Proposed Architecture
For Secure Mobile Payment System
Mobile payment is a potential application for future mobile networks. The technology war in the mobile payment domain is boosted by tough competition between mobile network operators, financial institutes, and payment service providers. The major driving forces for the successful mobile payment are user acceptance ratio that refers to an increased convenience and efficiency in performing payment transactions compared to online payments and Security.

The existing mobile payment systems and solutions support a wide variety of services including payments related to its business. These systems meet operational requirements such as scalability, extensibility, and fault tolerance. Additionally, the quality requirements and constraints of the system are also captured as input. However, those operational and quality requirements will differ from application to application, technology to technology and domain to domain. While adopting the payment services to other areas like academic side, it fails to meet the actual requirements of the users or provides overheads. The literature survey also reveals that a system for the payment of fees through mobile phone in higher academic institutions has never been thought off either in the national level or in the international level. Hence an urgent need arises to design a technology for mobile payment system with the support of End-to-End security in financial transactions between the client and the banks. In view of this lacuna, this thesis proposes to
design the architecture for Secure Mobile Payment System using **Public Key Infrastructure** (PKI).

This chapter presents the proposed architecture for secure mobile payment system and its security framework.

### 3.1 Proposed Architecture for Secure Mobile Payment System

The architecture defines the structure of components, their interrelationships, the principles and guidelines governing their design and evolution over time. Figure 3.1 depicts the proposed architecture for Secure Mobile Payment System that consists of **Mobile Client (MC)**, **Institution Server (IS)**, **Mobile Payment Consortia System (MPCS) Server**, **Banks Server** and the **Payment Interface**. **General Packet Radio Service (GPRS)** is used as the standard communication protocol between the mobile client and the institution server. The proposed Mobile Payment System provides the necessary technical infrastructure such as acquiring user information, connectivity, authentication, and communication to facilitate m-payments and payment gateway which acts as an intermediary between the banks, the institution and the clients. The MPCS provides support for strong authentication and non-repudiation by employing digital signatures. Confidentiality and message integrity are also provided using digital signature.

The proposed architecture for mobile payment system, functionalities, PKI based security algorithms and the performance of the model are proposed by the author [Britto et al. 2009]. It is a unique one, which has been accepted by International Research Forums like Association for Computing Machinery (ACM), Springer,
IEEE and International Journal of Information Processing and Management (IJIPM) [Britto et al. 2008, Britto et al. 2010a, Britto et al. 2010b].

The various functionalities of the architecture are well presented in the subsequent sections.
3.1.1 The Mobile Client (MC)

The Mobile Client (MC) is a student or parent who must have an account in a bank and the client must register to the institution, where he/she studies. The client has made a request to the IS for registration using SMS service. The institution server also allows the clients for registration over the Internet or directly. Once the registration process is completed, the institution server sends the AcCode, username and password to the student’s personal mail id together with the payment software. The client can download the payment application software using GPRS or through data cable. During the payment process, the client device sends the username and password for authentication.

3.1.2 General Packet Radio Service (GPRS) Network

The General Packet Radio Services (GPRS) of the GSM is designed to provide packet data services over the radio interface. GPRS supports common packet data protocols like IP and X.25. Being a packet data service, GPRS optimizes the use of network and radio resources. GPRS offers the possibility to charge by amount of data transferred rather than the time the mobile phone is attached to the network. It provides a greater benefit to the mobile users economically. In order to provide a cost effective payment system, the proposed architecture makes use of GPRS technology. The mobile clients need to open the GPRS service by applying to the telecom companies. The main advantages of the GPRS are that it can exchange large quantum of data with high speed and the connection is stable. GPRS supports several qualities of data services such as reliability and response time.
3.1.3 Institution Server (IS)

The Institution Server (IS) keeps a record of all registered clients. The authentication of the client is done by the institution server by verifying the username and password. After authentication, the IS establishes connection with student’s database for retrieving their details such as student name, account number, bank name and sends the student’s profile to the MPCS server. The information flow on the network is secure by encrypting and decrypting the message using public key infrastructure. The mobile payment server setup at the institution consists of mobile server, web server and database server. The mobile server provides alternatives to the users for payment service registration through SMS. The web server accepts all the payment requests that come from the mobile clients and handles device and user authentication. The database server keeps the student and payment related information.

