PAYMENT VIA MOBILE PHONES

By:

Akash Mantry

Rajvardhan Deshmukh

Shamanth Kumar P

Outline

- Motivation
- Background
- Security Problem
- Security Method
- Proposed Project Algorithm
- Implementation Details
- Project Application Demo
- Testing
- References

Motivation

- Throughout history human beings have relied on some sort of payment system to purchase the goods or services we wanted or needed. Mobile devices have changed business and now possibly the way financial transactions of all kinds are made. Consumers are willing to utilize mobile phones for payment purposes as it provides an efficient and easier way to process financial transactions.
- Even though widely used, many security concerns are also involved with payment through mobile phones. One of the biggest threats can be the interception of the traffic when the mobile payment is in process. This can lead to identity theft, identification disclosure and replay attacks. Poor data protection controls also lead to data disclosure and privacy infringement. If mobile payment is not done securely then any intruder can obtain the credit card details of the user and use it to transfer illegal funds.

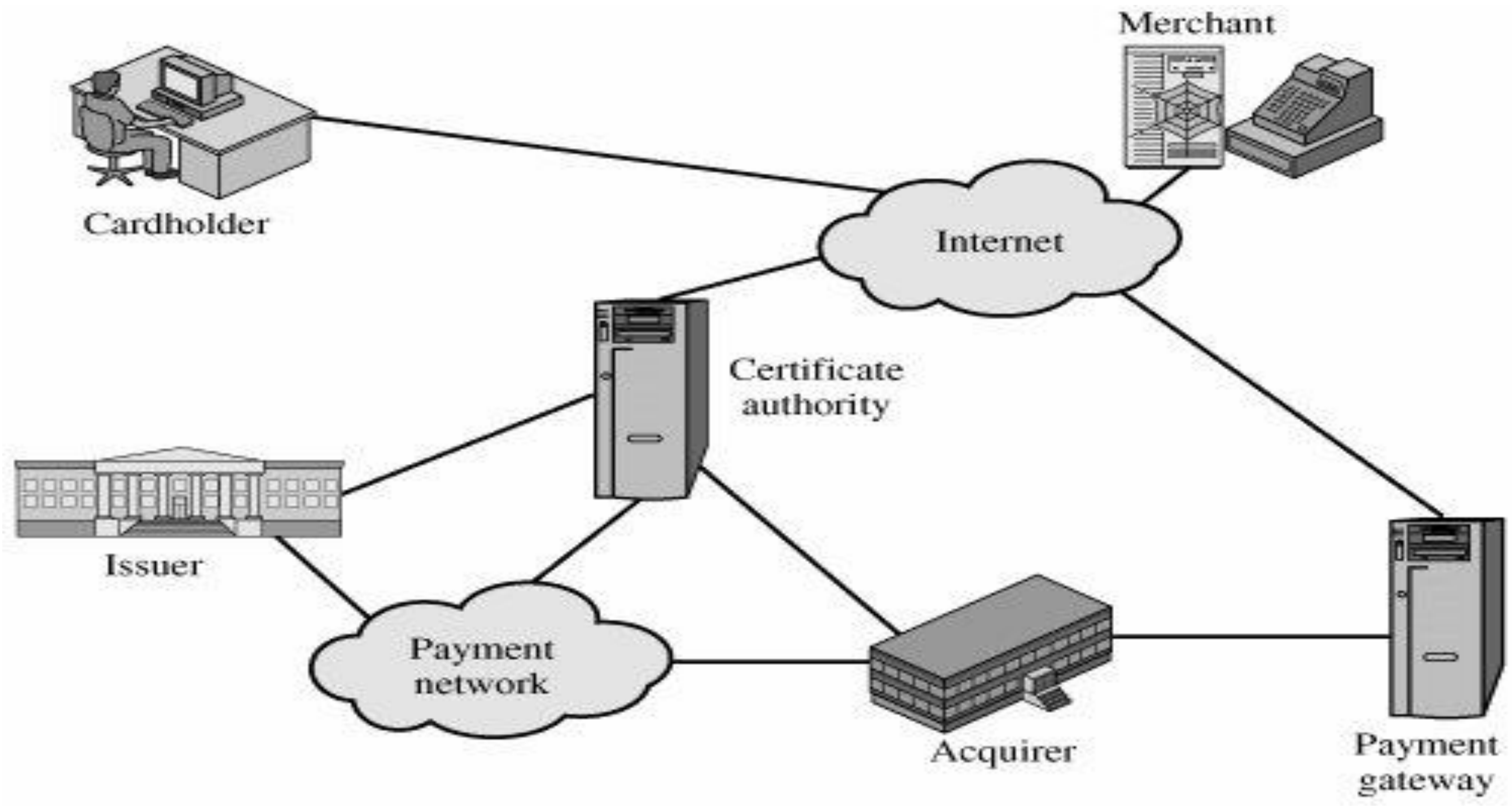
Background

- Payment through mobile phones has become one of the major methods of financial transaction used all around the world. With the advent of technology there has also been numerous risks and threats have also increased which hamper the way mobile payment takes place. There is urgent need for the security issue of mobile payment to be addressed.
- We are using a method widely known as the Secure Electronic Transaction to solve the security issues of mobile payment. The main purpose of the project it to solve issues like interception of payment info of the user and using such information for malicious purposes. SET achieves this by using the concept of Dual Signature. With SET, the user is given a digital certificate and a transaction is conducted and verified using a combination of digital signature and digital certificates among the user, the merchant and the user's bank.

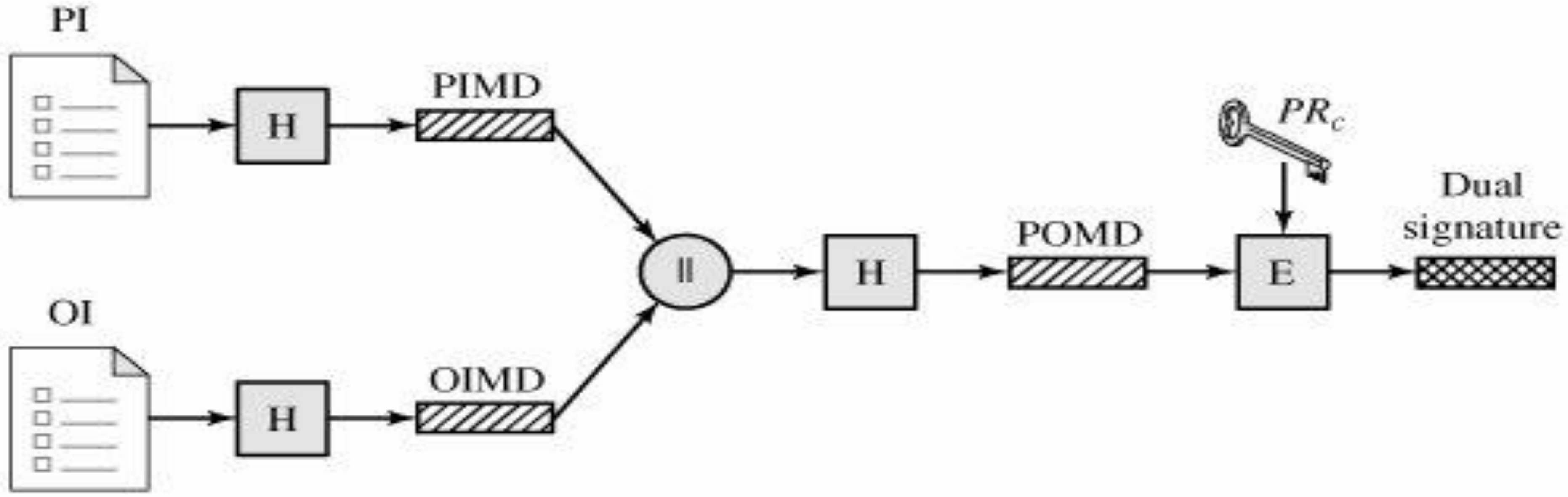
Security Problem in Payment Via Mobile Phones

- Interception of Traffic
- Identity Theft
- Information Disclosure
- Masquerade Attacks
- Illegal transfer of funds
- Replay Attacks

Security Method : Secure Electronic Transaction



Dual Signature in SET



PI = Payment information
OI = Order information
H = Hash function (SHA-1)
|| = Concatenation

PIMD = PI message digest
OIMD = OI message digest
POMD = Payment order message digest
E = Encryption (RSA)
PR_c = Customer's private signature key

Implementation Details of the Project

Proposed Project Algorithm

Components:

- Client –The person who orders product online.
- Merchant Server – The server corresponding to the merchant.
- Payment Gateway – The payment gateway for payment of the bill corresponding to the products ordered.
- Certification Authority – The trusted certification authority which provides the valid Digital signature.

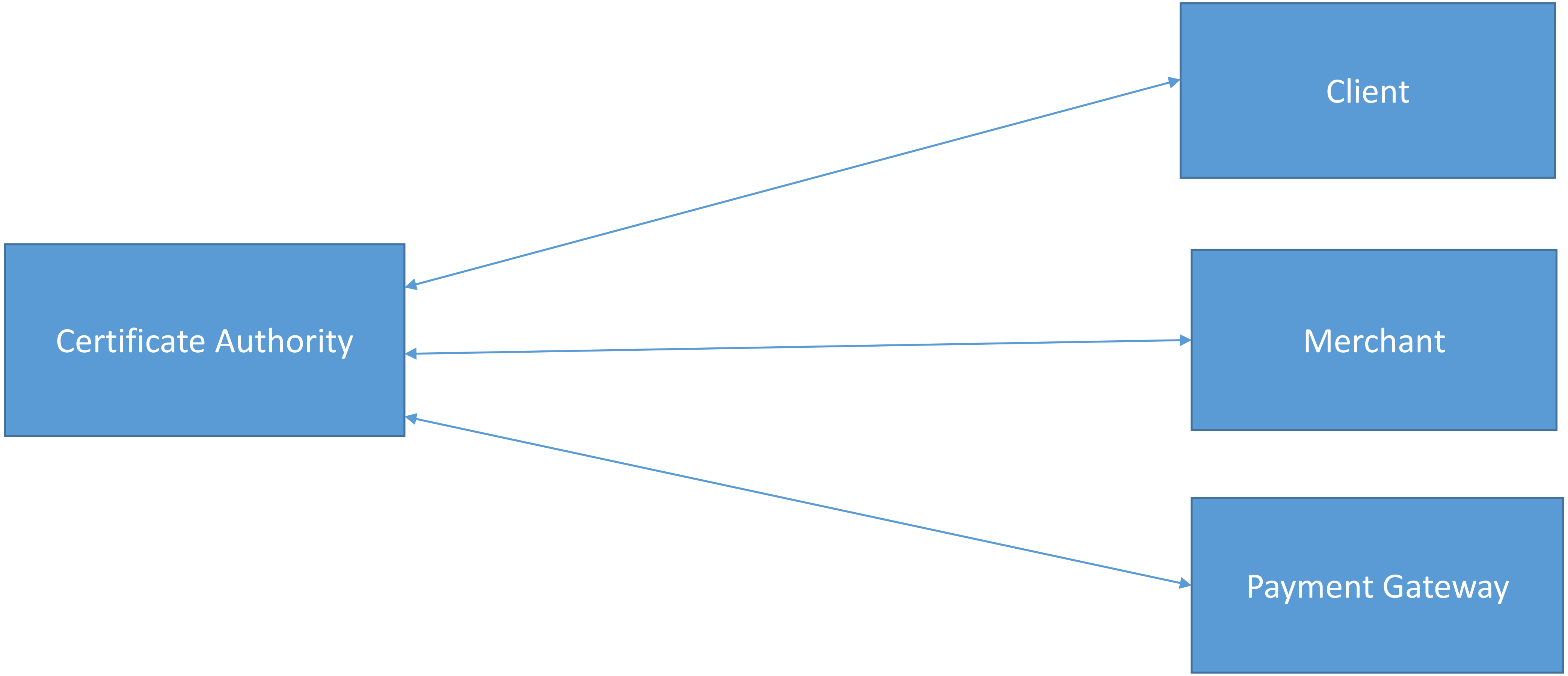
Assumptions made in the Project

1. Every component has the Certification Authority’s public key.
2. CA can be trusted.
3. Duplicate Session Key cannot be created.
4. Encrypted Timestamp cannot be altered by the attacker.
5. Database is combined for both merchant and bank.
6. Client can choose 1 product at a time.
7. OTP is sent to only registered phone number in Twilio.
8. Email notification is secure since it is handled by Gmail.
9. Product Name and Card Number are being sent to Merchant and Bank Server respectively.
10. No database is maintained for the credit card details.

