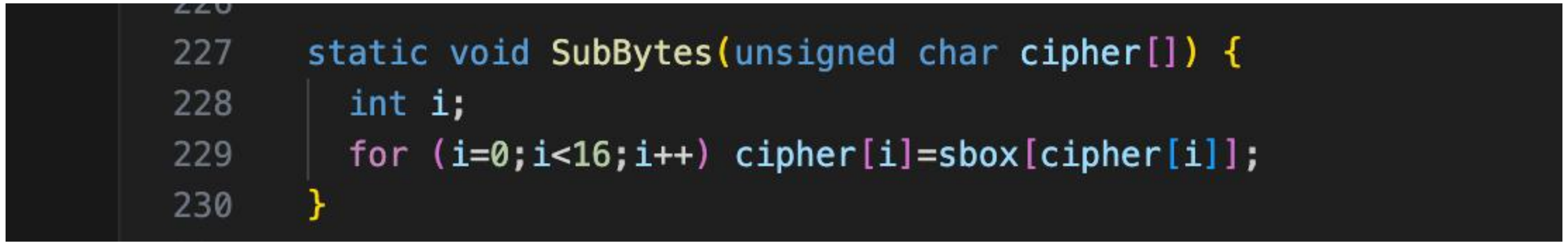
AES Implementation

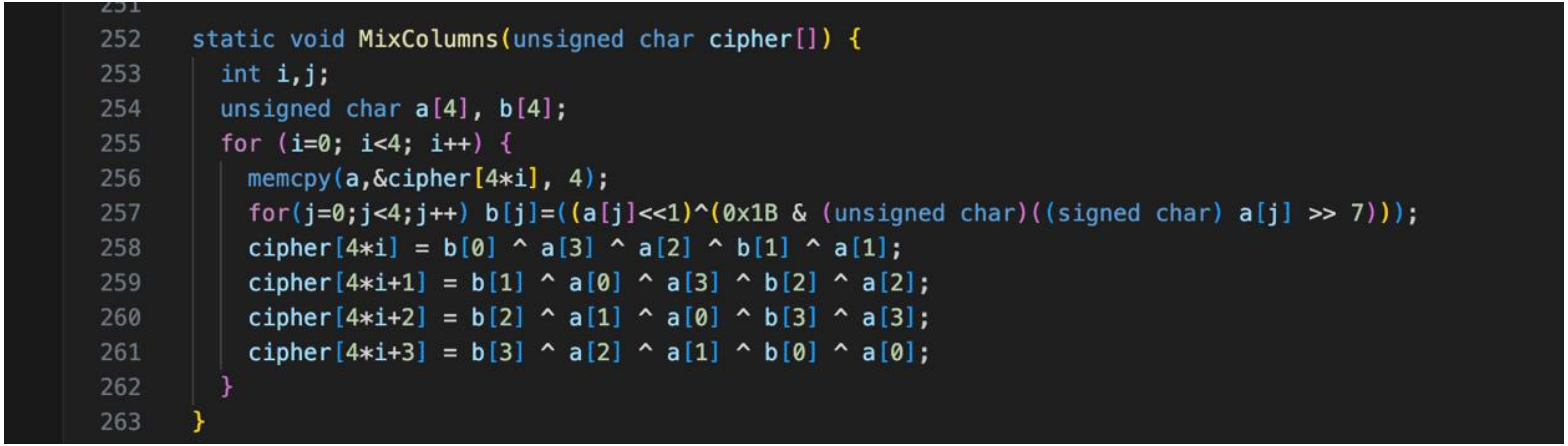
Applied Cryptography - 6240

TEAM
Ramnarayanan Sankar (801409708)
Poojitha Jayareddygari (801426875)
Shashank Kolluru (801421839)

Source Code before changes - Encryption

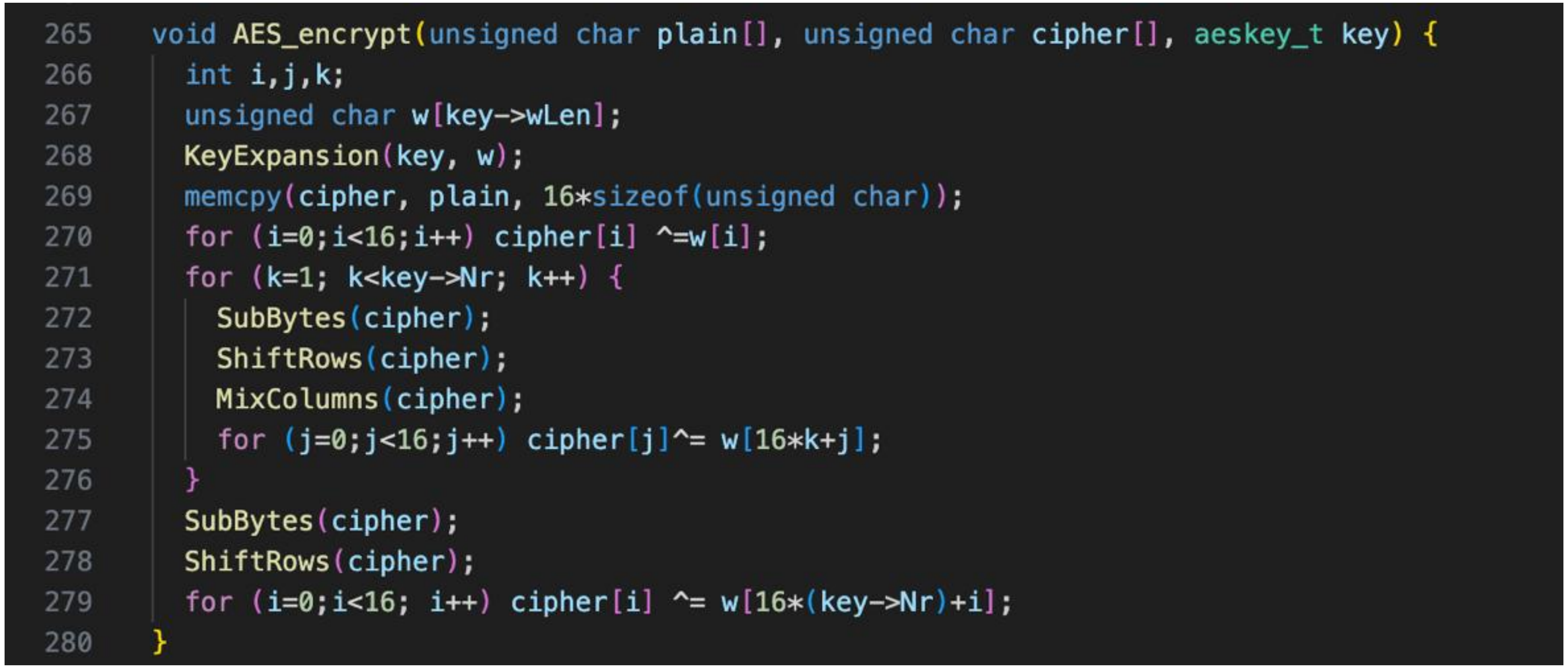


```
static void SubBytes(unsigned char cipher[]) {
  int i;
  for (i=0;i<16;i++) cipher[i]=sbox[cipher[i]];
}
```



