I Encontro Acadêmico de Modelagem Computacional do LNCC
Petrópolis, Rio de Janeiro, Brasil
10–11 de Março de 2008

Aplicações de $GF(2^8)$ no algoritmo AES

Raquel de Souza, **Fábio Borges**
Laboratório Nacional de Computação Científica, Petrópolis/RJ
E-mail: {rasouza,borges}@lncc.br

O algoritmo de criptografia AES *(Advanced Encryption Standard)*, conhecido também como Rijndael, trabalha sobre o Corpo de Galois $GF(2^8)$, ou seja, todas as operações matemáticas utilizadas são realizadas sobre este corpo, e usamos o polinômio irredutível $m(x) = x^8 + x^4 + x^3 + x + 1$. O motivo de se trabalhar sobre corpos de Galois, ou corpos finitos, em criptografia, é a garantia da existência de uma operação inversa para cada etapa, que é fundamental no processo de decifragem. No caso de se utilizar $GF(2^n)$, com $n \in \mathbb{N}^*$, temos ainda outro aspecto interessante: a soma coincide com o XOR (denotado por $\oplus$), operação muito rápida computacionalmente. Para a criptografia, em particular, o uso de $GF(2^8)$ é bastante adequado, visto que esse corpo tem $2^8 = 256$ elementos, o mesmo número de caracteres da tabela ASCII *(American Standard Code for Information Interchange)* estendida. Assim, é possível cifrar e decifrar qualquer mensagem.

Todas as etapas, exceto a *ShiftRows*, que é um rotacionamento circular dos bytes do estado, utilizam como base o corpo $GF(2^8)$. A seguir, apresentamos cada etapa do AES (exceto a ShiftRows) e sua relação com o corpo de Galois $GF(2^8)$.

SubBytes - Consiste em uma substituição dos bytes do estado por outros contidos em uma caixa de substituição (S-box). A S-box do Rijndael (S_{RD}) é gerada pela composição de duas funções f e g, sobre $GF(2^8)$.

MixColumns - Pode ser representada como uma multiplicação de matrizes, onde os elementos do estado são considerados polinômios sobre $GF(2^8)$. Multiplica-se uma matriz fixa C pela matriz S que representa o estado, e obtemos a matriz S'.

$$\begin{bmatrix} S'_{1,1} & S'_{1,2} & S'_{1,3} & S'_{1,4} \\ S'_{2,1} & S'_{2,2} & S'_{2,3} & S'_{2,4} \\ S'_{3,1} & S'_{3,2} & S'_{3,3} & S'_{3,4} \\ S'_{4,1} & S'_{4,2} & S'_{4,3} & S'_{4,4} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \odot \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} & S_{1,4} \\ S_{2,1} & S_{2,2} & S_{2,3} & S_{2,4} \\ S_{3,1} & S_{3,2} & S_{3,3} & S_{3,4} \\ S_{4,1} & S_{4,2} & S_{4,3} & S_{4,4} \end{bmatrix},$$

onde $\odot$ é o produto matricial em $GF(2^8)$, ou seja, a multiplicação é feita módulo $m(x)$ e a soma corresponde ao XOR.

AddRoundKey - É um XOR byte a byte entre o estado e a chave de rodada. Isto quer dizer que se $s_{x,y}$ é um byte do estado S e $k_{x,y}$ um byte da chave, o byte $s'_{x,y}$ do novo estado S' será igual a $s_{x,y} \oplus k_{x,y}$. Como já foi dito, a soma e o XOR coincidem quando usamos um corpo de característica 2. A transformação inversa à AddRoundKey também consiste em um XOR entre o estado e a chave de rodada, ou seja, AddRoundKey é sua própria inversa, visto que $(s_{x,y} \oplus k_{x,y}) \oplus k_{x,y} = s_{x,y}$.

Keywords: *criptografia, AES, corpos de Galois*