A Secure Payment Model Using Representational State Transfer (REST) Web service, Quick Response (QR) Code and Hash-Based Message Authentication Code (HMAC)

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Shirisha Yellu
Fall 2014

Committee Members

Dr. Mario Garcia  
Committee Chairperson

Dr. David Thomas  
Committee Member
ABSTRACT

With the improvement of living standards, stores are growing bigger with variety of available products. Therefore, building a simple, fast and convenient shopping guide system has become a concern for both merchants and customers. Security has become a major concern for the consumer providing credit card details. Major threats are reported at both onsite and online transactions. So it is important to create a system to establish secured payment method. Proposed system is the implementation of a secured operational model for mobile payment in which access control is based on Service-Oriented architecture. A consumer uses his/her mobile device to get authorization and generate a QR code as the payment certificate to pay for the products to purchase. Using REST web service for communication between different applications would help in instant messaging, which increase in performance of the system. User details are encrypted using HMAC encryption technique.
TABLE OF CONTENTS

Abstract .................................................................................................................................................. ii
Table of Contents ........................................................................................................................................ iii
List of Figures ........................................................................................................................................ iv
List of Tables ......................................................................................................................................... v
1. Background and Rationale .................................................................................................................. 1
   1.1 Near Field Communication ........................................................................................................ 3
   1.2 Mobile Wallet ............................................................................................................................ 4
   1.3 QR-Code ..................................................................................................................................... 4
   1.4 Representational State Transfer (REST) Web Service ............................................................ 5
   1.5 Comparing QR-codes to NFC technology ................................................................................ 5
   1.6 QR-Codes Vs Bar-codes .............................................................................................................. 6
   1.7 SOAP vs REST Web Service .................................................................................................... 7
   1.8 Existing Application- Mobile Device as a Payment Tool .......................................................... 8
      1.8.1 Starbucks Card Mobile Application .................................................................................... 8
      1.8.2 Google Wallet .................................................................................................................... 9
      1.8.3 EASYCARD ...................................................................................................................... 9
   1.9 Android ...................................................................................................................................... 9
      1.9.1 Eclipse Integrated Development Environment ........................................................................ 10
      1.9.2 Android Software Development Kit .................................................................................. 11
      1.9.3 Android Development Plugin Tools for Eclipse ............................................................... 11
   1.10 Android Emulator .................................................................................................................... 11
4.3 Bank Application ........................................................................................................34
  4.3.1 Login Page ........................................................................................................34
  4.3.2 Account Operations Page ..................................................................................35
  4.3.3 Create Account ..................................................................................................36
  4.3.4 Withdraw ...........................................................................................................37
  4.3.5 Deposit ................................................................................................................37
  4.3.6 Check Balance ..................................................................................................38
  4.3.7 Transfer Amount ..............................................................................................39
  4.3.8 View Report ......................................................................................................39

4.4 Android Application .................................................................................................40
  4.4.1 Login Page .........................................................................................................40
  4.4.2 Input Merchant ID and Amount .........................................................................41
  4.4.3 QR Code Generation .........................................................................................42
  4.4.4 Scanning QR Code to get Verification Code ....................................................43

5. Testing and Evaluation ................................................................................................45
  5.1 Test Cases ...............................................................................................................46

6. Conclusion and Future Work .....................................................................................58

Bibliography and References ..........................................................................................59

Appendix 1 .....................................................................................................................62
LIST OF FIGURES

Figure 1.1 A Sample QR Code for Wikipedia Main Page...................................................5
Figure 2.1 Transaction in Online Shopping.................................................................14
Figure 2.2 Cost of Card Fraud....................................................................................15
Figure 2.3 Prepaid Mobile Contactless Payments: Transactions and Total Value........16
Figure 2.4 Application Development Trends..............................................................17
Figure 3.1 Proposed System Architecture .................................................................18
Figure 3.2 Flow Chart Diagram for the Proposed System.............................................23
Figure 3.3 QR Encoding Process................................................................................23
Figure 3.4 Communication between Service Requester and Service Provider ..........24
Figure 3.5 Code Snippet for Calling Web Service to Authenticate Bank User..........25
Figure 3.6 Code Snippet for Calling Web Service to Generate QR Code.................26
Figure 3.7 Code Snippet for Calling Web Service to Verify Payment Code ...............27
Figure 4.1 Shopping Website Login Page .................................................................30
Figure 4.2 Shopping Website Home Page.................................................................31
Figure 4.3 Available Products in the Shopping Website................................................32
Figure 4.4 Items in the User Cart...............................................................................33
Figure 4.5 Checkout Page...........................................................................................33
Figure 4.6 Order Confirmation....................................................................................34
Figure 4.7 Login Page of the Bank............................................................................35
Figure 4.8 Home Page of Bank after Successful Login of User.............................36
Figure 4.9 Creating New Account by User.................................................................36
Figure 4.10 Withdraw Amount from User Account ..........................................................37
Figure 4.11 Deposit Amount to User Account ...............................................................38
Figure 4.12 Checking Available Balance in User Account ..............................................38
Figure 4.13 Transfer Amount .........................................................................................39
Figure 4.14 View Transaction Report.............................................................................40
Figure 4.15 Login Page on Android Application.............................................................41
Figure 4.16 Input Merchant ID and Amount ..................................................................42
Figure 4.17 Generated QR Code....................................................................................43
Figure 4.18 Verification Code on the User Device..........................................................43
Figure 5.1 Fields Validation on Login Page of Shopping Website ..................................47
Figure 5.2 Validating Username and Password on Android App..................................48
Figure 5.3 Screen Prompting for Merchant id and Amount after Successful Login ......49
Figure 5.4 Generated QR Code on Android Application on Emulator .........................50
Figure 5.5 Verification Code Table .................................................................................51
Figure 5.6 Scanning QR Code to get Verification Code ................................................53
Figure 5.7 Payment Screen with Input Merchant ID and Code......................................53
Figure 5.8 Payment Confirmation Page..........................................................................54
Figure 5.9 Database Rows after Deletion of Payment Success Transaction..................55
Figure 5.10 Transactions Report in Online Bank ............................................................54
Figure 5.11 QR Code for a Different Transaction ..........................................................56
Figure 5.12 Scanning QR Code to get Verification Code for Different User.................57
LIST OF TABLES
1. BACKGROUND AND RATIONALE

