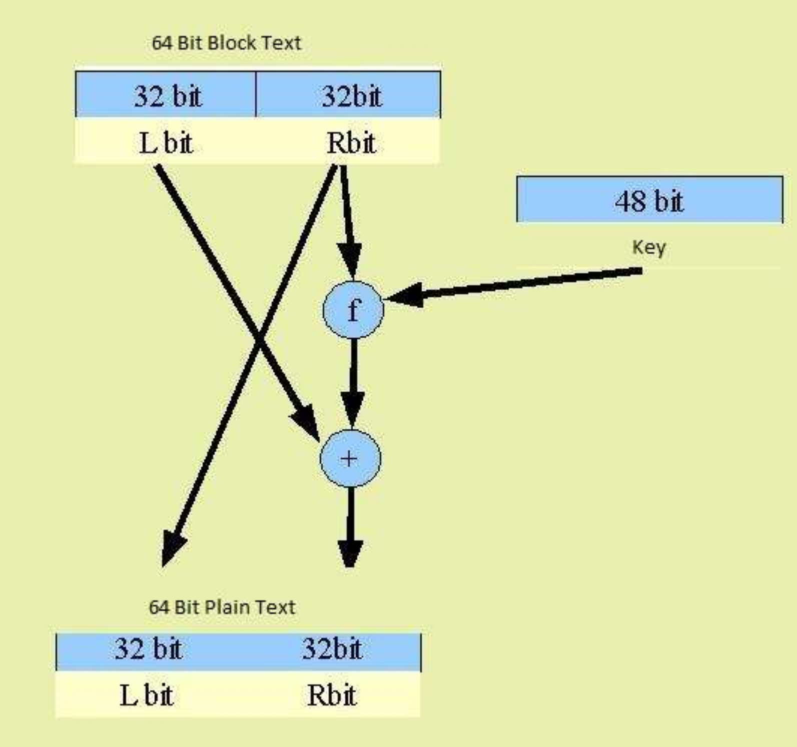
DES (Data Encryption Standard)

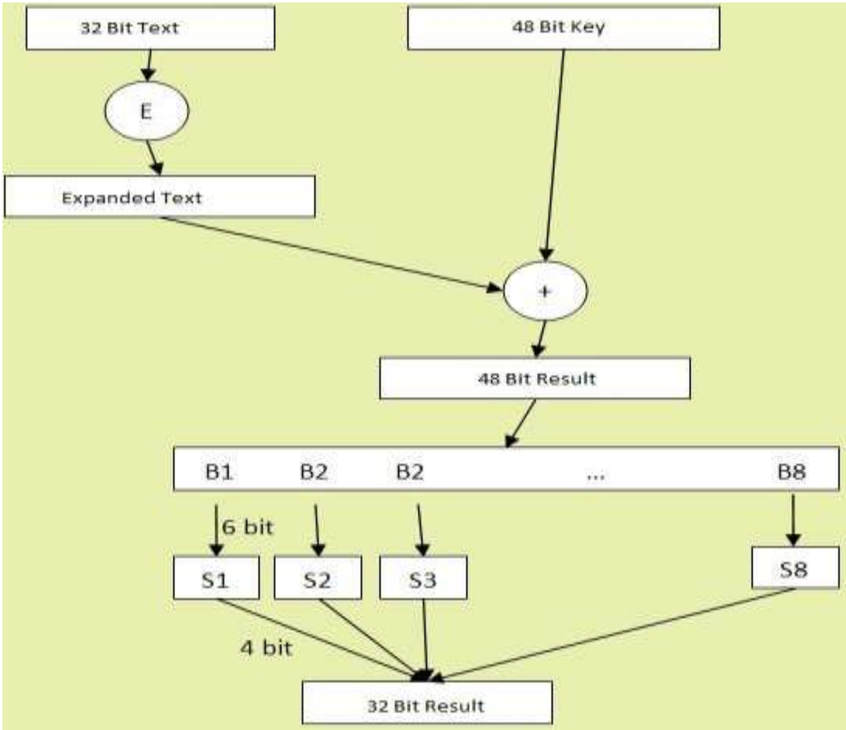
This algorithm has been developed with the improvement of LUCIFER algorithm. DES algorithm's encryption process is performed using the method called block encryption. Block cipher works by dividing plain text into blocks in certain lengths. All blocks divided according to this method will be encrypted separately and the resulting cipher text will be obtained by the arrangement of these blocks.