3.1.4 Mobile Payment Consortia System (MPCS) Server

The Mobile Payment Consortia System (MPCS) server acts as an intermediary between the institution and the banks. MPCS is responsible for authorizing the clients for payment service, generating the payment request, creating the transaction logs, routing and delivering the notifications to the remaining entities. MPCS plays a role as a Payment Gateway (PG) for the institution. MPCS also authorizes the payment services with the client by verifying option-id and the fee amount. MPCS database server keeps information about authenticated mobile users, payment requests and service logs. This information is used to complete the verification process at a later time.
3.1.5 Banks Server

To be a successful payment system, the cooperation of banks is very essential. The banks generally have their own infrastructure to support authentication and payment service. However, the MPCS server must co-operate with those existing authentication facilities. In the proposed model, the institution payment gateway i.e. MPCS server is tightly coupled with institution authorized bank to handle the payment services for students. When the payment requests come from the MPCS server, the institution authorized bank processes them in a regular way by establishing communication with student bank. Once the fee amount is credited to the institution account, the institution authorized bank sends the payment confirmation message to the MPCS server.

3.1.6 Payment Interface

MPCS makes a request for payment to the institution authorized bank using the secure payment interface. The bank identifies the clients with their Bank-ID and makes confirmation for the payment using their payment application. Finally, the requested amount is debited from the clients’ account and credited to the institution authorized bank account. Once the transaction is completed electronically, a confirmation message is sent by the institution authorized bank to the institutional server.

3.2 Major Phases of Proposed Payment System

There are two major phases involved in the proposed payment system to carry out the payment transactions such as registration phase and payment phase.
3.2.1 Registration Phase

In this phase, the client (student) may authenticate by registering oneself in the institution server where one studies. At the time of registration, the client provides his personal data such as roll number, name, course, mobile number, mail ID, bank name, branch name, account holder name, account number and IMEI number. The MPCS allows the students to register either through website or directly. After completing the registration, the client receives AcCode and UName/PWD through personal mail in a secure way. The client then avails the payment services by downloading the payment software in any one of the following ways:

a) Using GPRS

MPCS allows the clients to download the payment software from their mobile phones using GPRS. The client sends an SMS with <RollNo><Course Name> to the payment service number 56565. If the client is a valid student, the institution server sends an encrypted registration interface as a Uniform Resource Locator (URL) link. The client can activate the registration interface by entering the <AcCode> and can download the payment software via GPRS.

b) Through Internet

The mobile clients can also download the payment software on their PC or laptop from the institution website over the Internet. The client then stores the payment application to his mobile phone using data cable. If the mobile handset and the laptop have Bluetooth facility, the client can transfer the payment module via Bluetooth.
c) In Person

The MFCS also allows the client to download the payment software with the help of institution employees. After providing the required information to them, the client can download the payment application using data cable from the institution server directly.

3.2.2 Payment Phase

The payment phase includes various sub-phases such as device authentication, client and client/server authentication, authorization, payment request generation and payment confirmation. Once the registration process is completed, the client enters AcCode for activating the payment software at mobile device. If the AcCode is valid, then the authentication interface enables user’s device, otherwise, the payment software permits the user to enter the AcCode with three attempts. If the user fails in all attempts, URL for the payment software will be removed from the client’s device. While entering the AcCode, the client device’s International Mobile Equipment Identity (IMEI) number is also verified for device identity. When the user activates the payment software successfully, the client device sends the X.509 user certificate to the server for client device authentication.

After the client device is identified by the institution server, it sends server certificate to the client device. Now, the client verifies the server certificate with trusted Certificate Authority using Online Certificate Status Protocol (OCSP) mechanism. In the proposed model, X.509 user certificate is applied to authenticate the communicating devices. Once the device authentication process is completed, the client device generates secret key using AcCode and IMEI number. At this time,
the user interface decrypts the server's public key from the server certificate. If the interface decrypts the IS's public key successfully, the student is initially authenticated to the IS.

After completing the initial authentication, the payment user enters user name (UName) and password (PWD) and sends it to the institution server as an encrypted format using PKI. If the institution server decrypts the UName and PWD successfully, the server is authenticated to the client and also the second level user authentication is completed. Now, the IS server compares the decrypted UName and PWD with IS database, where IS database contains information of all registered students. If both of them match, finally, the client is authenticated to the IS. In case of failure, the interface allows the student to enter UName and PWD with three attempts. When the user fails in all the three attempts, the user account is blocked and the institution server sends an "Access Denied" message to the user. The proposed model applies multi-factor authentication scheme to increase the assurance of the communicating parties using Accode, IMEI number, UName/PWD and digital certificates.

