CHALMERS | UNIVERSITY OF GOTHENBURG

Online Based Authentication and Secure Payment Methods for M-Commerce Applications

Master of Science Thesis in the Programme Secure and Dependable computer systems

TAIWO DAYO AJAKAIYE
KARL SENANU KUDZO KRAUSE

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, July 2011

Online Based Authentication and Secure Payment Methods for M-Commerce Applications

Taiwo D. Ajakaiye
Karl S. K. Krause

Examiner: Tomas Olovsson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden July 2011

Preface

This thesis work was done in fulfillment of the requirements for a Swedish master’s degree at both Chalmers University of Technology and Gothenburg University. It contains work that has been done from January to June 2011. This work was solely carried out by us and builds on findings from other related studies. We acknowledge the contribution of the authors of these related studies to our work and the research community in general.

Undertaking this thesis work has been challenging because we had to gather and study information from different areas. This challenge turned out to be what we needed as we have acquired useful knowledge about these areas which will help us in our future careers.

We would like to thank our patient and helpful supervisor, Dr. Tomas Olovsson for his guidance throughout the entire period of this thesis work. We would also like to thank our families for their support during this period.

Abstract

The widespread use of the Internet has contributed enormously towards the growth of e-commerce. Technological advances in mobile phones (e.g. Smartphones) have also made it possible to carry out e-commerce via mobile phones (m-commerce). M-commerce involves the use of mobile devices such as mobile phones and PDA's in carrying out electronic transactions. Applications in this domain range from normal information consumption to high security financial electronic transactions. Just like e-commerce, the security of m-commerce applications is critical, especially when it involves applications that deal with user sensitive data such as credit cards details, medical details etc.

This thesis introduces a platform (e.g. Symbian, iPhone OS and Android OS) independent way of carrying out secure authentication from a mobile device. This was done by designing, prototyping and evaluating a platform-independent authentication method called OSP. An investigation and prototype implementation of how m-commerce applications can include secure payment capabilities was also presented. Questions that were answered in this study include; how do we verify that a user is who he claims to be and how do we carry out financial transactions in a secure way.

Keywords: OTP, PCI DSS, Platform, SMS, SSO

TABLE OF CONTENTS

1. Introduction

Mobile-commerce, also known as the next generation e-commerce, can be defined as any electronic transaction or interaction conducted using a mobile device such as a mobile phone or personal digital assistant (PDA) [1]. Carrying out electronic based services is becoming quite common via mobile devices. This is due to the emergence of Smartphones (mobile phone + PDA) with reasonable computing resources e.g. mini browsers, security primitives (certificates, encryption etc.) and so on. The fact that our mobile devices are always with us and rarely turned off makes m-commerce an attractive field for businesses. Thus, m-commerce has become a business model that serious enterprises can no longer afford to neglect.

Smartphones has opened up new opportunities for enterprises within m-commerce and it has also provided users an easy way of carrying out online transactions. However, the security issues that arise with the growth in this field cannot be neglected. For example, how does one ensure that participants in an m-commerce transaction are who they claim to be (authentication)? Also, how does one support secure financial transactions in m-commerce businesses? These are the issues this paper will be addressing in the coming chapters.

1.1. Background

The following history of Mobile commerce was adopted and freely interpreted from Wikipedia [2].

"Mobile commerce was born in 1997 when the first two mobile-phone enabled Coca Cola vending machines were installed in the Helsinki area in Finland. The machines accepted payment via SMS text messages. The first mobile phone-based banking service was launched in 1997 by Merita Bank of Finland, also using SMS." "Mobile-commerce-related services spread rapidly in early 2000. Norway launched mobile parking payments. Austria offered train ticketing via mobile device. Japan offered mobile purchases of airline tickets."
"PDAs and cellular phones have become so popular that many businesses[specify] are beginning to use mobile commerce as a more efficient way to communicate with their customers. In order to exploit the potential mobile commerce market, mobile phone manufacturers such as Nokia, Ericsson, Motorola, and Qualcomm are working with carriers such as AT&T Wireless and Sprint to develop WAP-enabled smartphones. Smartphones offer fax, e-mail, and phone capabilities."
"Since the launch of the iPhone, mobile commerce has moved away from SMS systems and into actual applications."

