

**A Project Report On**

**‘Secured Data Encryption’**

**Subject** – Cyber Security

**Team Members :**

Amreeta Koner- 17BCE2033

Amisha- 17BCB0022

**Abstract**

This report is an introduction to the Secure Encryption techniques for ATM Transactions in Python programming. Anybody, who doesn’t know even the basics of ATM Transactions in Python, will be certainly able to understand and gain great knowledge from this report. The core theme of the project focuses on the development of Security in ATM Transaction System in Python using Hash and RSA algorithms.

The report also contains the strategy used in making ATM Transactions, Comparison with different kinds of algorithm, advantages of Hash algorithm.

**Problem Statement**

ATTACKS ON VARIOUS POINTS:

The digital banking faces many robbery cases which are divided into the following main categories:

1. Attack on the digital infrastructure for accessing information about different funds transfer.
2. Digital Infrastructure attack on ATM management
3. Clients Side attacks while performing e-banking

Attacks related to ATM

1. Fraud

Fraud is done by the usage of fake cards. Skimming is one of the method by which fraud is performed. The attackers install the skimmers in the ATM’s card reading slots. These skimmers are not visible to the users of the ATM. The person coming for service of the machine checks these skimmers manually.

1. Physical

As the name suggests the attackers physically try to loot the cash out from the ATM. This is the most common attack. These are prevented deploying security personnel for the ATM.

1. Logic

These attacks mainly include use of malware. Skimmer is one of the examples of the logic attack. The attacker tries to get the knowledge about the algorithm from which the bank is performing the transactions.

The security and vulnerability are opposite sides of the same coin, an Automated machine becomes vulnerable due to weakness of its security. Automated teller machine manufactures go on adding and strengthening security features of Automated teller machine so that customer can carry banking transactions hassle free and without any fear of siphoning of amount from their account and the same frauds works with similar speed to crack the innovated security feature so that they can have access over the Automated teller machine to exploit the accounts of bank customers.

Automated teller machine security problems and Automated teller machine fraud problems typically follow some distinct patterns based mostly upon the situation of the Automated teller machine security attacks or Automated teller machine fraud incidents. Automated teller machine fraud problems within the most half involve MasterCard fraud and open-end credit fraud.

Because of the issues encountered with ATM card forgers, banks tried to cipher the PIN on the cardboard, or derive it from the account range, or offer another suggests that of checking it, in an exceedingly means that they hoped wouldn’t be too obvious to criminals and hackers. In fact, if one has got a Barclaycard or Barclay charge card that dates back a couple of decades, to detect that the primary and fourth digits of your PIN add up to an equivalent because the second and third, or that the primary and third add up to an equivalent because the second and fourth.

However this kind of security wasn’t abundant sensible against a bright forger, and this brings us to the second contribution that Automated teller machine created to laptop science and thus fostered industrial development of cryptography, that is that the study of codes and ciphers.

**Introduction**

**What is ATM?**

ATM stands for Automated Teller Machine. It’s a specialized computer that makes it convenient to manage your money. For example, almost all ATMs allow you to withdraw money, and many allow you to make deposits. At some ATMs, you can print a statement (a record of your account activity or transactions); check your account balances (the amount of money in your accounts right now); transfer money between your accounts; and even purchase stamps. You can usually access the most services at an ATM that’s operated by your own bank.

**Why use ATMs?**

ATMs are a safe and convenient way to manage your money. There are millions of ATMs worldwide and you can use many ATMs 24 hours a day, 7 days week. Some allow you to select the language you want to use.

**Is there a fee for using an ATM?**

Check with your bank to see if they charge any ATM fees to customers. Almost all banks do charge a fee to non-customers who use their ATMs. Keep in mind that even though using ATMs may cost you money, it’s much less expensive than using a check cashing service.

**Literature Review**

[1] The literature of behavioural studies on ATMs has mainly focused on adoption and diffusion of technology, impact of technology adoption from customers’ perspective, supplier’s perspective, and bankers perspective.

Major Technology Adoption Models as per the study of Norris and Yin (2008), Technology Adoption Research is almost twenty five years old and there are around eight important theories of adoption. All these eight theories are derived from the foundation of innovation diffusion and Technology acceptance model. The exception among eight models is the Social Cognitive Theory. Technology acceptance model is quite individual focused among eight adoption models. However, other models focus on how diffusion of innovation takes place within the firm. Oliveira and Maria (2011) have explored that there is a dearth of academic literature on reviews of adoption model at firm level used in Information Technology literature.

Author discussed Diffusion of Innovation Theory and Technology, Organization, and Environment (TOE)[7].For more complex new technology adoption, it is important to integrate more than one theoretical model to achieve a better understanding of the adoption phenomenon. But in many cases, technology adoption research is a replication without substantive theoretical advances. However, there are ample opportunities to make theoretical advances using our current knowledge as the starting point[5]. These are the conclusions of the authors based on their review and comparison of major milestones of technology adoption research, Job Satisfaction research and Theory of Planned behaviour. Most of the theories have emphasized on the factor influencing adoption behaviour and the process of diffusion of technology. These models also reveal the individual difference and cultural difference in technology adoption.

The first developed algorithm for the ATM transactions used to extract features of the bank, for e.g. Cash withdrawal, deposit or balance enquiry. These features were selected in a region form to perform transaction [4]. There was only simple encryption technique used like for example playfair cipher technique. This method was good but didn’t showed great results because of less amount of cryptography used.