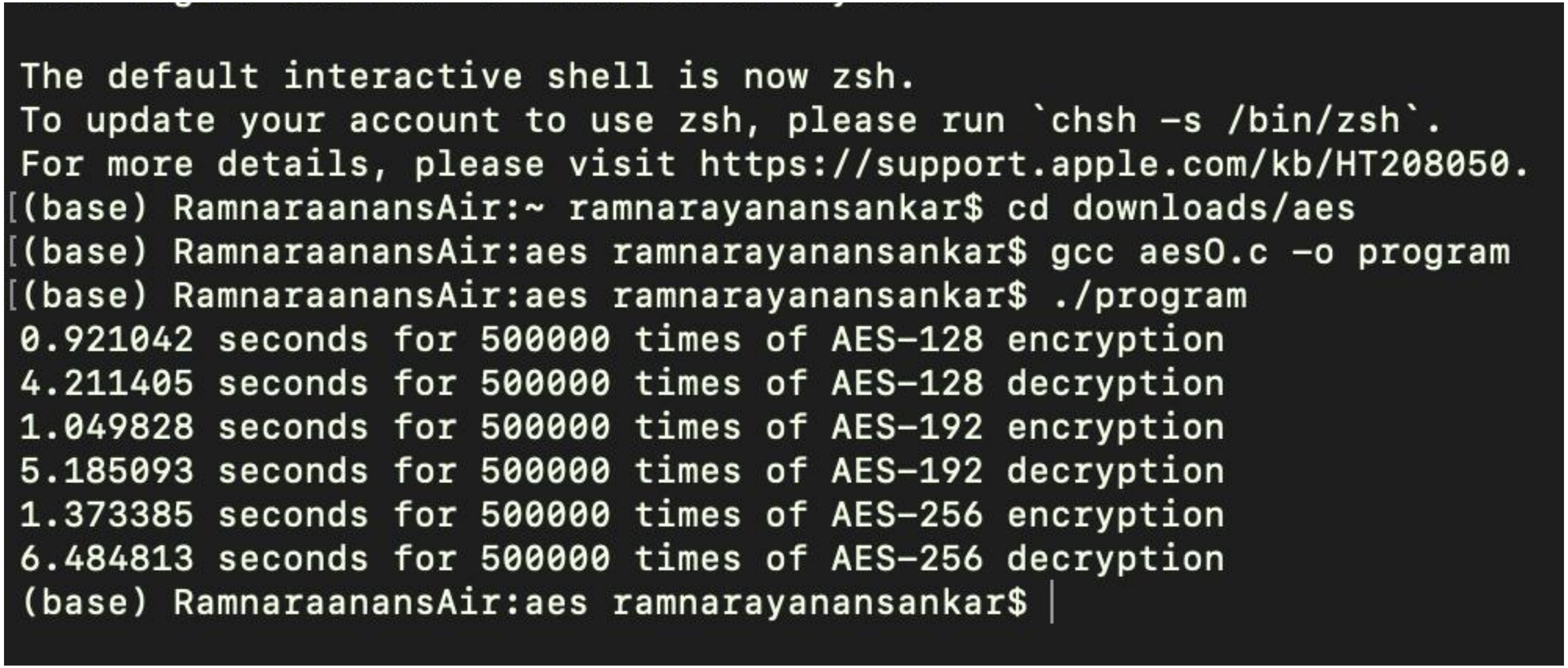
```
static void MixColumns(unsigned char cipher[]) {
  int i,j;
  unsigned char a[4], b[4];
  for (i=0; i<4; i++) {
    memcpy(a,&cipher[4*i], 4);
    for(j=0;j<4;j++) b[j]=((a[j]<<1)^(0x1B & (unsigned char)((signed char) a[j] >> 7)));
    cipher[4*i] = b[0] ^ a[3] ^ a[2] ^ b[1] ^ a[1];
    cipher[4*i+1] = b[1] ^ a[0] ^ a[3] ^ b[2] ^ a[2];
    cipher[4*i+2] = b[2] ^ a[1] ^ a[0] ^ b[3] ^ a[3];
    cipher[4*i+3] = b[3] ^ a[2] ^ a[1] ^ b[0] ^ a[0];
  }
}
```