In the DES algorithm; the text to be encrypted is divided into 64-bit blocks. The algorithm also takes a key of length 64 bits. However, the 8 bit of this 64 bit length key is used as parity bit. Parity is used to understand whether the data received in serial signal communications is sent correctly. There are two kinds of parity. ODD (odd) and EVEN (even). If this total is double, the parity is bit 1 and the sum of 1s in the datum is equal to odd number.

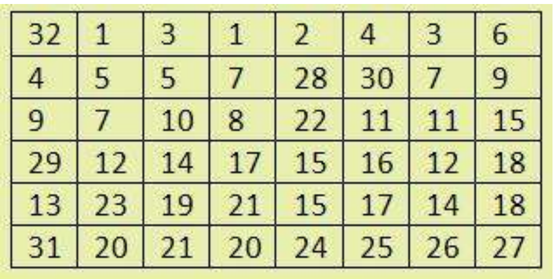


The 64-bit length block is divided into two equal parts. The remaining 32 bits of text on the right side are written directly to the left side of the message. Again with this 32 bit fragment on the right, the 48 bit key is inserted into the f function. The result is put into XOR operation with the 32-bit long piece from the left.

The key is actually 64 bits long. However, since the 8-bit portion of the key is used as a parity, the length of the key drops to 56 bits. After the length of the key has dropped to 56 bits; the key is inserted into the function of key generator and its length is reduced to 48 bits. In this case, a different key is provided for each operation of the algorithm.



When we analyze the f function; we see that the first step is to use the expansion table to extract the 48 bits from the 32 bits. There are various expansion tables used for this process.



32	1	3	1	2	4	3	6
4	5	5	7	28	30	7	9
9	7	10	8	22	11	11	15
29	12	14	17	15	16	12	18
13	23	19	21	15	17	14	18
31	20	21	20	24	25	26	27

Sample Expansion Table

The message, which is 48 bits long, is processed by XOR operation with the 48 bit key. The resulting 48-bit result is divided into 8 parts. So we have 6 bits of messages in 8 pieces. These 6-bit lengths enter the boxes called S-Box (Substitution box). These boxes generates 4 bit length results from the 6 bit length entries. In an example S-Box table; the first and last bits can correspond as the row, the bits in the middle may correspond to the column number. In such a case, the position information corresponding to the coordinates is taken and the result is given. Finally, the 4 bit length pieces are written next to each other to produce the final result.

A key generator that reduces the 56-bit key to 48 bits; each time selects different parts over the key of length 56 bits. In this case different keys are used in each iteration of the algorithm. Additionally, this algorithm works with the Feistel network principle.

All these operations are repeated 16 times. This makes difficult to resolve messages. To decrypt a text encrypted with DES, it is enough to give the same algorithm with the same key as the cipher text.

After breaking the DES algorithm with brute force methods, a more reliable 3-DES algorithm has been developed. This method works by performing DES ciphering three times in a row. The length of the key is 168 bits. The data is encrypted with the first 8 bytes of the 3-DES algorithm's key. Then the data is decoded by the middle 8 bytes of the key. Finally, the key is encrypted with the last 8 bytes, resulting in an 8-byte block.

DES algorithm and its derivatives; it is not used very much today due to such issues as security vulnerability and speed.