In ATM Transactions we only focus on ATM features and mainly on security. The feature transaction is a topic of deep interest in the Finance domain. The system should be good enough to represent the ATM in feature form. Features of ATM are in two modules – admin and user. Admin module features include creation of new accounts, pin creation and updating them in the database. Admin module feature category includes taking the input profile information.

Today there are number of methods to extract information from an ATM to the Server. This method uses RSA and Hash function algorithm to implement this feature. This method is often termed as One way encryption method. Both these methods fall under admin module feature category[2]. Then this data is decrypted to information set that represents user’s bank information to server. Encryption is a multistage technique. Encryption uses all the relevant information from the ATM output information that is given by a user that is to be sent to the server. It forms a encrypted text of the account number, pin where the output of other is sent to the server using RSA.

User module features include pin updating, withdrawal, balance enquiry etc. The method used to encrypt this user information is RSA and hash function algorithm. We have implemented this system user friendly and totally secure to protect user information.

[7] RSA algorithm is used to encrypt the data providing security to the concerned user to be the one to access it. By securing the data, we are not allowing unauthorized access to it. User data is encrypted first and then it is stored in the Cloud. When required, user places a request for the data for the Cloud provider; Cloud provider authenticates the user and delivers the data. RSA is a block cipher, in which every message is mapped to an integer. RSA consists of Public-Key and Private-Key. In our Cloud environment, Pubic-Key is known to all, whereas Private-Key is known only to the user who originally owns the data. Thus, encryption is done by the Cloud service provider and decryption is done by the Cloud user or consumer. Once the data is encrypted with the Public-Key, it can be decrypted with the corresponding Private-Key only

[8] Investigation upon implementation of RSA cryptosystem using the proposed RNS Montgomery multiplication algorithm, and design an RSA LSI to conﬁrm the performance and feasibility of the proposed algorithm. As implementation methods, RSA decryption procedures without and with the Chinese Remainder Theorem (CRT) are presented. The Cox-Rower Architecture is characterized by the scalability for operating time and chip size depending on the number of Rower units. In implementation, the relation between the number of Rower units and the base size in RNS representation becomes important for the performance, because operations for each base element are performed in parallel at Rower units. For an LSI prototype using 0.25 µm CMOS, we obtain 4.2 msec for 1024-bit RSA cryptosystem without CRT and 2.4 msec with CRT. This result is comparable with the present best performance of commercial chips

[9] New algorithm is described to prevent fault attacks on RSA signature algorithms using the Chinese Remainder Theorem (CRT-RSA). This variant of the RSA signature algorithm is widely used on smartcards. Smartcards on the other hand are particularly susceptible to fault attacks like the one described in Recent results have shown that fault attacks are practical and easy to accomplish Therefore, they establish a practical need for fault attack protected CRT-RSA schemes. Starting from a careful derivation and classification of fault models, we describe a new variant of the CRT-RSA algorithm. For the most realistic fault model described, we rigorously analyze the success probability of an adversary against our new CRT-RSA algorithm. Thereby, we prove that our new algorithm is secure against the Bellcore attack.

[10] With the increase of automated teller machine (ATM) frauds, new authentication mechanisms are developed to overcome security problems of personal identification numbers (PIN). Those mechanisms are usually judged on speed, security, and memory in comparison with traditional PIN entry systems. It remains unclear, however, what appropriate values for PIN-based ATM authentication actually are. We conducted a field study and two smaller follow-up studies on real-world ATM use, in order to provide both a better understanding of PIN-based ATM authentication, and on how alternative authentication methods can be compared and evaluated. Our results show that there is a big influence of contextual factors on security and performance in PIN-based ATM use. Such factors include distractions, physical hindrance, trust relationships, and memory. From these findings, we draw several implications for the design of alternative ATM authentication systems, such as resilience to distraction and social compatibility.

[11] To prevent ATM fraud and other ATM security related problems, one of the most recent technologies adopted by banking industries is two factor authentication which often combines the use of PIN and One Time Password (OTP) for user’s authentication. Since the OTP technology includes the use of SMS for delivery of the OTP from banks to customers, its security is based on the security of SMS which is extremely vulnerable to variety of attacks as they are transmitted as plain text. Mary Agyi and Devrim Seral have proposed an additional layer of security that protects the existing authentication system. Their system provides a secure end-to-end communication of OTP to customer’s by encrypting the SMS using Elliptic curve encryption. This is an asymmetric encryption technique.The authors chose this encryption method due to its ability of using smaller key size to obtain same security as compared to other asymmetric encryption techniques.

[12] The researchers from Sun Microsystems Laboratories compared and analysed the performance of Elliptic Curve Point Multiplication (ECC) and RSA on two 8-bit micro-controllers. They observed that ECC-160 point multiplication outperforms the RSA-1024 private-key operation by an order of magnitude and is within a factor of 2 of the RSA-1024 public-key operation.And thus, as compared to RSA, the widely-used public-key scheme, Elliptic Curve Cryptography offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings and is thus better suited for small devices. Also, to accelerate multiple-precision multiplication, they have proposed a new algorithm to reduce the number of memory accesses. Their proposed algorithm led to a 25% performance increase for ECC point multiplication on the Atmel AVR platform.

