CARD PAYMENT SECURITY USING RSA

Team Members:
19BCI0173(Devansh)
19BCI0198(Chaitanya)
19BCI0231 (Kartik)
19BEC0804(Aryan)

Report submitted for the
First Project Review
of

Course Code: CSE1011 – Cryptography Fundamentals

Slot: C1

Professor: Dr. Ilanthenral Kandasamy

ABSTRACT

RSA algorithm is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and Private key is kept private. This project proposed an implementation of a complete and practical RSA encrypt/decrypt solution on CARD PAYMENT SECURITY, the study of RSA public key algorithm. In addition, the encrypt procedure and code implementation is provided in details.

KEY WORDS - RSA, Credit Card, Security

INTRODUCTION

Because of expanding web based business movement these days, there is a requirement for some encryption method to guarantee security and an approach to guarantee that the client's information are safely put away in the database. In this way the framework presents RSA for this reason. The RSA calculation is a sort of lopsided encryption calculation which showed up in 1978. The calculation is open key encryption calculation which is a broadly acknowledged and actualized by open. The utilization of RSA in this the framework makes the procedure safer. Presently the bank exchanges should be possible safely without agonizing over assailant gaining admittance to the database as the information will be in scrambled structure.

RSA is the principal calculation referred to be reasonable for marking just as encryption, and one of the main incredible advances in broad daylight key cryptography. It is named for the three MIT mathematicians who created it — Ronald Rivest, Adi Shamir, and Leonard Adleman.

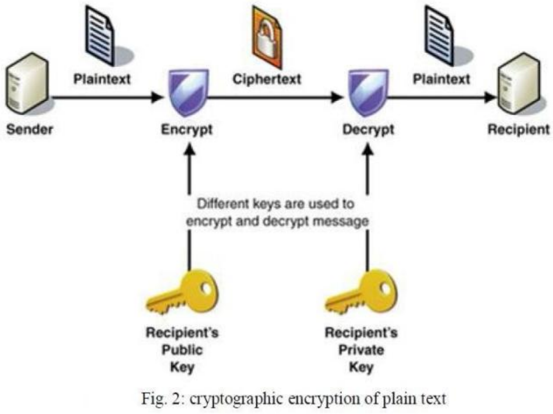


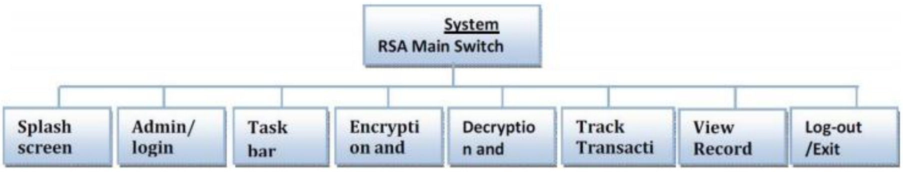
Fig. 2: cryptographic encryption of plain text

The design of the RSA security software partly evolved from the need for an all embracing information security system and partly from the need for a user friendly package that can fulfil any large ecommerce organization's information security needs.

LITERATURE SURVEY

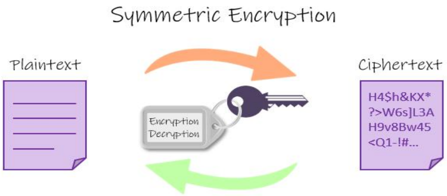
Authors and Year (Reference)	***Title (Study)***	***Concept / Theoretical model/ Framework***	***Methodology used/ Implementation***	***Dataset details/ Analysis***	***Relevant Finding***	***Limitations / Future Research/ Gaps identified***
Sistu Sudheer Kumar, A. Srinivas Reddy (May - 2015)	*ATM Security*	*Design of ATM system that will improve the Authentication of customer*	*"One Time Password (OTP)" and "Personal Identification Number (PIN)"*	*Bank Customer Dataset*	*OTP and PIN combination in order to improve authentication*	*ATM card falling into wrong hands by knowing PIN number*
Ezeofor C. J, Ulasi A. G. (December - 2014)	*Network Data Encryption and Decryption techniques*	*Analysis of Network Data Encryption and Decryption*	*Data Encryption and Decryption technique*	*Network Dataset*	*Unsecured data that travels through different networks are open to many types of attack and can be read, altered or forged*	*Data can still be attacked by hackers using deciphering algorithms*
Nentawe Y. Goshwe (July - 2013)	*Design of Data Encryption and Decryption in a network environment*	*Data Encryption and Decryption in a Network Environment using RSA algorithm with a specific message block size*	*RSA Implementation*	*Network Dataset*	*RSA allows a message sender to generate a public keys to encrypt the message and the receiver is sent with a generated private key using a secured database*	*An incorrect private key will still decrypt the encrypted message but to a form different from the original message.*

