

Rijndael Encryption

The Rijndael algorithm (given plaintext M) proceeds as follows:

1. Initialize **State** with M :

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$		m_0	m_4	m_8	m_{12}
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	←	m_1	m_5	m_9	m_{13}
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$		m_2	m_6	m_{10}	m_{14}
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$		m_3	m_7	m_{11}	m_{15}

where M consists of the 16 bytes m_0, m_1, \dots, m_{15} .

2. Perform **ADDROUNDKEY**, which XOR's the first **RoundKey** with **State**.
3. For each of the first $N_r - 1$ rounds:
 - Perform **SUBBYTES** on **State** (using a substitution, or S-box, on each byte of **State**),
 - Perform **SHIFTRROWS** (a permutation) on **State**,
 - Perform **MIXCOLUMNS** (a linear transformation) on **State**,
 - Perform **ADDROUNDKEY**.
4. For the last round:
 - Perform **SUBBYTES**,
 - Perform **SHIFTRROWS**,
 - Perform **ADDROUNDKEY**.
5. Define the ciphertext C to be **State** (using the same byte ordering).

Rijndael Decryption

To decrypt, perform cipher in reverse order, using inverses of components and the reverse of the key schedule:

1. ADDROUNDKEY with round key N_r
2. For rounds $N_r - 1$ to 1 :
 - INVSHIFTRows
 - INVSubBytes
 - ADDROUNDKEY
 - INVMIXColumns
3. For round 1 :
 - INVSHIFTRows
 - INVSubBytes
 - ADDROUNDKEY using round key 1

Note: Straightforward inverse cipher has a different sequence of transformations in the rounds. It is possible to reorganize this so that the sequence is the same as that of encryption (see A2).