Elizabeth Arikawa

Network Security

Advanced Encryption Standard (AES)

1. **Key and Message Selection**
   1. The key is a 128 bit array of data given at the beginning of the main method. Each byte is given in hex.
   2. The message to be encrypted is also a 128 bit array given at the beginning of the main method. Similar to the key, the message is encoded in hex
2. **Key Expansion**
   1. Before the encryption can begin, the 16 byte key previously given must be expanded to 176 bytes, 4 for each round plus one extra.
   2. The first round key is simple since it just matches the initial key given in the main method. But for following rounds, the key must be manipulated.
   3. These expanded keys rely on previous keys to be created using the formula:

Except when R mod 4 == 0, where the following formula determines the expanded key

* + 1. Sub word uses an array SBox to substitute the current byte for the corresponding one in the sbox.
    2. RotWord shifts all the bytes in the word one to the right.
    3. Rcon
    4. EK (Offset) returns the 4 bytes of the expanded key after the given offset. So if given EK(4) then EK(4), EK(5), EK(6) and EK(7) would be returned.
    5. XOR refers to exclusive or operator which operates in the following way:
       1. 0 XOR 0 = 0
       2. 1 XOR 0 = 1
       3. 0 XOR 1 = 1
       4. 1 XOR 1 = 0
    6. R refers to the current round. There are 44 rounds to find the expanded keys.

1. **Encryption**
   1. To represent the message through the encryption process I used a 4 by 4 integer matrix. This allows for easier manipulation of it.
   2. For a 16 bit key and message there are ten rounds to encrypt the message. The four steps during each round except the tenth are byte substitution, shift row, mix columns and add round key.
      1. Byte Substitution
         1. This step uses the sbox mentioned previously to change the current bytes of the state matrix to their relative ones from the sbox.
      2. Shift Rows
         1. This step shifts each of the rows to the right such that the first row is unchanged, the second moves one, the third moves two and the fourth moves three.

{1, 2, 3, 4}

{2, 3, 4, 1}

{3, 4, 1, 2}

{4, 1, 2, 3}

After Shift Rows

{1, 2, 3, 4}

{1, 2, 3, 4}

{1, 2, 3, 4}

{1, 2, 3, 4}

Original

* + 1. Column Mixing
       1. This step is a bit more complicated than the others. The state matrix is multiplied by a constant matrix over a Galois Field.

{1, 2, 3, 4}

{1, 2, 3, 4}

{1, 2, 3, 4}

{1, 2, 3, 4}

State Matrix

{2, 3, 1, 1}

{1, 2, 3, 1}

{1, 1, 2, 3}

{3, 1, 1, 2}

Constant Matrix

* + - 1. To find b1 of the new state matrix:

Which the E and L tables can also be used to solve:

* + - * 1. E and L tables are lookup tables similar to sbox
      1. Lastly, if the resulting number is larger than 0xff we must subtract 0xff to make sure the number is the correct size.
    1. Add Round Key
       1. Adding the round key to the state is pretty straightforward. At the end of each round the expanded round key for the current round is added to the current state.

1. Results
   1. Key: 0 1 2 3 4 5 6 7 8 9 a b c d e f

Message: 0 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff

Text after encryption: 69 c4 e0 d8 6a 7b 04 30 d8 cd b7 80 70 b4 c5 5a

* 1. Key: ff ee dd cc bb aa aa bb cc dd ee ff ff ee dd cc

Message: 10 20 30 40 50 60 70 80 80 70 60 50 40 30 20 10

Text after encryption: 2c 9f fd d6 29 3f 6d d6 ba 18 3a 0c c5 2c a0 cf