[13] Selvaraju and G.Seka have proposed an embedded Crypto-Biometric authentication scheme for ATM banking system. Their proposed model fuses cryptography and biometric techniques together for user authentication to improve the security level. They have used Advanced Encryption Standard (AES), a 128 bit private key algorithm, to encrypt the user’s fingerprint required during a transaction.They chose AES for encryption purpose since it is fast in both software and hardware, relatively easy to implement, and requires little memory.

[14] Navneet Sharma and Vijay Singh Rathore have discussed the different types of encryption methods and standards which are used between Auto teller machine and bank server during the transmission of private data. They concluded that 3DES (a revised version of DES) and AES are more safer and most of the banks are using the AES to protect their data from hackers.Yet, there are security issues in the use 3DES, as chances of assaults with the use of 232 Known-plaintexts, 2113 steps, 2 90 DES-solving, and 288 memory capacity exist.

[15] In his paper, Nentawe Y. Goshwe has presents a design of data encryption and decryption in a network environment using RSA algorithm (with a specific message block size).The RSA algorithm can be used for both key exchange and digital signatures.With the use of RSA algorithm a message sender will generate a public key to encrypt the message and the receiver will be sent a generated private key using a secured database. An eavesdropper or hacker who tries to break into the message will get a meaningless message. Also, Java and MySQL have been used to create a GUI for the user

[16] In their paper, Dr. Prerna Mahajan and Abhishek Sachdeva have implemented three encryption algorithms, AES, DES and RSA algorithms, and compared their performance based on the analysis of their encryption and decryption times.It was observed that AES algorithm consumes least encryption and RSA consume longest encryption time. The decryption of AES algorithm was found to be much better than the other two algorithms. And thus, from the simulation result, they have concluded that AES algorithm is much better than DES and RSA algorithm.

[17] Authenticated session key establishment is a central issue in network security. This paper addresses the question of whether we can design a compact, efficient and authenticated key establishment protocol that has the following two properties: (1) each message exchanged between two participants can be transferred in a short packet such as an ATM cell whose payload has only 384 bits, and (2) messages that carry key materials are unforgeable and non repudiatable without the involvement of a trusted key distribution center. We discuss why the answer to this question is negative if one follows the currently standard approach to key establishment, namely employing secret/public key encryption and, possibly, digital signature. We then present a number of protocols that represent a positive answer to the question. Our protocols are all based on a cryptographic primitive called "sign cryption" that fulfils both the functions of digital signature and public key encryption with a cost far smaller than that required by "digital signature followed by encryption".

# [18] According to the present invention, the solution includes the hardware hash algorithm block to automatically generate data to hash from its initialization values and to run unassisted instead of needing a continuous supply of additional input data. This approach according to the present invention solves the above shortcomings of related solutions by eliminating the need to continuously feed input data to be hashed to obtain a high fault coverage. This reduces the sizes of the firmware and test vectors necessary to test the hardware. Also, since the hardware autonomously generates new data to hash, other hardware modules can be tested in parallel. This reduces the overall test time and cost. To remove the requirement of inputting multiple fixed length sub-blocks, additional sub-blocks are created from the initial sub-block using a hardware expansion function, and the hardware continues to run unattended for some predetermined number of sub-blocks.

[19] Disclosed are exemplary techniques for compressing ATM cells by substituting reduced headers for the original headers. A header table may be maintained at the transmitting end whereby headers are stored in table entries based on a table index generated from a hash function applied to the headers. A copy of the header table may be maintained at the receiving end and may be updated by populating idle cells with header table contents. Upon receipt of a reduced ATM cell at the receiving end, the reduced header may be replaced with a full header obtained from the copy of the header table maintained at the receiving end. The present invention further provides techniques for registration and management of hash collisions.

[20] RSA is a family of trapdoor permutations. Mihir Bellare and Phillip Rogaway have proposed a new RSA-based signing scheme and signing with message recovery, namely, PSS and PSS-R. The signing process includes RSA decryption and some hashing. The verification process includes encryption with and hashing. The proposed scheme is a probabilistic generalization of Full Domain Hash(FDH) Scheme. It is as efficient as the original FDH but provides a better security bound. The security of PSS is purely based on the security of RSA algorithm.

**Implementation**

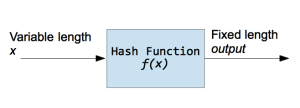
**ALGORITHM USED IN OUR PROJECT**

In ATM Transactions system, we need to secure sensitive data of user like PIN, etc.

To maintain this security, we implemented this system cryptography using hash function and RSA algorithm.

**HASH FUNCTION ALGORITHM**

A hash function is a function that takes input of a variable length sequence of bytes and converts it to a fixed length sequence. It is a one way function. This means if *f* is the hashing function, calculating *f(x)* is pretty fast and simple, but trying to obtain *x* again will take years. The value returned by a hash function is often called a hash, message digest, hash value, or checksum. Most of the time a hash function will produce unique output for a given input. However depending on the algorithm, there is a possibility to find a collision due to the mathematical theory behind these functions.



Hash functions are used inside some cryptographic algorithms, in digital signatures, message authentication codes, manipulation detection, fingerprints, checksums (message integrity check), hash tables, password storage and much more.

