st98 / solve.c
Created a year ago

angstromCTF 2016 - [crypto 160] My Accountant

solve.c

```c
// gcc -O3 solve.c -o solve
#include <stdio.h>
int sBox[4][16] = {
  { 2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9 },
  { 14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6 },
  { 4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14 },
  { 11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3 }
};
int sBoxInv[4][16] = {
  { 13, 3, 0, 10, 2, 9, 7, 4, 8, 15, 5, 6, 1, 12, 14, 11 },
  { 9, 7, 2, 12, 4, 8, 15, 5, 14, 13, 11, 1, 3, 6, 0, 10 },
  { 14, 2, 1, 13, 0, 11, 12, 6, 7, 9, 4, 3, 10, 5, 15, 8 },
  { 10, 4, 6, 15, 13, 14, 8, 3, 1, 11, 12, 0, 2, 7, 5, 9 }
};
int pBox[16] = { 6, 15, 3, 8, 2, 4, 9, 7, 13, 10, 0, 1, 5, 11, 14, 12 };
int pBoxInv[16] = { 10, 11, 4, 2, 5, 12, 0, 7, 3, 6, 9, 13, 15, 8, 14, 1 };
int P(int block, int permute[]) {
  int i, r = 0, bit;
  for (i = 0; i < 16; i++) {
    bit = (block & 1 << (15 - i)) != 0;
    r |= bit << (15 - permute[i]);
  }
  return r;
}
int S(int block, int sub[][16]) {
  int i, j, r = 0, bits;
  for (i = 0; i < 4; i++) {
    j = (4 * (3 - i));
    bits = (block & (0xf << j)) >> j;
    r |= sub[i][bits] << j;
  }
  return r;
}
int Eround(int block, int key) {
  int r = block ^ key;
  r = S(r, sBox);
  r = P(r, pBox);
  return r;
}
int Dround(int block, int key) {
  int r = P(block, pBoxInv);
  r = S(r, sBoxInv);
  r ^= key;
  return r;
}
int decrypt(int block, int key1, int key2, int key3) {
  int r = Dround(block, key3);
  r = Dround(r, key2);
  r = Dround(r, key1);
  return r;
}
int main(void) {
  int k1, k2, k3, t, i;
  int c[5] = { 0xc030, 0x4de9, 0x5847, 0xd776, 0xb7af }; // ciphertext
  int p[5] = { 0x4153, 0x5345, 0x5453, 0x0a43, 0x7572 }; // plaintext
  t = P(c[0], pBoxInv);
  t = S(t, sBoxInv);
  for (k1 = 0; k1 < 0x10000; k1++) {
    for (k2 = 0; k2 < 0x10000; k2++) {
        k3 = p[0];
        k3 = Eround(k3, k1);
        k3 = Eround(k3, k2);
```

```
63          k3 ^= t;
64          for (i = 0; i < 5; i++) {
65            if (decrypt(c[i], k1, k2, k3) != p[i]) goto end;
66          }
67          printf("%04x%04x%04x\n", k1, k2, k3);
68          end:;
69        }
70      }
71      return 0;
72  }
```