PROPOSED WORK AND IMPLEMENTATION

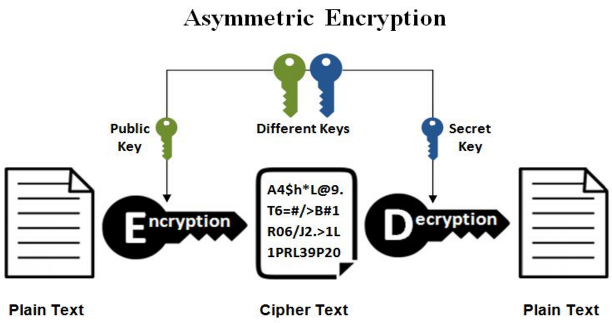


This necessitated the decomposition of the system into clearly defined subsystems such that the initial requirements specifications were met. The software system comprises the following subsystems: splash-screen subsystem, Admin/login subsystem, Task bar/Key generation subsystem, Encryption subsystem, Decryption subsystem, Track Transaction subsystem, View record subsystem, Log out/Exit subsystem.

The public key can be freely distributed without the key management challenges of symmetric keys since it can only encrypt and never decrypt data.



In a payment environment, the public key can be distributed to a merchant or to the end POS device, and that device can store the key in hardware or software. Even if that key is extracted by someone who shouldn't have rights to it, all that the person can do is encrypt data with the key; he can't decrypt anything. On the other hand, the corresponding private key where the decryption occurs must be handled very securely. The RSA algorithm is the most commonly used public key encryption algorithm in asymmetric cryptography. Two keys are used: Public Key and Private Key.



ADVANTAGES OF RSA

1. RSA is stronger than any other symmetric key algorithm.

2. RSA is very high speed of encryption .

3. Your sensitive details like your bank details are now secured using RSA.

4. Now the bank transactions can be done securely without worrying about attacker getting access to the database as the data will be in encrypted form.

5. RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total is considered infeasible due to the time it would take using even today's supercomputers.

6. RSA has overcome the weakness of symmetric algorithm i.e. authenticity and confidentiality.

7. Many protocols like secure shell, OpenPGP, S/MIME, and SSL/TLS rely on RSA for encryption and digital signature functions.

8. RSA signature verification is one of the most commonly performed operations in network-connected systems.

DISADVANTAGES OF RSA

1) RSA is a public key cryptosystem (asymmetric cryptography) which is slow compared to symmetric cryptography.

2) It requires a more computer power supply compared to single key encryption.

3) In this cryptosystem, if the private key is lost then all received message cannot be decrypted but security wise , it's great.

4) Complexity of algorithm i.e key is too large and calculation time is long.

RSA KEY GENERATION -

i. RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

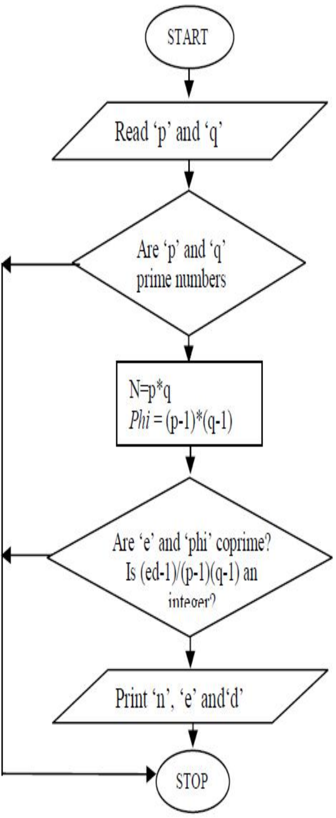

START
Read 'p' and 'q'
Are 'p' and 'q' prime numbers
N=p*q
Phi = (p-1)*(q-1)
Are 'e' and 'phi' coprime?
Is (ed-1)/(p-1)(q-1) an integer?
Print 'n', 'e' and'd'
STOP

Fig. 6: flow chart illustrating the RSA Key generation

ii. Choose two distinct prime number p and q.

iii. For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length.

iv. Compute $n = pq$.

n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.

v. Compute $\varphi(n)= \varphi(p)\varphi(q)=(p-1)(q-1)=n -(p+ q-1)$, where φ is Euler's totient function.

vi. Choose an integer e such that $1 < e <\varphi(n)$ and $\gcd(e, \varphi(n,$ $= 1$; i.e., e and $\varphi(n)$ are coprime. e is released as the public key exponent.

vii. Determine d as $d \equiv e-1 \pmod{\varphi(n)}$; i.e., d is the multiplicative inverse of e (modulo$\varphi(n)$).

