RC5 Encryption Algorithm

RC5 is a symmetric key block encryption algorithm designed by Ron Rivest in 1994. It is notable for being simple, fast (on account of using only primitive computer operations like XOR, shift, etc.) and consumes less memory.

RC5 is a block cipher and addresses two-word blocks at a time. Depending on input plain text block size, number of rounds and key size, various instances of RC5 can be defined and each instance is denoted as RC5-w/r/b where w=word size in bits, r=number of rounds and b=key size in bytes.
Allowed values are:

Parameter	Possible Value
block/word size (bits)	16, 32, 64
Number of Rounds	0 – 255
Key Size (bytes)	0 – 255

Note – Since at a time, RC5 uses 2 word blocks, the plain text block size can be 32, 64 or 128 bits.
Notation used in the algorithm:

Symbol	Operation
x <<< y	Cyclic left shift of x by y bits
+	Two's complement addition of words where addition is modulo
^	Bit wise Exclusive-OR

Step-1: Initialization of constants P and Q.
RC5 makes use of 2 magic constants P and Q whose value is defined by the word size w.

Word Size (bits)	P (Hexadecimal)	Q (Hexadecimal)
16	b7e1	9e37
32	b7e15163	9e3779b9
64	b7e151628aed2a6b	9e3779b97f4a7c15

For any other word size, P and Q can be determined as:

$P = Odd((e-2)2^w)$

$Q = Odd((\emptyset-2)2^w)$

Here, Odd(x) is the odd integer nearest to x, e is the base of natural logarithms and $\emptyset$ is the golden ratio.

Step-2: Converting secret key K from bytes to words. Secret key K of size b bytes is used to initialize array L consisting of c words where c = b/u, u = w/8 and w = word size used for that particular instance of RC5. For example, if we choose w=32 bits and Key k is of size 96 bytes then, u=32/8=4, c=b/u=96/4=24.

L is pre initialized to 0 value before adding secret key K to it.

```
for i=b-1 to 0
    L[i/u] = (L[u/i] <<< 8) + K[i]
```

Step-3: Initializing sub-key S.
Sub-key S of size t=2(r+1) is initialized using magic constants P and Q.

```
S[0] = P
for i = 1 to 2(r+1)-1
    S[i] = S[i-1] + Q)
```

Step-4: Sub-key mixing. The RC5 encryption algorithm uses Sub key S. L is merely, a temporary array formed on the basis of user entered secret key. Mix in user's secret key with S and L.

```
i = j = 0
A = B = 0
do 3 * max(t, c) times:
    A = S[i] = (S[i] + A + B) <<< 3
    B = L[j] = (L[j] + A + B) <<< (A + B)
    i = (i + 1) % t
    j = (j + 1) % c
```

Step-5: Encryption.
We divide the input plain text block into two registers A and B each of size w bits. After undergoing the encryption process the result of A and B together forms the cipher text block.

RC5 Encryption Algorithm:

1. One time initialization of plain text blocks A and B by adding S[0] and S[1] to A and B respectively. These operations are mod .
2. XOR A and B. A=A^B
3. Cyclic left shift new value of A by B bits.
4. Add S[2*i] to the output of previous step. This is the new value of A.
5. XOR B with new value of A and store in B.
6. Cyclic left shift new value of B by A bits.
7. Add S[2*i+1] to the output of previous step. This is the new value of B.
8. Repeat entire procedure (except one time initialization) r times.

```
A = A + S[0]
B = B + S[1]
for i = 1 to r do:
    A = ((A ^ B) <<< B) + S[2 * i]
    B = ((B ^ A) <<< A) + S[2 * i + 1]
return A, B
```

15. Alternatively, RC5 Decryption can be defined as:

```
for i = r down to 1 do:
    B = ((B - S[2 * i + 1]) >>> A) ^ A
    A = ((A - S[2 * i]) >>> B) ^ B
B = B - S[1]
A = A - S[0]
return A, B
```