M-commerce (mobile commerce) is trading of products and services through wireless devices. A few devices to term are personal digital assistants (PDAs) and cellular telephone. It is also termed next-generation E-commerce [2]. It was originally coined in 1997 which means "the delivery of electronic commerce capabilities directly into the consumer’s hand, anywhere, via wireless technology” [3]. Mobile devices are powerful and connected as personal computers or laptops. In recent years mobile phone technical capabilities have advanced rapidly and number of people using smart phones continued to increase [1].

Mobile commerce is worthy US231$ billion, with Asia representing almost half of the market. It is believed to reach US700 $ billion in 2018. According to Business Intelligence, almost 30% of users of mobile make purchases by 2015. Bankers estimates $68.1 billion in purchases will be made through mobile devices in 2015 by US and European users. Retailers of the mobile in United Kingdom are expected to rise their revenue to 31% FY 2012–14. A considerable increase in number of consumers can use their mobile phones as keys, cameras, and TVs. If mobile can be everything why can’t it be a payment device [3].

The main objective of this project is to create a system where consumer uses their mobile devices to pay for a wide range of services and digital or material goods instead of paying by cash, check, or credit cards. However, customer's security concerns are a significant boundary to wide selection and utilization of mobile payments. In this project, a secure operational model is outlined for mobile payment in which access control is based on service-oriented architecture [1]. A consumer uses his mobile device to get authorization from the bank application which generates QR Code as a payment Certificate. User can use this QR-Code in payment process for the purchases. The communication between different applications happens through standard
secure REST web service protocol. Confidential information like account number and passwords are encrypted using HMAC technique before transferred from android application to bank application for authentication to avoid man-in-the-middle (MITM) attacks. As stated in [1], such a system has the capability to overcome recent frauds which affected 40 millions of users across the globe in target store for thanksgiving in year 2013 where consumer credit card details have been hacked by intruders by introducing malware into their system [30]. Home Depot is another example of credit card fraud which affected 60 million users where hackers stole credit and debit card information which was revealed on September 2 2014 [29].

Importance of the project

- Safety and Security: As payment is done using mobile device which is safer and more secure when compared to other forms of payment. Moreover, if the card is lost or stolen the user can still use his online credentials of the bank to use to make the transaction. It also avoids the chance of “skimming”, which reduces the chance of stealing the card details by dishonest employee either at a restaurant or at a billing counter in a shopping location.

- Light Weight Application: With REST as a mode of communication the application becomes light weight. It relies on the HTTP standard. It is great to get a useful web service up and running quickly.

- Validity of Code: QR- Code generated for the payment process is based on Merchant ID, Account Number, and Time Stamp. So it is valid only for particular time stamp (15 minutes is used in this project) and can be accepted with only that particular merchant. With this, though the QR- Code is hacked by some hacker, it cannot be used in different
transaction. There are less chances for the hacker to use the code with that particular merchant in valid time period.

- **Overcome MITM attacks**: Man-in-the-Middle attack is overcome using HMAC technique. Using this technique Message Authentication Code for Account number and Password is calculated by using Cryptographic Hash function in combination with cryptographic secret key and then transferred through the network.

**M-Commerce Technologies**

1.1 Near Field Communication

Near Field communication (NFC) is a type of short-range remote communication. Antenna utilized here is much littler than the wavelength of the carrier signal. NFC permits two electronic devices set inside a couple of centimeters of one another to exchange information. In place for this system to work, both gadgets must be outfitted with a NFC chip. In this present reality, there are basically two approaches to say how this functions [4].

**Two-way communication**: It makes utilization of two gadgets which can read and read. For example, utilizing NFC, you can exchange data like pictures, features, or contacts from two android gadgets by touching together.

**One-way communication**: Here reading and writing to an NFC chip happens through a powered device like credit card reader or phone. So, the NFC-powered terminal deducts amount from the balance when you tap your commuter card on the terminal.
1.2 Mobile Wallet

Mobile wallets make use of near-field communication (NFC) chips inside mobile smart phones and tablets to transmit payment information. Consumer open an app on their handheld devices when he is ready to pay using mobile wallet. Then he enters a PIN and inputs the payment account they want to use. They simply tap their device to an enabled payment terminal at the time of payment. Then transmission of payment information is done. The payment method stays with the customer, increasing security. Faster payments are a way to keep customers happy. Business can be more secure by lessening handling of payments in cast [5].

1.3 QR-Code

Quick response codes (QR Codes) are two-dimensional barcodes designed to share encoded information in a variety of formats. It is generally a square black with white background with pixel box. It can encode variety of information which can be simple text, graphics, or a simple redirection of users to certain web page. It can hold a large amount of information than a bar code. The way of consumers paying for the goods and services are changing with these technologies. With this consumers are making use of their credit cards rather their smart phones. Market is extended to mobile shopping with many merchants and hypermarkets providing the mobile applications or QR Code for consumers [31].