Once the user authentication process is completed, the institution server sends the authenticated user profile to the MPCS server for payment authorization service as an encrypted format. The user profile includes student name, bank name, branch name, account holder name, account number, amount (fee yet to be collected), institute bank account number, mobile number, secret key and IMEI number to MPCS server.
After receiving the user profile, MFCS uploads the payment authorization interface to the client's mobile device. The interface prompts for option like tuition fee, semester fee, examination fee and fee amount. When the user selects the payment fee option and amount, both of them are encrypted and sent to the MPCS server. The MPCS server validates the response with database records and considers the response as the payment confirmation. If the fee amount matches with MPCS database, the mobile user is authorized by the MPCS to proceed with the payment service.

After the payment service is authorized for client, the MPCS server creates a log file for each payment request with the following parameters: log number, time at which the log is created, transaction id, transaction status, etc. During the payment process, the transaction status may falls into 'active', 'success' or 'failure' mode. MPCS now prepares and sends the Payment Request (PR) with the necessary information to the institution authorized bank. In this case, MPCS and the institution authorized bank make an agreement for fund transfer in preorder.

The institution bank starts the payment process by forwarding the PR to the respective client's bank. The client bank identifies the payer based on his/her Bank-ID and the account number. If the requested amount is available in the client account, the client's bank transfers the amount to institution authorized bank as regular way. The transaction status falls into 'success', when the fee amount transfers from user's account to institution account successfully, otherwise, it is 'failure'. The transaction status 'active' represents that the transaction is still in progress mode.
Once the requested fee amount is debited from the user’s account and credited to the institution account, the institution bank confirms the payment transaction and sends a confirmation message to the MPCS server. After receiving the payment confirmation message by the MPCS, it removes the transaction log file and forwards the same message to the institution server. Finally, the payment user receives the payment confirmation message from both the institution as well as from the user bank.

3.3 Security Framework for Proposed System

A framework is a logical structure that classifies and organizes the components and artifacts of the system. A comprehensive framework has a consistent naming of the components and elements and an expressive set of graphical representations for each component and element. The proposed security framework for MPCS provides well-defined and coherent components and artifacts of the system. Additionally, it also provides the necessary technical infrastructure such as acquiring user information, connectivity, communication and security to facilitate m-payments and acts as an intermediary between the banks, the institution and the clients. This framework is specifically developed for the academic related payments for the registered clients through mobile devices. The security framework for MPCS is designed according to the software architectural standards and presented in Figure 3.2.
Figure 3.2: Framework for Mobile Payment Consortia System

The functional components involved in the proposed MPCS security framework are illustrated in the following sections.
3.3.1 Functional Components for Mobile Client (MC)

The client handles a variety of mobile devices like Nokia, Sony Ericsson and others. The different mobile devices have different technical and physical characteristics in terms of performance and functionality. The modern mobile phones like smartphones provide much of the same functionality that is supported by the desktop computer. This makes them potentially reliable for mobile payments with the support of security. There are several components available for the mobile client to carry out the mobile payments.

Designing the client-side user interfaces for mobile payment application, presents a new set of challenges for the interface designers and software developers due to the limitation of mobile devices such as limited screen size, power, memory, battery life and computational capabilities as well as the constraints of wireless networks such as low bandwidth, high latency, and unpredictable availability and stability.

The main responsibilities of UserInterfaceManager are to acquire the specification different mobile devices and to validate the data entered by the client based on business and security logic. The Java 2 Micro Edition (J2ME) is a preferred development platform for mobile payment application interface due to portability of the Java code, ability to reduce the network traffic since Java can process the data locally, and ability to establish a different security policy at the client device. J2ME Wireless Tool Kit (WTK) includes the standard set of Application Programming Interfaces (APIs) like Connected Limited Device Configuration (CLDC), Mobile Information Device Profile (MIDP), Wireless Messaging API (WMA), etc. The MIDP contains several API methods to implement the mobile client request on the...
different mobile devices. The mobile client starts to communicate with the IS by entering AcCode at mobile device. The UserInterfaceManager handles various interfaces such as authentication interface and payment authorization interface.

The DisplayManager maps the size and resolution of the client device and based on them it provides application interface at client device. While entering the value to the interfaces, the CommunicationManager (CM) establishes and manages the connectivity with IS and MPCS server. It allows the exchange of information in a secure way between the communicating parties. The CM maintains the account for source and destination of communication, information about communicators, the time at which the communication is initiated. Other communication issues such as performance, security and privacy, litigation are also handled by the CM. TaskSwitchingManager keeps the information about payment transactions with TransID. The major responsibilities of SynchronizationManager are sending and receiving the data between the client and IS and with the MPCS server, authenticating the device using authentication interface, and downloading the payment application updates. During the data synchronization, the client requests are encrypted and decrypted with the help of SecurityManager. The SecurityManager is also responsible for authentication, signing the request, certificate management, secret key management and distribution etc. The SecurityManager consists of various security functional elements to support Public Key Infrastructure (PKI) and Bouncy Castle API (BC-API) so that the data entered by the client is sent in a secure way. The DeviceAppManager keeps the information about payment application Business Logic, validations and alerts, and authentication information. The BackupManager supports atomic transaction in case
of network disconnection. When the network is disconnected, the failed transaction is picked up from the restore point and resumes the data, instead of restart again.

