Scribe: Cryptography and Network Security (Week6.Class5.A)

Juluri Shree Shiva Teja

10-Oct-2020

1 Introduction

This scribe mainly describes about the construction of Advanced Encryption Standard AES using Rijndael Algorithm.

2 Round functions of Rijndael algorithm

The plaintext is converted into bits (0 and 1) and 128 bits are given as input to cipher every time. The output and the intermediate state of the cipher also have 128 bits. The round functions involved in it are as follows-

2.1 Byte Sub

It is a Substitution box which is responsible for non-linear transformation in AES providing confusion. Here, the state indicates 128 bits. Each byte (8 bits) of state gets substituted using the S-box giving byte(8 bits) output. So, we require substitution computation 16 times for every round.

2.1.1 Multiplicative Inverse

We say that for every non-zero element a $\in$ GF(2^8), there is always a unique element b $\in$ GF(2^8) such that ab = 1 modulo (irreducible polynomial). Then we define b to be the multiplicative inverse of a. We define the multiplicative inverse of 0 to be 0 by definition.

2.1.2 Fixed points and Opposite fixed points

Let x be the input to S-box and S(x) be the output of S-box. Then for a specific input value of x, if
x $\oplus$ S(x) = 0 gives fixed point.
x $\oplus$ S(x) = FF gives opposite fixed point.
Occurence of fixed points or opposite fixed points can cause a loophole in the

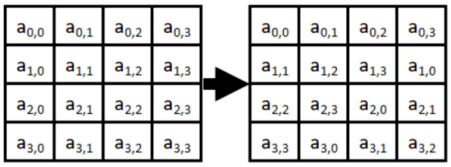
S-box leading to attack on algorithm. So, we aim for
$x \oplus S(x) \neq 0$, FF for all x.

2.1.3 S-box construction

S-box is constructed logically involving many properties including non-linearity (even for any linear combination of output vectors), bijective for invertibility, diffusion etc.
It is constructed using Affine mapping of the multiplicative inverse of input byte in GF(2^8) using constant matrices. Affine mapping is used for S-box inorder to eliminate fixed points or opposite fixed points.

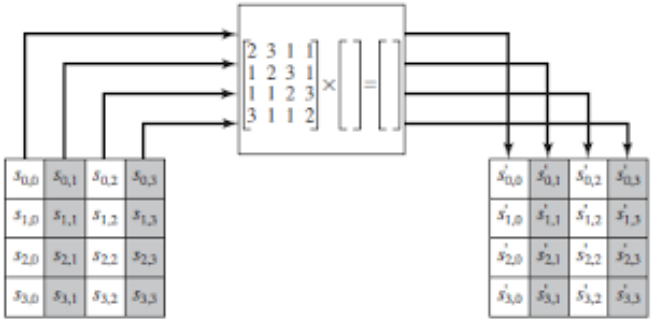
2.2 Shift Row

Each of the rows in the input state is shifted to the left by a set amount providing diffusion. The top row is not shifted at all, the next row is shifted by one, the third row is shifted by two and the last row is shifted by three.



2.3 Mix Columns

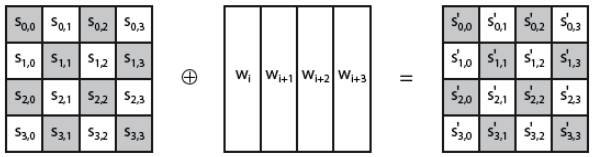
The input state is multiplied by constant matrix to produce the output state as per Galois Field multiplication in GF(2^8) providing diffusion. The coefficients of the constant matrix are designed such that the change in one byte of input data will change all the four bytes in the output data thus maximising the diffusion.



Note: For some disturbance given in a byte, Shift Row will propagate it among the row where as Mix Columns will propagate it among the columns. Repeated application of both will spread the disturbance all the over the ciphertext which is an essential property of the algorithm.

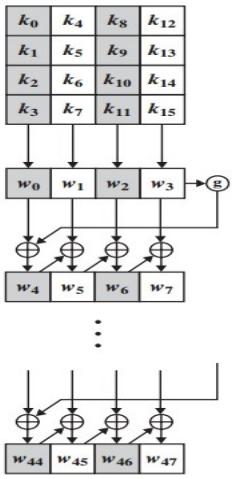
2.4 Add Round Key

In this AddRoundKey operation, the input to the round is exclusive-ored with the round key. It is the only phase of AES encryption that directly operates on the AES round key.



3 Key Scheduling Algorithm

The Key Scheduling Algorithm is used to produce a set number of round keys from the initial key. In AES, the initial key is used in the initial round as input to the AddRoundKey operation. In AES-128, the initial key is 128 bits in length and 10 round keys are produced as shown in the diagram. The g function includes S-box transformation, a permutation(left shift of each byte) and XOR. At each round, the round constant which is specific to the round number is XORed to obtain the round key.



4 AES Encryption

In AES-128, we have 10 rounds of encryption where each round includes Byte Sub, Shift Row, Mix Columns and Add Round Key except the last round. Before the first round, the actual plaintext is XORed with the initial key and the output is passed to the first round. The last round in the encryption has Byte Sub, Shift Row and Add Round Key where the Mix Columns is absent.
Removal of Mix Columns in last round of encryption will make both encryption and decryption exactly the same but with reverse order of round keys applied. This is because of the following-
1) Commutative property of Shift Row and Byte Sub: The output doesnt vary if the substitution is done before or after row shift because row shift only provides diffusion but not confusion and hence input byte to S-box remains the same.
2) Interchangeability of Add Round Key and Mix Columns: When the plaintext is XORed with key and given as input to Mix Columns, the output remains the same even if the plaintext undergoes Mix Columns operation and then XOR with equivalent key (obtained through Mix Columns on round key). This is because the linear transformation of Mix Columns can be pushed through an XOR.

5 Conclusion

The implementation of AES is convinient because the encryption and decryption functions are effectively the same though the key used has minor changes. During the actual implementation in the hardware, the possible operations like Byte Sub, Shift Row, Mix Columns are merged together to have the ease of accessing from combined lookup table. Encryption and decryption are as follows-