2020201098

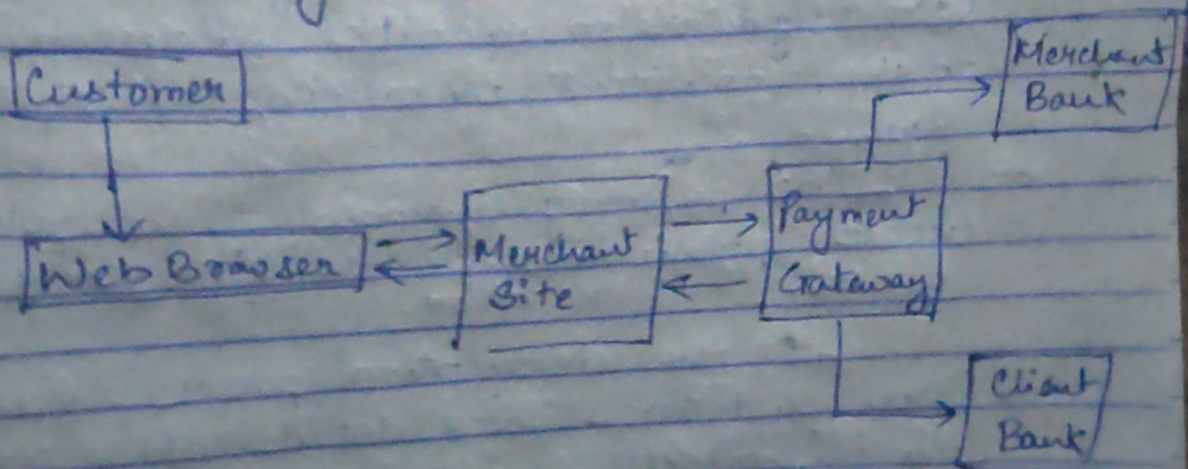
SNS Open Book

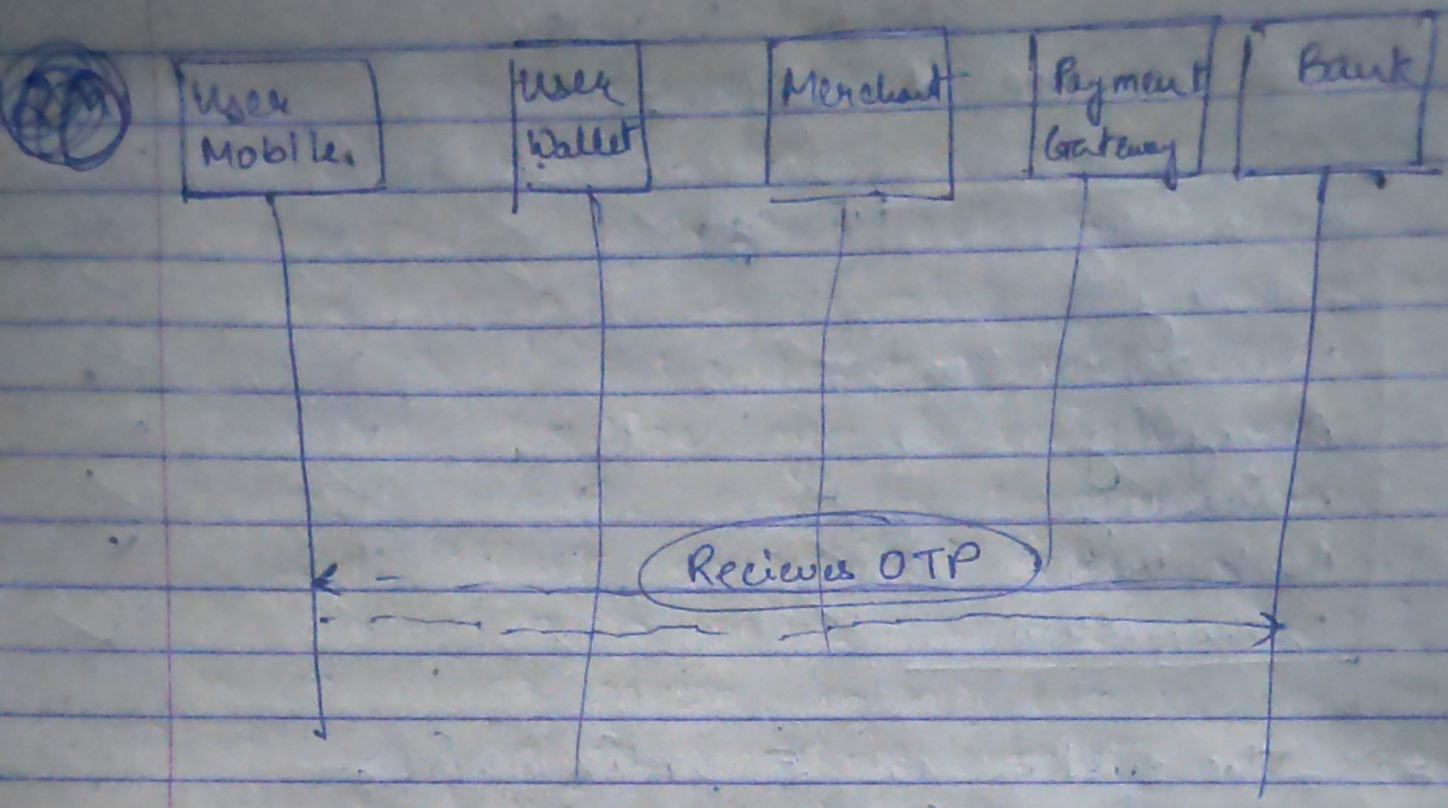
(Q2) In the proposed system :-

	Our system	Azhar	Nwoye	Kyaw
Confidentiality	✓	✓	✓	✓
Integrity	✓	✓	✓	✓
Non-repudiation	✓	✗	✗	✓
Anonymity	✓	✗	✓	✗
Availability	✓	✓	✓	✓
Authentication	✓	✗	✗	✗
Authorization	✓	✓	✓	✓

⇒ The system has 5 preliminaries :-

→ Customer / Client
→ Merchant
→ Client Bank
→ Merchant Bank
→ Payment Gateway





Whenever the client visits the site, this protocol makes a temporary identity of the client to provide an extra layer of security.

The client Bank along with the client use RSA signature to execute an electronic signature on the document by making use of the private key.

The primary purpose of the system is to create public and private keys for traders and Banks.

In the decryption process. RSA collects the customer's card details after reciept of the ciphertext from the customer and decrypts the cyphertext. The payment gateway validates the authorization for the payment phase after the customer's card details have been decrypted.

The Transaction is :-

Step 1 :

Start :

Client → Merchant :- The client sends a request to the merchant using his/her temporary identity (TIDc)

Merchant → Client :- The merchant sends back to the client the identity of the product and goods details including price, date, transaction identific^{n} (IDG, G)

:

:

Step 5 :-

Client Bank - Client Phone :- The Client Bank sends the verific^{n} OTP to the client's mobile phone; then the client responds to the OTP verific^{n} code. Once verific^{n} is complete the client sends an ack to the merchant Bank.

Client-Bank → Client : OTP req

Client → Client Bank : OTP res

Merchant → Payment Gateway :- Ack

↳ Ensures prevention from ^ impersonation User attack.

→ Only approved users must be eligible on the basis of electronic transfers, and only authorized users must therefore be able to access the details exchanged for payments. On the other hand strong client authentication should shield the initiation of online payments besides access. In the proposed method, before the payment procedure. The client Bank first asks for client authentication to ensure that the prospect is an authorized person who will recieve the verifi" (OTP) code to transfer a certain amount from his/her account to the merchant bank. Therefore through these steps User Impersonation Attacks are prevented.

→ For merchant impernation :-