Once the device authentication is completed, the UserInterfaceManager provides UName and PWD for client authentication, while entering the values for UName and PWD by the client that are encrypted using SecurityManager and sent to the IS as an encrypted message. At this time, DeviceAppManager keeps information like authentication reports and alert messages.

3.3.2 Functional Components for Institution Server (IS)

The ConnectionManager of IS establishes the connection with mobile client and with MPCS server. The RequestQueueManager is responsible for defining a queue for all the payment requests and for server's responses. Once the request is received by the server, the request is sent to the SecurityManager of the IS. The SecurityManager decrypts and encrypts the PR that come from client mobile device to the IS and vice versa using asymmetric key scheme. The SecurityManager of the IS also contains security elements to support secure communication. The main responsibility of the AuthenticationManager is to authenticate the client device and the mobile user. The KeyGenerationManager is responsible for generating and keeping the secret keys for all participating entities and private/public key for IS. The CertificateManager handles and maintains the X.509 client certificates for all payment users to support authentication. Once the initial authentication is completed by the SecurityManager by verifying the IMEI of the client device, the requests are sent to the ServiceManager through ConcurrencyManager for processing.

An Architecture for Secure Mobile Payment System using Public Key Infrastructure
The ConcurrencyManager handles the multiple services that come from the various mobile clients. Since the ConcurrencyManager keeps the technique of parallelism, it supports quick response time and reduces latency period. To improve the performance, the ConcurrencyManager finds the IMEI number as well as UName/PWD in parallel, instead of processing sequentially. The ConcurrencyManager executes the queries of mobile user, with the help of ServiceManager.

The ServiceManager receives all the requests from clients and creates a session for requests of each client. Then, the ServiceManager communicates with AreaTrackingManager to identify the location of the user, if necessary. This information may be useful to provide value added services to the mobile clients. The DatabaseManager keeps the records of the profiles of all clients including their secret key information. Once the institution server is connected to the client device, the IS receives the client request and decrypts them using SecurityManager. The AuthenticationManager validates the UName and PWD with the help of DatabaseManager via ServiceManager. After completing the authentication process by the institution server, it establishes a connection with MPCS Server and sends the client's profile as an encrypted format with the support of SecurityManager.

3.3.3 Functional Components for MPCS Server

The CommunicationManager of the MPCS server is connected to the IS and receives the profile of the authenticated clients including client's mobile number for payment authorization. To carry out the payment processes successfully, the Payment Request (PR) is routed through banking network for funds transfer. In the
proposed model, MPCS server acts as a payment gateway and responsible for maintaining the log information, authorizing the payments and forwarding the PR to institution authorized bank.

To authenticate IS and MPCS, both of them apply challenge-response mechanism with the help of SecurityManager. Once the authentication is completed, the MPCS server receives the client’s profile from IS and decrypts them using SecurityManager. The MPCS server then, uploads the user interface to client device for payment authorization along with log number via ServiceManager. The responsibility of the LogManager is to create a log file for each client request with the parameters of log number, time at which the log is created, transaction id, transaction status and so on. The transaction status may fall into any one category such as ‘active’, ‘success’ or ‘failure’.

The MPCS Server authorizes the user by verifying the value for option-id and fee amount with the help of DatabaseManager, where it keeps information of all payment users. Once the payment authorization is over, the ServiceManager prepares the PR using PRRequestGenerator, where PRRequestGenerator follows the PR format defined by the institution authorized bank. After creating PR, the MPCS server establishes the communication with the institution authorized bank using CommunicationManager and sends the payment request.

### 3.3.4 Secure Banking Network

Once the PR reaches the institution authorized bank, it starts the payment process with student bank. The banking network is more secure one. The banks are communicating with each other via their proprietary network. In the same way, the
institution authorized bank provides a secure communication with MPCS server which is out of the scope of this research. At the time of payment, the institution authorized bank accepts all the payment requests and sends the secure interface to client bank for fee amount collection. When the requested amount is debited from the client account and credited to the institution account, the institution authorized bank sends the payment confirmation message to the MPCS and it forwards the same message to the IS. Finally, the IS sends the payment transaction status to the mobile client. The client's bank also confirms the payment with mobile client. Thus, the client request is carried out successfully by the proposed system.