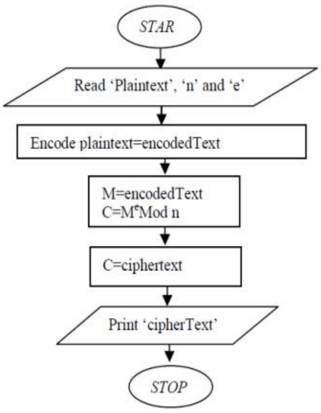
viii. The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and $\varphi(n)$ must also be kept secret because they can be used to calculate d

ix. After getting the public and private key the main thing is how to encrypt and decrypt using RSA

RSA ENCRYPTION -

Devansh transmits his public key (n, e) to Kartik and keeps the private key d secret. Kartik then wishes to send message M to Devansh. He first turns M into an integer m, such that 0 ≤m < n by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text c corresponding to :-
$c=m^{e}(\mathrm{mod}\ n)$

This can be done quickly using the method of exponentiation by squaring. Kartik then transmits c to Devansh.

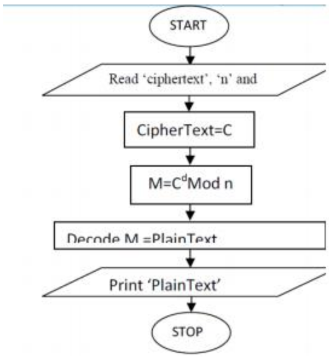


RSA DECRYPTION -

Devansh can recover m from c by using his private key exponent d via computing m where :-

$$m= c^{d}(\mathrm{mod}\ n)$$

Given m, he can recover the original message M by reversing the padding scheme.



CODE IMPLEMENTATION

```
#include<iostream>
#include<string.h>
#include <stdio.h>
#include <fstream>
#include <string.h>
#include <iostream>
#include <stdlib.h>
#include <ctype.h>
#include <conio.h>
using namespace std;

struct user
{
string user_id;
string user_name;
int user_ccn;
int encrypted_ccn;
string user_address;
int user_balance;
string user_password;
struct user *next;
}*first=NULL;

void enqueue();
void display();
void user_account();

int gcd(int a,int b)
{
return b==0 ? a:gcd(b,a%b);
}

int check_banker_id(string id)
{
string check_id;
int count=0;
cin.ignore();
cout<<"\nENTER BANKER ID: ";
getline(cin,check_id);
if(id.length()==check_id.length())
{
for(int i=0;i<id.length();i++)
{
if(id[i]==check_id[i])
count++;
else
break;
}
if(count==id.length())
return 1;
else
return 0;
}
else
return 0;
}

int check_banker_password(string password)
{
string check_password;
cout<<"Enter 10 digit password: "; char a;
for(int i=0;i<10;i++)
{
a=getch();
check_password=check_password+a;
cout<<"*";
}
int count=0; if(password.length()==check_password.length())
{
for(int i=0;i<check_password.length();i++)
{
if(password[i]==check_password[i])
count++;
else
break;
}
if(count==check_password.length())
return 1;
else
return 0;
}
else
return 0;
}
```

```
int main()
{
string banker_id="0000";
char password[11]="qpasswordq";
char a;
string check_password;
int check,select;
while(1)
{
cout<<"SELECT YOUR CHOICE\n\n1 Banker \n2 User \n3 Exit \n\n";
cout<<"ENTER YOUR CHOICE : ";
cin>>select;
if(select==1)
{
check=check_banker_id(banker_id);
if(check==1)
{
check=check_banker_password(password);
if(check==1)
{
system("cls");
cout<<"ADMINISTRATOR PAGE ";
cout<<"\n";
while(1){
cout<<"\nSELECT YOUR CHOICE\n\n1 Add User\n2 View User Details\n3
Main Menu\n\nENTER YOUR CHOICE : ";
cin>>select;
cout<<endl;
switch(select)
{
case 1:
{
enqueue(); break;
}
case 2:
{
display(); break;
}
case 3:
{
cout<<endl;
main();
}
default:
{
cout<<"\nINVALID INPUT!!!!!!\nTRY AGAIN\n";
}
}
}
}
else
{
cout<<"\nWRONG BANKER PASSWORD !!! TRY AGAIN !!!\n\n";
}
}
else
{
cout<<"WRONG BANKER ID !!! TRY AGAIN !!!\n\n";
}
}
else if(select==2)
{
user_account();
}
else
exit(0);
}
return 0;
}

void enqueue()
{
int p,q,n,toitent,c,msg,e,z,k,i,d;
cout<<"Enter Two Large Prime Numbers (p and q):"<<endl;
cout<<"Enter Prime Number p : ";
cin>>p;
cout<<"Enter Prime Number q : ";
cin>>q;
n=p*q;
toitent=(p-1)*(q-1);
cout<<endl<<"Values Of e are : ";
for(i=1;i<toitent;i++)
{
z=gcd(i+1,toitent); if(z==1)
cout<<i+1<<" ";
}
cout<<endl<<"Enter the Chosen Value Of e : ";
```

```
cin>>e;
string id,name;
string address,password;
int ccn,balance=0;
system("cls");
cout<<"ADD USER \n"<<endl;
cout<<"Enter The Following Details\n";
cout<<"\nEnter User Name: ";
cin>>name;
cout<<"Enter User ID: ";
cin>>id;
cout<<"Enter ccn: ";
cin>>ccn;
cout<<"Enter Address: ";
cin>>address;
cout<<"Create New password(5): ";
cin>>password;
cout<<"Enter The Initial Balance: ";
cin>>balance;
struct user *temp;
temp=(struct user *)malloc(sizeof(struct user));
msg=ccn;
for(int i=0;i<e;i++)
{
k=k*msg;
k=k%n;
}
cout<<"\nUnencrypted ccn : "<<ccn;
cout<<"\nEncrypted ccn is : "<<k<<endl;
temp=new(struct user);
temp->user_id=id;
temp->user_name=name;
temp->user_address=address;
temp->user_ccn=ccn;
temp->encrypted_ccn=k;
temp->user_balance=balance;
temp->user_password=password;
temp->next=NULL;
if(first==NULL)
first=temp;
else
{
struct user *s;
```