Source Code before changes - Encryption



```c
void AES_encrypt(unsigned char plain[], unsigned char cipher[], aeskey_t key) {
  int i,j,k;
  unsigned char w[key->wLen];
  KeyExpansion(key, w);
  memcpy(cipher, plain, 16*sizeof(unsigned char));
  for (i=0;i<16;i++) cipher[i] ^=w[i];
  for (k=1; k<key->Nr; k++) {
    SubBytes(cipher);
    ShiftRows(cipher);
    MixColumns(cipher);
    for (j=0;j<16;j++) cipher[j]^= w[16*k+j];
  }
  SubBytes(cipher);
  ShiftRows(cipher);
  for (i=0;i<16; i++) cipher[i] ^= w[16*(key->Nr)+i];
}
```

Original RUN TIME before changes Encryption and Decryption



```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[(base) RamnaraanansAir:~ ramnarayanansankar$ cd downloads/aes
[(base) RamnaraanansAir:aes ramnarayanansankar$ gcc aesO.c -o program
[(base) RamnaraanansAir:aes ramnarayanansankar$ ./program
0.921042 seconds for 500000 times of AES-128 encryption
4.211405 seconds for 500000 times of AES-128 decryption
1.049828 seconds for 500000 times of AES-192 encryption
5.185093 seconds for 500000 times of AES-192 decryption
1.373385 seconds for 500000 times of AES-256 encryption
6.484813 seconds for 500000 times of AES-256 decryption
(base) RamnaraanansAir:aes ramnarayanansankar$ |
```

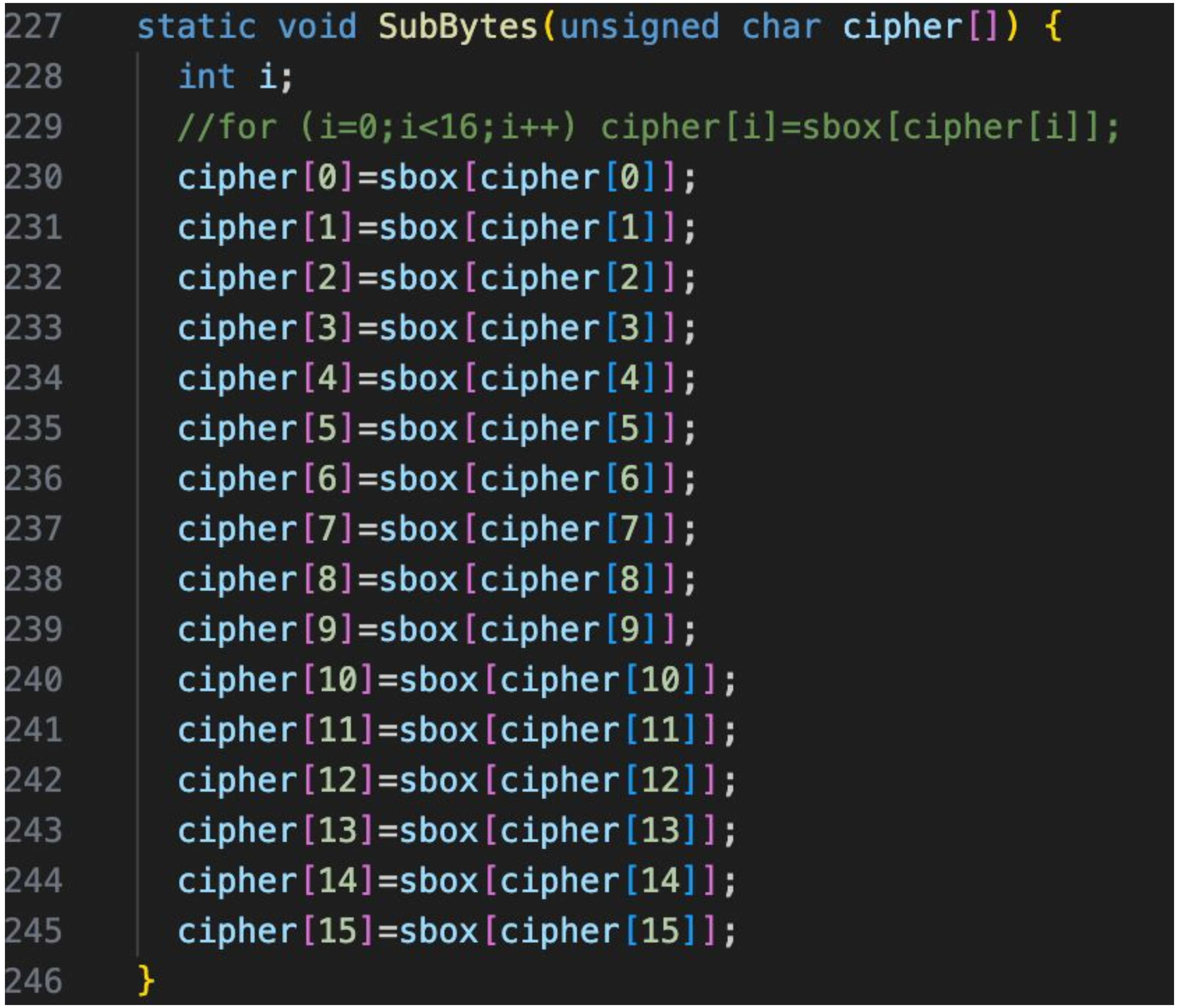
Original Source code Run time

Explanation

Source code changes for Encryption and Decryption:

