#include <stdio.h>

#include <stdint.h>

#include <string.h>

#define w 32

#define r 12

#define b 16

#define c (b / 4)

#define t (2 \* (r + 1))

#define Pw 0xB7E15163

#define Qw 0x9E3779B9

typedef uint32\_t WORD;

WORD S[t]

void RC5\_key\_schedule(unsigned char \*K) {

WORD L[c];

for (int i = 0; i < c; i++) {

L[i] = ((WORD)K[4\*i]) | ((WORD)K[4\*i+1] << 8) | ((WORD)K[4\*i+2] << 16) | ((WORD)K[4\*i+3] << 24);

}

S[0] = Pw;

for (int i = 1; i < t; i++) {

S[i] = S[i - 1] + Qw;

}

WORD A = 0, B = 0;

int i = 0, j = 0;

for (int k = 0; k < 3 \* t; k++) {

A = S[i] = ((S[i] + A + B) << 3) | ((S[i] + A + B) >> (w - 3));

B = L[j] = ((L[j] + A + B) << (A + B)) | ((L[j] + A + B) >> (w - (A + B)));

i = (i + 1) % t;

j = (j + 1) % c;

}

}

void RC5\_encrypt(WORD \*pt, WORD \*ct) {

WORD A = pt[0] + S[0];

WORD B = pt[1] + S[1];

for (int i = 1; i <= r; i++) {

A = ((A ^ B) << (B & 0x1F)) | ((A ^ B) >> (w - (B & 0x1F)));

A += S[2 \* i];

B = ((B ^ A) << (A & 0x1F)) | ((B ^ A) >> (w - (A & 0x1F)));

B += S[2 \* i + 1];

}

ct[0] = A;

ct[1] = B;

}

void RC5\_decrypt(WORD \*ct, WORD \*pt) {

WORD B = ct[1];

WORD A = ct[0];

for (int i = r; i >= 1; i--) {

B -= S[2 \* i + 1];

B = ((B >> (A & 0x1F)) | (B << (w - (A & 0x1F)))) ^ A;

A -= S[2 \* i];

A = ((A >> (B & 0x1F)) | (A << (w - (B & 0x1F)))) ^ B;

}

pt[1] = B - S[1];

pt[0] = A - S[0];

}

void print\_block(WORD \*block) {

printf("%08X %08X\n", block[0], block[1]);

}

int main() {

unsigned char key[b] = "ExampleRC5Key123";

WORD pt[2] = {0x12345678, 0x9ABCDEF0};

WORD ct[2];

WORD decrypted[2];

RC5\_key\_schedule(key);

RC5\_encrypt(pt, ct);

RC5\_decrypt(ct, decrypted);

printf("Plaintext: ");

print\_block(pt);

printf("Encrypted: ");

print\_block(ct);

printf("Decrypted: ");

print\_block(decrypted);

return 0;

}