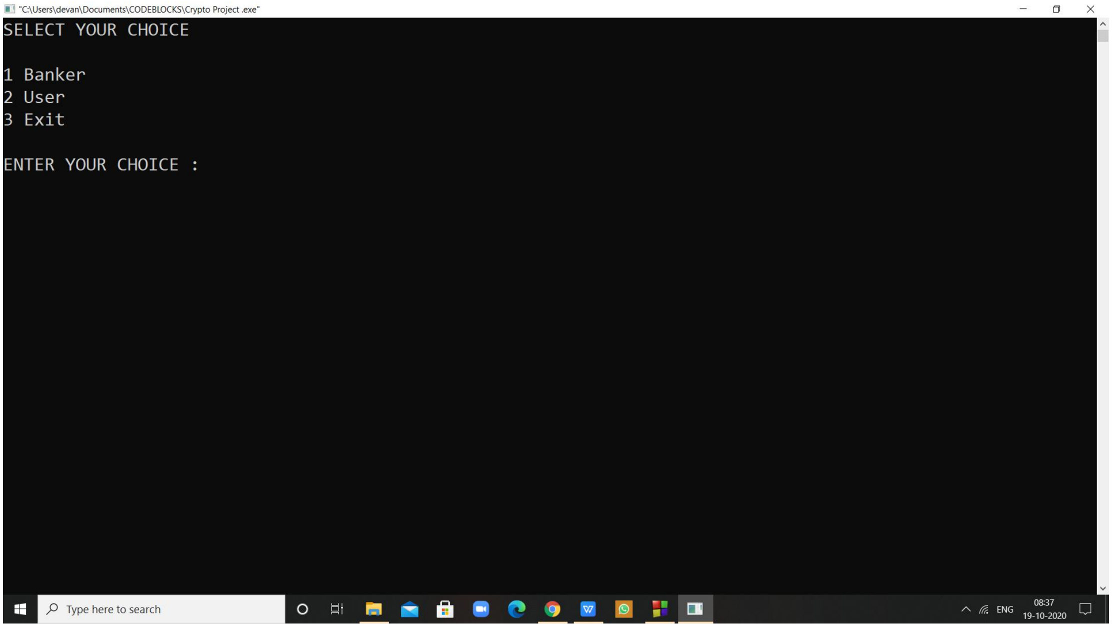
```
s=(struct user*)malloc(sizeof(struct user));
s=new(struct user);
s=first;
while(s->next!=NULL) s=s->next;
s->next=temp;
}
}

void display()
{
struct user *temp;
temp=(struct user*)malloc(sizeof(struct user));
if(first==NULL)
cout<<"\nNo Records Available\n";
else
{
int num;
cout<<"SELECT YOUR CHOICE\n\n1 View All Records \n2 View Specific Record\n";
cout<<"ENTER YOUR CHOICE : ";
cin>>num;
if(num==1)
{
temp=first;
while(temp!=NULL)
{
cout<<"\nUser ID : "<<temp->user_id;
cout<<"\nUser_name : "<<temp->user_name;
cout<<"\nCredit Card Number : "<<temp->user_ccn;
cout<<"\nAddress : "<<temp->user_address;
cout<<"\nBalance : "<<temp->user_balance;
cout<<"\nPassword : "<<temp->user_password;
cout<<"\n\n\n";
temp=temp->next;
}
}
else if(num==2)
{
int flag=0; string want_id;
cout<<"\nEnter User ID You Want to Search : ";
cin.ignore();
getline(cin,want_id);
temp=first;
while(temp!=NULL)
```

```
{
if(temp->user_id==want_id)
{
cout<<"\nRECORD FOUND!!!";
cout<<"\nUser ID : "<<temp->user_id;
cout<<"\nUser Name : "<<temp->user_name;
cout<<"\nccn : "<<temp->user_ccn;
cout<<"\nAddress : "<<temp->user_address;
cout<<"\nBalance : "<<temp->user_balance;
cout<<"\nPassword : "<<temp->user_password; cout<<endl;
flag=1;
break;
}
temp=temp->next;
}
if(flag==0)
cout<<"\n!!! RECORD NOT FOUND !!!\n";

}
else
{
cout<<"WRONG INPUT !!! TRY AGAIN !!!";
display();
}
}
}

void user_account()
{
if(first==NULL)
cout<<"\nNo User Record!!!\n"<<endl;
else
{
int flag=0,eccn;
string id;
struct user *temp;
temp=new(struct user);
cout<<"\nEnter The User ID : ";
cin>>id;
temp=first;