Some of the Hash function algorithm examples are :

* MD5
* SHA (SHA1,SHA224,SHA256,SHA384,SHA512)

**RSA ALGORITHM**

Key Generation :-

1. Generate two large prime numbers suppose p & q.
2. n = p\*q
3. m = Φ(n) = (p-1)\*(q-1)
4. Choose a small number e , co prime to m and GCD(m , e) = 1 such that 1<e< Φ(n)
5. Find d such that (d\*e)mod Φ(n) = 1

* Cipher text = (plain text)e mod n
* Plain text = (Cipher text)d mod n

The ATM sends the pin using RSA encryption to the server. The Server decrypts that and uses hash function for further processing.

**System Model and Methods**

**MODULE COMPONENTS:**

In this Project, we will be using two modules :

* Admin
* ATM (user)

**ADMIN Module:**

If there comes a new user, he needs to create a new account/profile where the account number will be given serial wise and the user needs to enter his/her personal details. The given details by user are stored in encrypted form in database. After creating new account, the new user is provided with the card number for using ATM service.

After creating and giving his/her details to the server, if he/she needs to update his/her details, he/she should enter card number to login and to check validity. He/she enter the updated details and the same details is updated in the database.

**Code**

import sqlite3

def checK(ck):

conn=sqlite3.connect("E:\Okay Crypto\sql.db")

cmd="select \* from details where Card\_No="+str(ck)

cursor=conn.execute(cmd)

profile=None

for row in cursor:

profile=row

conn.close

return profile

def gcd(a, b):

while(b > 0):

c = a % b

a = b

b = c

return a

def mulinv(phi,b):

r1 = phi

r2 = b

t1=0;

t2=1;

while(r2>0):

q = r1 / r2

r=r1 -q\*r2;

r1=r2

r2=r

t=t1-q\*t2

t1 = t2

t2 = t

if(r1==1):

if(t1<0):

t1=t1+phi

return t1

else:

return t1

p=353

q=281

phi=(p-1)\*(q-1)

n=p\*q

e=2

while(e<phi):

if(gcd(e,phi)==1):

break

else:

e=e+1

d=mulinv(phi,e)

def decrypt(c):

y=int((c\*\*d)%n)

return y

def encrypt(msg):

c=int((msg\*\*e)%n)

return c

def update(cardNo,pin):

conn=sqlite3.connect("E:\Okay Crypto\sql.db")

cmd="select \* from details where Card\_No="+int(cardNo)

cursor=conn.execute(cmd)

ifRecordExist=0

for row in cursor:

ifRecordExist=1

if(ifRecordExist==1):

cmd="update details SET Pin="+int(pin)+" where Card\_No="+int(cardNo)

conn.execute(cmd)

conn.commit()

conn.close()

def updateB(cardNo,balance):

conn=sqlite3.connect("E:\Okay Crypto\sql.db")

cmd="select \* from details where Card\_No="+int(cardNo)

cursor=conn.execute(cmd)

ifRecordExist=0

for row in cursor:

ifRecordExist=1

if(ifRecordExist==1):

cmd="update details SET Balance="+int(balance)+" where Card\_No="+int(cardNo)

conn.execute(cmd)

conn.commit()

conn.close()

def server(cardNo,pin\_user):

pin\_user=decrypt(pin\_user)

profile=checK(cardNo)

if(profile!=None):

pin=profile[5]

Pin=int(pin)

balance=int(profile[4])

if(Pin == 0):

pin=int(input('Create a new Pin: '))

if(len(pin)!=4):

print('Create a correct Pin')

pin=int(input('Enter the Pin: '))

print('Pin Created Successfully')

pin=hash(pin)

update(cardNo,pin)

else:

pin\_user=int(pin\_user)

pin\_user=hash(pin\_user)

if(Pin==pin\_user):

if(balance>Amt):

balance=balance-Amt

print ('Transaction Success')

print ('Available Balance',balance)

updateB(cardNo,balance)

else:

print('Not Sufficient Balance')

else:

print('Enter the correct Pin')

if(profile==None):

print ('Enter Correct Card No')

def server2(cardNo,pin\_user):

pin\_user=decrypt(pin\_user)

profile=checK(cardNo)

if(profile!=None):

pin=profile[5]

balance=profile[4]

if(pin == 0):

pin=int(input('Create a new Pin: '))

if(len(pin)!=4):

print('Create a correct Pin')

pin=int(input('Enter the Pin: '))

print('Pin Created Successfully')

pin=hash(pin)

update(cardNo,pin)

else:

pin\_user=int(pin\_user)

pin\_user=hash(pin\_user)

if(pin==pin\_user):

print('available Balance :',balance)

else:

print('Enter the correct Pin')

if(profile==None):

print ('Enter Correct Card No')

def server3(cardNo,pin\_user):

pin\_user=decrypt(pin\_user)

profile=checK(cardNo)

if(profile!=None):

pin=profile[5]

if(pin==0):

pin=int(input('Create a new Pin: '))

if(len(pin)!=4):

print('Create a correct Pin')

pin=int(input('Enter the Pin: '))

print('Pin Created Successfully')

pin=hash(pin)

update(cardNo,pin)

else:

pin\_user=int(input('Enter Old Pin: '))

pin\_user=hash(pin\_user)

if(pin==pin\_user):

newpin=int(input('Create a new Pin: '))

if(len(newpin)!=4):

print('Create a correct Pin')

newpin=int(input('Enter the Pin: '))

print('Pin Created Successfully')

pin=hash(newpin)

update(cardNo,pin)

else:

print ('Enter the correct pin')

if(profile==None):

print ('Enter Correct Card No')

print ('\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Welcome\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n')

a=int(input('\n1. Withdrawl \n2. Check Balance \n3. Change Pin \n4. Exit\nEnter the choice: '))

while(a!=4):

if(a==1):

cardNo=int(input('Enter the Card No.: '))

pin\_user=int(input('Enter the pin: '))

pin\_user=encrypt(pin\_user)

amnt=int(input('Enter the Amount to be Withdrawn: '))