Today, M-commerce applications can be used for different services such as Mobile ticketing, Mobile vouchers/coupons/loyalty cards, content purchase and delivery, location-based services, information services, mobile banking, mobile store front, mobile brokerage, auctions, mobile marketing and advertisement etc.

1.2. Problem statement

M-commerce is used for a variety of products and services ranging from basic applications such as mobile marketing to high security mobile payment applications. Mobile payments are now becoming a widely used medium for carrying out financial transactions. Ericsson, a telecommunication giant and a major player in the mobile payment industry, estimates that the mobile payment market will yield a profit of 20 billion Euros by 2015 and a turnover of 600 billion Euros [3]. A mobile payment application must provide means for carrying out secure authentication and financial transactions.

Authentication and secure payment is a major security issue when it comes to carrying out mobile financial transactions remotely. Developers of such applications are always faced with questions such as; how do we ensure that the person requesting to carry out a financial transaction is who he claims to be? How do we carry out secure financial payments from a mobile device? There are several mobile payment applications (see chapter 4) providing some form of authentication/payment function, and installed on various Smartphones (iPhone, BlackBerry, Android phone, etc.) today.
However, most existing solutions are platform dependent and each has its unique implementation for secure authentication and payment. For example, a solution implemented in java for an Android phone will have to be re-implemented in Objective C in order to be used on an iPhone due to language restrictions. Another question which is obvious at this point is; how do we implement a method for secure authentication or payment which is compatible with all Smartphones?

1.3. Purpose

The objective of this thesis work is to propose a secure platform-independent authentication and payment method for m-commerce applications. To achieve this, the following research questions were looked into:

1. What are the security threats that are currently faced by m-commerce systems?
2. What are the necessary security requirements that must be met by a platform-independent authentication and payment system?
3. What are the current authentication methods/solutions available?
4. What are the current payment methods/solutions available?

1.4. Organization of thesis

Chapter 2: Previous studies in the area of authentication and payments are presented in this chapter. Here, we illustrate the relation and relevance of our studies to previous related work that have been done in the research community.

Chapter 3: This chapter describes the method that what adopted during this thesis work. It gives an overview of the different stages involved in the research process. These stages include data collection, analysis, validation and evaluation.

Chapter 4: There are several authentication and payment products/systems in use today. In this chapter, we investigated the security and architecture used in these systems, with the aim of understanding how a secure platform independent authentication and payment method can be designed.

Chapter 5: This chapter describes the various threats that are currently faced by the mobile community. In this section, we have created a threat model which helped us to identify and understand the possible threats that are faced by m-commerce applications and ways of mitigating them.

Chapter 6: A method for carrying out platform independent authentication and payment via mobile phones is described in this chapter. This method was influenced by previous studies and current existing products. A prototype implementation of these methods as part of an m-commerce application is also presented.

Chapter 7: An evaluation of the authentication method and the two payment approaches presented in chapter 6 are presented here.

Chapter 8: Conclusions and future work for this thesis work are presented in this final chapter.

2. Related work

We have studied various research works that have been done in the area of authentication with a focus on mobile devices. The studies were conducted on how a secure financial transaction is carried out. The essence of this section of the thesis work is not to only cite some of the important results that were obtained, but to also see their relevance to the research problem.

2.1. Two-factor authentication

Fadi Aloul, S.Z and Wassim El-Hajj addressed the problem of carrying out secure authentication via mobile devices [4]. They proposed the use of a two-factor method of authentication which makes use of something you have (mobile phone) and something you know (one-time password). The method involves the use of a mobile phone for the generation of a one-time password (OTP), or the use of SMS in retrieving a remotely generated OTP from a server. Results showed this two-factor authentication method to be a more secure form of verifying users than traditional password systems. They also showed how this method can be used to eliminate the problems that one-factor authentication methods (e.g. passwords) face. Their method provides a cheaper alternative to current two-factor authenticating systems (tokens, cards) widely used today. It does this by making use of the users' mobile phone for OTP generation, therefore eliminating the extra cost involved in purchasing additional tokens and cards.

2.2. Single sign-on system

When a user has several user accounts with different service providers, he would need to remember and use different user-ids and passwords while connecting to those accounts. The single sign-on (SSO) mechanism relieves users of having to undergo unnecessary multiple authentications for each service. In the paper titled "The study of multi-level authentication-based single sign-on system" [5], the authors pointed out that systems which have a single sign-on experience, assign the same level of security to each service providers within a distributed network. This according to the authors is not really secure. If one of the service provides within the distributed network becomes compromised, then the single sign-on experience will tend to pose a threat to other service providers that require a higher level of security. The authors proposed a multi-level authentication mechanism (MLA-SSO), in which different security levels that are required by different service providers can be automatically analyzed and assigned by a server. This improves the flexibility, performance and security of the network.