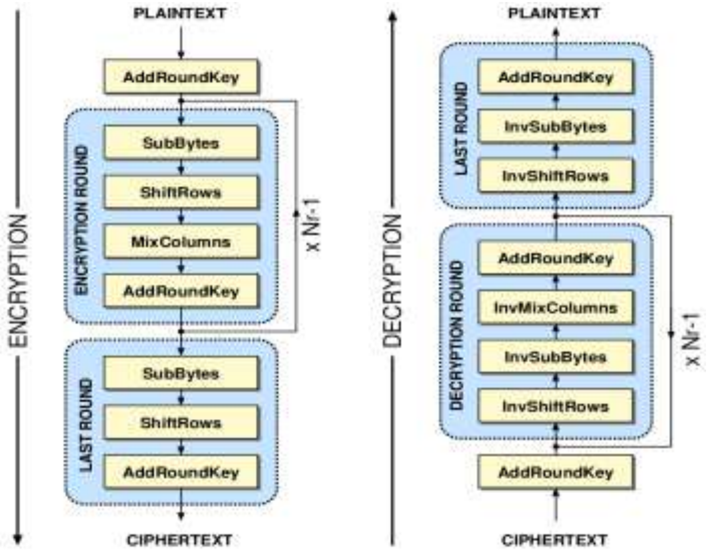
AES (Advanced Encryption Standard)

Once it was realized that the DES algorithm could be broken by brute force, a new encryption standard was needed. AES was developed as a result of the studies carried out for this purpose.

AES uses 128-bit-length blocks for encryption. This algorithm can use 128,192 or 256 bit length keys. This algorithm uses a symmetric cryptography which is a method the key must be known by both parties. Unlike DES, this algorithm does not use the Feistel network system. It has significant advantages in terms of security, speed, flexibility and memory.

This algorithm works at if the 128-bit key is 10 repetition, if the 192-bit key is 12 repetition and if the 256-bit key is 14 repetition.

The operations that the algorithm uses during its operation are shown on the right side. After the number of repetitions is determined according to the key length, sub-bytes, shift rows, mix columns and add round key operations are applied sequentially for n-1 rounds. In the last round, these operations are performed except for the key addition process. In the case of decryption, this procedure is applied from the reverse side.

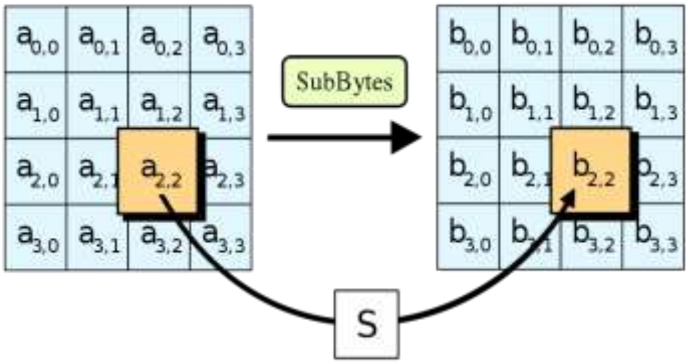


After 10 rounds, the incoming data comes out encrypted. In the first round, the key is joined as the first position, while in other rounds the newly produced keys are inserted.

The AES algorithm has 128 bits of input, output and matrices. The matrix consists of 4 rows, 4 columns (4 × 4), and 16 divisions. This matrices is called 'state'. One byte of data is dropped in each partition of the state. It brings up a 32-bit word in every row.

The text to be encrypted is divided into 128-bit long blocks. Each block is placed in the state matrix. In the same way, this is handled as a matrix in the 128-bit key already received.

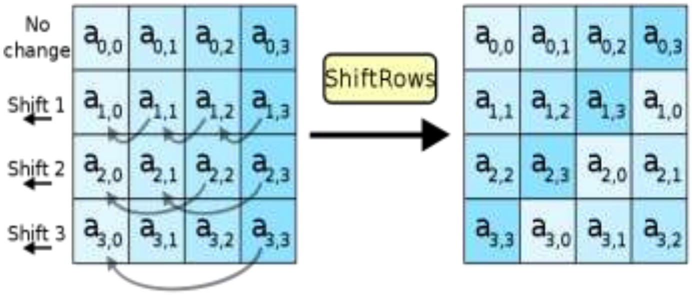
In the Sub-Byte step, the value of each byte in the state matrix is updated using an 8-bit substitution box. The status matrix contains one byte, or 8 bits, in each cell. This 8-bit portion is divided into two 4-bit portions. Then, the values of these 4-bit portions are calculated separately. Then the corresponding values are found on the S-Box from the row-column coordinates corresponding to these values. Each cell of the S-Box also has 8-bit values.