Figure 1.1 shows a sample QR code for Wikipedia main page [23].
1.4 Representational State Transfer (REST) Web Service

REST stands for representational state transfer which is an architectural pattern for developing web service. It is communicated over the HTTP specification which makes use of main methods like GET and POST, or media types like json, html, plain text, or HTTP URI syntax like paths, parameters, and HTTP response codes. Basically HTTP Requests are sent from the client who identifies the location of a resource. The resource type is passed using the Uniform Resource Locator (URL). A typical HTTP REST URL contains protocol, host name, path and query string [27].

1.5 Comparing QR-codes to NFC technology

In present two types of mobile technology are famous in the charge to race for mobile wallet supremacy. All the other players in the field are keeping efforts to replace credit card, with their mobile payment application with focus on NFC technology and QR codes [6].
QR-code:

Mobile device’s cameras scan a code to unlock the user payment information and complete a transaction. Statistics shows that over 29% of US adults used QR code in 2012. There is a significant increase in smartphone users leveraging QR code technology. QR codes can be used by all existing smartphone users [6].

NFC:

Mobile payment system using NFC activates radio connection from one device to other. A statistic show that, between 2011 and 2016 there is an expected growth of usage of NFC by 38%. In 2016, the NFC market expected to reach more than 10 billion dollars. As of May 2013 NFC is available in mobile phones from 9 out of 10 of the world's major manufacturers. It is estimated there will be 500 million NFC-enabled phones on the market by 2014 [6].

1.6 QR-Codes Vs Bar-codes

A barcode is an optical machine-readable representation of information identifying with the article to which it is connected. Initially barcodes efficiently represented information by changing the widths and spacing of parallel lines, and may be alluded to as direct or one-dimensional (1d). QR code encoders provide a number of improved features over conventional bar codes which include the following [7]:

1. **High capacity encoding**: It permits storage of numerous characters with which high limit of encoding of information is attained. Whereas bar code has a cutoff of storing 20 digits.

2. **Improved efficiency of information storage**: QR code stores data in both the directions – horizontal and vertical, which means, it uses one-tenth of space [7]. QR code can store more data as a result of this as contrasted with bar codes.
3. **Capabilities to store Kana and Kanji characters:** QR codes can encode JIS level 1 and 2 character set and a full width Kanji character as contrasted with other 2-dimensional images. It can store 20% more data as contrasted with others.

4. **Damage resistance:** QR codes are damage resistant. It has error correcting abilities which makes it to restore regardless of the fact that it is dirty. One can restore up to 30% of data which is represented to in code word in an incompletely harm QR code. Code word is the unit that develops the information range. When code word holds 8 bits in QR encoding framework. The information may not be completely restored if there should arise an occurrence of great harm.

5. **Omni-directional encoding:** QR codes are readable from various directions. By this, you can read information either from the upper images or the lower images and it permits high speed reading. QR codes attain incredible tasks through position identification designs. They are found in 3 corners of the picture. Quick reading is guaranteed by position detection system furthermore the impact of background interference is diminished.

6. **Multiple data areas:** Data in the QR codes can be planned into single data structure and QR codes are examples which can be separated into numerous data zones. Data symbol can be partitioned into 16 images.

1.7 SOAP vs REST Web Service

Simple Object Access protocol is a set of rules for design of exchange of information in the web services in computer networks. It depends on application layer protocols like Hyper Text Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP). It uses the XML information set for message format. REST is an architectural style in building client-server
applications. SOAP is a set of rules for exchange of data between two endpoints [11].

If you want to perform transactions using a REST system then a new resource is created. Creating a new resource whenever you run into a problem with a REST system generally solves most problems. Compare to SOAP, REST system is more interoperable. Problem with SOAP is that it has large number of different standards to choose from. At the point when a specific seller decides to implement a specific standard, that seller frequently gives a usage that is simply marginally unique in relation to an alternate seller's (or all others). This prompts issues at whatever point you need to cross seller limits (languages and operating system). In terms of platforms, REST has the advantage because all you need to use REST is an HTTP stack [11].

1.8 Existing Application- Mobile Device as a Payment Tool

There are many pre-existing applications which use mobile device as a payment tool. A few among them are:

1.8.1 Starbucks Card Mobile Application

In the Starbucks Card Mobile application, clients enter their Starbucks Card number and their advanced mobile phone turns into their Starbucks Card. This application utilizes standardized barcode technology. In the select stores, buyers can pay with the application. When the card number has been entered on the smartphone device, it will show a scanner tag that is utilized the same way a Starbucks Card is utilized to make purchases. Users can pay and reload the card. It has additionally helped the organization remain solitary as the only large-scale mobile payments provider [8].

1.8.2 Google Wallet

Google Wallet is a mobile payment system developed by google. It allows users to store
credit cards, debit cards, and gift cards. It is also used to redeem sales promotions on their mobile phone. It can use (NFC) to make payments fast and secure. By simply tapping the phone on any PayPass-enabled terminal at checkout it is made convenient [9]. Payment information is transmitted via NFC. After certain time, it is noticed that a transaction with merchant name and dollar amount is recorded on phone. Google Wallet keeps information safe and secure. The app has its own PIN, and if you lose your phone, you can remotely disable your mobile wallet [1].

1.8.3 EASYCARD

EASYCARD is Taiwan's most standard contactless payment service that has been extensively used for taking the subway or getting a charge out of each day shopping/devouring activities. In 2011, TAISYS Technologies Co., Ltd. reported the world's first Android based near-field payment application at the "Open Platform Value-Added Service Conference" composed by the Committee of Communications Industry Development of Taiwan's Ministry of Economic Affairs. It empowers "EASYCARD" utilize that tricks advanced graphical user interface so clients may adequately perform balance checks, micro-installment trades, e-wallet top-up, et cetera by method for their Android handsets [1].