2.3. Strong authentication

In the Research carried out by Do van Thanh et al [6], the authors introduced the concept of using the mobile phone device as a token for authentication instead of a traditional hardware token. The overall cost of using an additional device to carry out authentication is very high for organizations that have to maintain thousands of tokens. Also, users will have to carry around hardware tokens whenever they need to carry out authentication on the fly. The authors proposed the use of mobile phones as a replacement for hardware tokens as a way of solving the various issues described above. They also discussed various ways that the mobile phone could be used as device tokens in a secure two-factor authentication process.

2.4. Social Authentication

A study [7] carried out by two researchers from McGill University in Canada proposed an additional authentication factor to an already existing two-factor authentication (see 2.1). The authors Muthucumaru Maheswaran and Bijan Soleymani suggested that this additional authentication factor (someone you know), should be highly dependent on the social network the particular individual belongs to. That is, every individual who uses a mobile device as an authenticator needs to belong to a particular social network. In the case when a member of that particular network has lost his secret credentials or the mobile device, that person will require someone to vouch for him. During the process of vouching for someone, the secret credential is not sent to the voucher but to the individual who needs to be vouched for. This maintains the secrecy and privacy of the credentials and thus adds an additional level of security to the already existing system.

2.5. ECC-based Wireless Local Payment Scheme

Gianluigi Me and Maurizio A. Strangio conducted a study [8] which was driven by the problem of insecurity involved in the use of mobile phones for localized financial transactions. They studied the security issues present in mobile phone wireless communication technologies (Bluetooth, IrDA and Wi-Fi etc.) such as Blue bugging, Bluesnarfing, Blue jacking etc. [9, 10]. They then presented a local payment scheme via mobile phones based on a Public key infrastructure (PKI). Security was ensured in the scheme by using secure cryptographic primitives and a standard compliant key agreement.

The scheme covers the secure communication between a payer, payee and their respective banks. A payer requests for an electronic check from his bank. The bank securely sends the electronic check with the payers' signature bound to it. Payer and payee establish a secure connection, exchange public key certificates, and agree on a secret session key used for authentication. Payer on receipt of an encrypted invoice from payee returns an equally encrypted and digitally signed e-check. Payee signs, submits the check to his bank and receives a message on whether the check was accepted or not. The result of this study was a prototype based on Bluetooth protocol stack and Elliptic Curve based cryptographic primitives.

2.6. Online payment service providers

Alan D. Smith conducted a study [11] on the use of online payment service providers such as PayPal. This was done by conducting interviews with 190 working adults from 18 different companies. The interview took place within the metropolitan area of Pittsburgh US, over a period of 4 months. The author pointed out certain disadvantages of using online payment service providers. These disadvantages include:
"*PayPal is not a bank and, therefore, is not subject to regulatory and internal audits and the funds are not federally insured*" [11]
"*PayPal relies heavily on security and service software that have showed to be vulnerable in the past. For example*, "*In the summer of 2000, PayPal and other online payment services were attacked by Russian hackers*" [11]

However, based on the result from the survey conducted in the study, it was also shown that online payment service providers are widely used and have tremendous potential for continued growth.

2.7. Summary

Several studies have been conducted in the area of authentication and payments. Some studies [4 - 7] talked about the different techniques that can be used to build a secure authentication method. These techniques include two-factor, single sign-on, strong and social authentication. However, most research has been focusing on platform (operating systems e.g. iPhone OS, Android etc.) dependent authentication solutions, while less attention have been paid on platform independent solutions. One may conclude from this trend that platform dependent solutions are more secure. With this study, we showed that using a platform independent authentication method is adequate without compromising the security of the authentication solution. For example, using multiple factors increases the security of the authentication process. This concept of using multiple factors to strengthen the authentication method is supported in several studies [4, 6, and 7]. It is also widely used in current existing authentication systems (see chapter 4).