while(temp!=NULL)
{
if(temp->user_id==id)
{
```

```
flag=1;
break;
}
else
temp=temp->next;
}
if(flag==0)
cout<<"\nUser ID NOT FOUND\n";
else
{
flag=0;
string password;
cout<<"Enter The 5 Digit Password : ";
cin>>password;
cout<<endl;
while(temp!=NULL)
{
if(temp->user_password==password)
{
flag=1;
break;
}
temp=temp->next;
}
if(flag==0)
cout<<"\nInvalid Password ";
else
{
cout<<"!!! USER FOUND !!!\n\n";
cout<<"User ID : "<<temp->user_id<<endl;
cout<<"User Name : "<<temp->user_name<<endl;
cout<<"User ccn : "<<temp->user_ccn<<endl;
cout<<"User Address : "<<temp->user_address<<endl;
int num;
cout<<"\nSELECT YOUR CHOICE\n\n1 Payment Process\n2 Main Menu\n3
Exit\n\n";
cout<<"ENTER YOUR CHOICE : "; cin>>num;
if(num==1)
{
string id;
cout<<"\nEnter Receipient's ID : "; cin>>id;
struct user *target;
target=new(struct user);
target=first;
flag=0;
while(target!=NULL)
{
if(target->user_id==id)
{
flag=1;
break;
}
target=target->next;
}
if(flag==0)
{
cout<<"\nINVALID ID ";
}
else
{
cout<<"\n!!! RECEIPIENT ID FOUND !!!\n\n";
cout<<"Current Amount In User Credit Card : "<<temp->user_balance<<endl;
int amount;
cout<<"Enter The Amount You Want To Pay : "; cin>>amount;
cout<<"Enter Your Encryted ccn : "; cin>>eccn;
if(eccn==temp->encrypted_ccn)
{
if(amount > temp->user_balance)
cout<<"\nInsufficient Balance";
else
{
temp->user_balance-=amount;
target->user_balance+=amount;
cout<<"\n!!! PAYMENT SUCESSFUL !!!\n";
cout<<"Current Amount In User Credit Card After Payment :
"<<temp->user_balance<<endl<<endl;
}
}
else
{
cout<<"Wrong Encrypted ccn"<<endl;
cout<<"!!! PAYMENT TERMINATED !!!"<<endl<<endl;
}
}
}
}
}
```

