```
1 #include "prototypes.h"
2
3
4
 5 /* Encrypt a single block with 10 rounds */
 6 void AesEncrypt(unsigned char *blk, unsigned char *key, int Nr)
7 {
        printf("Round 0:\n");
8
 9
       printf("----Start: ");
       for (int i = 0; i < 16; i++)
10
            printf("%02x ", blk[i]);
11
12
       printf("\n");
13
14
       printf("----Output: ");
15
       AddRoundKey(blk, key, 0);
16
       for (int i = 0; i < 16; i++)
17
18
            printf("%02x ", blk[i]);
19
20
       for (int x = 1; x \leftarrow (Nr - 1); x++)
21
22
            SubBytes(blk);
23
            ShiftRows(blk);
24
            MixColumns(blk);
25
            AddRoundKey(blk, key, x);
26
            printf("\nRound %d:\n", x);
            printf("----Output: ");
27
28
            for (int i = 0; i < 16; i++)
29
                printf("%02x ", blk[i]);
30
31
       }
32
       printf("\nRound 10:\n");
33
34
       SubBytes(blk);
35
       ShiftRows(blk);
36
       AddRoundKey(blk, key, Nr);
37
       printf("----Output: ");
       for (int i = 0; i < 16; i++)
38
            printf("%02x ", blk[i]);
39
40 }
41
42
43
44
45
46
47 /* The AES Substitution Table */
48 static const unsigned char sbox[256] = {
       0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, →
49
           0xd7, 0xab, 0x76,
50
       0xCA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0, 0xAD, 0xD4, 0xA2, 0xAF, 0x9C, >
           0xA4, 0x72, 0xC0,
```

```
0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC, 0x34, 0xA5, 0xE5, 0xF1, 0x71,
51
          0xD8, 0x31, 0x15,
       0x04, 0xC7, 0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A, 0x07, 0x12, 0x80, 0xE2, 0xEB, >
52
          0x27, 0xB2, 0x75,
53
       0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0, 0x52, 0x3B, 0xD6, 0xB3, 0x29, >
          0xE3, 0x2F, 0x84,
54
       0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A, 0xCB, 0xBE, 0x39, 0x4A, >
          0x4C, 0x58, 0xCF,
55
       0xD0, 0xEF, 0xAA, 0xFB, 0x43, 0x4D, 0x33, 0x85, 0x45, 0xF9, 0x02, 0x7F, 0x50, →
          0x3C, 0x9F, 0xA8,
       0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5, 0xBC, 0xB6, 0xDA, 0x21, 0x10, >
56
          0xFF, 0xF3, 0xD2,
       0xCD, 0x0C, 0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17, 0xC4, 0xA7, 0x7E, 0x3D, 0x64, →
57
          0x5D, 0x19, 0x73,
       0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A, 0x90, 0x88, 0x46, 0xEE, 0xB8, 0x14, 0xDE, >
58
          0x5E, 0x0B, 0xDB,
       0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C, 0xC2, 0xD3, 0xAC, 0x62, 0x91, →
59
          0x95, 0xE4, 0x79,
       0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x4E, 0xA9, 0x6C, 0x56, 0xF4, 0xEA, 0x65, →
60
          0x7A, 0xAE, 0x08,
       0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6, 0xE8, 0xDD, 0x74, 0x1F, 0x4B, →
61
          0xBD, 0x8B, 0x8A,
       0x70, 0x3E, 0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E, 0x61, 0x35, 0x57, 0xB9, 0x86, >
62
          0xC1, 0x1D, 0x9E,
       0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94, 0x9B, 0x1E, 0x87, 0xE9, 0xCE, →
63
          0x55, 0x28, 0xDF,
       0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, >
64
          0x54, 0xbb, 0x16 };
65
66
67
   static const unsigned char modifiedsbox[256] = {
       0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, →
68
          0xd7, 0xab, 0x76,
                               // 0
       0xCA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0, 0xAD, 0xD4, 0xA2, 0xAF, 0x9C, →
69
          0xA4, 0x72, 0xC0,
                               // 1
       0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC, 0x34, 0xA5, 0xE5, 0xF1, 0x71, →
70
          0xD8, 0x31, 0x15,
                               // 2
       0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5, 0xBC, 0xB6, 0xDA, 0x21, 0x10, >
71
          0xFF, 0xF3, 0xD2,
                               // 3 ----- swap with 7
       0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0, 0x52, 0x3B, 0xD6, 0xB3, 0x29, →
72
          0xE3, 0x2F, 0x84,
                               // 4
       0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A, 0xCB, 0xBE, 0x39, 0x4A, →
