CS 4780 **Summer 2015**

System Admin/Network Security

Project 1

I want you to implement the RC5 encryption algorithm for a $w = 32$ bit (= 4 byte) word size, $r = 12$ rounds, and a $b = 8$ byte key K. We will write $t = 2 \times r + 2 = 24$ and $c = \lceil b/u \rceil = 2$. [$u$ is the number of bytes per word]

The program is to be named `rc5` and it must take a single command line argument of the key in hex [16 hex characters]. It then reads the message as ASCII characters from standard input until end-of-file on standard input. Each block is 8 characters [64 bits] split into 2 32-bit pieces [A and B] ["little-endian" = A is the lower half of the 32 bit word]; the final block is to be extended with bytes of 0 [the C character `'\0'`] if needed. The output is to be printed to the screen as hexadecimal bytes.

To do the encryption, we will have prepared an array $S[2r+2]$ as below. Here $+$ is ordinary 32-bit addition, $\leftarrow$ is left shift, and $\oplus$ is xor. The array S is of size t and is generated from the inputed key. The encryption then is:

$A = A + S[0]$; $B = B + S[1]$
for $i = 1$ **to** r **do**
    $A = ((A \oplus B) \leftarrow B) + S[2 \times i]$; $B = ((B \oplus A) \leftarrow A) + S[2 \times i + 1]$

To generate the S array, we first set $P =$ `0xb7e15163` and $Q =$ `0x9E3779B9`. We first set $S[0] = P$ and for $1 \le i \le t - 1$ we set $S[i] = S[i - 1] + Q$. We create an array $L[c]$ copying K into L in little-endian order; we can do: zero the array L and for $b - 1 \ge i \ge 0$ set $L[i/u] = L[i/u] \leftarrow 8 + K[i]$. [We've created a word array from a byte array.] Finally, we combine L and S by

$i = j = C = D = 0$
do $3 \times \max(t, c)$ **times**
    $C = S[i] = (S[i] + C + D) \leftarrow 3$; $i = i + 1 \bmod t$
    $D = L[j] = (L[j] + C + D) \leftarrow (C + D)$; $j = j + 1 \bmod c$

The S array is now ready for use.

I want this due 09 Jul 2015 at 2359. Turnin directions on the web site.