Two other studies titled ECC-based Wireless Local Payment Scheme [8] and online payment service providers and customer relationship management [11] present two different ways to make financial transactions. The first study presented a local payment scheme via mobile phone based on public key infrastructure. This does provide a solution for making transactions via mobile phones; however, the solution is platform dependent and will not fulfill the purpose (see chapter 1.3) of this study. In the second study, the authors identified online payment services providers (e.g. PayPal) to be widely used, and with tremendous potential for continued growth. Online payment service providers can provide platform independent solutions; therefore our study will evaluate different ways in which such services can be integrated into m-commerce applications.

3. Methodology

This chapter describes the research approach used in this study. It involved researching previous studies that were conducted in the area of authentication, as well as reviewing what underlining techniques current existing authenticating systems use. In addition various policies and regulations in the payment industry and current existing payment methods were studied in order to propose a payment method for m-commerce applications that is both secure and platform independent. This approach was adopted in order to clearly understand and define a solution to the research problem. The diagram below depicts an overview of the research method.

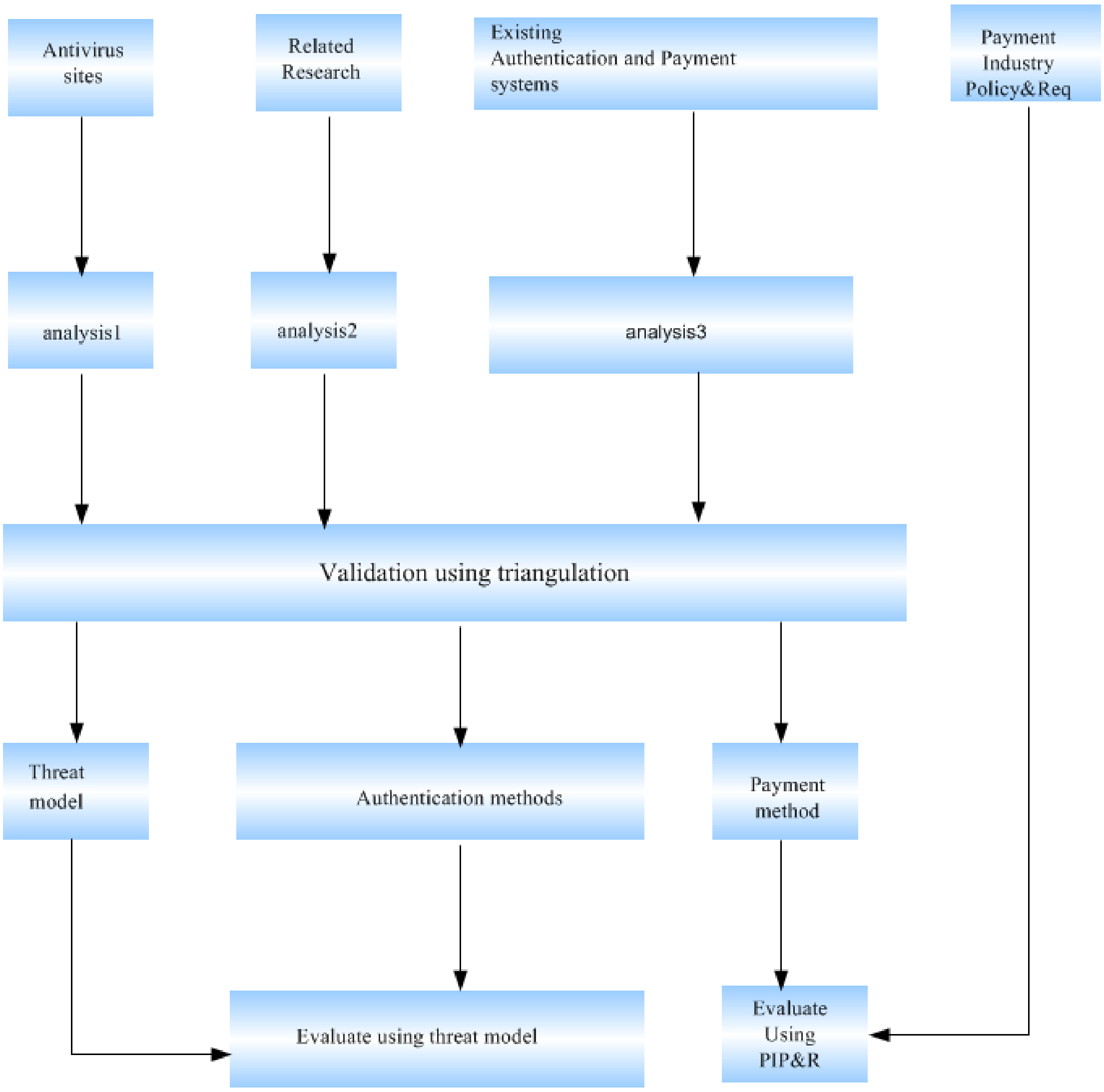

Antivirus sites
Related Research
Existing Authentication and Payment systems
Payment Industry Policy&Req
analysis1
analysis2
analysis3
Validation using triangulation
Threat model
Authentication methods
Payment method
Evaluate using threat model
Evaluate Using PIP&R

Figure 1: Overview of thesis methodology

3.1. Data Collection

The process of gathering data was done from a number of important sources. Some of the sources included antivirus websites, research libraries etc. Most of the data collected were in the form of written text, PowerPoint presentations and videos, while the collection type was in the form of documents and media files. See the table below for more details.

	Data source	Type of data	Data form	Collection type
1	Antivirus/security sites [12 – 14,31,34]	Threat documentation	Written text	Documents
2	Related research [4 – 7, 8, 11]	Articles	Written text	Documents
3	Existing authentication systems [16-20]	Videos, Articles and Demos	Written PowerPoint presentations, Written text and media formats	Documents and media files
4	Existing payment systems [21-23]	Videos, Articles and Demos	Written PowerPoint presentations, Written text and media formats	Documents and media files
5	Payment industry policy and requirement (PIPR)[53]	Documentation	Written text	Documents

Table 1: Data Collection

3.2. Analysis

The collected data from data source 1 to 4 was analyzed and refined with the aim of obtaining useful information for the research problem at hand. Data source 5 was on the order hand used directly as criteria for evaluating the payment approaches proposed in chapter 6.2.

Data source 1