Topics used in our Project

- RSA encryption with 512 bit key
- Certification Authority (X.509v1 Certificates)
- Dual Signature
- Session Keys
- Salt
- MAC(MD-5)
- HMAC SHA-1 – Password
- TimeStamps

Certificate and Key Distribution



Client

Certificate Authority

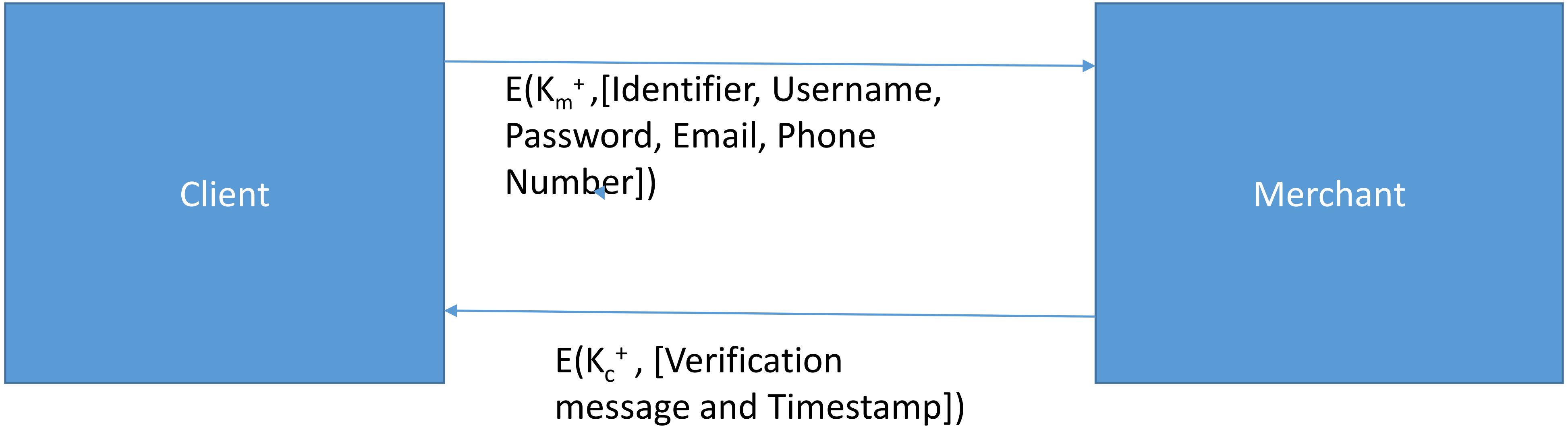
Merchant

Payment Gateway

Flow 1 : Certification Authority → Client, Server and Payment Gateway

- CA distributes its public key to client, server and PG through one of the key exchange methods.
- CA encrypts the message known by everyone with its private key and provides the digital signature.
- Client, Server and PG can decrypt it using the public key of the CA which was distributed.
- C, S and PG recognize CA and send their public certificates to the CA.
- CA stores the digitally signed certificate for client, server and PG.

1) Registration of New User



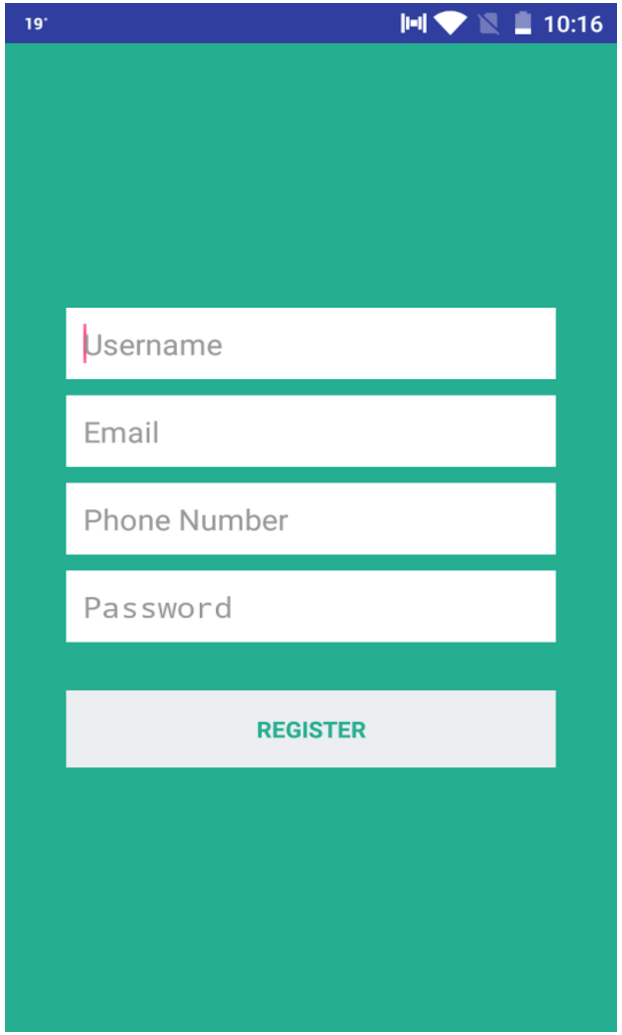
Client

E(K_m^+,[Identifier, Username, Password, Email, Phone Number])

E(K_c^+, [Verification message and Timestamp])

Merchant

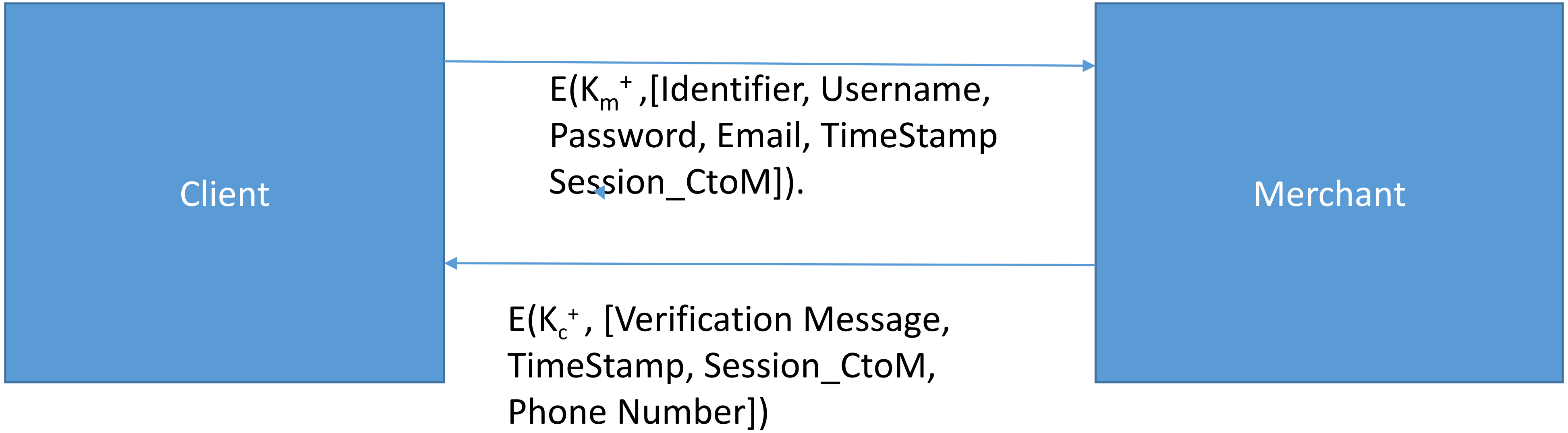
Registration of New User Page



Implementation of Registration Page:

- Register new user using the fields – Username, Email, Phone Number and Password.
- The registration details are sent along with the Identifier(“Registration”).
- All the fields sent are encrypted with merchant’s public key hence providing security.
- A table called “User” table maintains the list of the users in the database.
- When a user registers and the user is found to be unique the registration process succeeds and the newly registered user details is updated in the “User” table.
- The merchant sends the verification message(“Successful” or “Unsuccessful”) and the timestamp as reply. The message and the timestamp are encrypted with the public key of the client.
- If the user already exists or if the time condition fails then the registration process fails.
- After successful registration, the user is directed to Login Page.

2) User Login Page



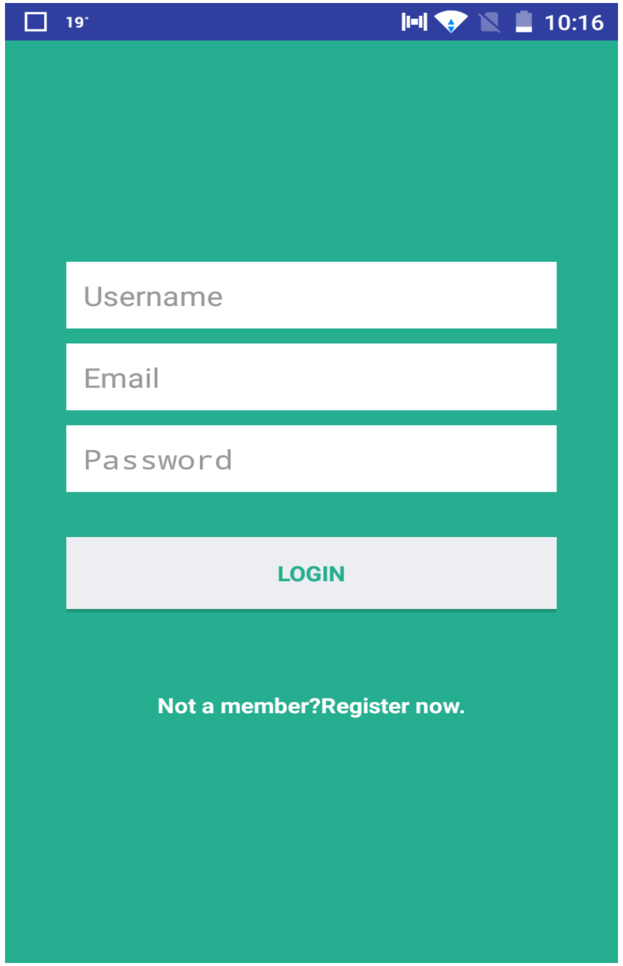
Client

E(K_m^+ ,[Identifier, Username, Password, Email, TimeStamp Session_CtoM]).