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
S
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

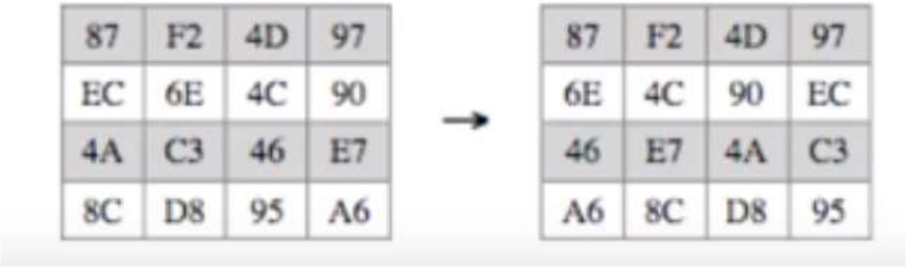
Sub – Byte Operation

For example, we assume that a cell of the state matrix has bits 10101101. We divide these 8 bits into two parts as 1010 and 1101. Then we compute the decimal values of these binary values. 1010 = 10, 1101 = 13. Then we find the value at S-Box's 10.row and 13.column. And we use this value for next process.

In the shift row step, the bytes of each row on the rows of the status matrix are shifted by a certain number. The first line is not changed, the second line is shifted left 1, the third line is shifted left 2, and the last line is shifted left 3. The overhanging divisions are added to the beginning of the shift.


No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

Shift Row Operation



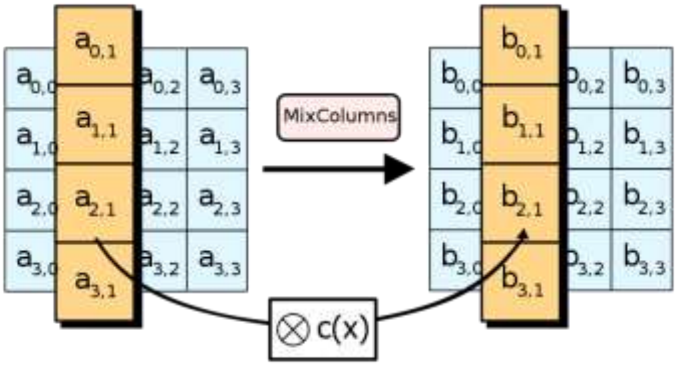
87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

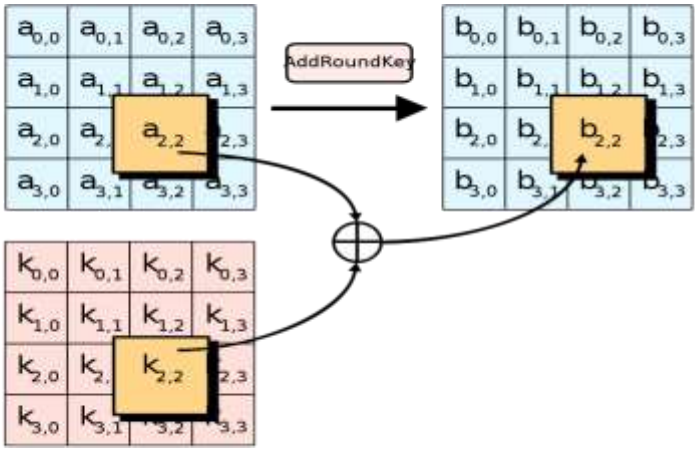
Example of Shift Row Operation

In mix columns step, the new column is obtained by using the elements of the old column. Each column is taken separately and multiplied by a previously prepared matrix. The resulting single column result is written in the same column of the new matrix. This process takes 4 bytes of input and outputs 4 bytes, allowing each input byte to affect each byte value in the output.



Mix Columns Operation

In add round key step, the resulting 128-bit state matrix and the 128-bit-long key are inserted into the XOR operation. The same coordinate elements of the state and key matrices are processed and the result is written on the same coordinate. Since the key matrix changes every round, the matrix that is processed changes every round.