Antivirus/Security sites carry out one primary function, and that is developing programs or providing security information that helps protect computer users against viruses and other computer threats. Antivirus companies maintain up to date documentation about various computer and mobile phone threats, thus this was a good source for gathering threats that exist against mobile devices. Information about mobile threats was obtained from the following antivirus companies; F-secure [12], Symantec [13] and McAfee [14] and other security sites [31, 34].

Analysis 1

Various categories of threats that occur in mobile devices were collected. The threats where studied to understand how they infect and propagate to other mobile devices. Ways of mitigating it was also researched and documented (see chapter 5.4).

Data source 2

Various related studies have been conducted in the area of authentication and payment via mobile devices. Some studies talked about using two-factors based authentication methods due to the low level of security provided by one-factor authentication methods. Other studies propose authentication techniques such as single sign-on (see chapter 2.2), social authentication (see chapter 2.4), and so on. Due to the payment aspect of the research problem, studies such as “ECC-based wireless local payment scheme” and “Online payment service providers and customer relationship management" were also studied. All these studies were chosen because they brought a great deal of knowledge about how similar research problems were approached in the past.

Analysis 2
Various related work was studied to understand the best way to handle our research problem; how to carry out secure platform-independent authentication and payment via mobile phones. This involved learning about the various research methodologies and technologies used.

Data source 3 & 4
Existing authentication and payment systems/products implement various authentication and payment methods. This makes it possible to investigate the underling methods and techniques used by secure authentication and payment systems.

Analysis 3 & 4
The architecture and technologies used in the existing authentication and payment systems were analyzed with the aim of exploring how to develop a secure platform-independent authentication and payment method for m-commerce applications (see chapter 4)

3.3. Validation

The results of all the analyses conducted on data obtained from the antivirus/security sites, related research, existing authentication and payment systems/products (see 3.2) were validated by conducting Triangulation (see below).

Triangulation
Triangulation [15] is a method of validating data collected from different data sources especially when it comes to exploratory studies such as this work. Thus, this method was adopted based on its suitability to this study. For example, information obtained from one antivirus site was validated by examining contradictory and agreeable information from other antivirus sites. The information is valid in the case where agreeable sites are more than contradictory sites and vice-versa.

3.4. Evaluation

A separate evaluation was carried out on the proposed authentication method and the payment approaches.