Merchant

E(K_c^+, [Verification Message, TimeStamp, Session_CtoM, Phone Number])

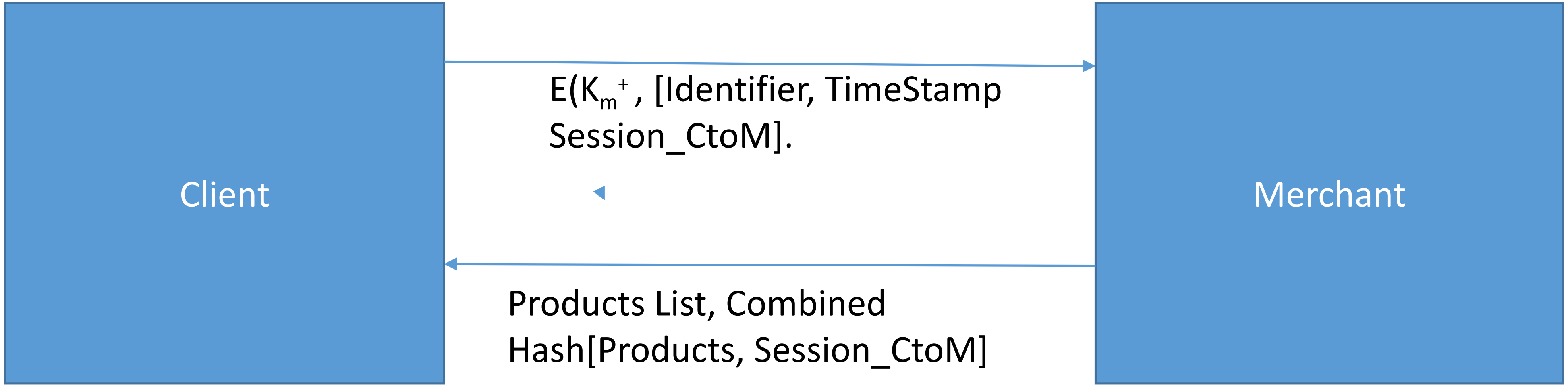
User Login Page



Implementation of Login Page:

- Registered User can login on the Login page using the following fields – Username, Email and Password.
- The login details are sent along with the Identifier(“Login”), TimeStamp and Session Key(Session_CtoM).
- All the fields sent are encrypted with merchant’s public key hence providing security.
- A table called “User” table maintains the list of the users in the database.
- When a user logs in and the user’s login credentials are correct, the login process succeeds and the details are updated in the “Sessions” table.
- The merchant sends the verification message(“Successful” or “Unsuccessful”), TimeStamp, Session_CtoM and phone number as reply. Everything is encrypted with the public key of the client.
- After successful login the user is directed to the Products page.

3) Product Selection Page



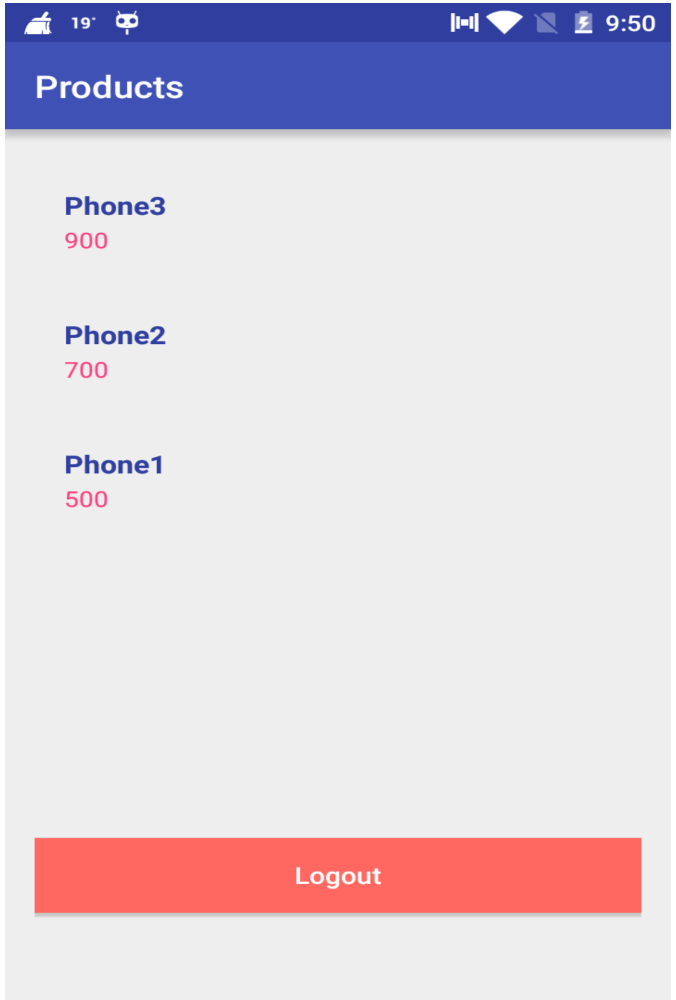
Client

$E(K_m^+$, [Identifier, TimeStamp Session_CtoM].

Merchant

Products List, Combined Hash[Products, Session_CtoM]

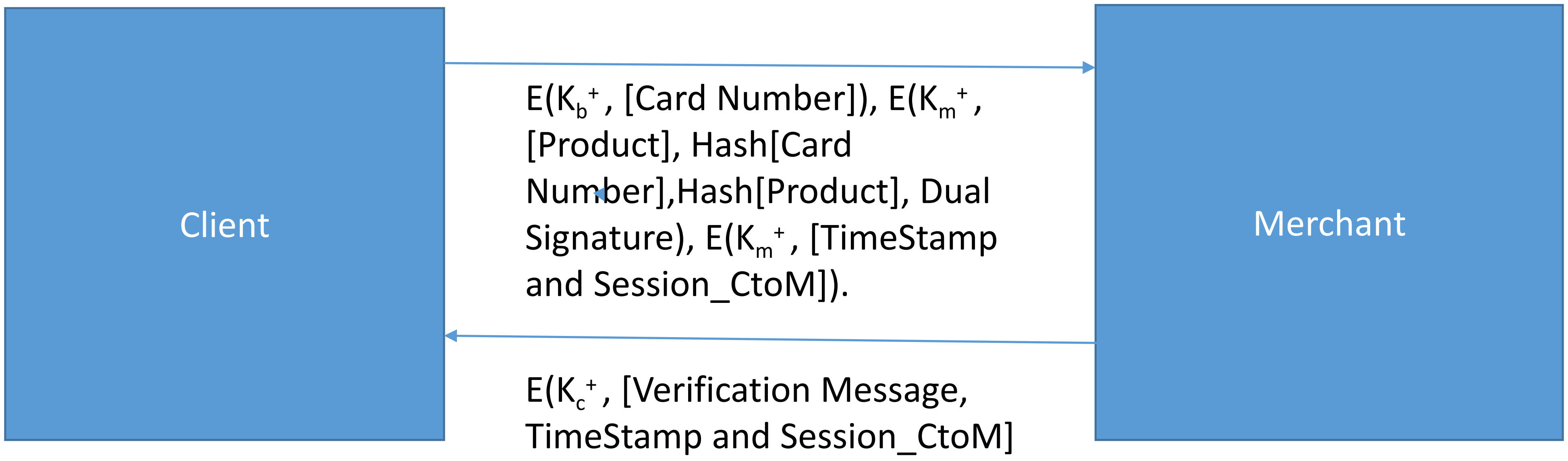
Select Product Page



Implementation of Select Products Page:

- The product request is sent along with the Identifier("Products"), TimeStamp and Session Key(Session_CtoM).
- All the fields sent are encrypted with merchant's public key hence providing security.
- A table called "Products" table maintains the list of the products in the database.
- When a user requests, the product list is sent along with combined hash of products and Session key.
- The client verifies whether the hash is correct. If it is correct the product list is displayed.
- The users selects the product available and moves to the payment page.

4) Client Merchant Exchange



Client

Merchant

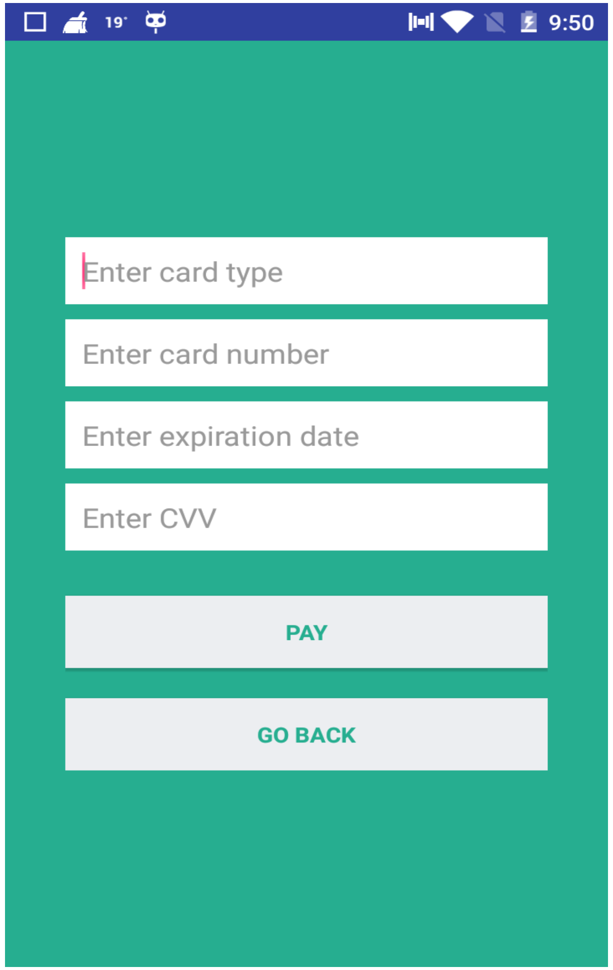
E(K_b^+, [Card Number]), E(K_m^+, [Product], Hash[Card Number],Hash[Product], Dual Signature), E(K_m^+, [TimeStamp and Session_CtoM]).

