**MAC5723 e MAC336 - CRIPTO - 1a. Prova - abril 2005**

# Questão 1 (50%)

(A) Especificar a função inversa da função RC5 de $r$ rounds dada abaixo.
(B) Demonstre que a sua especificação é de fato a função inversa.

**Parâmetros do RC5**

1. $w = 16, 32$, ou 64 bits, é o número de bits em cada variável (*i.e.*, cada posição de memória);
2. $r$ é o número de iterações (*rounds*);
3. $b$ é o número de bytes na chave.

**Operações básicas do RC5**

As operações básicas usadas são todas sobre operandos de $w$ bits, e são:

1. $v \boxplus u$ é a soma dos inteiros $v, u$ de $w$ bits, resultando um valor de $w$ bits (*i.e.*, soma $\bmod 2^w$);
2. $v \oplus u$ é o ou-exclusivo (XOR) de $v, u$ de $w$ bits, resultando um valor de $w$ bits;
3. $v \lll t$ é o deslocamento circular (*i.e.*, rotação) de $t$ posições para a esquerda dos bits em $v$.
4. $v \ggg t$ é o deslocamento circular (*i.e.*, rotação) de $t$ posições para a direita dos bits em $v$.

A entrada é de $2w$ bits, e a saída também. As duas metades da entrada $A$ e $B$ são alteradas a cada iteração. São usadas 2 subchaves antes da primeira iteração, e 2 subchaves em cada iteração, tendo-se no total $2r + 2$ subchaves. A chave $K$ é de $b$ bytes, da qual são geradas as subchaves $K_j$ de $w$ bits cada.

**Algoritmo de geração de subchaves RC5**

Você pode supor dado o algoritmo que expande os $b$ bytes da chave $K$ para $2r + 2$ subchaves de $w$ bits $K_0, K_1, K_3, ...K_{2r+1}$.

**Algoritmo de criptografia RC5**

Este algoritmo supõe que o computador armazena os bytes em modo *little-endian*, ou seja tanto o texto legível como o ilegível são armazenados da seguinte forma nas variáveis $A$ e $B$, para $w = 32$: os primeiros quatro bytes do texto (i)legível $(x_0, x_1, x_2, x_3)$ são armazenados em $A$ na seqüência $(x_3, x_2, x_1, x_0)$, e os quatro bytes seguintes do texto (i)legível $(x_4, x_5, x_6, x_7)$ são armazenados em $B$ na seqüência $(x_7, x_6, x_5, x_4)$.

**Algoritmo RC5**
**Entrada**: chave $K$ de $b$ bytes, texto legível de $2w$ bits $(A, B)$;
**Saída**: ilegível de $2w$ bits $(A, B)$

1. Calcular $2r + 2$ subchaves $K_0, K_1, K_3, ...K_{2r+1}$ (* pode supor dado este algoritmo *)

2. $A \leftarrow A \boxplus K_0; B \leftarrow B \boxplus K_1;$

3. **para** $j = 1, 2, 3, ...r$ **faça**:

$$A \leftarrow ((A \oplus B) \lll B) \boxplus K_{2j}; B \leftarrow ((B \oplus A) \lll A) \boxplus K_{2j+1}$$

4. A saída é o valor $(A, B)$.

# Questão 2 (50%)

Baseado no Teorema Chinês do Resto, podemos projetar diversos criptossistemas parecidos com o RSA, de chave pública. Um deles é da seguinte forma: sendo $n$ produto de dois primos $p, q$ tais que $(p + 1)$ e $(q + 1)$ são divisíveis por 4, Alice calcula um código $c$ de uma mensagem $m$ pela expressão:

$$c = m(m + b) \bmod n$$

onde $b$ é uma constante tal que $0 < b < n$. Por exemplo, se $m = 17, p = 7, q = 11, b = 13$, então $n = 77, c = 17(17 + 13) \bmod 77 = 48$

Pergunta-se: como Beto que tenha recebido $c$ da Alice pode recuperar $m$? Em outras palavras:

1. qual é o algoritmo para calcular a chave pública de Beto $(b, n)$ e a chave particular de Beto em função de $p, q, b$?

2. e qual é o algoritmo de decriptografia com essas chaves?

SUGESTÃO:

1. Calcular inicialmente (i.e., você pode supor dado um algoritmo para calcular) $d$ tal que $2d \bmod n = b$.

2. E então demonstrar que $(m+d)^2 \bmod n = (c+d^2) \bmod n$

3. Beto teria que resolver a equação seguinte, com incógnita $x = (m+d)$: $x^2 \bmod n = a \bmod n$ com $a = (c+d^2)$, i.e., $x$ é raiz quadrada de $a$ mod $n$.

4. A seguir calcular $x_1, x_2$ tais que $(x_1)^2 \bmod p = a$ e $(x_2)^2 \bmod q = a$.

5. Aplicando o algoritmo do Teorema Chinês do Resto, combinar as soluções $x_1$ e $x_2$ para obter uma solução para $x^2 \bmod n = a \bmod n$.

Lembrete: (Teorema Chinês do Resto) São dados, para: $i = 1, ...r : a_i = N \bmod m_i$ com $mdc(m_i, m_j) = 1$, se $i \neq j$.

Solução é $N = \sum_{i=1}^{r} a_i M_i y_i \bmod (m_1 ... m_r)$ onde $M_i = (m_1 ... m_r)/m_i$ e $y_i = M_i^{-1} \bmod m_i$.