Authentication method
After analyzing the threats obtained from the various antivirus/security sites, we were able to come up with a threat model (see chapter 5). The threat model was used to understand how potential threats can compromise a system from an attackers point of view. In our case, the threat model was then used to evaluate the security of the authentication method, by investigating how well it mitigates attacks from the threat model.

Payment methods
On analyzing the existing systems, two approaches where payment can be made from mobile devices without disregarding the platform independent requirement were

identified (see chapter 6.2). The security of each approach was evaluated by analyzing it against standards and polies in the credit card industry (see chapter 7.2). The card industry standards and requirements were used because it is legally required for any m-commerce application that wishes to transmit, store and process credit card details.

4. Existing systems

In the second Chapter, we reviewed the studies that have been carried out in the area of authentication and performing payments in mobile and other devices. We were able to identify various methods of carrying out authentication and secure payments, and most of these methods are currently used in existing commercial systems/products today. Thus, this chapter documents our analysis of these existing systems with the aim of finding out the most secure and realistic way to authenticate and perform payments via mobile phones.

4.1. Authentication systems

During the explorative process, four different kinds of systems were identified in the market that claims to offer a secure authentication process.

4.1.1. PayPal password login

This is a system which makes use of a one-factor authentication. The strength of this system lies in the underlying fact that users are mandated to choose strong passwords (e.g. combination of different data types, restriction on short passwords etc.). Although a strong password can be chosen, such one-factor systems have been shown to be vulnerable to attacks such as password cracking and is not considered secure enough.

How it works
PayPal [16] ensures that users register their personal details (e.g. password) which are then used for verification during the login process. A PayPal user intending to purchase goods from an m-commerce store is redirected to PayPal where he is asked to supply his User ID and Password. PayPal verifies the authenticity of the credentials entered by the user. It allows the user to proceed and finalize the payment.

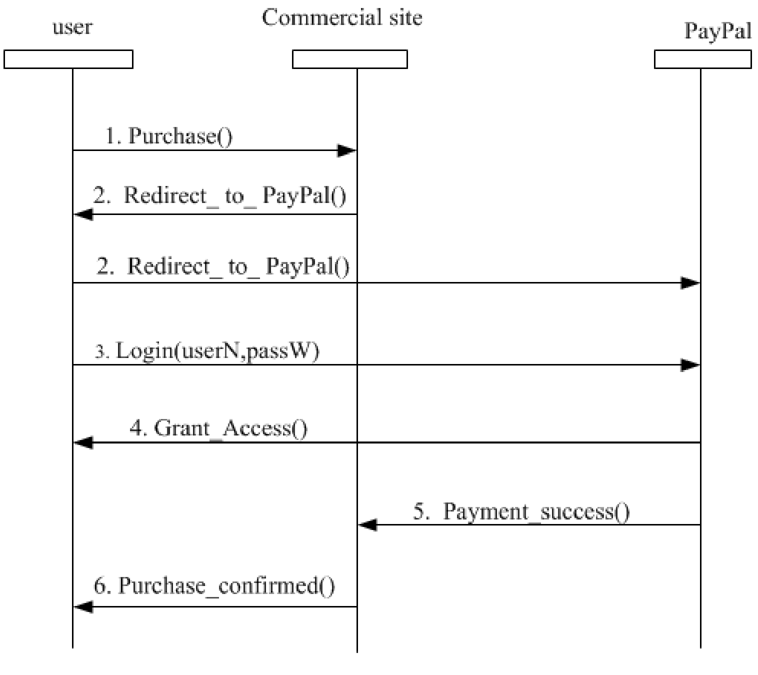


Figure 2: PayPal authentication process flow

4.1.2. WebSEAL Single Sign-on

The WebSEAL authentication system [17] relies on a single sign-on mechanism to give subscribers premium access to services through a portal on their mobile devices. All subscribers of a particular telecom operator are already authenticated by their trusted telecom gateway (for example, WAP or i-mode gateway). At the same time when users want to access services through the portal on their mobile devices, they would need to also provide another form of credential. This therefore prolongs the process of authentication for mobile clients. WebSEAL Single sign on helps to eliminate the inconveniences that are caused by authentication process. In the case where a WAP gateway is used, WAP traffic is converted to HTTP traffic by the gateway and login credentials are embedded into to HTTP headers. The WAP gateway makes use of a Radius server to perform client authentication. For a successful user authentication, client's personal details (e.g. telephone number, MSIDSN) are extracted from the SIM card.