Add Round Key Operation

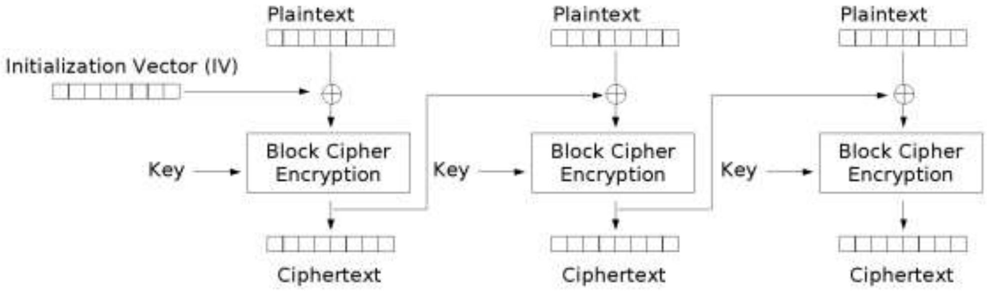
The AES algorithm takes the key and generates a key by the number of operations by passing it through a series of operations. This number is 10 for a 128-bit length. 10 different keys are generated and the resulting last key becomes the first key used to decode the encryption. When generating a key, each newly created new key is obtained by itself using the previous keys.

The encrypted text generated in the AES algorithm can be easily decoded by inverse operations to obtain the input text. The operations for this decoding operation are reverse row shift, pass through the inverted S box, and reverse column shuffle and adding round key. For decoding; in sub-byte step, we use a different S-Box table. In the shift row step, rows are shifted to right with same number. And In the mix columns step, columns are multiplied by a different matrix.

With all these features, AES is safer and faster than the previous examples. Because of this reason, the algorithm is used frequently nowadays.

CBC (Cipher Block Chaining)

It is one of the methods used in block cipher encryption. In this method, the message to be encrypted is processed with the previous ones.



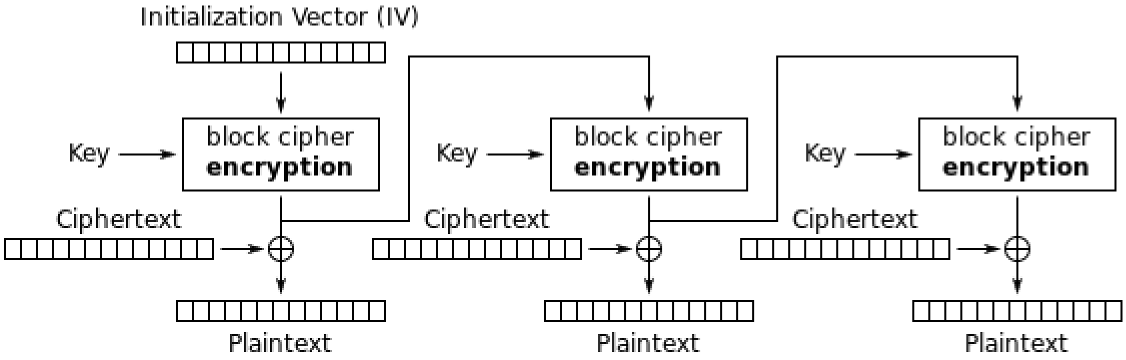
Cipher Block Chaining (CBC) mode encryption

Open message is divided into blocks. Each block is encrypted separately and consequently the encrypted message is a combination of messages obtained by encrypting each block separately. Thus, each block becomes dependent on the previous blocks. To ensure that a message is encrypted again under the same key, the first bloke initialization vector must be used.

During the opening of the message is applied the reverse process. Although this method is used frequently, it cannot be processed in parallel because each block is dependent to each other.

OFB (Output Feedback)

It is one of the methods used in block cipher encryption. It generates keystream blocks, which are XOR performed with the plaintext blocks to get the ciphertext.



Output Feedback (OFB) mode decryption

In this mode, the value entered in the encryption algorithm is the initialization vector. The value obtained as the result of encryption is used as the initial vector of the next block. Encrypted message blocks are inserted into XOR operation with open blocks, ciphering algorithm output.