Amt=int(amnt)

server(cardNo,pin\_user)

if(a==2):

cardNo=int(input('Enter the Card No.: '))

pin\_user=int(input('Enter the pin: '))

pin\_user=encrypt(pin\_user)

server2(cardNo,pin\_user)

if(a==3):

cardNo=int(input('Enter the Card No.: '))

pin\_user=int(input('Enter the pin: '))

pin\_user=encrypt(pin\_user)

server3(cardNo,pin\_user)

z=int(input('\nDo you want to continue:\n1. Yes\t2.No \nEnter the Choice: '))

if(z==1):

a=int(input('\n1. Withdrawl \n2. Check Balance \n3. Change Pin \n4. Exit\nEnter the choice: '))

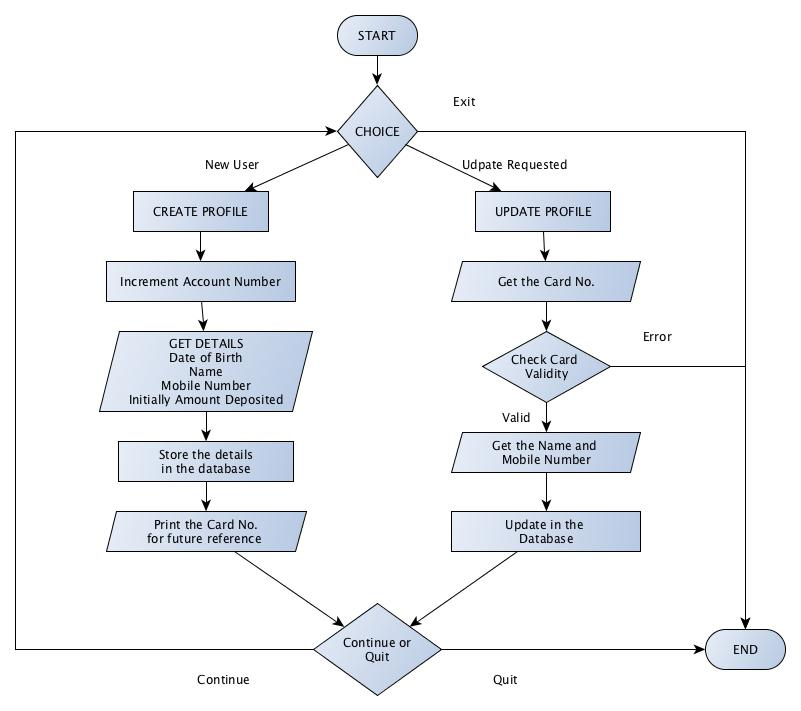
if(z==2):

break

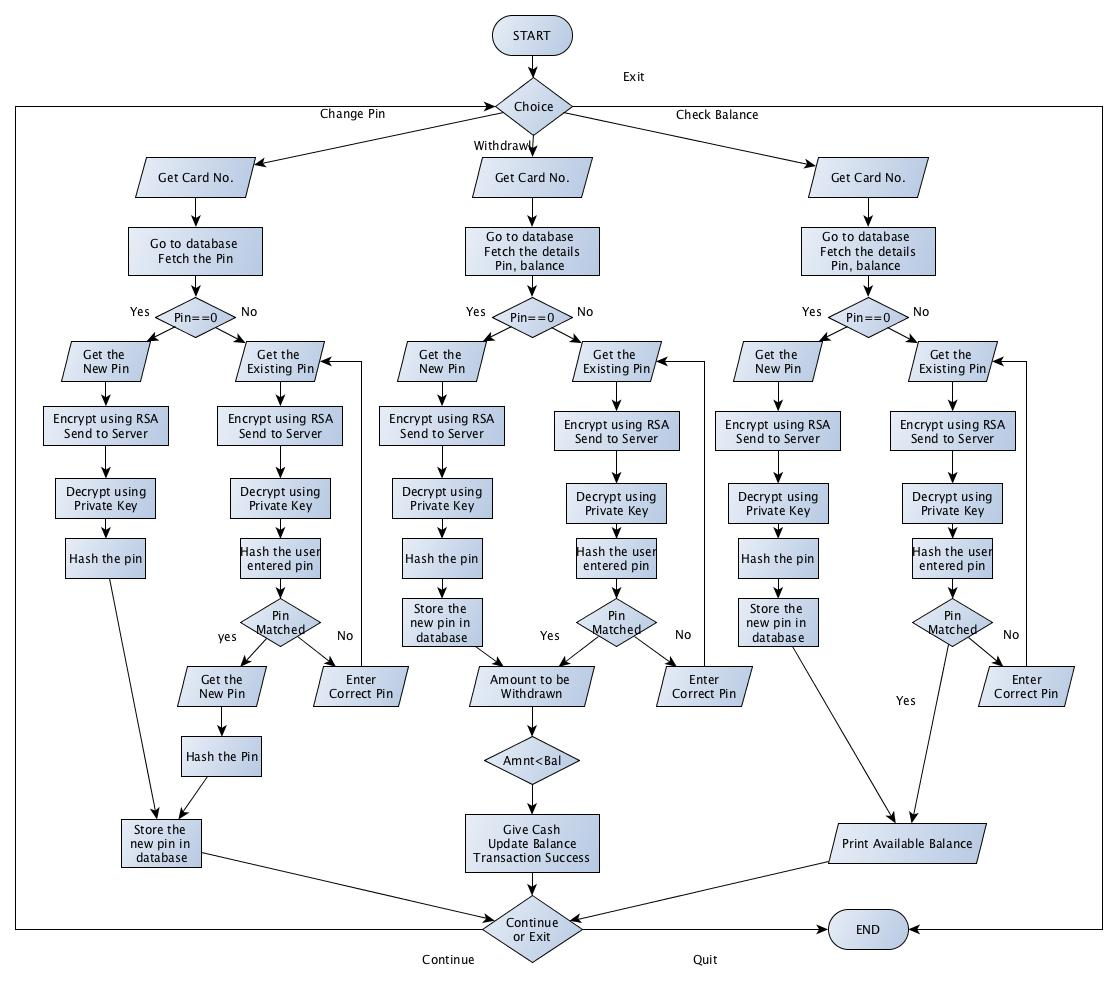
if(a==4 or z==2):