1.9 Android

Smartphone is a gadget that can give different computational and communicational services. There has been an incredible change in the technology of smartphone devices in recent years. Various services are provided inside these little gadgets, which are utilized as a part of our everyday life. With increase in use of smartphones, improvement of smartphone applications has additionally expanded. At present there are millions of applications, which are produced for
different platforms, for example, Android, iOS, Symbian and Windows and so on. Out of all these stages Android occupies over 70% of the smartphone market. It powers a large number of smartphones in more than 190 nations [32].

Android is a Linux-based operating system intended for smartphones and tablets. The user interface of android is based off touch inputs that compares to realistic activities like swiping, tapping to control on-screen objects [34]. It is created by Android Inc., which was gained by Google in July 2005. The openness of the platform is one advantage of creating android applications. It was found in a review conducted in April-May 2012 that it is the most prevalent stage for engineers, utilized by 71% of mobile designer population [32]. There are right now more than 850,000 applications for android [33]. Android comes in different forms, for example, 1.x, 2.x, 3.x, 4.x etc.[34]. The greater part of the android applications is produced in Java, which is a standout amongst the most broadly utilized programming dialects. The Applications created in Java runs on mixture of gadgets, as Java is platform independent. The tools that are required to develop an android app in Java are as follows.

- Eclipse Integrated Development Environment (IDE).
- Android Software Development Kit (SDK).

1.9.1 Eclipse Integrated Development Environment

Eclipse is a Multi-Language Integrated Development Environment (IDE) that involves plugin system and workspace that is utilized to create different applications in distinctive programming languages like C, C++, Java, COBAL, PHP and so on [34]. It is generally used to
write Java applications. It incorporates different development tools that help the coder. It is free and open source programming.

1.9.2 Android Software Development Kit

The methodology of creating new applications for Android is called Android Software Development. The applications are created utilizing Android programming improvement unit that incorporates software development tools [34]. It embodies debuggers, emulators, Application programming Interface (API) tools, tutorials, libraries and example code. The android applications are created in Java utilizing android software development kit as a part of Eclipse.

1.9.3 Android Development Plugin Tools for Eclipse

The Android Development Tools (ADT) for Eclipse enables us to utilize the Android software development kit tools in the Eclipse Integrated Development Environment to create applications in android stage. Android Development tools help the coders to create the tasks rapidly, include packages based on android framework and debug the code and so forth. The use of Android improvement tools with Eclipse is one of the speediest approaches to create the applications in android stage [34].

1.10 Android Emulator

The Android SDK incorporates a virtual mobile device emulator that runs on your machine. The emulator gives you a chance to model, create and test android applications without utilizing a physical gadget. The android emulator mimics the greater part of the hardware and software features of a typical smartphone, aside from that it can't put real telephone calls. It
provides a variety of navigation and control keys, which you can "press" utilizing your mouse or console to create occasions for your application. It also provides a screen in which your application is shown, together with whatever other dynamic android applications.

The emulator uses Android Virtual Device (AVD) configuration to model and test the application. It runs a full Android system stack, down to the kernel level that includes a set of preinstalled applications, (for example, the dialer) that you can access from your applications. User can pick what version of the Android framework he needs to run in the emulator by designing Advs, and he can likewise modify the mobile phone skin and key mappings. At the point when dispatching the emulator and at runtime, one can utilize a variety of commands and choices to control its behavior [28].

1.11 Web Service

A web service is any bit of programming code that makes it accessible over the web and uses a standardized XML/JSON messaging system. XML/JSON is used to encode all communications to a web service. For instance, a customer conjures a web service by sending a XML/JSON message, and after that holds up for a relating XML/JSON response. Since all correspondence is in XML/JSON, web services are not fixed to any one operating system or programming language, Java can chat with Perl, windows applications can talk with UNIX applications.
2. PREVIOUS RESEARCH, MOTIVATION

2.1 Previous Research

Speed pass is the first of contactless payment which came in 1997. Mobil gas stations offered contactless payment devices that clipped onto a key ring. The customer waved the gadget over a named square at the gas pump and paid right away. Today Exxon Mobil still offers this administration, and different service stations are joining contactless payment technologies into their installment decisions. [19]

There has been some previous research done by many scholars to provide security. M.A Qadeer, N. Akhtar, S. Govil, and A. Varshney have done research on contactless payment using RFID-enabled smart SIMcard [20]. P. Urien has proposed a payment model in 2014 which makes use of a Near Field communication technology to secure credit card numbers [14]. Y. Jing has proposed multiple payment models in his paper on online payment and security of E-commerce published in 2009[13]. K. Dey, V. Mankar, and S. Mukherje has proposed an interoperable mobile wallet service published in 2013[15]. Z.Zareh , E. Barkhodari has proposed a system makes use of real time authentication and one time password to provide security[16]. Y. Song, K. Xu, and K. Liu have done research on web instant messaging using REST web service in 2010. H.E. Michail, A.P. Kakarountas, A. Milidonis, and C.E. Goutis have done a research to send the message more securely on a private network using HMAC technique [17].

Traditional Payment Method

Consumers select items from the store and reach the payment lane, where he can pay for the products to purchase. The clerk uses a scanner to scan the bar-code on goods bought by the
consumer. The total price and transaction data are kept in a specific data format and asks the customer to swipe the card. After swipe is performed the card details are collected in order to process a payment and transferred them to a bank electronically. Bank verifies if the customers have sufficient funds to process the payment. If customer is authentic and has enough funds then transaction is committed and bank deposits the money into merchant account [23]. Figure 2.1 shows flow of activities in a traditional online payment system.

