PRÁCTICA: RIJNDAEL

Objetivo: Implementar el algoritmo Rijndael

Desarrollo:

1. Implementa el cifrado en bloque para bloques y claves de 128 bits (que se representan como matrices de estado de 4x4 bytes):
   - Expansión de la clave obteniendo 10 subclaves a partir de la clave original usando la constante de iteración
   - Etapa inicial: AddRoundKey entre el bloque de entrada y la clave original
   - 9 Iteraciones de:
   1. SubBytes — usando la S-Caja
   2. ShiftRow
   3. MixColumn — multiplicando una matriz por los cuatro bytes de cada columna, operando con bytes.
   4. AddRoundKey — con la subclave correspondiente a la iteración
   - Etapa final:
   1. SubBytes
   2. ShiftRows
   3. AddRoundKey — con la última subclave

S-Caja

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Matriz para multiplicación en MixColumn

2 3 1 1

1 2 3 1

1 1 2 3

3 1 1 2

Ejemplo implementación de Mixcolumn:

```
void gmix_column(unsigned char *r) {
        unsigned char a[4];
        unsigned char b[4];
                unsigned char c;
                unsigned char h;
                /* The array 'a' is simply a copy of the input array 'r'
         * The array 'b' is each element of the array 'a' multiplied by 2
         * in Rijndael's Galois field
         * a[n] ^ b[n] is element n multiplied by 3 in Rijndael's Galois field */
                for(c=0;c<4;c++) {
                        a[c] = r[c];
                        h = r[c] & 0x80; /* hi bit */
                        b[c] = r[c] << 1;
                        if(h == 0x80)
                                b[c] ^= 0x1b; /* Rijndael's Galois field */
                }
                r[0] = b[0] ^ a[3] ^ a[2] ^ b[1] ^ a[1]; /* 2 * a0 + a3 + a2 + 3 * a1 */
                r[1] = b[1] ^ a[0] ^ a[3] ^ b[2] ^ a[2]; /* 2 * a1 + a0 + a3 + 3 * a2 */
                r[2] = b[2] ^ a[1] ^ a[0] ^ b[3] ^ a[3]; /* 2 * a2 + a1 + a0 + 3 * a3 */
                r[3] = b[3] ^ a[2] ^ a[1] ^ b[0] ^ a[0]; /* 2 * a3 + a2 + a1 + 3 * a0 */
}
```

Constantes de iteración para Expansión de claves: 32 bits correspondientes a: $x^{\text{iteración-1}}$, seguido de 24 bits nulos.

2. El programa debe solicitar:

   - Clave (16 bytes en hexadecimal)
   - Bloque de Texto Original (16 bytes en hexadecimal)

   Debe devolver:

   - Las 10 subclaves
   - El Estado resultado de cada iteración
   - Bloque de Texto Cifrado (16 bytes en hexadecimal)

   Ejemplo:
   Clave: 000102030405060708090a0b0c0d0e0f
   Bloque de Texto Original: 00112233445566778899aabbccddeeff

   R0 (Subclave = 000102030405060708090a0b0c0d0e0f) = 00102030405060708090a0b0c0d0e0f0
   R1 (Subclave = d6aa74fdd2af72fadaa678f1d6ab76fe) = 89d810e8855ace682d1843d8cb128fe4
   R2 (Subclave = b692cf0b643dbdf1be9bc5006830b3fe) = 4915598f55e5d7a0daca94fa1f0a63f7
   R3 (Subclave = b6ff744ed2c2c9bf6c590cbf0469bf41) = fa636a2825b339c940668a3157244d17
   R4 (Subclave = 47f7f7bc95353e03f96c32bcfd058dfd) = 247240236966b3fa6ed2753288425b6c
   R5 (Subclave = 3caaa3e8a99f9deb50f3af57adf622aa) = c81677bc9b7ac93b25027992b0261996
   R6 (Subclave = 5e390f7df7a69296a7553dc10aa31f6b) = c62fe109f75eedc3cc79395d84f9cf5d
   R7 (Subclave = 14f9701ae35fe28c440adf4d4ea9c026) = d1876c0f79c4300ab45594add66ff41f
   R8 (Subclave = 47438735a41c65b9e016baf4aebf7ad2) = fde3bad205e5d0d73547964ef1fe37f1
   R9 (Subclave = 549932d1f08557681093ed9cbe2c974e) = bd6e7c3df2b5779e0b61216e8b10b689
   R10 (Subclave = 13111d7fe3944a17f307a78b4d2b30c5) = 69c4e0d86a7b0430d8cdb78070b4c55a

   Bloque de Texto Cifrado: 69c4e0d86a7b0430d8cdb78070b4c55a