3.4 Security Architecture for MPCS

The security is the integral part of proposed MPCS system. The MPCS architecture is designed with high level security using Public Key Infrastructure (PKI). The PKI provides strong security services and allows distribution and use of public keys and certificates. The PKI offers a great deal of advantages with Digital Signature and standard encryption/decryption algorithms using asymmetric key cryptosystem. Both privacy and authentication can be provided using the combination of encryption and digital signature. The message digest algorithm can ensure the message integrity effectively. The critical security factor of non-repudiation is well supported by the digital signature. The standard encryption algorithms are used to ensure the strong data confidentiality.

To carry out the mobile payment successfully, the proposed system has to provide security at three stages. Figure 3.3 depicts the layered security architecture for MPCS. The first stage is to protect the communication between the mobile client...
and IS. The secure communication that is required in the second stage is between the IS and MPCS server. In the third and final stage, the proposed system has to protect the information exchange between MPCS server and mobile client. The MPCS server is connected with the institution authorized bank via Virtual Private Network (VPN) in the wireless environment using ISO 8583 interface. The remaining entities of the proposed system have secure communication using Hypertext Transfer Protocol over SSL (HTTPS) method.

Figure 3.3: Layered Security Architecture for MPCS
The security architecture for the proposed system provides the software elements that make up application security. These elements perform various kinds of security tasks. The security architecture elements are cohesive and loosely coupled to simplify reuse and maintenance. The client-side User Interface (UI) elements provide a mechanism for mobile clients to interact with payment application deployed in institution server as well as with MPCS server. These elements acquire and validate data entered by the mobile clients at device itself. UI process elements drive security processes to provide secure user interface for authentication, authorization, etc. and offer communication security with the help of Security Entities. The Security Entities manage the user credentials such as client and server certificates and encryption keys. The Data Access Elements are used to access those security credentials that reside at mobile database.

The Institution server presents the unique feature of request façade that is used to optimize the user’s payment requests. Security Logic elements keep the security logic for the payment application. These elements apply security rules on payment interfaces and perform various security services. Once the payment requests are received by the IS, the additional processing steps are required such as CA authentication service. Security Workflow elements define, control and implement the security services with the external systems for each payment request using customized third party packages. The Security Entities manage the all user’s credentials and implement the security properties in payment process. The Data Access Elements abstract the logic required to access the underlying institution data store.
The MPCS server maintains the service façade that optimizes the IS requests. However, the MPCS server processes the user requests in real-time sent by the IS. The Service Interface Elements exposes the business logic as well as security logic into the all payment services. Security Workflow elements of the MPCS server establish and maintain the security services with banks while accessing the payment requests. The Data Access Elements provides the secure way to access the MPCS database.

The payment application is required to ensure the specific security functionality that can be accessed from the components in any layer. These common security elements perform authentication, authorization and validation. The elements for operational management task implement exception handling policies, performance analyzer, configuration and tracking. The Communique Elements manage the communications with other applications and tools like certification validation tool. The security architecture provides a secure open technological solution to mobile payment transactions and offers convenience for mobile users, service providers like institution and financial institutions such as banks. The security architecture is implemented by PKI that facilitates the secure payment transactions. The proposed security architecture offers security features such as authentication, authorization, confidentiality, data integrity and non-repudiation, and resolves frauds against institutional payments. The authentication is a process of verifying authenticity of the user to access a system by one or more authentication mechanisms. The MPCS supports entity authentication and user authentication. The device authentication is achieved by X.509 standard public key certificate and IMEI number of the client device, whereas the client/server authentication is verified by AcCode, and

An Architecture for Secure Mobile Payment System using Public Key Infrastructure
Authorization is the property by which the user’s properties can be associated to the payment service access. MPCS authorizes the users by validating the fee amount. Confidentiality is the process that keeps the information secret from an unauthorized person, process and device. The proposed system supports data confidentiality by using standard encryption mechanism. Integrity ensures that the information and systems have not been altered or corrupted by outside parties, since the message digest algorithm MD5 is revealed already and thus SHA-1 algorithm is adopted in MPCS to achieve message integrity. Finally, the important property of non-repudiation prevents the communicating parties from denying their actions after the transaction has completed by issuing proof. The digital signature supports and ensures the non-repudiation mechanism in the Secure Mobile Payment System.

The proposed mobile payment security architecture provides a solution for securing payment’s sensitive data over the GPRS network, irrespective of the underlying transport protocol used for transporting this data. However, MPCS uses SSL/TLS protocol for secure data transmission. The only requirement is that of having a MIDP-complaint device. This architecture helps the students to make the payment to the college account and to perform secure transactions anytime, anywhere through their mobile devices.