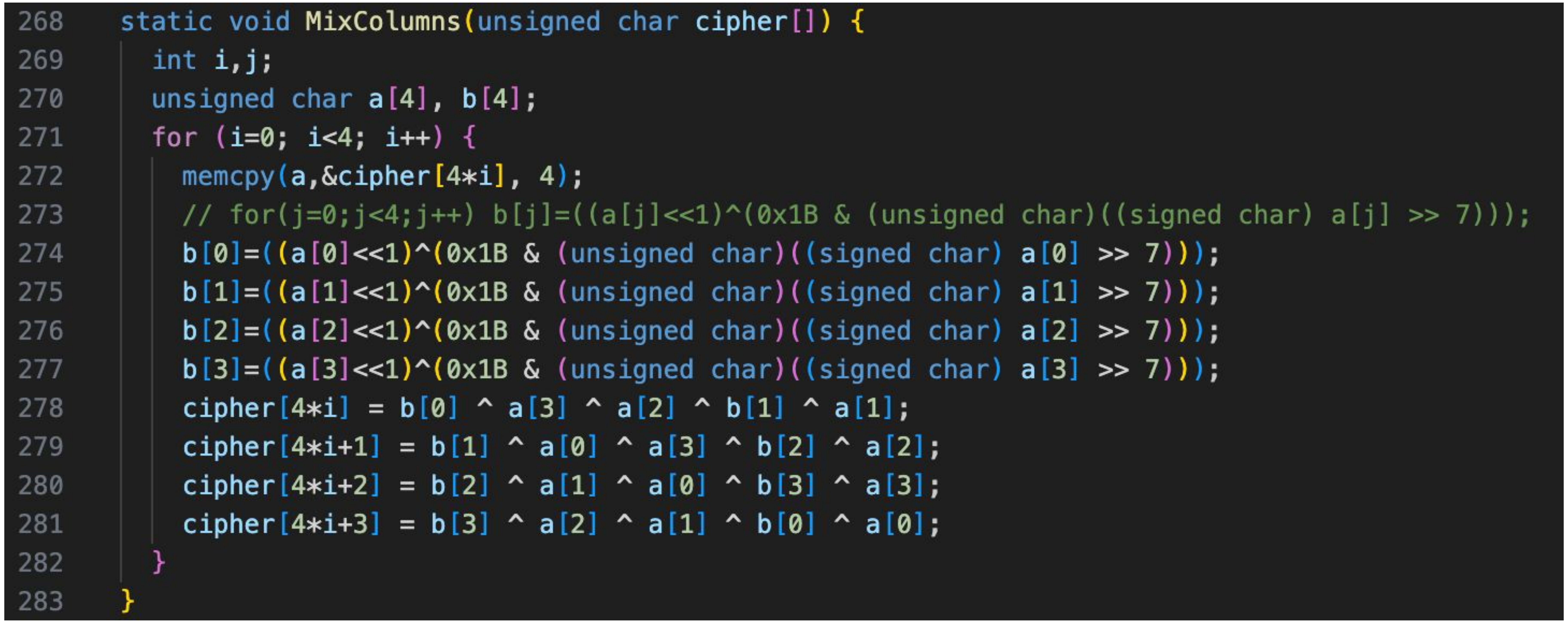
We eliminated all 'for' loops within the SubBytes, MixColumns, and AES_Encrypt blocks, opting instead to assign values directly to the respective variables. This approach aims to improve efficiency and streamline the execution of these blocks.

Source Code after changes - Encryption



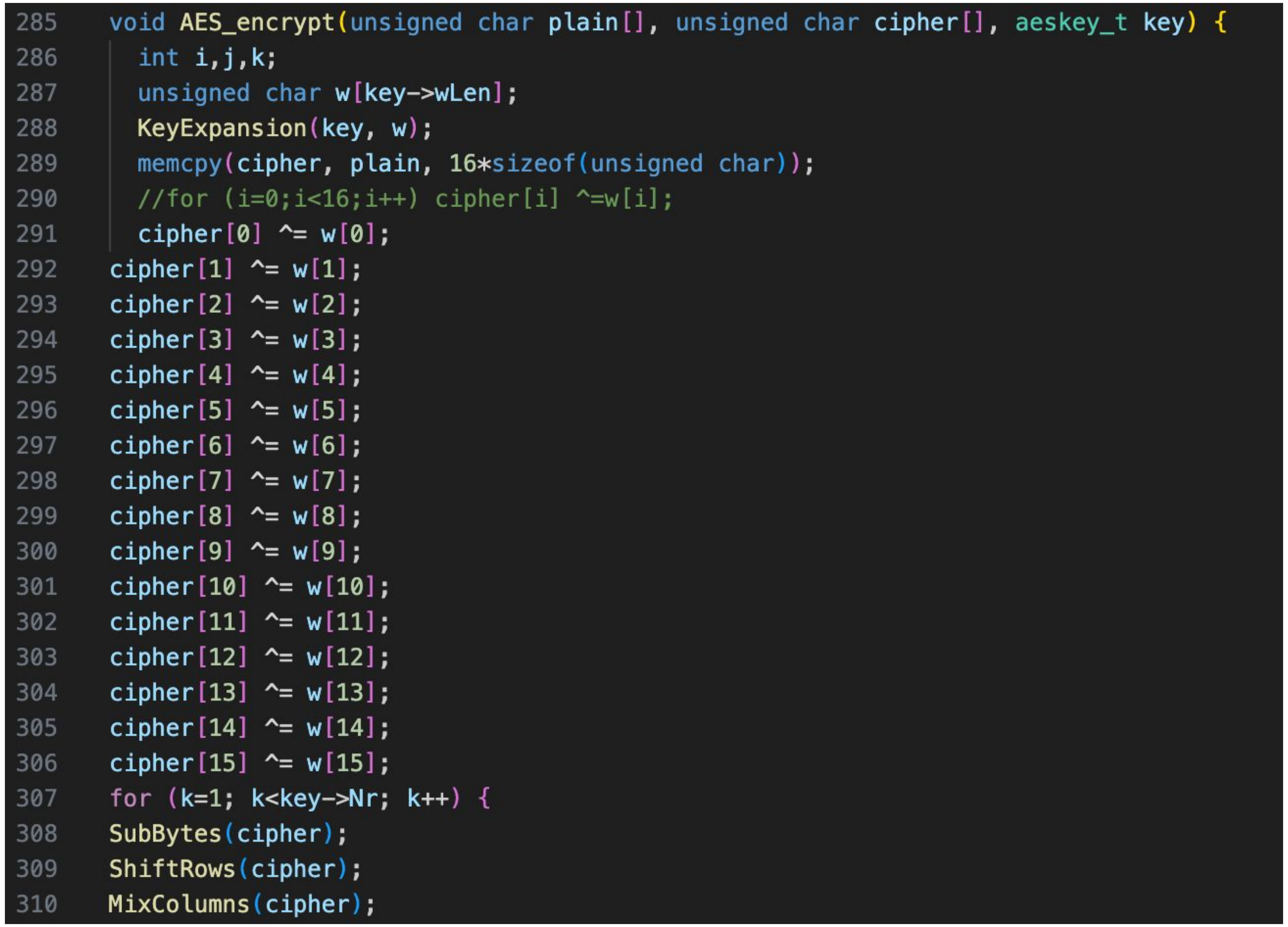
```c
static void SubBytes(unsigned char cipher[]) {
    int i;
    //for (i=0;i<16;i++) cipher[i]=sbox[cipher[i]];
    cipher[0]=sbox[cipher[0]];
    cipher[1]=sbox[cipher[1]];
    cipher[2]=sbox[cipher[2]];
    cipher[3]=sbox[cipher[3]];
    cipher[4]=sbox[cipher[4]];
    cipher[5]=sbox[cipher[5]];
    cipher[6]=sbox[cipher[6]];
    cipher[7]=sbox[cipher[7]];
    cipher[8]=sbox[cipher[8]];
    cipher[9]=sbox[cipher[9]];
    cipher[10]=sbox[cipher[10]];
    cipher[11]=sbox[cipher[11]];
    cipher[12]=sbox[cipher[12]];
    cipher[13]=sbox[cipher[13]];
    cipher[14]=sbox[cipher[14]];
    cipher[15]=sbox[cipher[15]];
}
```

