**NEX encrypt algorithm**

**Description**

NEX encrypt is symmetric-key algorithm for the encryption text using python3. This algorithm adapted from DES algorithm. Created for project#1 “Crack me if you dare” 322327 Computer Network Security [1/2561]

**Input qualification**

Plaintext: Unicode 16 bits per character, 64 bits per block

Key: ascii 8 bits per character, maximum 64 bits

**Encryption**

**Overall**

Divide the text into block, each block contains 4 character (64 bits). Each block will encrypt 8 rounds. And each round will encrypt block with algorithm depend on subkeys.

**Generate subkeys**

We will generate subkeys from input key by step:

1. Convert input key to binary 64 bits. Append 0 to 64 bits if input key less than 64 bits and remove bits more than 64th bit

“thiskey” => “0111010001101000 0110100101110011 0110101101100101 0111100100000000”

1. Divide key from step 1 into 4-part call “ABCD”. ‘A’ is first part, ‘B’ is second part. Then permuted each part with following A1 B1 C1 D1 A2 B2 C2 D2 … A16 B16 C16 D16   
   ( A1 is first char of A ) . We call the permuted bits is KR(0)

KR(0) => “0000111111111001 0111100000100111 0000111011100100 1000001001000110”

1. Then create KR(1) -> KR(8) by KR(n+1) = KR(n) shift right ( KR(n)63 + KR(n)64 )2 +1 times

KR(0)63 = 1 and KR(0)63 = 0 then => (10)2 + 1 = 3 KR(1) = KR(0) shift right 3 times

KR(1) => “1100000111111111 0010111100000100 1110000111011100 1001000001001000”

1. Then we get 8 subkeys KR(1) -> KR(8)

**Encrypt block of plaintext**

Encrypt each block with 8 rounds by following step:

1. Divide block of plaintext into 4-parts call “ABCD”. Then permuted each part with following C1 A1 D1 B1 C2 A2 D2 B2 … C16 A16 D16 B16

Example round 1:

Plantext = “LOVE” => “0000000001001100 0000000001001111

0000000001010110 0000000001000101”

Plantext permuted = “0000000001010110 0000000001001100

0000000001000101 0000000001001111”

1. Divide result from step 1 into 4-parts call “OPQR” and divide KR(round) from into 4-parts call “WXYZ”. And pair permuted plaintext with each permuted subkeys “OW”, “PX”, “QY”, “RZ”
2. Check each KR that paired with plaintext, let KR(round)1 + KR(round)12 = KA and KR(round)15 + KR(round)16 = KZ

If KZ = ‘00’ and KA = ‘00’ Then permuted paired plaintext with following sequence

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 7 | 11 | 1 |
| 13 | 2 | 8 | 12 |
| 9 | 3 | 5 | 15 |
| 14 | 6 | 4 | 10 |

If KZ = ‘00’ and KA = ‘01’ Then permuted paired plaintext with following sequence

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | 14 | 7 | 11 |
| 10 | 6 | 8 | 13 |
| 15 | 4 | 2 | 3 |
| 0 | 1 | 5 | 12 |

If KZ = ‘00’ and KA = ‘10’ Then permuted paired plaintext with following sequence

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | 14 | 8 | 10 |
| 0 | 5 | 6 | 7 |
| 2 | 1 | 4 | 13 |
| 15 | 3 | 11 | 12 |

If KZ = ‘00’ and KA = ‘11’ Then permuted paired plaintext with following sequence

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 10 | 11 | 4 |
| 6 | 0 | 2 | 13 |
| 15 | 3 | 1 | 9 |
| 14 | 8 | 12 | 5 |

If KZ = ‘01’ Then XOR paired permuted plaintext and KR

If KZ = ‘10’ Then complement permuted plaintext

If KZ = ‘11’ Then left shift permuted plaintext (KA)2 + 1 times

**Overview**

**KR(8)**



**Plaintext**

**1**

**2**

**3**

**4**

**KEY**

**1**

**2**

**3**

**4**

**KR(0)**

**KR(1)**



**function**

**PT1**

**PT2**

**Encrypted**

**Code**

GitHub :<https://github.com/akaradat/NEXEncrypt>

**Referent**

* J. Orlin Grabbe. *The DES Algorithm Illustrated*. Retrieved 27 September, 2018, Available : <http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm>
* *Data Encryption Standard*. Retrieved 26 September, 2018, Available :<https://en.wikipedia.org/wiki/Data_Encryption_Standard>