The security architecture of MPCS supports authentication mechanisms in three different levels and tested successfully. They are:

- Device authentication
- Client Authentication
- Client/Server Authentication
3.5 Secure Communication between Mobile Client and IS

The proposed model, MPCS has been developed based on Public Key Infrastructure that supports all the security services such as authentication, privacy, confidentiality, data integrity and non-repudiation. The proposed system applies the most widely used digital certificates (X.509 standards) to provide recognized proof of the person’s (or entity) identity. The Digital certificates are one part of a set of components of PKI. The PKI also includes Certification Authorities (CA) that sign, issue, manage, and revoke digital certificates. The Certificate Revocation List (CRL) is another crucial PKI component that lists the certificates which are formerly valid. The PKI certificates are revoked, when the keys within the certificate are compromised or access privileges for user/device are lost or by theft of mobile devices. The device authentication process is shown in Figure 3.4.

3.5.1 Device Authentication

![Device Authentication Diagram]

Figure 3.4: Device Authentication

An Architecture for Secure Mobile Payment System using Public Key Infrastructure
In the MPSCS model, X.509 Mobile User Certificate (MUC) is bound with MPSCS.jar file, which is used to verify the user mobile device by the Institution Server. While activating the interface, a separate thread sends X.509 based client certificate and URL to the institution server over the GPRS connection. After receiving the client's certificate, the institution server checks that the certificate has been presented within the validity period and verifies client's certificate with its public key. Then the institution server obtains the user certificate from CA through the client URL.

To verify whether the certificate is on revocation list, the institution server obtains CRL Distribution Point (CDP), appear as a URL, from the client certificate and retrieves CRL from the CDP. The institution server examines CRL for the serial number of the user certificate. Then IS sends query to the CDP for 'fresh' CRL. The CDP processes the query and responds to the IS with the updated CRL. The new CRL is verified by the institution server for the validity of the client certificate. When the validity of the user certificate and public key of the user is matched with the user certificate, sent by the mobile user as well as the certificate obtained from the CA, the client device is identified and authenticated.

Once the client device is identified and authenticated, the institution server sends X.509 based server certificate to the mobile user for validating the server certificate. To verify the server certificate, the mobile user uses Online Certificate Status Protocol, which offers an online mechanism for determining certificate validity. The establishment of certificate validity by checking CRLs is an effective one, however it may not be adequate due to reasons such as large quantity of memory.
obtained by the large number of revoked certificates, high-bandwidth utilization for
downloading the large CRLs causing thereby network congestion, frequent
expiration of CRLs, and immediate notification of certificate revocation that is
required by the most sensitive financial applications like mobile payment
application. OCSP is a lightweight certificate validation scheme and the OCSP
servers are able to query the certificate database directly for certificate revocation
status. To check the revocation status of the server certificate, the mobile user
transmits a query to the OCSP server with the serial number of the server certificate.
The OCSP Server then checks the certificate revocation status and responds to the
mobile user with “revoked”, “good’, or “unknown”. If the OCSP server response is
good, then the server is identified and authenticated to the mobile user. OCSP offers
a more real-time mechanism to send the status of the certificate without consuming
large quantities of the network bandwidth. Once the user device and the institution
server are identified and authenticated by each other, the secure connection is
established between them.

3.5.2 Client and Client/Server Authentication

The client and client/server authentication process is presented in Figure 3.5. Over a
secure connection, the MPCS.jar user-interface can start and prompt for AcCode.
At the time of registration for mobile payment services, the institution server (IS)
generates a secret key for each user by using both IMEI number and AcCode. The
generated secret key is used for client authentication as well as for further secure
communication. The institution server encrypts IS’s public key using this secret
key and binds the encrypted public key with server certificate that is signed by the
CA. The encrypted public key server certificate is sent to the mobile user through the GPRS for client authentication, after the user device is identified and authenticated.

When the encrypted server certificate is received by the user device, the user interface sent by the IS generates a secret key using AcCode and IMEI number. The user interface then attempts to decrypt the server’s public key which resides in the server’s certificate. If the server’s public key is decrypted successfully, then the client authentication is completed initially. If not, the error message for invalid AcCode is displayed at the mobile device and the interface prompts for AcCode. The user allows three attempts to enter AcCode. If the user fails in all the attempts, the respective mobile payment service account is blocked.