Source Code after changes

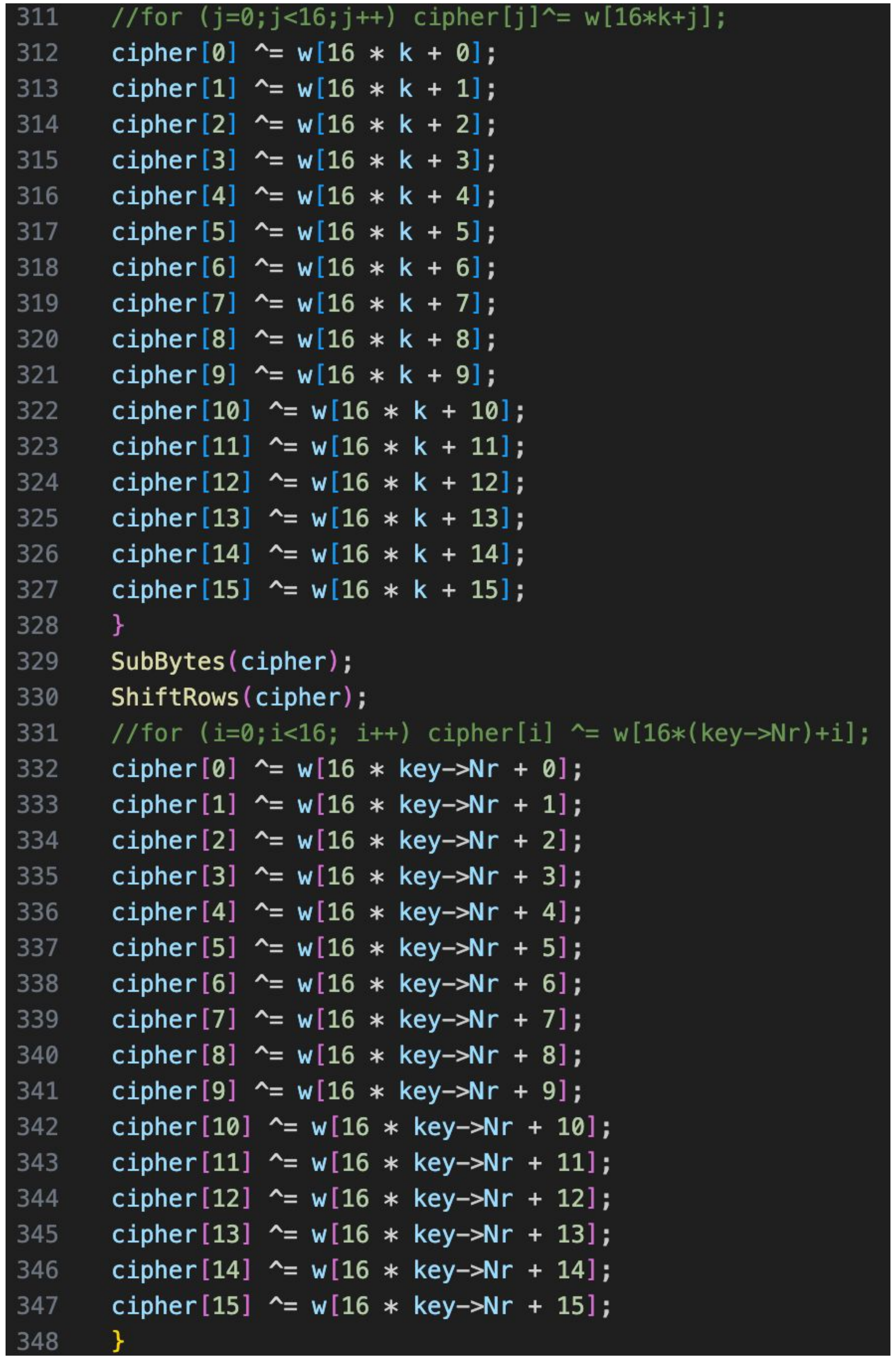


```c
static void MixColumns(unsigned char cipher[]) {
  int i,j;
  unsigned char a[4], b[4];
  for (i=0; i<4; i++) {
    memcpy(a,&cipher[4*i], 4);
    // for(j=0;j<4;j++) b[j]=((a[j]<<1)^(0x1B & (unsigned char)((signed char) a[j] >> 7)));
    b[0]=((a[0]<<1)^(0x1B & (unsigned char)((signed char) a[0] >> 7)));
    b[1]=((a[1]<<1)^(0x1B & (unsigned char)((signed char) a[1] >> 7)));
    b[2]=((a[2]<<1)^(0x1B & (unsigned char)((signed char) a[2] >> 7)));
    b[3]=((a[3]<<1)^(0x1B & (unsigned char)((signed char) a[3] >> 7)));
    cipher[4*i] = b[0] ^ a[3] ^ a[2] ^ b[1] ^ a[1];
    cipher[4*i+1] = b[1] ^ a[0] ^ a[3] ^ b[2] ^ a[2];
    cipher[4*i+2] = b[2] ^ a[1] ^ a[0] ^ b[3] ^ a[3];
    cipher[4*i+3] = b[3] ^ a[2] ^ a[1] ^ b[0] ^ a[0];
  }
}
```

