**Pseudocode**

|  |
| --- |
| **Key\_gen ( )**  **Input:**  Prime integers-p, q, r, s, t, u  **Output:**  Public key: {E, n}  Private Key :{ D, n}  **Method:**   1. n=p\*q\*r 2. m=s\*t\*u 3. N=n\*m 4. Pi(n)=(p-1)\*(q-1)\*(r-1) 5. Pi(m)=(s-1)\*(t-1)\*(u-1) 6. Pi(N)=Pi(n)\*Pi(m) 7. e1; 1<e1<Pi(n) and gcd(e1,Pi(n))=1 8. e2; 1<e2<Pi(m) and gcd(e2,Pi(m))=1 9. E1=e1^e2 mod N 10. E; 1<E<Pi(N)\*E1 and gcd(E,Pi(N)\*E1)=1 11. D=inverse(E) mod(Pi(N)\*E) |

|  |
| --- |
| **Encrypt( )**  **Input:**  Plain text-M ( < n)  Secret key-Ks  Public key- {E,n}  **Output:**  Encoded text-C  **Method:**   1. C1=M^E mod n 2. C=C1 EXOR Ks   **Decrypt ( )**  **Input:**  Encoded text-C  Secret key-Ks  Private key- {D,n}  **Output:**  Plain text-M  **Method:**   1. C1=C EXOR Ks 2. M=C1^D mod n |

The above pseudocode has the step by step procedure to generate the public and private keys and then encrypt and decrypt the data using the keys respectively.

**3.4. MKGTR-Example:**

Consider the following example.

**Key-generation:**

p=127

q=263

r=337

s=139

t=223

u=379

n<- 11256137 (p\*q\*r)

m<-11747863 (s\*t\*u)

N<-132235555385231 (n\*m)

pi(n)<-11092032 (126\*262\*336 )

pi(m)<-11580408 (138\*222\*378)

pi(N)<-128450256109056 (pi(n)\*pi(m))

gcd(e1,pi(n))=1; 1<e1<pi(n)

let e1=2761

gcd(e2,pi(m))=1; 1<e2<pi(m)

let e2=587

E1=2034188637405 (e1 pow e2 modN)

E ; 1<E<pi(N)\*E1 and gcd(E,pi(N)\*E1)=1

E=4425692186722853

D=inverse(E) mod pi(N)\*E1

D=106446335187214604086443437

**Encryption:**

C=M pow E mod n (M<n)

Let M=59

C=2944062

C1=C EXOR Ks

C1=2948633

**Decryption:**

C=C1 EXOR Ks

C=2944062

M=C pow D mod n

M=59

Thus, the plain text sent by the sender is received by the receiver without any modifications.