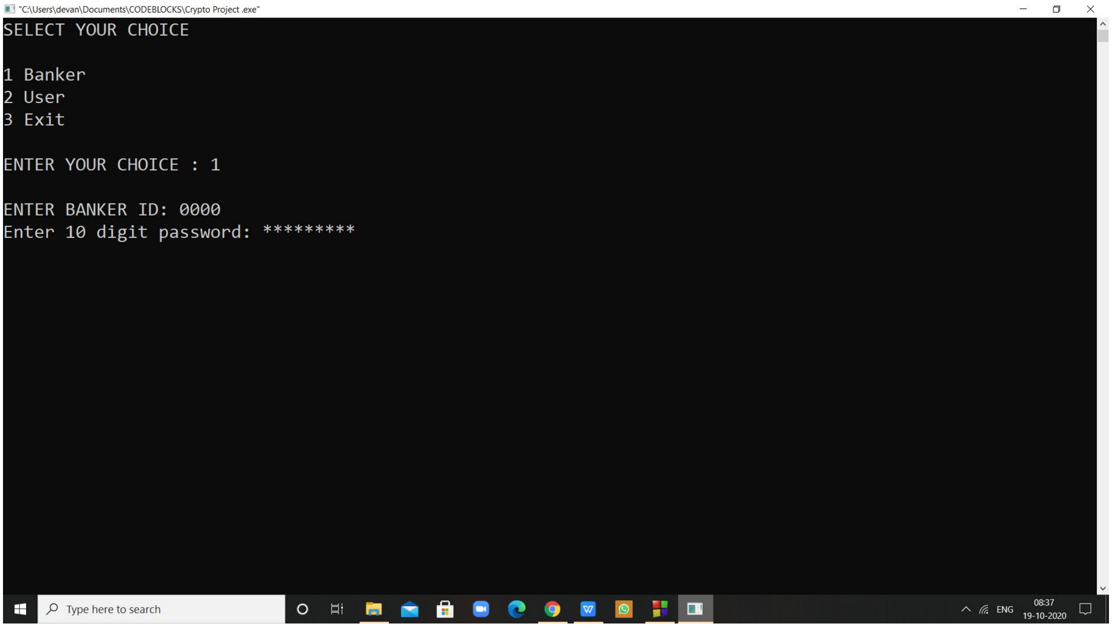
IMPLEMENTATION SCREENSHOTS

1 - MAIN MENU :-



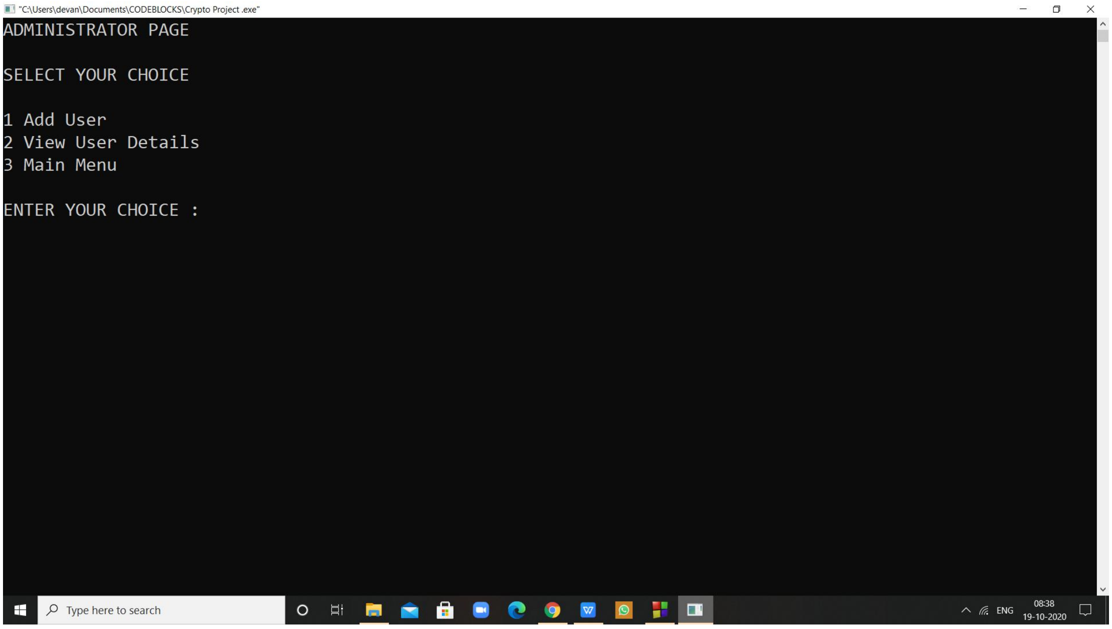
```
SELECT YOUR CHOICE

1 Banker
2 User
3 Exit

ENTER YOUR CHOICE :
```