Source Code after changes

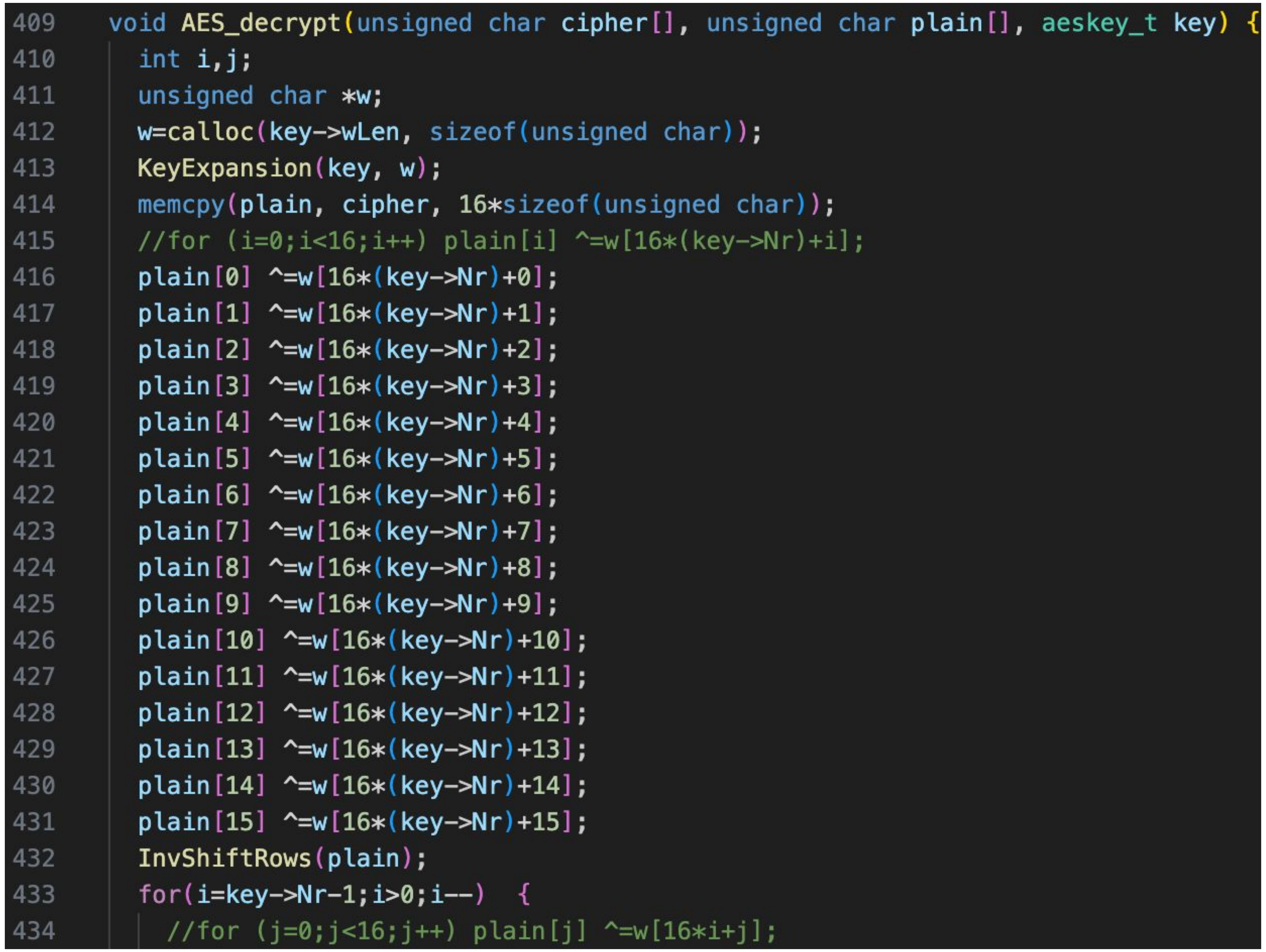


```
void AES_encrypt(unsigned char plain[], unsigned char cipher[], aeskey_t key) {
    int i,j,k;
    unsigned char w[key->wLen];
    KeyExpansion(key, w);
    memcpy(cipher, plain, 16*sizeof(unsigned char));
    //for (i=0;i<16;i++) cipher[i] ^=w[i];
    cipher[0] ^= w[0];
cipher[1] ^= w[1];
cipher[2] ^= w[2];
cipher[3] ^= w[3];
cipher[4] ^= w[4];
cipher[5] ^= w[5];
cipher[6] ^= w[6];
cipher[7] ^= w[7];
cipher[8] ^= w[8];
cipher[9] ^= w[9];
cipher[10] ^= w[10];
cipher[11] ^= w[11];
cipher[12] ^= w[12];
cipher[13] ^= w[13];
cipher[14] ^= w[14];
cipher[15] ^= w[15];
for (k=1; k<key->Nr; k++) {
SubBytes(cipher);
ShiftRows(cipher);
MixColumns(cipher);
```