![Flowchart of a traditional online payment system](image)

**Figure 2.1 Transaction in Online Shopping**

### 2.2 Motivation

In the past years, credit card breaches have dominated the headlines. 45 million credit card numbers have reported in theft in TJ Maxx’s lax security in 2007. The breaches cost organizations hundreds of millions in fraud. Luckily, they were big enough to recover and continue their business, but it is difficult to gain trust from the public in the face of such injury. Statistics of credit card frauds losses are shown in Figure 2.2.
In a breach, sellers are in charge of a legal examination, remediation cost, and harder to determine cost: loss of buyer confidence. To compound an already painful situation, they will face expanded exchange costs and now and again, the disavowal of charge card acknowledgement benefits. This could force a smaller company out of business [22]. Figure 2.2 shows the total amount of money which is accounted in the credit/debit card frauds.

![Global Card Fraud Losses ($Billions)](image)

**Figure 2.2 Cost of Card Fraud [22]**

In order not to repeat above issues and to diminish the chances of identity breach and to build the execution of the framework, there should be secure payment approach in which payment details are safely exchanged over the system electronically from merchant machine to bank. The trend in contactless payment transactions is also increasing as shown in Figure 2.3.
Also, recently SOA advancement standard has risen to concentrate on drastically enhancing the effectiveness of making, adjusting, amplifying, and repurposing answers for big business application reconciliation, process automation, and trading partner interchanges. The paradigm shift from traditional monolithic applications, followed by client server and distributed architecture has been a gradual one. With Monolithic applications being firmly coupled and coordinated, The thought behind SOA, which is later pattern in application advancement being inexact coupled, consistent and less unpredictable [21]. REST protocol in SOA gives much more preferences to the architecture. REST portrays a set of architectural patterns by which information can be transmitted over a standard interface, (for example, HTTP). REST does not contain an extra messaging layer and concentrates on configuration principles for making stateless services. A customer can get to the utilizing the one of a kind URI and a representation of the asset is returned. With every new resource representation, the customer is said to exchange state. While getting to Restful assets with HTTP convention, the URL of the resource serves as the resource identifier and GET, PUT, DELETE, POST and HEAD are the standard HTTP
operations to be performed on that resource [21]. Figure 2.4 shows the application trends in SOA architecture.

![Application Development Trends](image.png)

**Figure 2.4 Application Development Trends [25]**

### 2.3 My Contribution

My contribution towards this project is, implementation of three different applications (shopping Application, Bank Application, Android Application) to show the idea from 3 different sources – “A Secure Cloud-based Payment Model for M-Commerce”,” Research on Web Instant Messaging Using REST Web Service”, Efficient implementation of the keyed-Hash Message authentication Code (HMAC) using the SHA-1 Hash function. Each paper has a unique idea to improve the security and performance. The difficult part in here is communication between three applications. I used Representation State Transfer (REST) protocol by which instant messaging is possible.
3. ARCHITECTURE

Figure 3.1 shows the proposed system architecture. This architecture is divided into three applications – android application (mobile), bank application, E-commerce website application. Mode of communication between these applications is done using REST web services. Security for account number and password are provided using HMAC technique. User starts with purchasing items from the shopping application. Instead of giving the card details in payment process which is insecure, temporary code (decoded QR-code) is given which is valid for only that particular merchant and specific time frame.

![Figure 3.1 Proposed System Architecture]
3.1 QR-Code Encoding Process

Mode selection, this is the first step in encoding QR-code. The QR standard has four modes for encoding text: numeric, alphanumerical, byte, and Kanji. Every mode encodes the text as a string of bits (1s and 0s),

Every mode encodes the content as a string of bits (0s and 1s), however every mode utilizes an alternate technique for changing over the content into bits, and each one encoding system is optimized to encode the information with the most limited conceivable string of bits. This implies that after you make the string of data bits that represent your content, you should then utilize those bits to generate error correction code words utilizing a procedure called Reed-Solomon error correction. QR scanners read both the information code words and the error correction code words [26].

By comparing the two, the scanner can figure out whether it read the information effectively, and it can correct a few errors on the chance that it didn't read the information accurately. Data encoding, is the step results series of bits to part up into data code words that are 8 bits in length. Then the final message is structured. Data and error correction code words created in the past steps should now be organized in the best possible request. For large QR codes, the data and error correction code words are created in pieces, and these squares must be interleaved as indicated by the QR code detail. Next step in QR-encoding is module placement Matrix. Then place the bits in the QR code matrix. The code words are organized in the matrix in a particular manner. During this step, you will likewise put the examples that are regular to all QR codes, for example, the cases on the three corners. Data Masking, Certain patterns in the QR code matrix can make it troublesome for QR code scanners to effectively read the code. To balance this, the QR code specification characterizes eight mask patterns, each of which modifies
the QR code as per a specific example. One must figure out which of these mask patterns brings about the QR code with the least undesirable characteristics. Last step is to add variant data to the QR code by including pixels specifically zones of the code that were left clear in past steps [26].