Once the client is authenticated initially, the institution server commences second level user authentication and the user interface prompts for UName and PWD. To ensure the message integrity, the user creates 160-bits hash value, called Message...
Digest, for username and password using hash function SHA-1. The hashed UName and PWD are encrypted using the secret key in order to prevent eavesdropping from the intruders. The message digest is then encrypted using client's private key in order to create a digital signature by which the critical security component of non-repudiation is achieved. Finally, the mobile application encrypts the UName and PWD using the institution server's public key and sends both encrypted and signed UName and PWD to the institution server.

The server starts by extracting the user's public key as well as IS's public key. In order to extract the user's public key, as well as the IS's public key, the server identifies the requester's MSISDN number and finds the respective keys from the IS's database. Using the client's public key, the institution server de-signs the document and decrypts UName and PWD using the IS's public key respectively. Once decrypted, the institution server creates a message digest for UName and PWD using hash function SHA-1. The server then, decrypts the message digest using the shared secret key. To ensure the message integrity, the decrypted message digest is mapped with the digested UName and PWD, which is generated by the institution server. If both of them match, integrity of the message is ensured and the server validates the UName and PWD with the server database. If both match, the client/server authentication is completed successfully and the server sends welcome message to the mobile user. Otherwise, the server sends error message for invalid UName and PWD to the user device. The client and client/Server authentication process is diagrammatically explained in Figure 3.6.
Figure 3.6: Encryption/Decryption operations during client and client/server Authentication
3.6 Secure Communication between IS and MPCS

Once the user authentication process is completed by the institution server successfully, and then IS starts handshake processes with MPCS server for secure communication. The proposed model authenticates both the institution server and the MPCS server using challenge-response mechanism. When the institution server initiates the communication with the MPCS, the MPCS server sends an encrypted RAND-number using shared PIN number to the institution server. The institution server extracts the PIN number from its database and decrypts the RAND number given by the MPCS. If the RAND number is decrypted successfully by the institution server, then MPCS server is identified and authenticated and generates SRES based RAND number sent by the MPCS. Finally, the institution server encrypts the SRES using shared PIN number and sends it to the MPCS server. The IS and IS/MPCS authentication and secure communication are shown in Figure 3.7.

![Figure 3.7: Authentication and Secure Communication between IS and MPCS](image)

*An Architecture for Secure Mobile Payment System using Public Key Infrastructure*
Meanwhile, the MPCS server generates SRES based on RAND number which is sent to the IS. The MPCS server then, decrypts the SRES and compares the SRES which was sent by the IS and generated by the MPCS. If both of them match, the institution server is authenticated to the MPCS server. Once each entity is authenticated, the institution server and MPCS server exchanges their public keys to each other.

While sending the public key by the IS, the institution server encrypts its public key using PIN number where the PIN is shared between IS and MPCS server. The encrypted IS's public key is hashed by using SHA-1 algorithm and this hashed key is signed by the IS's private key. Finally the institution server concatenates both encrypted (M1) and signed (M2) IS’s public key and sends the combined key message to the MPCS Server.

The MPCS Server receives combined message and it splits the message into M1 and M2. The encrypted IS’s public key (M1) is decrypted by using shared PIN number and the signed message (M2) is decrypted using the IS’s public key. To verify the integrity of the message, the IS’s public key is digested again using SHA-1 algorithm. The new digest key message is checked with the digest message sent by the IS server. If both are matched, the integrity of the message is ensured by the MPCS server. Once the institution server sends its public key successfully, then MPCS server starts to send its public key to the IS in reverse procedure as secure way. The process of public keys exchange between IS and MPCS server is depicted in Figure 3.8.
After exchanging the public keys between the IS and MPCS server, the institution server sends the user profile for payment authorization that includes student name, bank name, branch name, account holder name, account number, amount (fee yet to be collected), institute bank account number, mobile number, secret key and IMEI number to Mobile Payment Consortia System in a secure way. While sending the user profile, the institution server encrypts user profile by using IS’s private key in order to create IS digital signature for the message. The signed user profile is then...
digested by using SHA-1 algorithm. This allows the MPCS server to detect any modification which has been done with user profile by intruders. The digest user profile is encrypted using MPCS’s public key to protect from eavesdropping by any third party. Finally, the institution server sends both the signed and encrypted digest user profile to the MPCS server.

![Diagram of Encryption and Decryption Process for Sending User profile](image)

**Figure 3.9: Encryption and Decryption Process for Sending User profile**

After receiving the messages, the MPCS server decrypts signed user profile using IS’s public key and stores the original user profile into MPCS’s database. Also, the
server decrypts encrypted digest user profile by using MPCS’s private key. If there are failures to decrypt both the messages, the MPCS server sends an error message back to the IS. Using SHA-1 algorithm, MPCS server creates digest for signed user profile and authenticates the integrity of the message by verifying newly created digest signed user profile with sent digest signed user profile. The process of encryption and decryption for sending user profile from IS to MPCS server is depicted in Figure 3.9.