E(K_c^+, [Verification Message, TimeStamp and Session_CtoM]

Flow 2 : Client → Merchant Server

- Order Info is hashed.
- Payment Info is hashed.
- Hashed OI and Hashed PI is concatenated.
- The concatenated Hashed OI and Hashed PI is hashed again.
- Dual Signature for the hashed concatenated info.
- Order Info is encrypted with public key of merchant and payment Info is encrypted with bank’s public key. This is sent to merchant server.
- Message = (encrypted{merchant's public key} order info)+(encrypted{bank's public key} payment info)+(hash of order info)+(hash of payment info)+(concatenated hash)

Payment Information Page



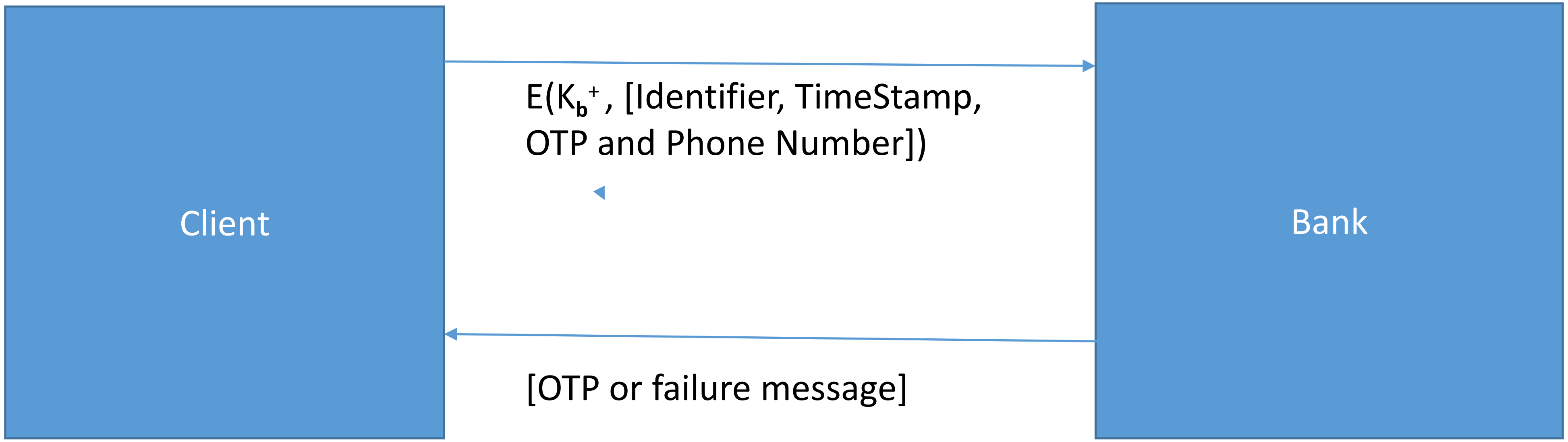
Implementation of Payment Information Page:

- After selecting the products, the user navigates to the Payment Information page where the user enters the following details – Card Type, Card Number, Expiration Date, CVV.
- The Identifier(“Message”),details, Session_CtoM and the TimeStamp are sent first to the merchant server.
- After the merchant verifies the above mentioned details it replaces the session_CtoM with Session_MtoB(newly generated) and the old timestamp.
- Session_MtoB is updated in the “Sessions” table.
- The details are forwarded to the bank server.
- The bank server verifies the timestamp and card details. If everything is correct, It sends the OTP to the client. It updates the same in the “OTP” table.
- Else it sends a failure message to the client.
- The client is directed to the OTP page.

Flow 3 : Merchant Server → Payment Gateway Server

- Server gets Order Info encrypted. (confidentiality)
- Server gets the hashed PI.
- The plaintext OI is hashed and concatenated with hashed PI.
- DS for the hashed concatenated info.
- This DS is matched with the DS obtained from client and verified.
- Server forwards the message to payment gateway.
- Message = (encrypted{merchant's public key} order info)+(encrypted{bank's public key} payment info)+(hash of order info)+(hash of payment info)+(concatenated hash)

5) Client - Payment Gateway Exchange



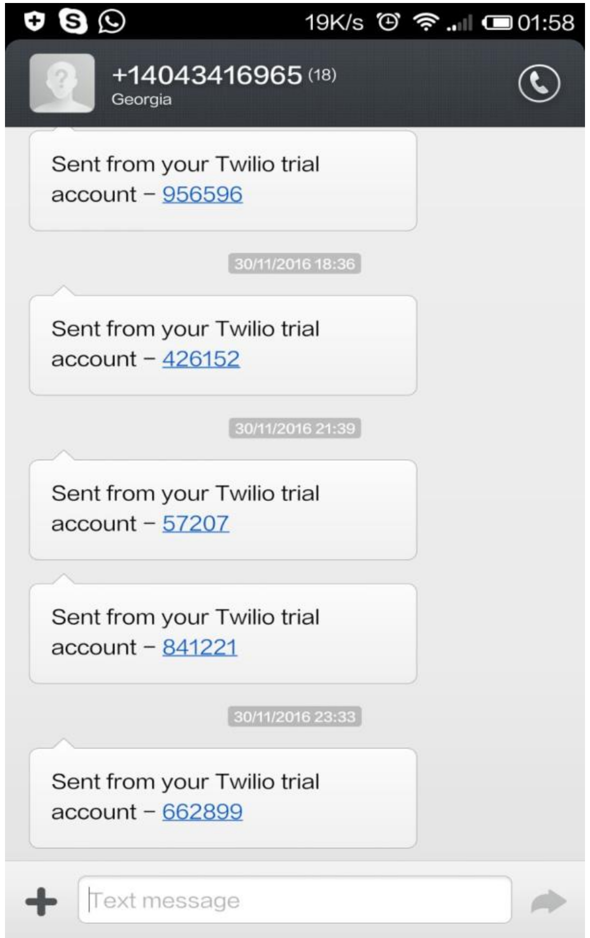
Client

Bank

E(K_b^+, [Identifier, TimeStamp, OTP and Phone Number])

[OTP or failure message]

OTP Page



+14043416965 (18)
Georgia

Sent from your Twilio trial account – 956596

30/11/2016 18:36

Sent from your Twilio trial account – 426152

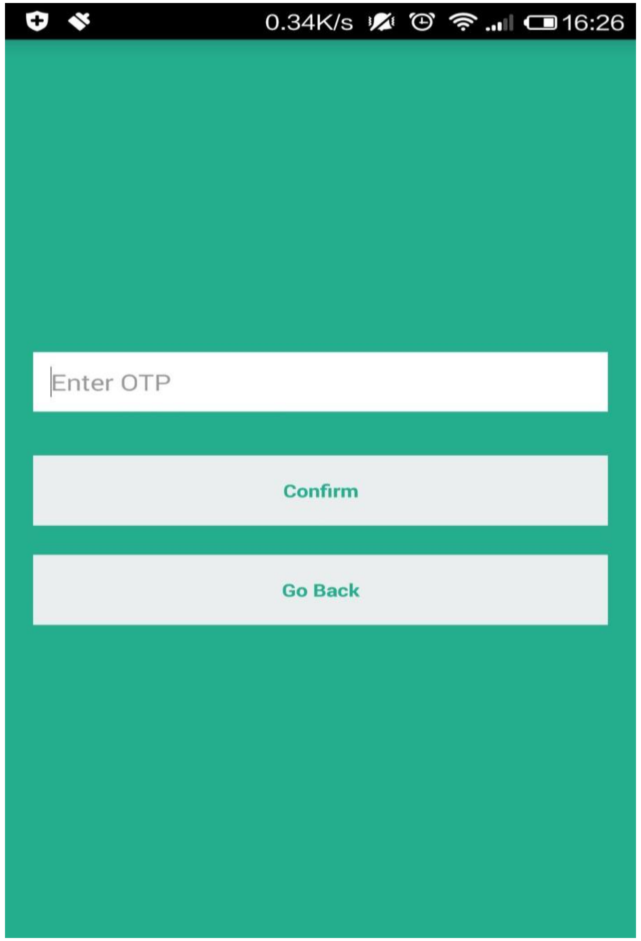
30/11/2016 21:39

Sent from your Twilio trial account – 57207

Sent from your Twilio trial account – 841221

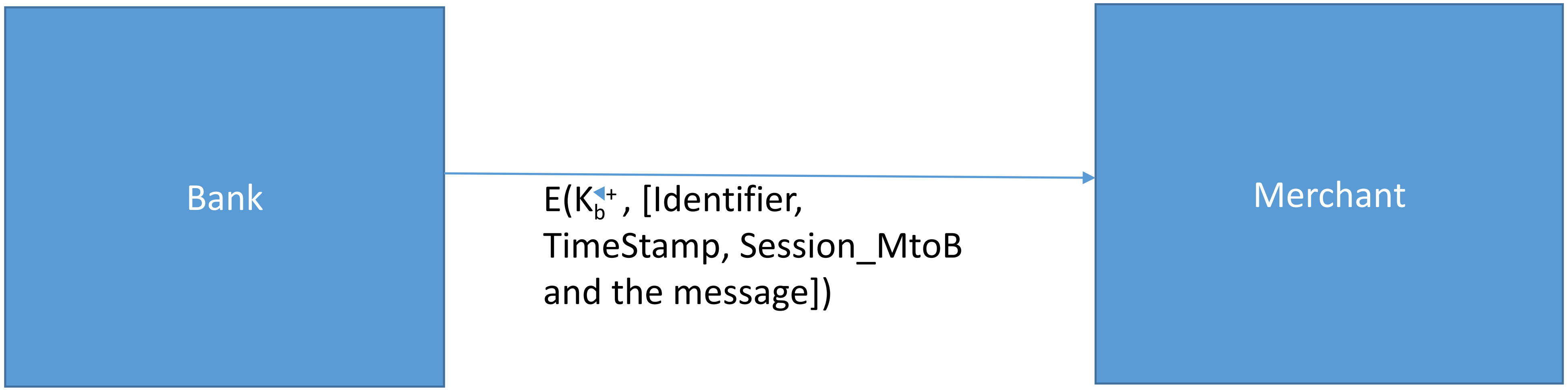
30/11/2016 23:33

Sent from your Twilio trial account – 662899



- After entering the payment details the user confirms it and navigates to the OTP page. On this page the user enters the OTP received on his/her phone number.
- The OTP is sent along with the Identifier(“OTP”), TimeStamp and Phone Number.
- The bank server receives the OTP and checks it in the database table “OTP”.
- Accordingly it sends a message to the merchant.

6) Bank Merchant Exchange



Bank

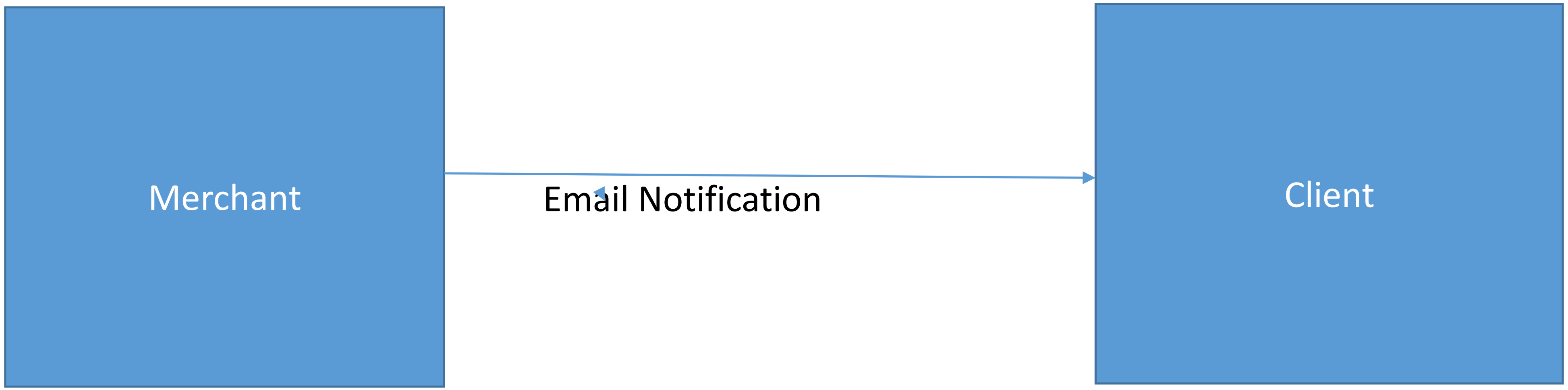
Merchant

E(K_b^+ , [Identifier, TimeStamp, Session_MtoB and the message])