print ("Good Bye")

exit



**ATM :**

****

ATM Services:

* Update PIN
* Withdrawal and update balance
* Print Available balance

**Code**

import sqlite3

class birth:

date = int

month = str

year = int

def update(Id):

conn=sqlite3.connect("E:\Okay Crypto\sql.db")

cmd="select \* from details where Card\_No="+str(Card)

cursor=conn.execute(cmd)

ifRecordExist=0

for row in cursor:

ifRecordExist=1

if(ifRecordExist==1):

print ('Enter the Details')

name=raw\_input("Enter the Name: ")

cmd="update Details set Name="+str(name)+" where Card\_No="+str(Card)

conn.execute(cmd)

phone=int(input("Enter the phone number: "))

cmd="update Details set Phone\_No="+str(phone)+" where Card\_No="+str(Card)

conn.execute(cmd)

conn.commit()

else:

print('Enter the Correct Details')

conn.close()

def insert(accNo):

conn=sqlite3.connect("E:\Okay Crypto\sql.db")

cmd="select \* from details where Account\_No="+str(accNo)

cursor=conn.execute(cmd)

cmd="insert into details (Account\_No,Name,Card\_No,Phone\_No,Balance,Pin) values("+str(v)+","+str(name)+","+str(cardNo)+","+str(phone)+","+str(bal)+","+str(pin)+")"

conn.execute(cmd)

conn.commit()

conn.close()

cursor=0

def count(cursor):

conn=sqlite3.connect("E:\Okay Crypto\sql.db")

cmd="select count(\*) from details"

cursor=conn.execute(cmd)

results=cursor.fetchone()[0]

conn.commit()

conn.close()

return results

def checK(ck):

conn=sqlite3.connect("E:\Okay Crypto\sql.db")

cmd="select \* from details where Card\_No="+str(ck)

cursor=conn.execute(cmd)

profile=None

for row in cursor:

profile=row

conn.close

return profile

cardNo=632000

print ('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Welcome\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

a=int(input('\n1. Create Profile \n2. Update Profile \n3. Exit\nEnter the choice: '))

while(a!=3):

if(a==1):

i=count(cursor)

accNo=i+1

v=accNo

accNo=birth()

print ('Enter the Details')

accNo.date=int(input('Enter the Day of birth: '))

accNo.month=str(input('Enter the Month of birth: '))

accNo.year=int(input('Enter the Year of birth: '))

name=raw\_input("Enter the Name: ")

phone=int(input("Enter the phone number: "))

bal=int(input('Enter the amount deposited: '))

y=accNo.date

x=accNo.month

j=accNo.year

pin=0

cardNo+=v

print ('Card No:',cardNo)

insert(v)

print ('Created')

if(a==2):

Card=int(input('Enter the Card No.: '))

profile=update(Card)

print("Updated Succesfully")

z=int(input('\nDo you want to continue:\n1. Yes\t2. No \nEnter the Choice: '))

if(z==1):

a=int(input('\n1. Create Profile \n2. Update Profile \n3. Exit\nEnter the choice: '))

if(z==2):

break

if(a==3 or z==2):

print ("Good Bye")

exit

**Processing :**

To implement ATM transactions we have divided our project into two parts. One module comprises of the Administrator and the other one is the ATM.

The Admin module is responsible for creation of the database, and handling the customers. Admin module just assigns the card no to the individuals that come to open a new bank account. The bank balance is updated and stored in the database. There is no pre-assigned pin for the users; the user sets his pin on his own when he goes to the ATM for the first time.