3.7 Secure Communication between MPCS and Mobile Client

After receiving the user profile, MPCS server starts the PR authorization service by sending the interface to the mobile user. When the MPCS initiates the communication with the mobile user, the server initially verifies the IMEI number of the user device for client device authentication. Then the user interface for PR authorization is enabled. The user interface consists of fee options such as semester fee, examination fee, etc., and fee amount. The payment user can choose an option from a list provided by the institution. The option-id and fee amount are hashed using SHA-1 algorithm to produce unique message digest that ensures integrity of the data. The hashed option-id and the fee amount are then encrypted using the client’s private key to create the digital signature for user authentication and non-repudiation.

In parallel, the MPCS server encrypts a copy of the option-id and fee amount message using the secret key to support high level security. Since the secret key is known only by the user and the institution, it authorizes the user to carry out the payment transactions and ensures data confidentiality. Both the digital signature and
the encrypted message are concatenated and encrypted using MPCS’s public key. This avoids data theft and eavesdropping. To protect from identity theft as well as to ensure data confidentiality, the MPCS’s public key is encrypted using part of an IMEI number. Finally, the payment application creates digital envelop with encrypted concatenated message (M1) as well as with encrypted IMEI number (M2) and sends it to the MPCS server that ensures message confidentiality.

Once the digital envelop is received, the MPCS server decrypts the message M2 using the part of an IMEI number and the message M1 using the MPCS’s private key.

Figure 3.10: Secure Communication between Mobile Client and MPCS
The MFCS server also verifies the remaining part of an IMEI number in order to validate the user and splits the encrypted message digest i.e. digital signature as well as encrypted option-id and fee amount. Then the server decrypts the encrypted message digest using the client’s public key and encrypted option-id and amount using the secret key, which in turn retrieves both the message digest as well as option-id and amount. Finally, the MPCS server creates message digest for option-id and amount and compares the message digest sent by the client as well as the newly generated message digest. If both are matched, the integrity of the message is ensured and if fee amount is matched with MPCS’s database successfully, the user PR is authorized. The Security functionalities between client and MPCS server are shown in Figure 3.10.

3.8 Significance of the Proposed Architecture

The proposed architecture supports end-to-end security by achieving the security properties such as confidentiality, authentication, authorization, data integrity, and non-repudiation.

Confidentiality

The confidential of the message is concerned with protecting the message content from disclosure to any party. The confidentiality of a message as well as the authenticity of the sender can be guaranteed using asymmetric key encryption. The shared secret key is generated using AcCode and IMEI number, which is used to encrypt the public keys. This secret key is only known to the involving parties in the payment transaction. Additionally, the proposed security algorithm using public key infrastructure promises the message confidentiality during the transmission. At the
time of payment authorization, MPCS gateway obtain the information from the client using digital envelop. This ensures an added confidentiality for the message.

**Authentication**

The authentication is the process of establishing or confirming the identity of communicating parties and enhances trust relationship among them. The proposed mobile payment system provides secure authentication infrastructure at different levels. The device authentication is achieved by verifying the validity of engaging party’s X.509 certificates with CA. The client is authenticated by decrypting the server’s public key using shared secret key. The client/server authentication is achieved by validating the AcCode and by verifying the UName and PWD.

**Authorization**

The payment gateway, namely MPCS server permits authenticated mobile clients to carry out payment transactions, in particular to confirm the payment service. The MPCS server ensures both authorization and payment confirmation by validating the option-id and fee amount with secret key. Although option-id and fee amount are revealed to anyone, the secret key is only known to IS and mobile client.

**Data Integrity**

The protocol used in the proposed system ensures that the message was not modified and not tampered by anyone during the radio transmission. The data integrity is achieved by creating a fingerprint for the message that is generated using SHA-1 hashing functions. If the message was altered or bribed by the malicious users, the receiver would get the different fingerprint for the original message and revealing that the message was compromised.
Non-repudiation

In the proposed system, both the sender and receiver could not repudiate the payment transaction. The important security service of non-repudiation is achieved using digital signature. The digital signature only can generate by the private key of the involving parties where the private key must be kept secret. If the private key is not properly safeguarded by the owner, digital forgery can become a major issue. However, the proposed system solves this problem by protecting client’s private key at client device using AcCode.

Hence, this architecture is a novel one when compared with the existing architecture designed with end-to-end security using PKI to carry out the financial transactions through the client’s mobile devices anywhere at any time. The security algorithms developed and the experimental study are presented in the forthcoming chapters.