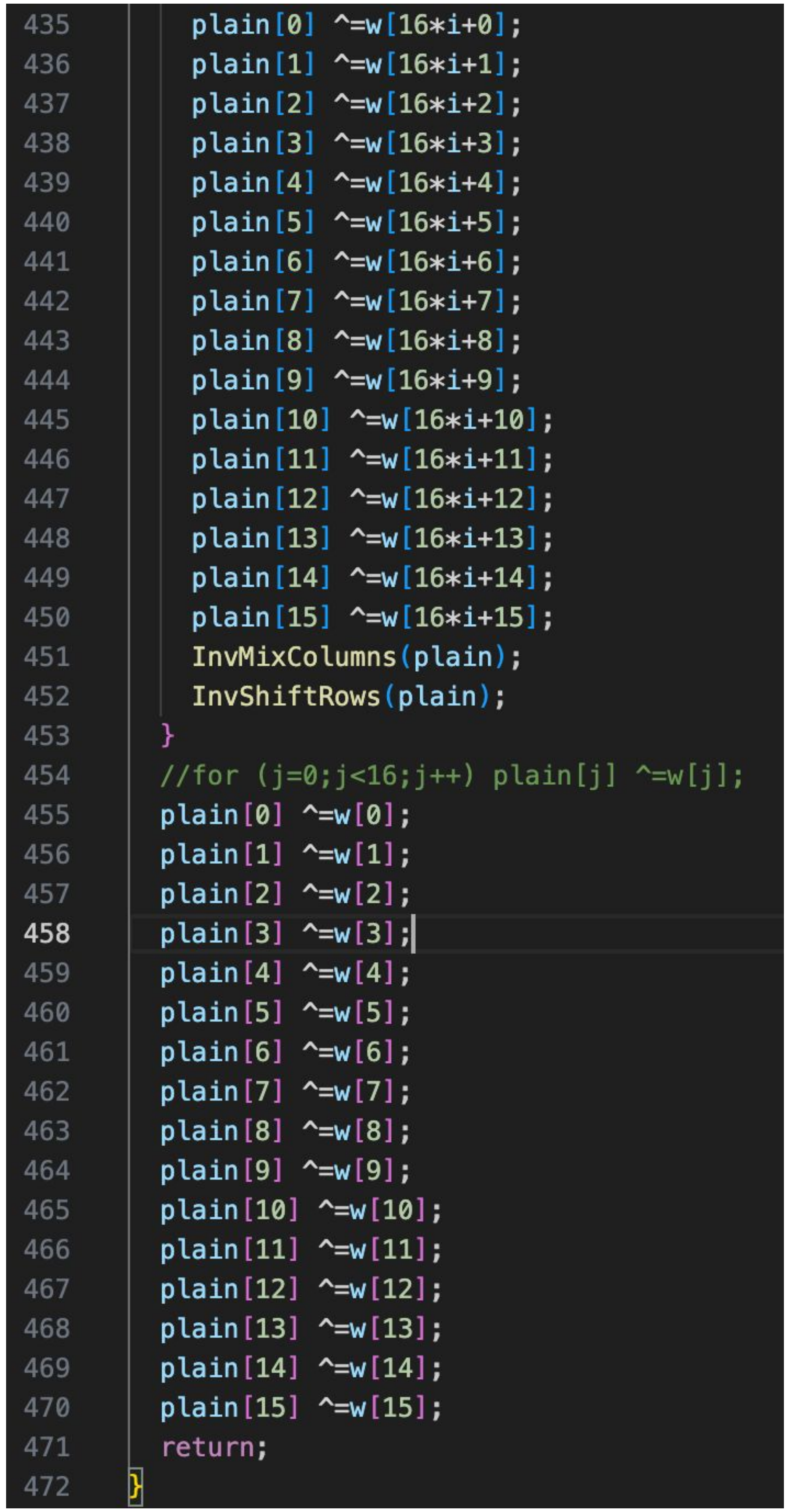


```c
//for (j=0;j<16;j++) cipher[j]^= w[16*k+j];
cipher[0] ^= w[16 * k + 0];
cipher[1] ^= w[16 * k + 1];
cipher[2] ^= w[16 * k + 2];
cipher[3] ^= w[16 * k + 3];
cipher[4] ^= w[16 * k + 4];
cipher[5] ^= w[16 * k + 5];
cipher[6] ^= w[16 * k + 6];
cipher[7] ^= w[16 * k + 7];
cipher[8] ^= w[16 * k + 8];
cipher[9] ^= w[16 * k + 9];
cipher[10] ^= w[16 * k + 10];
cipher[11] ^= w[16 * k + 11];
cipher[12] ^= w[16 * k + 12];
cipher[13] ^= w[16 * k + 13];
cipher[14] ^= w[16 * k + 14];
cipher[15] ^= w[16 * k + 15];
}
SubBytes(cipher);
ShiftRows(cipher);
//for (i=0;i<16; i++) cipher[i] ^= w[16*(key->Nr)+i];
cipher[0] ^= w[16 * key->Nr + 0];
cipher[1] ^= w[16 * key->Nr + 1];
cipher[2] ^= w[16 * key->Nr + 2];
cipher[3] ^= w[16 * key->Nr + 3];
cipher[4] ^= w[16 * key->Nr + 4];
cipher[5] ^= w[16 * key->Nr + 5];
cipher[6] ^= w[16 * key->Nr + 6];
cipher[7] ^= w[16 * key->Nr + 7];
cipher[8] ^= w[16 * key->Nr + 8];
cipher[9] ^= w[16 * key->Nr + 9];
cipher[10] ^= w[16 * key->Nr + 10];
cipher[11] ^= w[16 * key->Nr + 11];
cipher[12] ^= w[16 * key->Nr + 12];
cipher[13] ^= w[16 * key->Nr + 13];
cipher[14] ^= w[16 * key->Nr + 14];
cipher[15] ^= w[16 * key->Nr + 15];
}
```