3.2 Hash Message Authentication Code (HMAC)

In cryptography, a keyed-hash message confirmation code (HMAC) is a particular development for figuring a message validation code (MAC) including a cryptographic hash function in combination with a secret cryptographic key. Likewise with any MAC, it might be utilized to check both the data integrity and the authentication of a message. In the proposed model, while user login to the android application, his credentials alongside merchant ID are encoded in a message and message is decoded by the service of bank application server. HMAC opposes Man in the Middle attacks due to authentication of server and validates message contents between both client and server. [12]

An iterative hash function separates a message into pieces of a fixed size and repeats over them with a compression function. For instance, MD5 and SHA-1 work on 512-bit pieces. The size of the output of HMAC is the same as that of the basic hash work (128 or 160 bits on account of Md5 or SHA-1, individually), despite the fact that it can be truncated if wanted [12]. The HMAC function can be denoted as

\[
HMAC(K, m) = H((K \oplus \text{ipad}) | H((K \oplus \text{ipad}) | m)) [12]
\]

Where \( H \) is a cryptographic hash function,

\( K \) is a secret key padded to the right with extra zeros to the input block size of the hash function, or the hash of the original key if it's longer than that block size,
$m$ is the message to be authenticated,

$|$ denotes concatenation,

$\oplus$ denotes exclusive or (XOR),

$\text{opad}$ is the outer padding (0x5c5c5c…5c5c, one-block-long hexadecimal constant),

and $\text{ipad}$ is the inner padding (0x363636…3636, one-block-long hexadecimal constant).

Here secret key $k$, is shared with the bank and $m$ is account number and Password which must be passed securely from the android application to the bank application for authorization.

### 3.3 Service Oriented Architecture (SOA)

This model is based on SOA. SOA is an architectural design pattern. To achieve desired results service consumers makes use of a service, which is unit of work done by a service provider. Software agents on behalf of their owners act as both provider and consumer. These services are interoperable via different development technologies [10].

The fundamental building block of service-oriented architecture is a service. A service is defined as a program that can be interacted with through well-defined message exchanges. Availability and stability must be checked at the time of service design [10].

The proposed system, with three different applications being developed in different languages, the standard mode of communication between services happens through messages. So the communication between services through messages makes use of the SOA [10].
3.4 Working with REST web service

REST (Representational State Transfer) is a lightweight Web Service architecture design style, its usage and operation is less marked than SOAP and XML-RPC and it is more compact. It can be finished through the HTTP convention and by utilizing cache it can likewise enhance the reaction speed, execution, effectiveness which are altogether improved than SOAP convention. REST structure follows the CRUD standard. CRUD standard defines 4 main behavior for the resources: Create, Read, Update, and Delete. These are responsible for completion of its operation and handling tasks. These cannot be decomposed further, so they are atomic. Basic operations that can be performed on resources include create, access, modify and delete resources with actions to HTTP protocol of POST, GET, PUT, and DELETE methods respectively. This design and development approach focus on web services. They can reduce development complexity and increase system scalability [17].

3.5 Flow of Execution

Figure 3.2 shows the flow of execution of online payment system in the proposed system. The entire decision making situation that is involved in the project is also shown.
Figure 3.2 Flow Chart Diagram for the Proposed System

Figure 3.3 shows the steps involved in generating QR code.

Figure 3.3 QR Encoding Process

Figure 3.4 shows all the activities between the 3 applications, where activity 1, 6 are communicated/processed using REST Web Service.
Figure 3.4 Communication between Service Requester and Service Provider

Figure 3.5 shows the code snippet to authenticate user credentials from android application with bank application by calling “Authenticate Web Service” using REST protocol.
Figure 3.5 Code Snippet for Calling Web Service to Authenticate Bank User

```java
package com.example.qrpayment;

import java.io.IOException;

public class AuthUserAsynkTask extends AsyncTask<String, Void, String> {
    public String callWebService(String accountId, String password) {
        HttpClient client = new DefaultHttpClient();
        HttpEntity inputEntity = null;
        HttpEntity outputEntity;
        HttpResponse response = null;
        try {
            System.out.println();
            inputEntity = new StringEntity("{"accountID": ":" + accountId + "," + "password": ":" + password + "}"");
            post.setEntity(inputEntity);
            response = client.execute(post, context);
            outputEntity = response.getEntity();
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return "Unable to Process your request. Please try again";
    }
}
```

Figure 3.6 shows code snippet to generate QR code by calling “get Code Web Service” using REST protocol.
Figure 3.6 Code Snippet for Calling Web Service to Generate QR Code

Figure 3.7 shows the code snippet to verify payment details by calling “Make Payment Web Service” using REST protocol.
Figure 3.7 Code Snippet for Calling Web Service to Verify Payment Code
4. FUNCTIONALITY OF THE APPLICATION

Overall functionality of application for this system is divided into number of features that are implemented. Modular approach for development process is observed and implementation of each feature is considered as an application. There are 3 different applications - E-commerce application for user login, product selection and taking the decoded information of QR-code for further payment process to place the order. Bank Application Server provides the web service to generate QR-Code. Android Application, user logs in into the App by giving his Account number and Password. These login details are verified at the Bank Application Server. If user logs in successfully then screen displays asking for merchant ID and Amount to which QR-code must be generated. If user has credit at the bank then QR-code is generated for further payment process.

4.1 Applications

The proposed system makes use of 3 different applications.

4.1.1 E-commerce Application

E-commerce application is the portal which makes varied products available to the consumers. Consumer logins and selects the items of his/her choice and proceed to the payment, wherein he is asked to enter the merchant ID and security code. Consumer enters the security code after scanning QR code generated from the android application. Payment could be successful only when consumer enters valid code.

4.1.2 Android Application

User logs in into the android application with his bank account number and password. In background Hash based Message Authentication Code (HMAC) is generated for both account number and password and transferred through a secure REST (representational state transfer)
web service to bank application for authentication. If user is authorized then he is asked to enter the merchant ID and amount. Based on merchant ID, account number, time stamp Quick Response Code (QR-Code) is generated which is valid only for certain period of time (15 minutes) in payment process.

4.1.3 Bank Application

This application acts as intermediate between android application and shopping application. Android application communicates with the bank application server using REST web service provided by the bank to verify the user login details (Account number, Password). These login details are de-hashed here to check user’s authorization with the bank. Bank application sends authorization status message to android application. After the login is successful based on the inputs (Account number, Merchant ID, Time stamp) given by the user QR-code is generated.