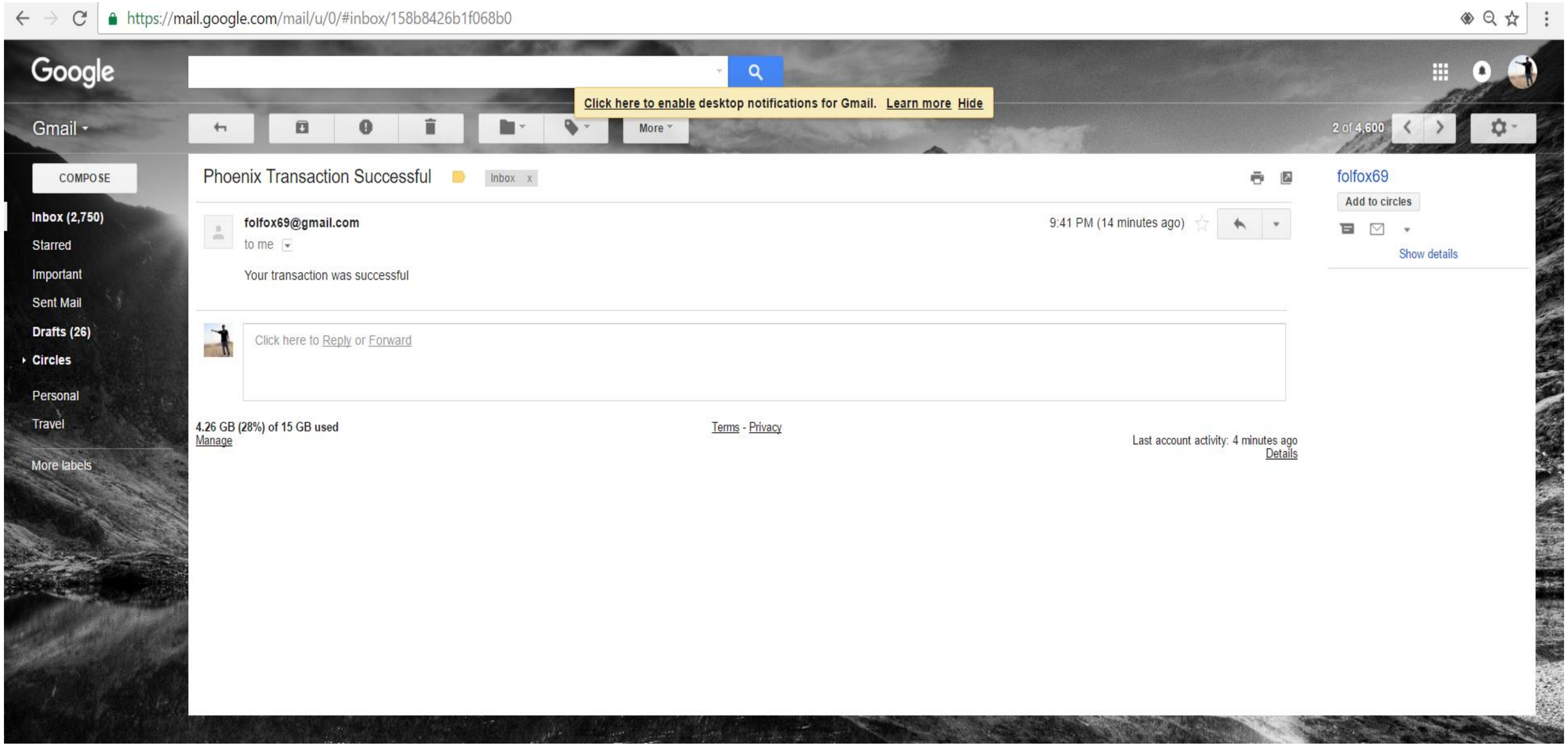
Flow 4 : Payment Gateway Server → Merchant Server

- PG verifies it and then sends payment verification message to merchant.

7) Merchant Client Exchange



9) Success Email Notification



Phoenix Transaction Successful

folfox69@gmail.com
to me

9:41 PM (14 minutes ago)

Your transaction was successful

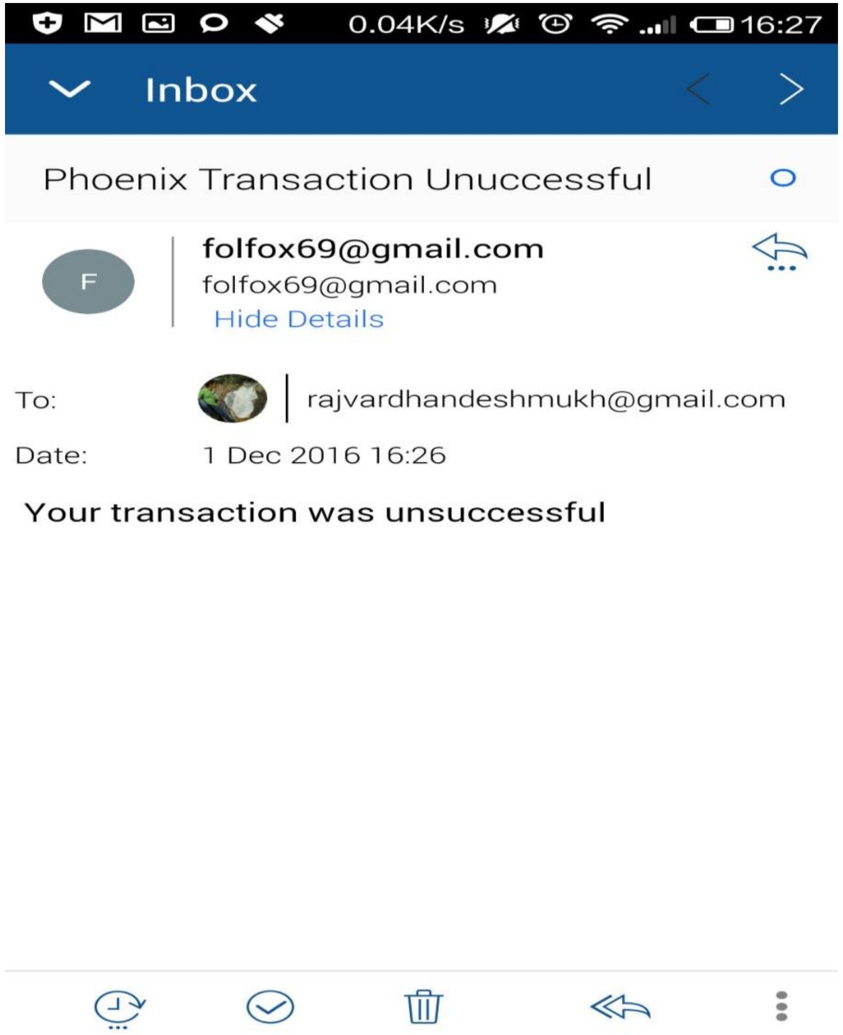
Click here to Reply or Forward

4.26 GB (28%) of 15 GB used
Manage

Last account activity: 4 minutes ago
Details

An email notification notifying that the transaction was successful is sent to the user’s registered email address.

Failure Email Notification



Phoenix Transaction Unuccessful

folfox69@gmail.com
folfox69@gmail.com
Hide Details

To: rajvardhandeshmukh@gmail.com

Date: 1 Dec 2016 16:26

Your transaction was unsuccessful

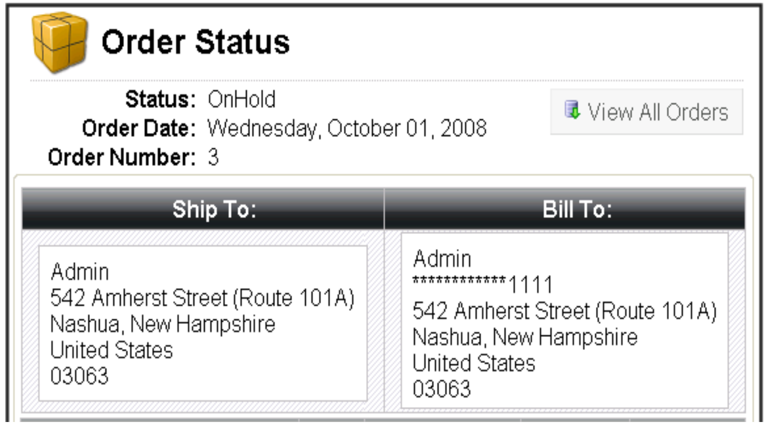
A failure email notification notifying that the transaction failed is sent to the user's registered email address.

Flow 5 : Merchant Server → Client

- Merchant Server notifies client that the process is completed.

High Level Implementation Details

User orders using Android App

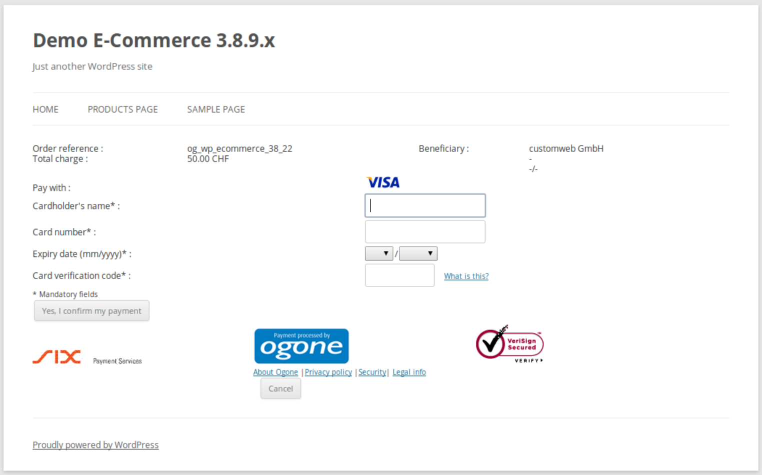


Order Status

Status: OnHold
Order Date: Wednesday, October 01, 2008
Order Number: 3

Ship To:	Bill To:
Admin 542 Amherst Street (Route 101A) Nashua, New Hampshire United States 03063	Admin ************1111 542 Amherst Street (Route 101A) Nashua, New Hampshire United States 03063

