x^6 & +\cdots+ & b_1x & + & b_0 \\ + & c_7x^7 & + & c_6x^6 & +\cdots+ & c_1x & + & c_0 \\ \hline & (b_7 \oplus c_7)x^7 & + & (b_6 \oplus c_6)x^6 & +\cdots+ & (b_1 \oplus c_1)x & + & (b_0 \oplus c_0) \end{array}$$

The sum of two polynomials taken in this manner yields another polynomial of degree 7. In other words, component-wise XOR of bytes is identified with this addition operation on polynomials.

(2) Multiplication: polynomial multiplication (coefficients are in $\{0,1\}$) modulo $m(x) = x^8 + x^4 + x^3 + x + 1$ (remainder when dividing by $m(x)$ — analogous to modulo arithmetic with integers). The remainder when dividing by a degree 8 polynomial will have degree ≤ 7. Thus, the "product" of two bytes is associated with the product of their polynomial equivalents modulo $m(x)$.

(3) Inverse: $b(x)^{-1}$, the inverse of $b(x) = b_7x^7 + b_6x^6 + \cdots + b_1x + b_0$, is the degree 7 polynomial with coefficients in $\{0,1\}$ such that

$$b(x)b(x)^{-1} \equiv 1 \pmod{m(x)} \ .$$

Note that this is completely analogous to the case of integer arithmetic modulo n. In this case the "inverse" of the byte $b = (b_7, b_6, \ldots, b_1, b_0)$ is the byte associated with the inverse of $b(x) = b_7x^7 + b_6x^6 + \cdots + b_1x + b_0$.

By associating bytes with polynomials, we obtain the above three operations on bytes. RIJNDAEL uses inverse as above in the ByteSub operation.

$GF(2^8)$ is the set of 256 bytes viewed as polynomials, together with the operations described above.

4-byte Vectors. In the MixColumn operation of RIJNDAEL, 4-byte vectors are considered as degree 3 polynomials with coefficients in $GF(2^8)$. That is, the 4-byte vector (a_3, a_2, a_1, a_0) is associated with the polynomial

$$a_3x^3 + a_2x^2 + a_1x + a_0,$$

where each coefficient is a byte viewed as an element of $GF(2^8)$ (addition, multiplication, and inversion of the coefficients is performed as described above). We have the following operations on these polynomials:

(1) addition: component-wise "addition" of coefficients (addition as described above)

(2) multiplication: polynomial multiplication (addition and multiplication of coefficients as described above) modulo $M(x) = x^4 + 1$. Result is a degree 3 polynomial with coefficients in $GF(2^8)$.

In MixColumn, the 4-byte vector (a_3, a_2, a_1, a_0) is replaced by the result of multiplying $a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ by the fixed polynomial

$$c(x) = 03x^3 + 01x^2 + 01x + 02$$

and reducing modulo $x^4 + 1$. The coefficients of $c(x)$ are given as bytes in hex notation.