Source Code after changes - Decryption



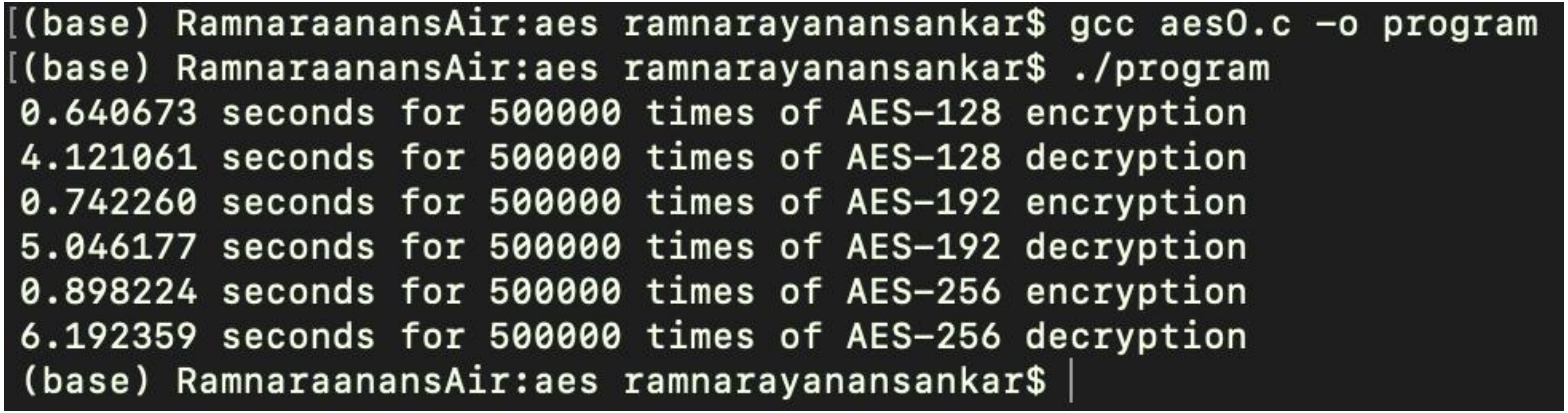
```c
void AES_decrypt(unsigned char cipher[], unsigned char plain[], aeskey_t key) {
    int i,j;
    unsigned char *w;
    w=calloc(key->wLen, sizeof(unsigned char));
    KeyExpansion(key, w);
    memcpy(plain, cipher, 16*sizeof(unsigned char));
    //for (i=0;i<16;i++) plain[i] ^=w[16*(key->Nr)+i];
    plain[0] ^=w[16*(key->Nr)+0];
    plain[1] ^=w[16*(key->Nr)+1];
    plain[2] ^=w[16*(key->Nr)+2];
    plain[3] ^=w[16*(key->Nr)+3];
    plain[4] ^=w[16*(key->Nr)+4];
    plain[5] ^=w[16*(key->Nr)+5];
    plain[6] ^=w[16*(key->Nr)+6];
    plain[7] ^=w[16*(key->Nr)+7];
    plain[8] ^=w[16*(key->Nr)+8];
    plain[9] ^=w[16*(key->Nr)+9];
    plain[10] ^=w[16*(key->Nr)+10];
    plain[11] ^=w[16*(key->Nr)+11];
    plain[12] ^=w[16*(key->Nr)+12];
    plain[13] ^=w[16*(key->Nr)+13];
    plain[14] ^=w[16*(key->Nr)+14];
    plain[15] ^=w[16*(key->Nr)+15];
    InvShiftRows(plain);
    for(i=key->Nr-1;i>0;i--)  {
        //for (j=0;j<16;j++) plain[j] ^=w[16*i+j];
```



```
    plain[0] ^=w[16*i+0];
    plain[1] ^=w[16*i+1];
    plain[2] ^=w[16*i+2];
    plain[3] ^=w[16*i+3];
    plain[4] ^=w[16*i+4];
    plain[5] ^=w[16*i+5];
    plain[6] ^=w[16*i+6];
    plain[7] ^=w[16*i+7];
    plain[8] ^=w[16*i+8];
    plain[9] ^=w[16*i+9];
    plain[10] ^=w[16*i+10];
    plain[11] ^=w[16*i+11];
    plain[12] ^=w[16*i+12];
    plain[13] ^=w[16*i+13];
    plain[14] ^=w[16*i+14];
    plain[15] ^=w[16*i+15];
    InvMixColumns(plain);
    InvShiftRows(plain);
  }
  //for (j=0;j<16;j++) plain[j] ^=w[j];
  plain[0] ^=w[0];
  plain[1] ^=w[1];
  plain[2] ^=w[2];
  plain[3] ^=w[3];
  plain[4] ^=w[4];
  plain[5] ^=w[5];
  plain[6] ^=w[6];
  plain[7] ^=w[7];
  plain[8] ^=w[8];
  plain[9] ^=w[9];
  plain[10] ^=w[10];
  plain[11] ^=w[11];
  plain[12] ^=w[12];
  plain[13] ^=w[13];
  plain[14] ^=w[14];
  plain[15] ^=w[15];
  return;
}
```

RUN TIME

Modified Run Time



```
[(base) RamnaraanansAir:aes ramnarayanansankar$ gcc aesO.c -o program
[(base) RamnaraanansAir:aes ramnarayanansankar$ ./program
0.640673 seconds for 500000 times of AES-128 encryption
4.121061 seconds for 500000 times of AES-128 decryption
0.742260 seconds for 500000 times of AES-192 encryption
5.046177 seconds for 500000 times of AES-192 decryption
0.898224 seconds for 500000 times of AES-256 encryption
6.192359 seconds for 500000 times of AES-256 decryption
(base) RamnaraanansAir:aes ramnarayanansankar$ |
```

Modified Run time for encryption and decryption

THANKYOU