The merchant also registers with the gateway. Whenever client sends a request a temp id is generated and if anything goes wrong during request processing or any malicious data are found, the protocol discards request and termin the entire transaction.

The anonymity of client is also helps in preventing counterfeting and blackmail.
The anonymity of the client is done first by providing temporary id to the client

Q1

LS has following info at Registration phase But, it was not mentioned that LS stores ID_{Ui} and PWD_{Ui} while a priviledged user can store the information while the users are registering.

→ Thus the priviledged user will now have ID_{Ui} and PWD_{Ui}

The hash function (h) accepts fixed length string and collision resistant.

The hash function is common among the system.

If the priviledged user has the user details, then they can use Brute force approach to find w, R, α of user.

→ Randomly pick R and α, lets say R_{rand} and α_{rand}

Now :- $X = h(R_{rand} \| ID_{Ui} \| \alpha_{rand})$

while ($X != PWD_{Ui}$)

→ Keep calculating new X for new R and α

Once, they have R and α then they can generate 'w' using the Rep() func^n.

Randomly generate (w_{rand}) and then put it in the GRep fun^{n}.

$$X = G_{Rep}(w_{rand}, \alpha)$$

while $(X != R)$

→ Repeat.

Thus, priviledged user will have the details w, R, α of any user.

Also,

(h) is collision-resistant

∴ one-one mapping

Thus,

The secret credential of users are calculated by using fuzzy extractor on w.

(b) During content key acquisition phase.

(Impersonator)

$r_1 \longrightarrow T_U$ timestamp.

Knows ID of content (ID_{DC})

$DIDU \rightarrow$ Random text only known to impersonator.

$e \rightarrow$ Public key of Server

Impersonator computes the following :-

$$M_1 = (T_U \,||\, DIDU \,||\, IDDC)^e$$

selects $K_U \rightarrow$ randomly (Known only to the impersonator

$$W_1 = h(K_U \,||\, DIDU \,||\, T_U)$$

$\langle M_1, W_1 \rangle$ —(Send to LS)→ The LS then extracts the contents T_U, DIDU, using (d) private key.

It verifies $\boxed{|T_U - T_U'| < \Delta T}$

T_U' - Server Timestamp.

The timedelay should not exceed a threshold to prevent Replay attack.

Thus, now, K_U is calculated as:-

$$K_U = h(DID_U \| d)$$

$$\boxed{W_1 = h(K_U \| DID_U \| T_U)}$$

∴ now the timestamp is not the issue the impersonator can repeatedly generate K_U and send to LS until it matches.

Thus, User impersonation can indeed happen in this system.