Rsa pratical 7

#include<stdio.h>

#include<math.h>

int gcd(int a, int h)

{

int temp;

while(1)

{

temp = a%h;

if(temp==0)

{

return h;

}

a = h;

h = temp;

}

}

void main()

{

double d, c, m, k = 2, p = 2, q = 8, n=p\*q, count, totient = (p-1)\*(q-1), e=2, msg = 12;

while(e<totient)

{

count = gcd(e,totient);

if(count==1)

{

break;

}

else

{

e++;

}

}

d = (1 + (k\*totient))/e;

c = pow(msg,e);

m = pow(c,d);

c=fmod(c,n);

m=fmod(m,n);

printf("Message data = %lf",msg);

printf("\np = %lf",p);

printf("\nq = %lf",q);

printf("\nn = p q = %lf",n);

printf("\ntotient = %lf",totient);

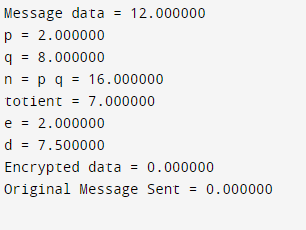
printf("\ne = %lf",e);

printf("\nd = %lf",d);

printf("\nEncrypted data = %lf",c);

printf("\nOriginal Message Sent = %lf",m);

}



Pratical 5: des

#include<stdio.h>

void main()

{

int p10[10] = {3, 5, 2, 7, 4, 10, 1, 9, 8, 6}, p8[8] = {6, 3, 7, 4, 8, 5, 10, 9};

int i, key1[10], key[10], ls1[5], ls2[5], temp, p1[10], j, temp1;

printf(" \n Enter Key Value: ");

for(i = 0; i < 10; i++)

{

scanf("%d", &key[i]);

}

printf(" \n P10 Value: ");

for(i = 0; i < 10; i++)

{

printf(" %d ", p10[i]);

}

printf(" \n P8 Value: ");

for(i = 0; i < 8; i++)

{

printf(" %d ", p8[i]);

}

for(i = 0; i < 10; i++)

{

temp = p10[i];

for(j = 0; j < temp; j++)

{

temp1 = key[j];

}

p1[i] = temp1;

}

printf(" \n After Applying Permutation P10: ");

for(i = 0; i < 10; i++)

{

printf(" %d ", p1[i]);

}

for(i = 0; i < 10; i++)

{

temp1 = p1[i];

if(i < 5)

{

ls1[i] = temp1;

}

else

{

ls2[i - 5] = temp1;

}

}

ls1[5] = ls1[0];

for(i = 0; i < 5; i++) { ls1[i] = ls1[i + 1]; }

ls1[4] = ls1[5];

printf(" \n After Left Shifting By 1 \n ");

printf(" \n First 5 Bits: ");

for(i = 0; i < 5; i++) { printf(" %d ", ls1[i]);}

ls2[5] = ls2[0];

for(i = 0; i < 5; i++) { ls2[i] = ls2[i + 1]; }

ls2[4] = ls2[5];

printf(" \n Second 5 Bits: ");

for(i = 0; i < 5; i++) { printf(" %d ", ls2[i]);}

for(i = 0; i < 10; i++)

{

p1[i] = ls1[i];

if(i > 5) { p1[i] = ls2[i - 5]; }

}

for(i = 0; i < 10; i++)

{

temp = p8[i];

for(j = 0; j < temp; j++) { temp1 = p1[j]; }

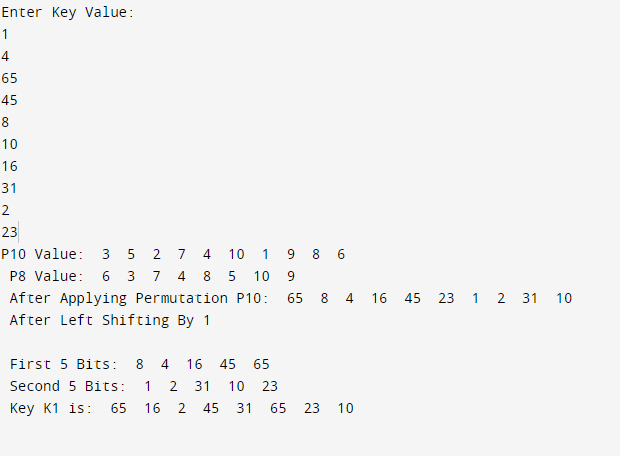
key1[i] = temp1;

}

printf("\n Key K1 is: ");

for(i = 0; i < 8; i++) { printf(" %d ", key1[i]); }

}



Pratical 7

Implement Diffi-Hellmen Key exchange Method

#include<stdio.h>

int power(int a, int m, int n)

{

int r, y=1;

while(m>0)

{

r=m%2; if(r==1) { y=(y\*a)%n;

}

a=a\*a%n; m=m/2;

}

return y;

}

void main()

{

int q=31,al=1, xa,xb,ya,yb,ka,kb;

xa=6; xb=15;

printf("Implement Diffie-Hellman Key Exchange Method.\n\n");

ya=power(al,xa,q); yb=power(al,xb,q);

ka=power(yb,xa,q); kb=power(ya,xb,q);

printf("q : %d\nalpha : %d\n",q,al);

printf("Xa : %d\nXb : %d\n",xa,xb);

printf("Ya : %d\nYb : %d\n",ya,yb);

printf("Ka : %d\nKb : %d",ka,kb);

}