User Interface

The proposed system is visually shown in the form of a web site which is designed using JavaScript.

4.2 E-commerce User

E-commerce user is a user whoever uses the shopping application.

4.2.1 Login

In the Login page (as shown in Figure 4.1), the user is prompted to enter authentication details. Here the user enters his valid authentication details - username and password. If the details
given by the user are valid (this is verified using authentication process), then the user will be taken to his account details and can be able to access various features of the system. Else an error message is popped by asking the user to enter valid username or password.

Figure 4.1 Shopping Website Login Page

4.2.2 Home Page

Successfully logged in user is redirected to Home Page of the shopping website as shown in Figure 4.2.
4.2.3 Add items to the Cart

User can search the items with identification number of the product. If the user simply clicks search without entering the identification number, all the products are displayed. User can add the item to cart after selecting the quantity. Figure 4.3 shows the available products on the shopping website.
4.2.4 Cart items

User can view the items in his cart; delete items from the cart. User can also add some additional products to the cart by clicking add more product button and proceed to checkout. Figure 4.4 shows items added into the cart by the user.
4.2.5 Checkout Page

Checkout page prompts user to enter the Merchant ID and decoded QR-code to place the order as shown in Figure 4.5.
4.2.6 Payment Confirmation Page

Once the user's provides Merchant ID and Code after generating it from the android app, a payment success message is displayed on the screen confirming the order as shown in Figure 4.6.

![Payment Success](image)

Figure 4.6 Order Confirmation

4.3 Bank Application

User on logging in on the shopping site has a bank account to look after his online transactions. User is provided with different functionalities on the bank application.

4.3.1 Login Page

User can login into the bank application with his online bank account details used at the time of registering as shown in Figure 4.7.
4.3.2 Account Operations page

Logged in user can perform operations like creating a new account, Deposit, withdraw, get balance, and transfer amount and view report. Figure 4.8 shows the home page of the bank once user is logged in to the bank website.
4.3.3 Create account

User can create a new account in the bank website as shown in Figure 4.9.
4.3.4 Withdraw

User can withdraw amount from his account as shown in Figure 4.10.

![Figure 4.10 Withdraw Amount from User Account](image)

4.3.5 Deposit

User can add some money to his bank account as shown in Figure 4.11.
4.3.6 Check balance

User can check available balance in his account through the bank website as shown in Figure 4.12.
4.3.7 Transfer Amount

User can transfer the amount from one account to the other account as shown in Figure 4.13.

![Online Bank Transfer Amount](image)

Figure 4.13 Transfer Amount

4.3.8 View report

User can view history of transactions performed – withdrawal or deposits of money from his account along with transaction date and time as shown in Figure 4.14.
4.4 Android Application

User makes use of the proposed system’s android application to generate a QR code which he uses for shopping.

4.4.1 Login Page

User logs in to the bank application provided by the bank with his username and password as shown in Figure 4.15.
4.4.2 Input Merchant ID and Amount

Once the user successfully logs into the application he needs to provide the merchant id and amount of the items user purchasing in the shopping website as shown in Figure 4.16.
4.4.3 QR Code Generation

Once the user submits input merchant id and amount a QR code is generated and displayed on the android application as shown in Figure 4.17.
4.4.4 Scanning QR Code to get Verification Code

User make use of free QR code scanner and scans the QR code to generate verification code which he uses at the shopping site to make a payment as shown in Figure 4.18.
Figure 4.18 Verification Code on the User Device
5. TESTING AND EVALUATION

Software testing is the process or set of steps to assess a software item to study variations between given input and expected results. Product quality is evaluated with software testing. Software testing is a process that is carried out during the development process. It can be termed even as verification and validation process.

Success of a system is known with software testing. Testing of the system make a logical assumption that if all parts of the system are right, the objective will be accomplished effectively. In the testing process the actual system is tired in an organization and accumulates errors from the new system by taking the activity to rectify them. System testing is the phase of implementation, which is planned to ensure that the system meets desires accurately and productively. The front-end and back-end connection are tested to verify that the new system meets expectations in full effectiveness.

The process of testing is considered as an important one as it reveals bugs from the system. For achieving this methodology legitimate information is given to the system. So the user ought to have more regard for information. It is key to give right inputs for proficient testing. Insufficient data or non-testing lead to lapses that may appear few months afterward. The accompanying issues can produce deficient testing situations.

- Time delay arises between the cause and appearance of the problem.

- The effect of the system errors can be observed on the files and records within the system.

The testing procedure concentrates on logical intervals of the product guaranteeing that all the announcements have been tried on the function intervals, where tests are directed to uncover
errors and ensure that the defined inputs will produce actual results that agree with the required results. Testing must be carried out utilizing the two normal steps Unit testing and Integration testing.

**Unit testing**

Unit testing is a software testing system by which singular units of source code, sets of one or more machine project modules together with related control information, utilization methods, and working strategies are tried to figure out whether they are fit for utilization.

**Integration Testing**

Integration Testing is the stage in software testing in which software modules are joined and tested as a group. It happens after unit testing and before validation testing.