Order Information



Demo E-Commerce 3.8.9.x

Just another WordPress site

HOME
PRODUCTS PAGE
SAMPLE PAGE

Order reference : og_wp_ecommerce_38_22
Total charge : 50.00 CHF

Beneficiary : customweb GmbH
-
-/-

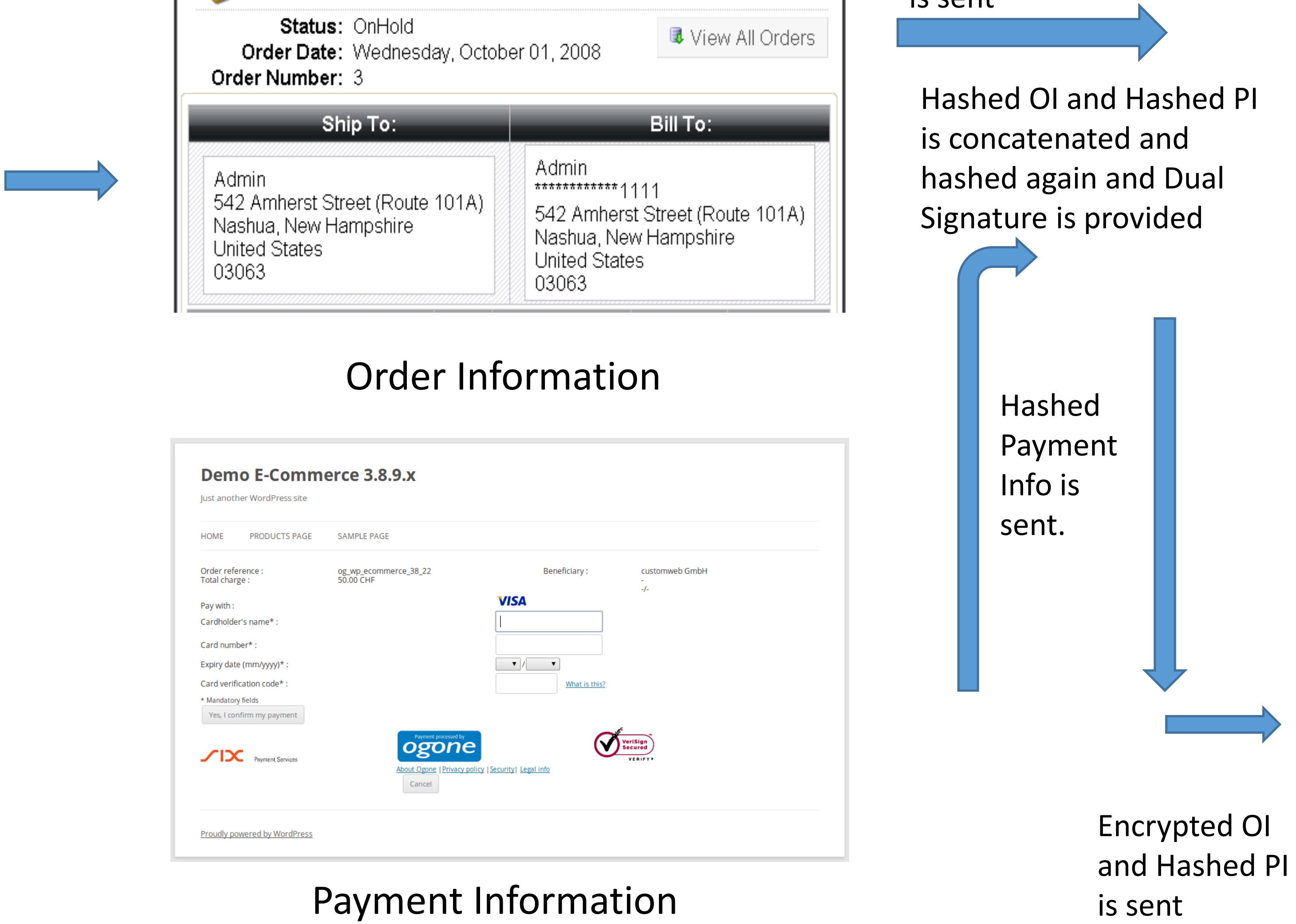
Pay with : VISA
Cardholder's name* :
Card number* :
Expiry date (mm/yyyy)* : /
Card verification code* : What is this?

* Mandatory fields

Yes, I confirm my payment
SIX Payment Services
Payment processed by ogone
About Ogone | Privacy policy | Security| Legal info
Cancel
VeriSign Secured VERIFY

Proudly powered by WordPress

Payment Information



Hashed Order Info is sent

Hashed OI and Hashed PI is concatenated and hashed again and Dual Signature is provided

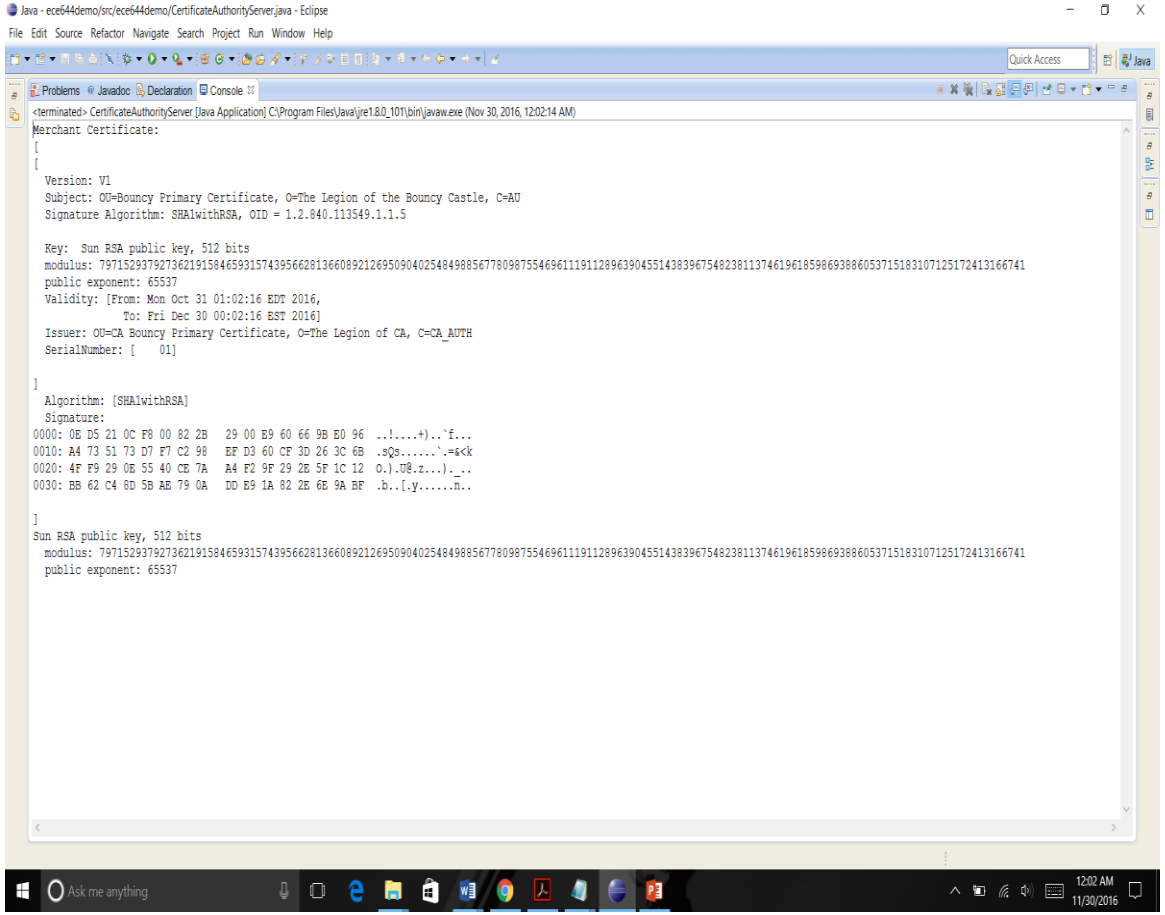
Hashed Payment Info is sent.

Encrypted OI and Hashed PI is sent

Merchant Server

Payment Gateway

Java Console Screenshots

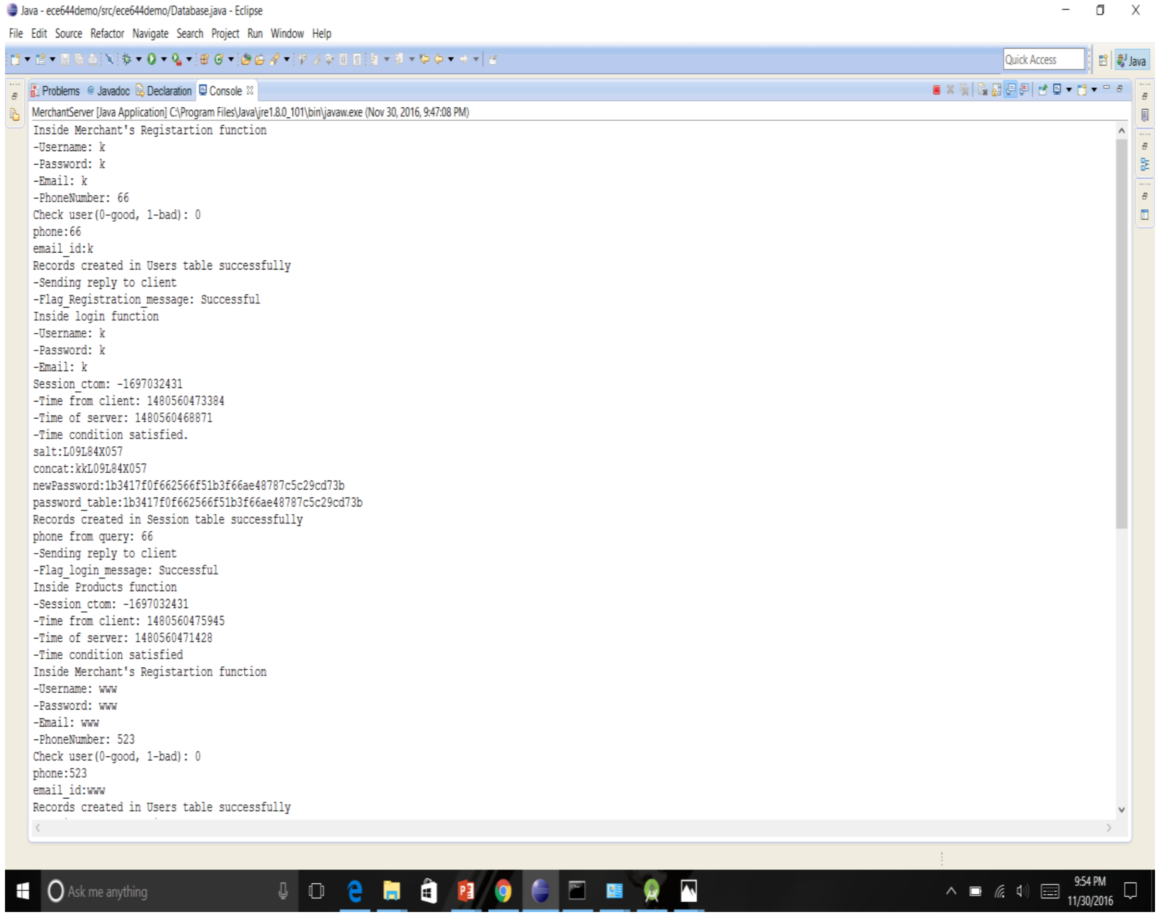

Java - ece644demo/src/ece644demo/CertificateAuthorityServer.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Problems Javadoc Declaration Console
Ask me anything
12:02 AM
11/30/2016

```
<terminated> CertificateAuthorityServer [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Nov 30, 2016, 12:02:14 AM)
Merchant Certificate:
[
[
  Version: V1
  Subject: OU=Bouncy Primary Certificate, O=The Legion of the Bouncy Castle, C=AU
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

  Key:  Sun RSA public key, 512 bits
  modulus: 7971529379273621915846593157439566281366089212695090402548498856778098755469611191128963904551438396754823811374619618598693886053715183107125172413166741
  public exponent: 65537
  Validity: [From: Mon Oct 31 01:02:16 EDT 2016,
               To: Fri Dec 30 00:02:16 EST 2016]
  Issuer: OU=CA Bouncy Primary Certificate, O=The Legion of CA, C=CA_AUTH
  SerialNumber: [    01]

]
  Algorithm: [SHA1withRSA]
  Signature:
0000: 0E D5 21 0C F8 00 82 2B   29 00 E9 60 66 9B E0 96  ..!....+)..`f...
0010: A4 73 51 73 D7 F7 C2 98   EF D3 60 CF 3D 26 3C 6B  .sQs......`.=&<k
0020: 4F F9 29 0E 55 40 CE 7A   A4 F2 9F 29 2E 5F 1C 12  O.).U@.z...)._..
0030: BB 62 C4 8D 5B AE 79 0A   DD E9 1A 82 2E 6E 9A BF  .b..[.y......n..

]
Sun RSA public key, 512 bits
  modulus: 7971529379273621915846593157439566281366089212695090402548498856778098755469611191128963904551438396754823811374619618598693886053715183107125172413166741
  public exponent: 65537
```