In the ATM when the user starts his transaction, the ATM sends the data to the server. The server then checks and compares them in the database. If the card number is wrong then it asks to enter the correct card else it asks for the pin.

If coming for the first time then it asks for the new pin generation else for his transaction code. The pin is encrypted by the RSA algorithm and sent to the Server, the server firstly decrypts the pin and then hashes the pin.

Hashing is used because it is one-way profile and no MIM attack can take place over it. If user comes for the first time, hashed pin is stored in the database. Else the hashed pin is then compared to the stored hashed pin. If the pin matches, it transacts the amount and the same is updated in the database. If pin doesn’t matches it asks for correct pin and aborts the transaction.

**Performance Evaluation**

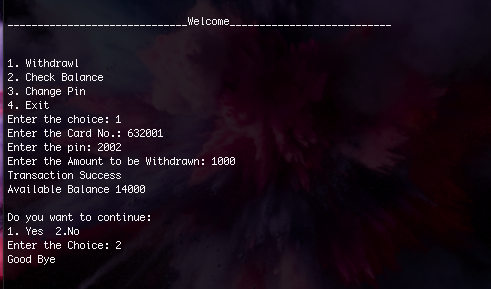
To implement ATM transactions we have divided our project into two parts. One module comprises of the Administrator and the other one is the ATM.

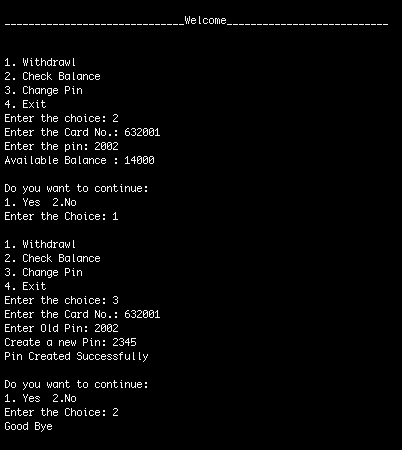
The Admin module is responsible for creation of the database, and handling the customers. Admin module just assigns the card no to the individuals that come to open a new bank account. The bank balance is updated and stored in the database. There is no pre-assigned pin for the users; the user sets his pin on his own when he goes to the ATM for the first time.

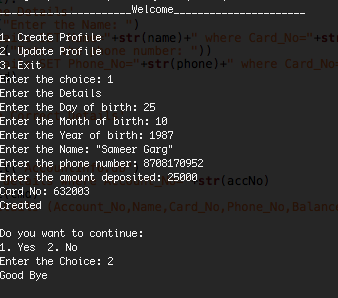
In the ATM when the user starts his transaction, the ATM sends the data to the server. The server then checks and compares them in the database. If the card number is wrong then it asks to enter the correct card else it asks for the pin.

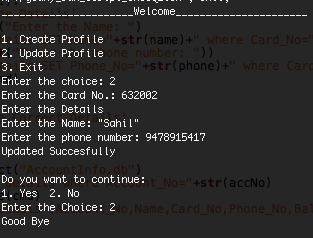
If coming for the first time then it asks for the new pin generation else for his transaction code. The pin is encrypted by the RSA algorithm and sent to the Server, the server firstly decrypts the pin and then hashes the pin.

Hashing is used because it is one-way profile and no MIM attack can take place over it. If user comes for the first time, hashed pin is stored in the database. Else the hashed pin is then compared to the stored hashed pin. If the pin matches, it transacts the amount and the same is updated in the database. If pin doesn’t matches it asks for correct pin and aborts the transaction.