73
          0x4C, 0x58, 0xCF,
                               // 5
       0xD0, 0xEF, 0xAA, 0xFB, 0x43, 0x4D, 0x33, 0x85, 0x45, 0xF9, 0x02, 0x7F, 0x50, →
74
          0x3C, 0x9F, 0xA8,
                               // 6
       0x04, 0xC7, 0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A, 0x07, 0x12, 0x80, 0xE2, 0xEB, >
75
          0x27, 0xB2, 0x75,
                               // 7 ----- swap with 3
76
       0xCD, 0x0C, 0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17, 0xC4, 0xA7, 0x7E, 0x3D, 0x64, →
          0x5D, 0x19, 0x73,
                               // 8
77
       0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A, 0x90, 0x88, 0x46, 0xEE, 0xB8, 0x14, 0xDE, >
          0x5E, 0x0B, 0xDB,
                               // 9
       0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C, 0xC2, 0xD3, 0xAC, 0x62, 0x91, >
78
```

```
0x95, 0xE4, 0x79,
                                 // 10
 79
        0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x4E, 0xA9, 0x6C, 0x56, 0xF4, 0xEA, 0x65, →
            0x7A, 0xAE, 0x08,
                                // 11
 80
        0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6, 0xE8, 0xDD, 0x74, 0x1F, 0x4B, →
            0xBD, 0x8B, 0x8A,
                                 // 12
        0x70, 0x3E, 0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E, 0x61, 0x35, 0x57, 0xB9, 0x86, >
 81
            0xC1, 0x1D, 0x9E,
                                // 13
        0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94, 0x9B, 0x1E, 0x87, 0xE9, 0xCE, >
 82
            0x55, 0x28, 0xDF,
                                 // 14
 83
        0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, >
            0x54, 0xbb, 0x16 }; // 15
 84
 85
 86 /* The key schedule rcon table */
 87 static const unsigned char Rcon[10] =
 88 { 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x1B, 0x36 };
 89
 90 /* The *x function */
 91 static unsigned char xtime(unsigned char x)
 92 {
 93
        if (x & 0x80) { return ((x << 1) ^ 0x1B) & 0xFF; }</pre>
 94
        return x << 1;
 95 }
 96
 97
 98
 99 /* MixColumns: Processes the entire block */
100 static void MixColumns(unsigned char *col)
101 {
102
        unsigned char tmp[4], xt[4];
103
104
        for (int x = 0; x < 4; x++, col += 4)
105
106
            xt[0] = xtime(col[0]);
107
            xt[1] = xtime(col[1]);
            xt[2] = xtime(col[2]);
108
109
            xt[3] = xtime(col[3]);
110
            tmp[0] = xt[0] ^ xt[1] ^ col[1] ^ col[2] ^ col[3];
            tmp[1] = col[0] ^ xt[1] ^ xt[2] ^ col[2] ^ col[3];
111
            tmp[2] = col[0] ^ col[1] ^ xt[2] ^ xt[3] ^ col[3];
112
            tmp[3] = xt[0] ^ col[0] ^ col[1] ^ col[2] ^ xt[3];
113
114
            col[0] = tmp[0];
            col[1] = tmp[1];
115
116
            col[2] = tmp[2];
117
            co1[3] = tmp[3];
118
        }
119 }
120
121
122
123
124 /* ShiftRows: Shifts the entire block */
```

```
125 static void ShiftRows(unsigned char *col)
126 {
127
        unsigned char t;
128
129
        /* 2nd row */
130
        t = col[1];
        col[1] = col[5];
131
        col[5] = col[9];
132
133
        col[9] = col[13];
134
        col[13] = t;
135
        /* 3rd row */
136
        t = col[2];
137
138
        col[2] = col[10];
139
        col[10] = t;
140
        t = col[6];
141
        col[6] = col[14];
142
        col[14] = t;
143
144
        /* 4th row */
145
       t = col[15];
       col[15] = col[11];
146
        col[11] = col[7];
147
148
        col[7] = col[3];
149
        col[3]
               = t;
150 }
151
152
153
154 /* SubBytes */
155 static void SubBytes(unsigned char *col)
156 {
157 #if DEMO
        for (int x = 0; x < 16; x++)
158
159
           col[x] = sbox[col[x]];
160 #else
161 for (int x = 0; x < 16; x++)
162
           col[x] = modifiedsbox[col[x]];
163 #endif // DEMO
164 }
165
166
167
168 /* AddRoundKey */
169 static void AddRoundKey(unsigned char *col, unsigned char *key, int round)
170 {
171
        for (int x = 0; x < 16; x++)
172
            col[x] \stackrel{\text{}}{}= key[(round << 4) + x];
173 }
```