2 - ENTERING BANKER ID AND PASSWORD:-



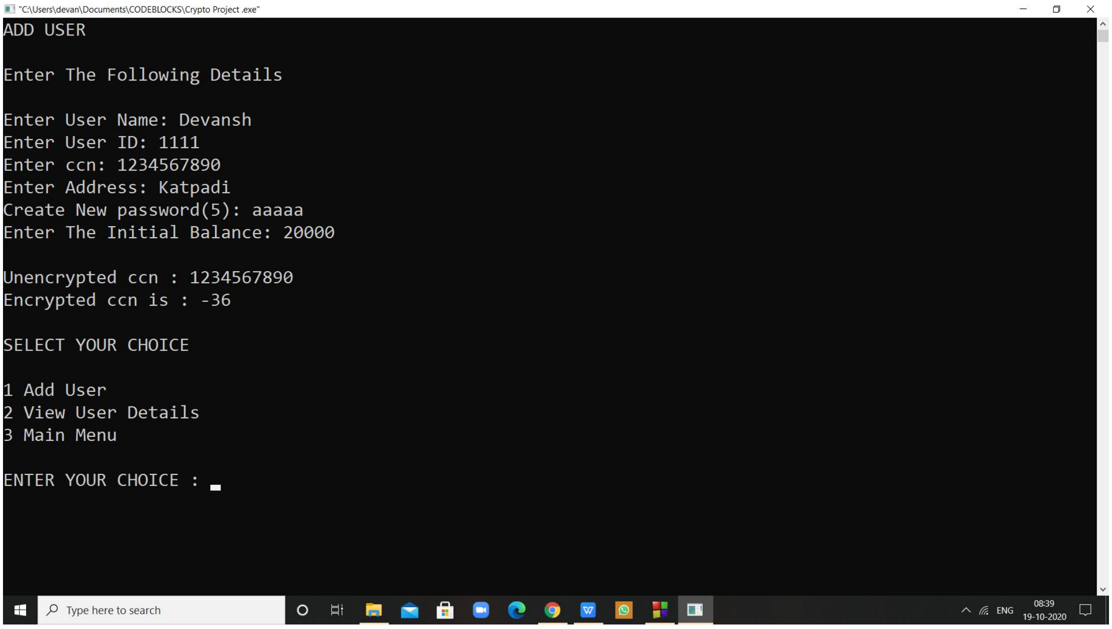
```
SELECT YOUR CHOICE

1 Banker
2 User
3 Exit

ENTER YOUR CHOICE : 1

ENTER BANKER ID: 0000
Enter 10 digit password: *********
```

3 - BANKERS PAGE:-



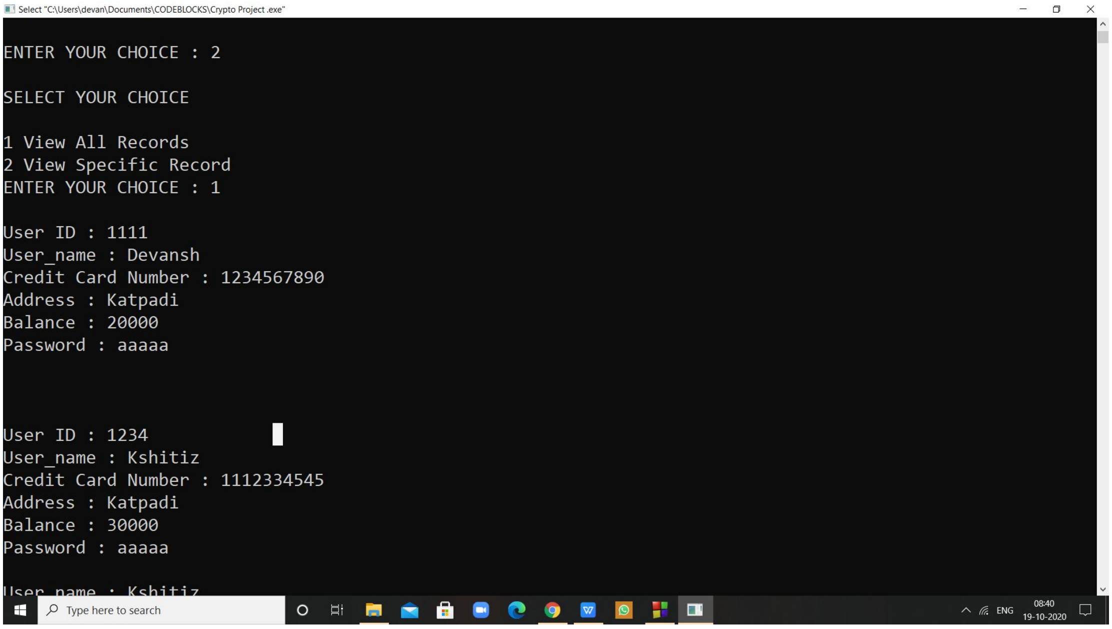
```
ADMINISTRATOR PAGE

SELECT YOUR CHOICE

1 Add User
2 View User Details
3 Main Menu

ENTER YOUR CHOICE :
```

4 - ADDING USERS:-



```
ADD USER

Enter The Following Details

Enter User Name: Devansh
Enter User ID: 1111
Enter ccn: 1234567890
Enter Address: Katpadi
Create New password(5): aaaaa
Enter The Initial Balance: 20000

Unencrypted ccn : 1234567890
Encrypted ccn is : -36

SELECT YOUR CHOICE

1 Add User
2 View User Details
3 Main Menu

ENTER YOUR CHOICE :
```