Java - ece644demo/src/ece644demo/Database.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Problems Javadoc Declaration Console
Ask me anything
9:54 PM
11/30/2016

```
MerchantServer [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Nov 30, 2016, 9:47:08 PM)
Inside Merchant's Registartion function
-Username: k
-Password: k
-Email: k
-PhoneNumber: 66
Check user(0-good, 1-bad): 0
phone:66
email_id:k
Records created in Users table successfully
-Sending reply to client
-Flag_Registration_message: Successful
Inside login function
-Username: k
-Password: k
-Email: k
Session_ctom: -1697032431
-Time from client: 1480560473384
-Time of server: 1480560468871
-Time condition satisfied.
salt:L09L84X057
concat:kkL09L84X057
newPassword:1b3417f0f662566f51b3f66ae48787c5c29cd73b
password_table:1b3417f0f662566f51b3f66ae48787c5c29cd73b
Records created in Session table successfully
phone from query: 66
-Sending reply to client
-Flag_login_message: Successful
Inside Products function
-Session_ctom: -1697032431
-Time from client: 1480560475945
-Time of server: 1480560471428
-Time condition satisfied
Inside Merchant's Registartion function
-Username: www
-Password: www
-Email: www
-PhoneNumber: 523
Check user(0-good, 1-bad): 0
phone:523
email_id:www
Records created in Users table successfully
```

Certification Authority

Merchant Server

Database

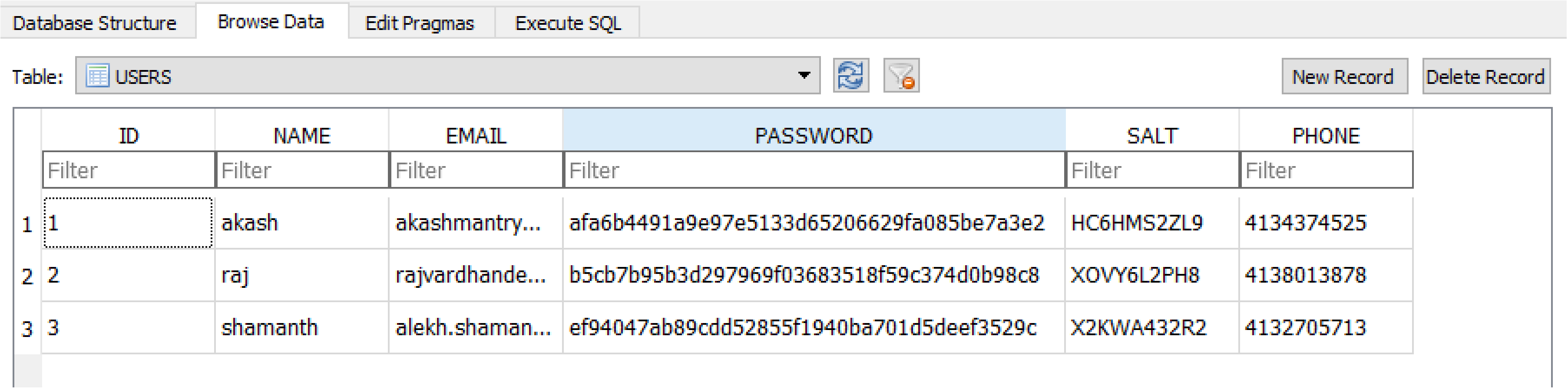


Table: USERS

	ID	NAME	EMAIL	PASSWORD	SALT	PHONE
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	akash	akashmantry...	afa6b4491a9e97e5133d65206629fa085be7a3e2	HC6HMS2ZL9	4134374525
2	2	raj	rajvardhande...	b5cb7b95b3d297969f03683518f59c374d0b98c8	XOVY6L2PH8	4138013878
3	3	shamanth	alekh.shaman...	ef94047ab89cdd52855f1940ba701d5deef3529c	X2KWA432R2	4132705713

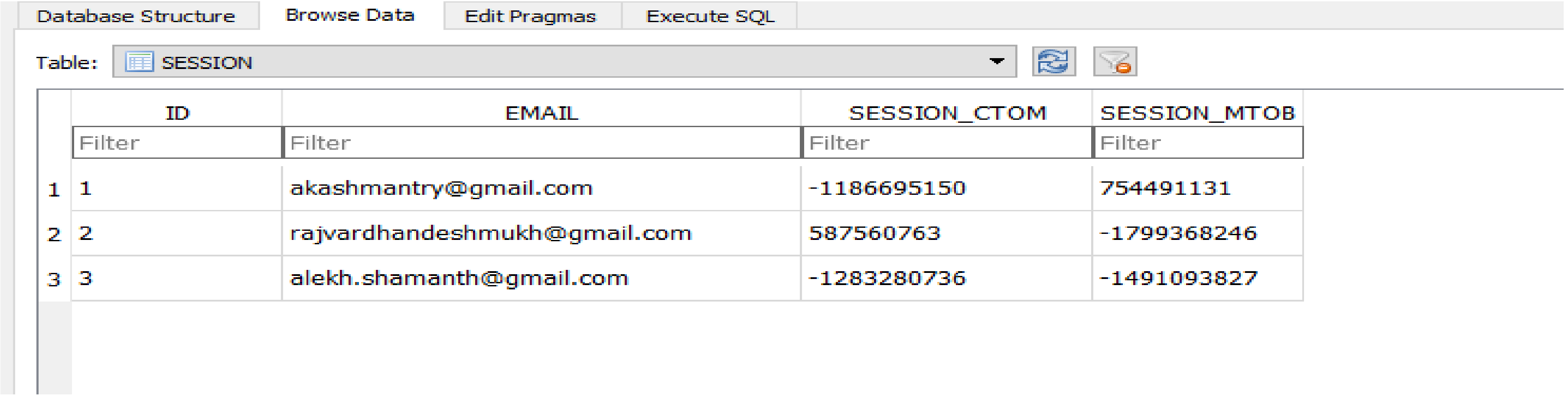


Table: SESSION

	ID	EMAIL	SESSION_CTOM	SESSION_MTOB
	Filter	Filter	Filter	Filter
1	1	akashmantry@gmail.com	-1186695150	754491131
2	2	rajvardhandeshmukh@gmail.com	587560763	-1799368246
3	3	alekh.shamanth@gmail.com	-1283280736	-1491093827