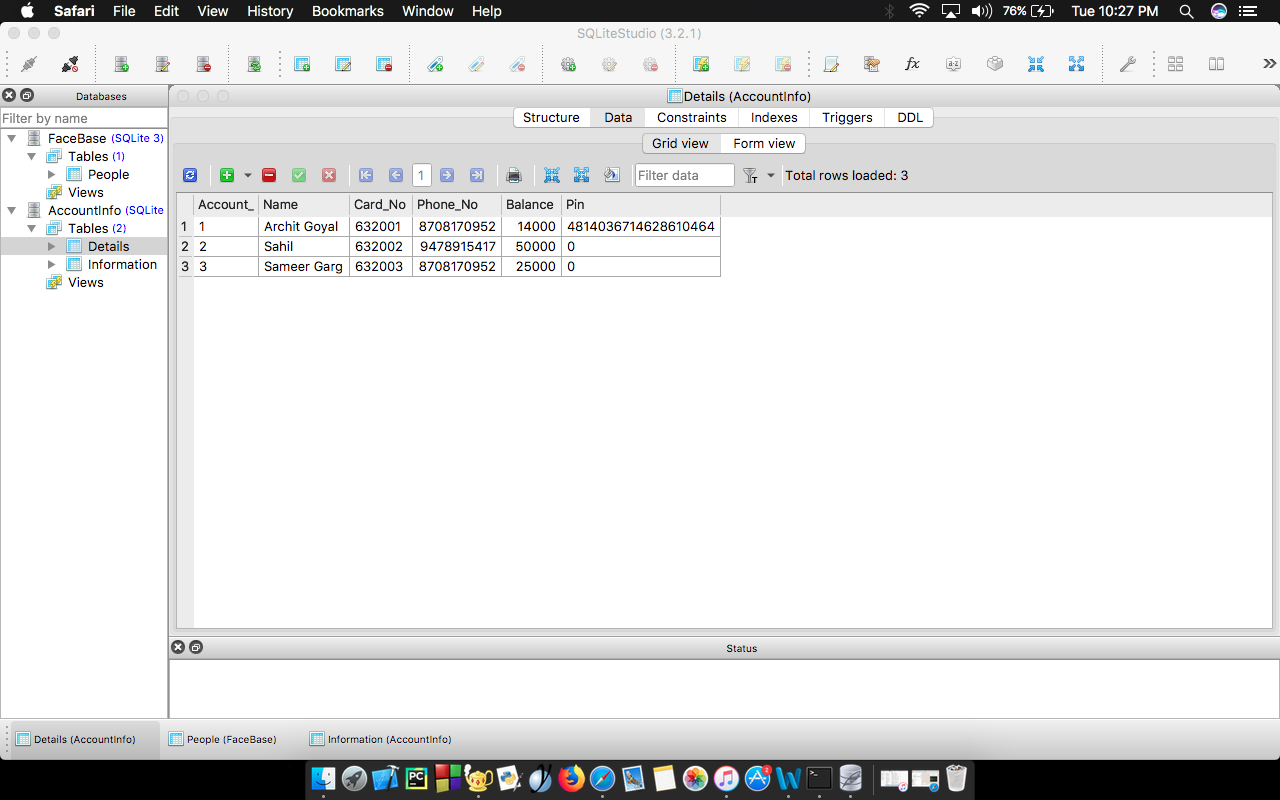












**Results and Discussion**

We have made Secured Data Encryption System using Python language and it is storing details in encrypted form of user ensuring security to user. The RSA technique used helps to prevent the third party from knowing the keys of the user. The Hash function prevents the Man in Middle attack.

It can implemented in real world for personal uses.

**REFERENCES**

[1]Jyotiranjan Hota, “Automated Teller Machines in India”, Proceedings of GLOGIFT 13, Dec-2013

[2]Oliveira, T. and Martins, M, F. (2011) Literature Review of Information Technology Adoption Models at Firm Level, The Electronic Journal Information Systems Evaluation, 14(1): 110-121.

[3] Agarwal, R. and Prasad, J. (1998) A Conceptual and Operational Definition of Personal Innovativeness in the Domain of Information Technology, Information Systems Research, 9(2): 204-215.

[5] Identifiers for Digital Identity Management.

[6] Devinaga, R. (2010). ATM risk management and controls.European journal of economic, finance and administrative sciences.ISSN 1450-2275 issue 21.

[7] Parsi Kalpana, Sudha Singaraju, “Data Security in Cloud Computing using RSA Algorithm”, International Journal of Research in Computer and Communication technology, IJRCCT, ISSN 2278-5841, Vol 1, Issue 4, September 2012

[8] Hanae Nozaki, Masahiko Motoyama, Atsushi Shimbo, and Shinichi Kawamura, “Implementation of RSA Algorithm Based on RNS Montgomery Multiplication”, SpringerLink, 20 September 2001

[9] Johannes Blomer, Martin Otto, Jean- Pierre Seifest, “A new CRT-RSA algorithm secure against bellcore attacks”, Proceedings of the 10th ACM conference on Computer and communications security, October 27 - 30, 2003

[10] Alexander De Luca, Marc Langheinrich, Heinrich Hussmann, “Towards understanding ATM security: a field study of real world ATM use”, Proceedings of the Sixth Symposium on Usable Privacy and Security, Article No. 16, July 14 - 16, 2010

[11] Mary Agoyi , Devrim Seral, “The use of SMS encrypted message to secure automatic teller machine, [Procedia Computer Science](https://www.sciencedirect.com/science/journal/18770509), Volume 3, 2011, Pages 1310-1314

[12] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz, “Comparing Elliptic Curve Cryptography and RSA on 8-Bit CPUs”,Part of the Lecture Notes in Computer Science book series (LNCS, volume 3156)

[13] N Selvaraju,G.Seka, “A Method to Improve the Security Level of ATM Banking Systems Using AES Algorithm ”, International Journal of Computer Applications (0975 – 8887)

[14] Navneet Sharma , Vijay Singh Rathore, “Different Data Encryption Methods Used in Secure Auto Teller Machine Transactions”, International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-1, Issue-4, April 2012

[15] Nentawe Y. Goshwe, “Data Encryption and Decryption Using RSA Algorithm in a Network Environment”, IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.7, July 2013

[16] Dr. Prerna Mahajan, Abhishek Sachdeva, “A Study of Encryption Algorithms AES, DES and RSA for Security”, Global Journal of Computer Science and Technology Network, Web & Security, Volume 13 Issue 15 Version 1.0 Year 2013

[17] Yuliang Zheng, H Imai, “Compact and unforgeable key establishment over an ATM network”, IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98)

[18] Matthew H. Childs, Thomas M. Norcross, “Automatic data generation for self-test of cryptographic hash algorithms in personal security devices”, National Semiconductor Corp

[19] Yan Wu, Song Bi, Timothy Bornemisza, “ATM header compression using hash tables”, Ikanos Communications Inc

[20] Mihir Bellare, Phillip Rogaway, “The Exact Security of Digital Signatures- How to Sign with RSA and Rabin”,Part of the Lecture Notes in Computer Science book series (LNCS, volume 1070)