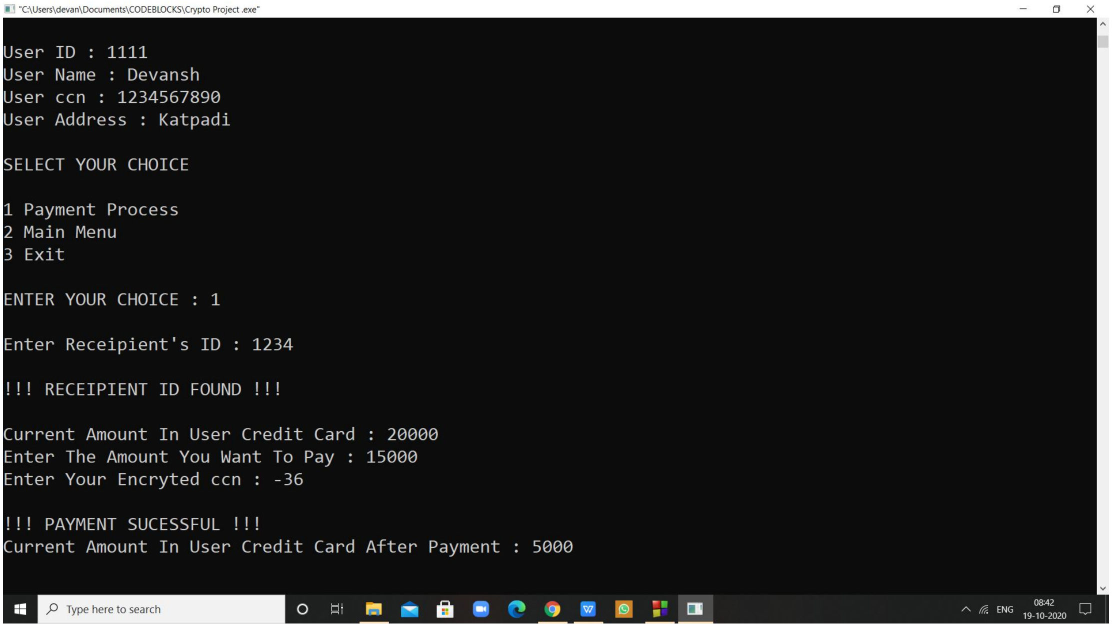
5 - VIEWING USERS:-



Select "C:\Users\devan\Documents\CODEBLOCKS\Crypto Project.exe"
Type here to search
ENG
08:40
19-10-2020

```
ENTER YOUR CHOICE : 2

SELECT YOUR CHOICE

1 View All Records
2 View Specific Record
ENTER YOUR CHOICE : 1

User ID : 1111
User_name : Devansh
Credit Card Number : 1234567890
Address : Katpadi
Balance : 20000
Password : aaaaa



User ID : 1234
User_name : Kshitiz
Credit Card Number : 1112334545
Address : Katpadi
Balance : 30000
Password : aaaaa

User_name : Kshitiz
```

6 - PAYMENT PROCESS:-



```
User ID : 1111
User Name : Devansh
User ccn : 1234567890
User Address : Katpadi

SELECT YOUR CHOICE

1 Payment Process
2 Main Menu
3 Exit

ENTER YOUR CHOICE : 1

Enter Receipient's ID : 1234

!!! RECEIPIENT ID FOUND !!!

Current Amount In User Credit Card : 20000
Enter The Amount You Want To Pay : 15000
Enter Your Encryted ccn : -36

!!! PAYMENT SUCESSFUL !!!
Current Amount In User Credit Card After Payment : 5000
```

EXPECTED RESULT

The Main Aim of our project is to implement Card Security using RSA algorithm and make it as secure as possible so that its not easily broken by hackers and our information and money is lost. Credit Card security is most important in this time as most transactions is being made by credit cards and the customer wants to send his money easily .

REFERENCES

[1] Sistu Sudheer Kumar, A. Srinivas Reddy , May-2015, " A survey on theft prevention during ATM transaction without ATM cards", eISSN: 2319-1163 | pISSN: 2321-7308 Volume: 04 Special Issue: 06 | NCEITCS-2015 | May- 2015.

[2] Ezeofor C. J, Ulasi A. G., December 2014, "Analysis of Network Data Encryption & Decryption T echniques in Communication

[3] Nentawe Y. Goshwe , July 2013, " Data Encryption and Decryption Using RSA Algorithm in a Network Environment" IJCSNS

[4] Sivakumar T, Gajjala Askok, k. Sai Venuprathap, August 2013, " Design and Implementation of Security Based ATM theft Monitoring system", International Journal of Engineering Inventions e-ISSN: 2278-7461, p-ISSN: 2319-6491 Volume 3, Issue 1 (August 2013) PP: 01-07