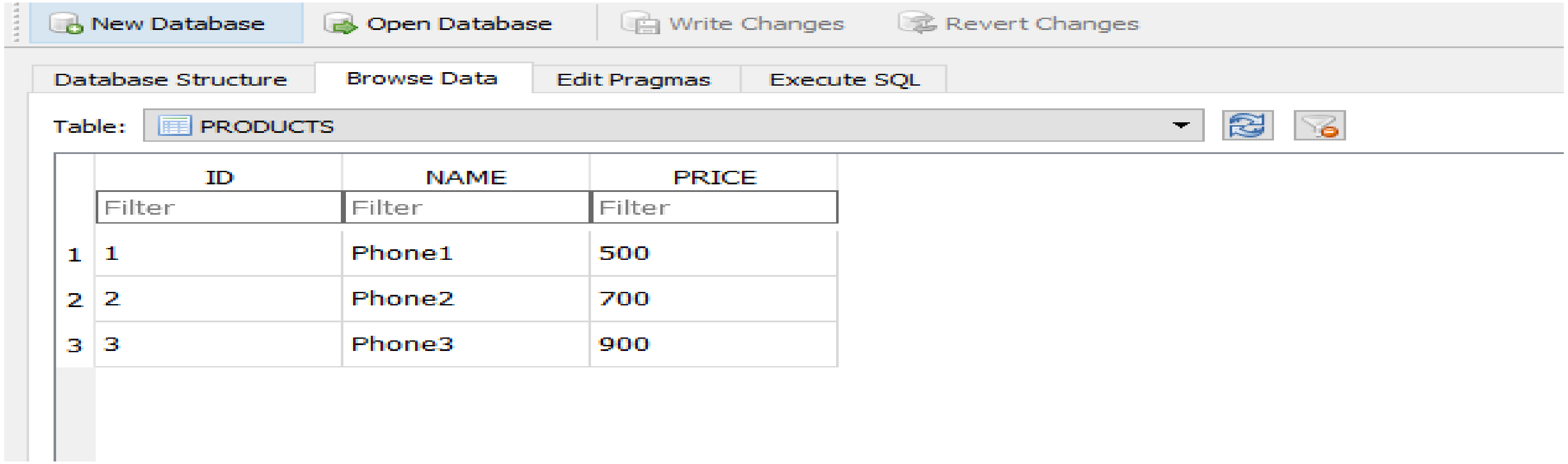


Table: PRODUCTS

	ID	NAME	PRICE
	Filter	Filter	Filter
1	1	Phone1	500
2	2	Phone2	700
3	3	Phone3	900

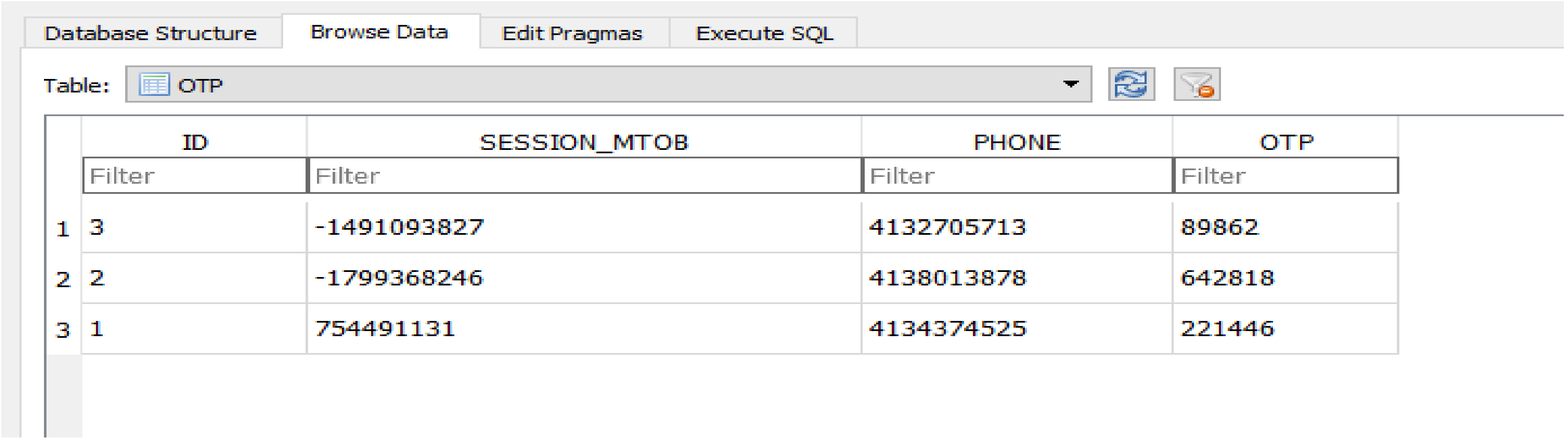
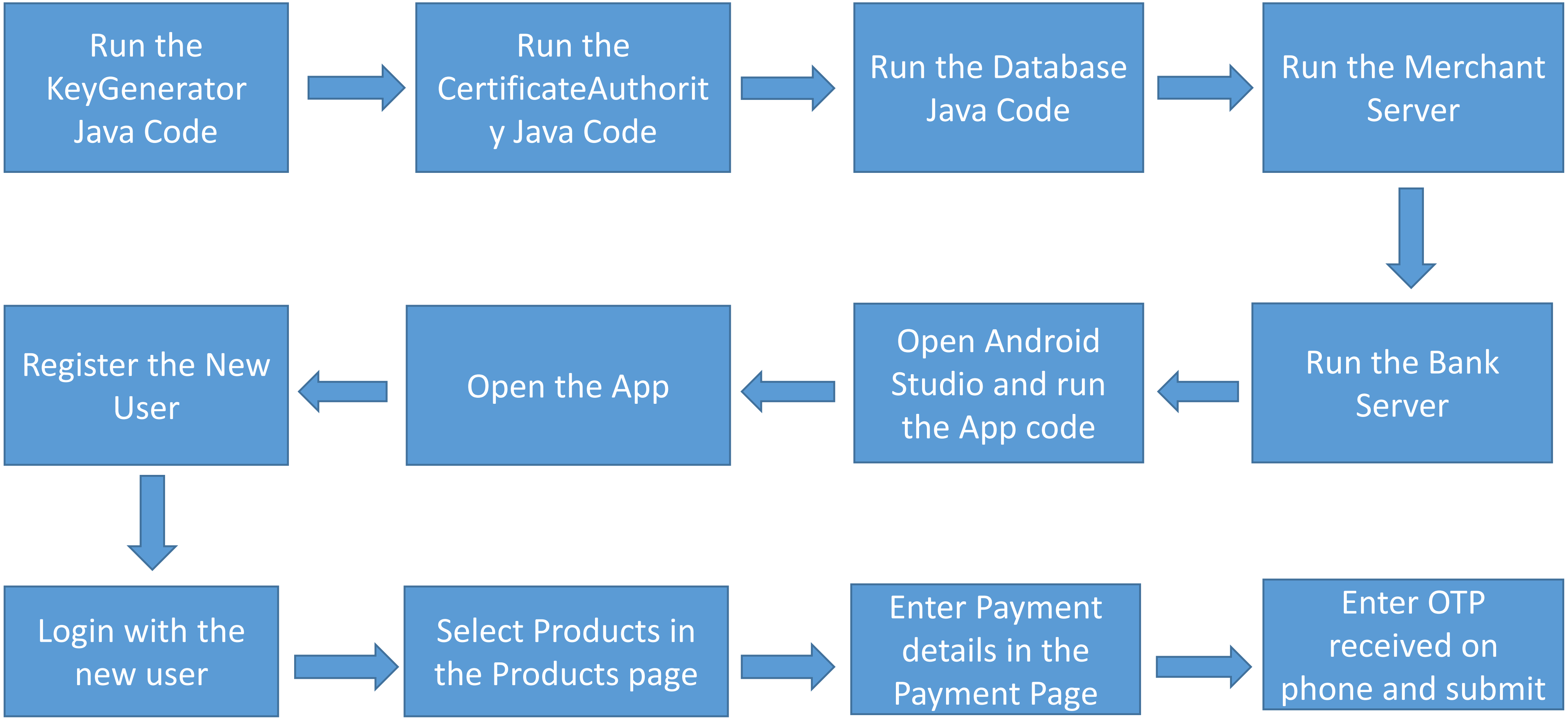


Table: OTP

	ID	SESSION_MTOB	PHONE	OTP
	Filter	Filter	Filter	Filter
1	3	-1491093827	4132705713	89862
2	2	-1799368246	4138013878	642818
3	1	754491131	4134374525	221446

Steps followed to execute the implementation



Run the KeyGenerator Java Code

Run the CertificateAuthority Java Code

Run the Database Java Code

Run the Merchant Server

Run the Bank Server

Open Android Studio and run the App code

Open the App

Register the New User

Login with the new user

Select Products in the Products page

Enter Payment details in the Payment Page

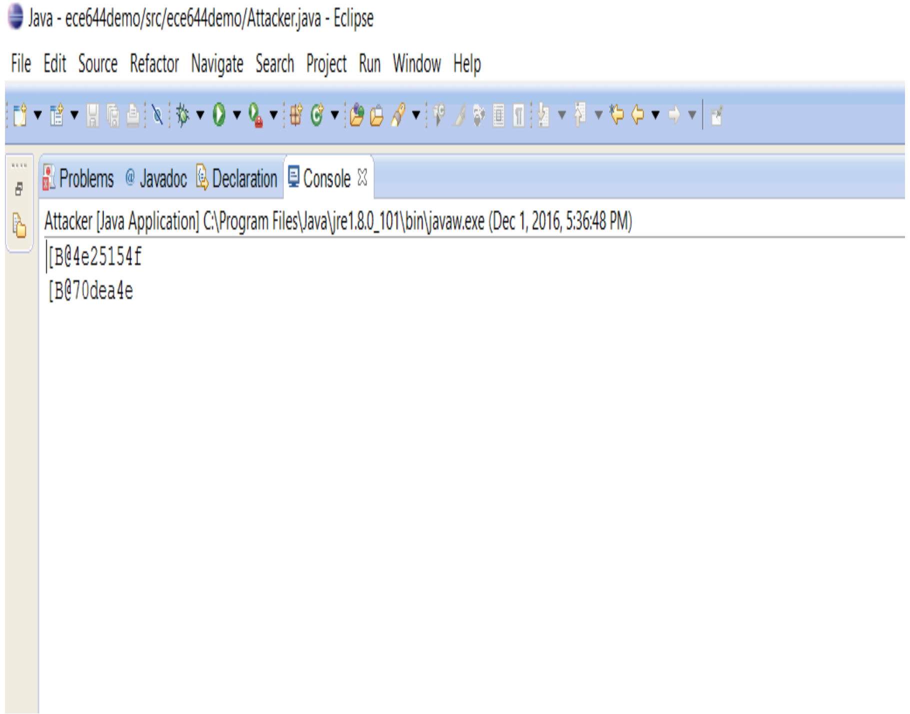
Enter OTP received on phone and submit

Testing of the implementation(Attacks)

- Replay Attack : The attacker captures the packet and replays it to get illegal services. The use of timestamp prevents this attack.
- Masquerade Attack : Attacker masquerades as merchant when the merchant server is down. Since the data is encrypted the attacker cannot obtain meaningful data.
- Man in the middle attack: When the merchant sends product list to the client the attacker captures this and modifies the product list and sends it to the client. This attack is prevented by using hash of the session key and the product list.

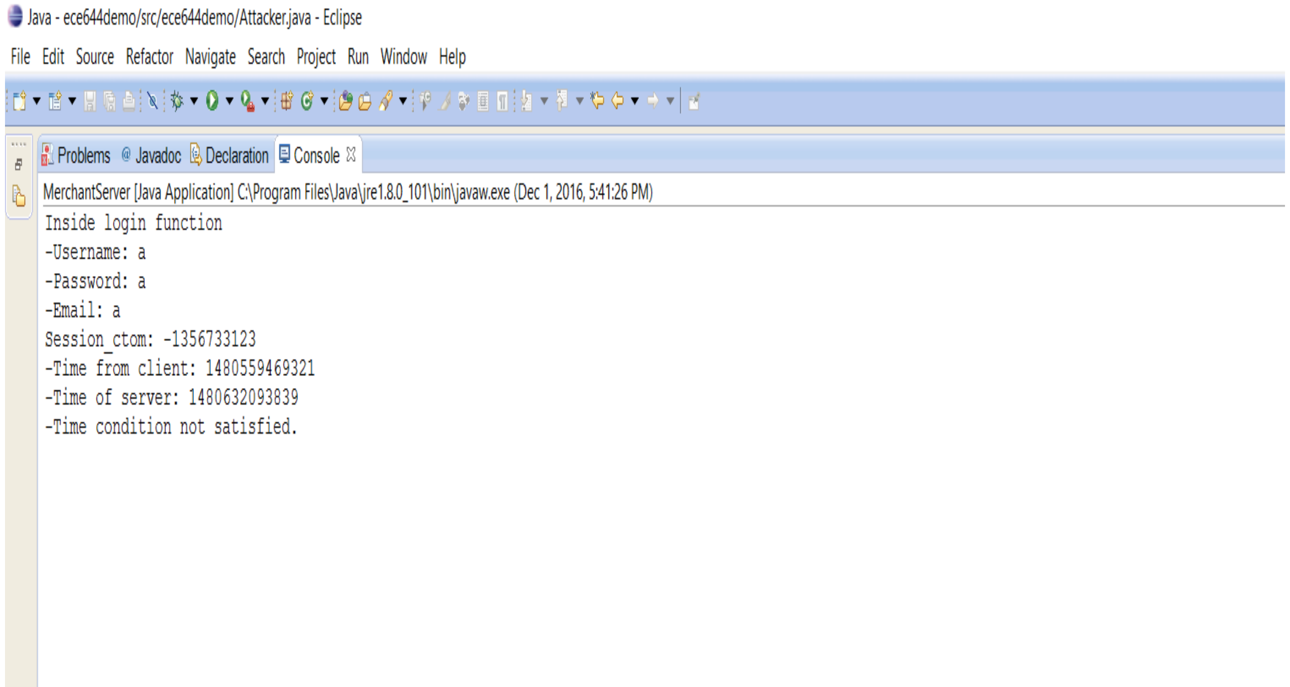
Attack Screenshots

Masquerade Attack


Java - ece644demo/src/ece644demo/Attacker.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console

```
Attacker [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Dec 1, 2016, 5:36:48 PM)
[B@4e25154f
[B@70dea4e
```

Replay Attack


Java - ece644demo/src/ece644demo/Attacker.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console

```
MerchantServer [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Dec 1, 2016, 5:41:26 PM)
Inside login function
-Username: a
-Password: a
-Email: a
Session_ctom: -1356733123
-Time from client: 1480559469321
-Time of server: 1480632093839
-Time condition not satisfied.
```

References

- YANG Rui-xia, “Design of Secure Mobile Payment System Based on IBC”, 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications.
- Yong Wang, Christen Hahn and Kruttika Sutrave, “Mobile Payment Security, Threats, and Challenges”,
- Android Integrating PayPal using PHP, MySQL http://www.androidhive.info/2015/02/android-integrating-paypal-using-php-mysql-part-1/
- S. Lu and S. A. Smolka, "Model checking the secure electronic transaction (SET) protocol," *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 1999. Proceedings. 7th International Symposium on*, College Park, MD, 1999, pp. 358-364
- M. C. Ruiz, D. Cazorla, F. Cuartero and J. J. Pardo, "A formal specification and performance evaluation of the purchase phase in the SET protocol," *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*