**5.1 Test Cases**

Testing in the proposed system is done by making the procedure level testing first. This is done by giving improper inputs; the errors occurred are noted and eliminated. It implements tested error-free system into real-life environment and make necessary changes, which runs in an online fashion. Following are the test cases used to test the working of the proposed project:

**Test case 1: Validating Username and Password Fields**

The user will be given username and password to login to the E-commerce application site. If the user tries to login to the site with incorrect or no password provided, then the system shows and error message and he is not able to login to the site (as shown in Figure 5.1).
Test Case 2: Account number and Password validation at Android App

Figure 5.2 shows the screen shot where user can login into Android App by giving Account number given by the bank and password. User must give valid account number and password which is given at the Bank Application.
Test case 3: Successful Login

On successful login to the Android App, user is prompted to enter merchant ID and amount based which QR-code is generated (as shown in Figure 5.3).
Figure 5.3 Screen Prompting for Merchant id and Amount after Successful Login

Test case 4: QR Code Generation

Figure 5.4 shows the screen shot of successful generation of QR-Code based on three attributes (Merchant ID, Account number, Time Stamp).
Test case 5: Inserting successfully generated Verification Code into database

Figure 5.5 shows the screen shot of the successful insertion of row into Bank Application database. On successful generation of the QR-code, text encoded in the QR-Code is saved in the bank database along with payment ID, Account number, Amount, Merchant ID, Time Stamp.
Test case 6: Scanning QR Code to get Verification Code

Figure 5.6 shows the screen for decoded QR-Code with the help of QR-code scanner.
Test case 7: Verifying Placing of Order

Figure 5.7 shows the screen shot of success placing of order when valid merchant ID, decoded QR-code is given at the payment screen in shopping application and the customer has sufficient funds in his account. Figure 5.8 shows the screen where the user can enter the merchant ID (same as used in generating code) and decoded code.
Figure 5.7 Payment Screen with Input Merchant ID and Code

Figure 5.8 Payment Confirmation Page
Test case 8: Deleting Successful Payment Transaction from Database

Figure 5.9 shows the screen shot of automatic deletion of row from the Bank Application database once the code has been used at shopping and payment is successful.

![Database Rows after Deletion of Payment Success Transaction](image)

**Figure 5.9 Database Rows after Deletion of Payment Success Transaction**

Test case 9: Payment Success Transaction on User Transactions Report

Figure 5.10 shows the screen shot of Report of the transactions performed by the user. It shows the latest payment transaction in the report.
Figure 5.10 Transactions Report in Online Bank

Test Case 10: QR Code Generation and scanning for different User

Figure 5.11 shows the QR code generated for a different user with different merchant and amount. The resulting QR code looks different for each transaction.
Figure 5.11 QR Code for a Different Transaction

Figure 5.12 shows the corresponding code after scanning the QR code generated on the android device.
Figure 5.12 Scanning QR Code to get Verification Code for Different User
6. CONCLUSION AND FUTURE WORK

Proposed project provides all the users a platform by implementing a secured payment system for M-commerce applications making use of QR code with HMAC encryption technique. Such a payment system working on proposed techniques can provide consumer data privacy and prevents customer related frauds. The method is concerned with prevention of identity theft, merchant related frauds, and customer data security. With the use of REST Web Service, performance of the system is improved with instant messaging. All the applications are tested as shown in the testing section.

There can be a security problem with other users getting the QR code by any means like image capture and can make use of the same code for their personal transactions with the bill having the same items. This can be overcome by adding item number as an input to the QR encoder.

There is huge scope for enhancement in this project. In case mobile is lost/stolen, services must be disabled remotely. The efficiency of the application with REST can be tested with the help of different tools. User can be notified with the left over time frame on the screen to use the code. In this security mechanism, a consumer should be able to preset the effective duration boundary of using the service, or preset the location of a supermarket where the consumer wants to use the service.
BIBLIOGRAPHY AND REFERENCES


The following code implements generation display activity of QR code on android application.

The credit for the source of this code goes to my Sahasra Yellu.

**DisplayCodeActivity.java**

```java
package com.example.payment;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Bitmap.Config;
import android.graphics.Color;
import android.net.Uri;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.google.zxing.BarcodeFormat;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.qrcode.QRCodeWriter;

public class DisplayCodeActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_code);
        TextView displayTextView = (TextView) findViewById(R.id.displayCodeText);
        ImageView imageView = (ImageView) findViewById(R.id.displayCodeImage);
        String code = getIntent().getStringExtra("code");
        displayTextView.setText(code);
        try {
            generateQRCode_general(code, imageView);
        } catch (WriterException e) {
            e.printStackTrace();
        }
    }

    private void generateQRCode_general(String code, ImageView imageView) throws WriterException {
        BitMatrix matrix = QRCodeWriter().encode(code, BarcodeFormat.QR_CODE, 200, 200);
        Bitmap bitmap = Bitmap.createBitmap(200, 200, Config.ARGB_8888);
        for (int x = 0; x < 200; x++) {
            for (int y = 0; y < 200; y++) {
                if (matrix.get(x, y)) {
                    bitmap.setPixel(x, y, Color.BLACK);
                } else {
                    bitmap.setPixel(x, y, Color.WHITE);
                }
            }
        }
        imageView.setImageBitmap(bitmap);
    }
}
```
private void generateQRCode_general(String data, ImageView img) throws WriterException {
    com.google.zxing.Writer writer = new QRCodeWriter();
    String finaldata = Uri.encode(data, "utf-8");
    BitMatrix bm = writer.encode(finaldata, BarcodeFormat.QR_CODE, 150, 150);
    Bitmap ImageBitmap = Bitmap.createBitmap(150, 150, Config.ARGB_8888);
    for (int i = 0; i < 150; i++) {  // width
        for (int j = 0; j < 150; j++) {  // height
            ImageBitmap.setPixel(i, j, bm.get(i, j) ? Color.BLACK: Color.WHITE);
        }
    }
    if (ImageBitmap != null) {
        img.setImageBitmap(ImageBitmap);
    } else {
        Toast.makeText(getApplicationContext(),
                getResources().getString(R.string.qrimage_error),
                Toast.LENGTH_SHORT).show();
    }
}