

**ALGORITHM PROJECT ON CRYPTOGRAPHY**

**SUBMITTED BY:-** **SUBMITTED TO:-**

SUYASH SOMANI-13103604 INDU CHAWLA

KARAN SINGLA-13103612 ANKITA WADHWA

CHETAN KANJANI-13103618 ALGORITHM LAB FACULTY

MANAV SHARMA-13103620

**PROBLEM STATEMENT:**

Different encryption algorithms like AES (also known as RJINDAEL algorithm and RSA algorithm) are used to encrypt and decrypt the data.

**DESCRIPTION:**

In [cryptography](http://en.wikipedia.org/wiki/Cryptography), encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption does not of itself prevent interception, but denies the message content to the interceptor. In an encryption scheme, the message or information, referred to as [plaintext](http://en.wikipedia.org/wiki/Plaintext), is encrypted using an encryption algorithm, generating [cipher text](http://en.wikipedia.org/wiki/Ciphertext) that can only be read if decrypted.

So, here we are implementing 2 encryption algorithm and showing that which is the best among them. AES algorithm is a symmetric encryption with 1 key for encryption.AES basically comprises of 4 main functions that are ROUNDKEYADDITION ,SUBBYTES,SHIFT ROWS,MIX COLUMNS and similar functions are used in decryption of the data also.

For RSA algorithm, there can be many different keys for encryption of a particular data. This is so because there are two keys which are known as private key and public key. These can vary from computer to computer.

**DATA STRUCTURES USED:**

We are using basic 1-d and 2-d array for the implementation of both the algorithms. For eg. State array, inv\_s\_box, rcon etc.

**ALGORITHM IMPLEMENTED:**

**RSA:**

In this encryption algorithm, we are taking 2 keys, 1 is private and other one is public key.

Public key comprises of 3 variables. In our code, we are naming them as "e,n". We are taking p and q as input and then calculating "e" further(co-prime b/w 1 and z).

Further, private key comprises of "d,n" where "d" satisfies the equation (d\*e)%z==1. Here, "z" is (p-1)\*(q-1).

Then taking input by the user that what is to be encrypted and the encryption is as: (input^e)%n.

Same method is followed while using decryption.

**AES:**

This algorithm is one of the best encryption algorithm that is used widely for many purposes.

|  |  |
| --- | --- |
| Key Expansion: | Routine used to generate a series of Round Keys from the Cipher Key. |
| Round Key: | Round keys are values derived from the Cipher Key using the Key Expansion routine; they are applied to the State in the Cipher and Inverse Cipher. |
| State | Intermediate Cipher result that can be pictured as a rectangular array of bytes, having four rows and ***Nb*** columns. |
| S-box | Non-linear substitution table used in several byte substitution transformations and in the Key Expansion routine to perform a one-for-one substitution of a byte value. |
| Word | A group of 32 bits that is treated either as a single entity or as an array of 4 bytes. |

This algorithm comprises of main 4 arrays that are **State array, Rcon array,get\_sbox\_value, get\_inv\_sbox \_value.**

State array is the main array in which all the operations are being carried out.

Rcon array is used in Key Expansion( ) function. This is a predefined array for AES algorithm.

Get\_sbox\_value and Get\_inv\_sbox\_value array are used to initially feed the state array. These are predefined arrays for AES algorithm.

This algorithm basically comprises of one main function which is further made up of 4 main functions( Round\_key\_Addition, Shift\_rows,Mix\_columns,Sub\_Bytes).

The whole Encryption algorithm is based on these functions.

For the AES algorithm**, the length of the input block, the output block and the State is 128 bits.** This is represented by ***Nb*** = 4, which reflects the number of 32-bit words (number of columns) in the State.

For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by ***Nr***, where ***Nr*** = 10 when ***Nk*** = 4, ***Nr*** = 12 when ***Nk*** = 6, and ***Nr*** = 14 when ***Nk*** = 8.

For AES algorithm, **the length of the cipher key K is 18,192,256 bits.**

The key length is represented by ***Nk*** = 4, 6, or 8, which reflects the number of 32-bit words (number of columns) in the Cipher Key.

The Cipher funtion works in the following order:

state = in (Input array)

Round\_key\_Addition(state, w[0, Nb-1])

for round = 1 step 1 to Nr–1

SubBytes(state)

ShiftRows(state)

MixColumns(state)

Round\_key\_Addition (state, w[round\*Nb, (round+1)\*Nb-1])

end for

SubBytes(state) ShiftRows(state) Round\_key\_Addition (state, w[Nr\*Nb, (Nr+1)\*Nb-1])

out = state (Output array)

Functions used:

* **addrkey():**

Transformation in the Cipher and Inverse Cipher in which a Round Key is added to the State using an XOR operation. The length of a Round Key equals the size of the State (i.e., for Nb = 4, the Round Key length equals 128 bits/16 bytes).

* **mixcolumns():**

Transformation in the Cipher that takes all of the columns of the State and mixes their data (independently of one another) to produce new columns.

* **rotword():**

Function used in the Key Expansion routine that takes a four-byte word and performs a cyclic permutation.

* **shiftrows():**

Transformation in the Cipher that processes the State by cyclically shifting the last three rows of the State by different offsets.

* **subbytes():**

Transformation in the Cipher that processes the State using a non­ linear byte substitution table (S-box) that operates on each of the State bytes independently.

* **subword():**

Function used in the Key Expansion routine that takes a four-byte input word and applies an S-box to each of the four bytes to produce an output word.