How it works

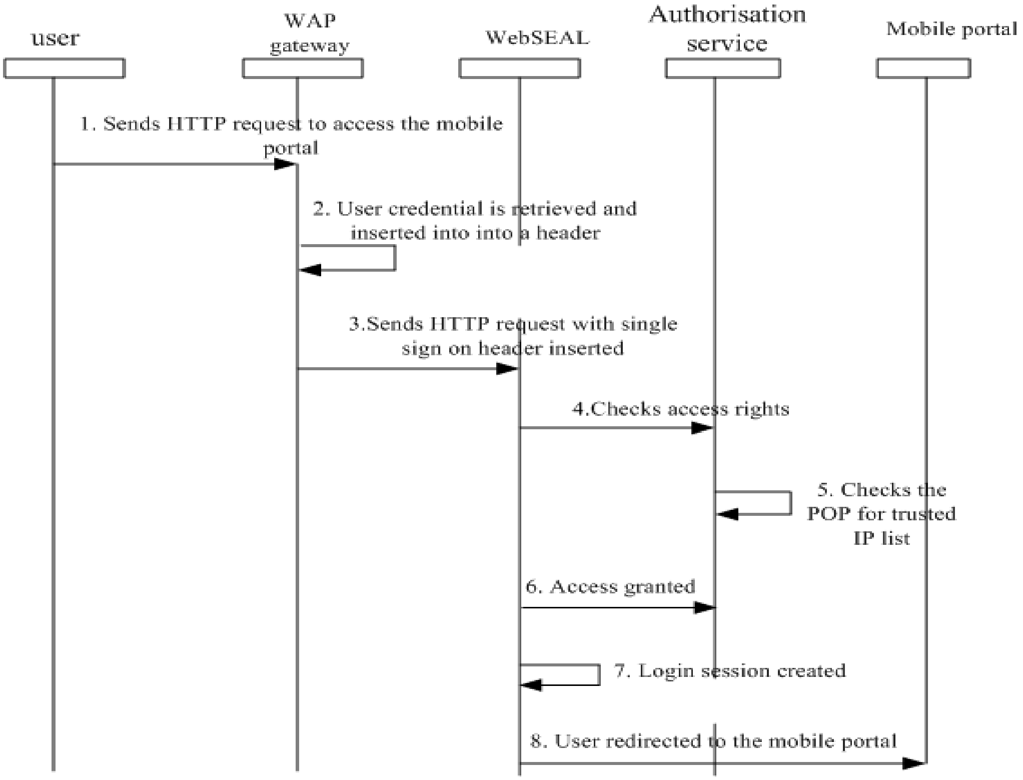

user
WAP gateway
WebSEAL
Authorisation service
Mobile portal
1. Sends HTTP request to access the mobile portal
2. User credential is retrieved and inserted into into a header
3.Sends HTTP request with single sign on header inserted
4.Checks access rights
5. Checks the POP for trusted IP list
6. Access granted
7. Login session created
8. User redirected to the mobile portal

Figure 3: WebSEAL authentication process flow

When a user wants to access a premium service or resource through a single portal or application on his mobile device, the user sends authentication requests to the WAP gateway. The user's information or credentials are retrieved and inserted into an HTTP header. HTTP requests are sent to the WebSEAL with the single sign-on HTTP header inserted into the streams of data. When WebSEAL receives the request packets, it retrieves the login credentials of the mobile clients and checks if a login session exists for that particular user. In the case when the user does not have a particular login session, WebSEAL then checks with the necessary authorization service if the user is within the trusted IP list. The trusted IP list includes all subscribers from a particular telecom gateway, and in that case, once authenticated by the trusted gateway, mobile clients IP is automatically included in the list of trusted IPs. If the client's IP is trusted, access is granted to the particular service and a login session is created. The user is then redirected to the mobile portal to enable the user to have access to the requested service.

4.1.3. AcrotOTP Mobile

This authentication system [18] employs two-factor authentication by using a mobile device as the hardware token. The mobile device possesses an OTP generator module that generates random one-time passwords. The AcrotOTP system is made up of a container which holds the Acrot keys used for generating Random OTP. The Acrot keys are protected by cryptographically strong encryption. The system is made up of an authentication server which is used to verify that the OTP entered by the user is indeed a valid token.

How it works

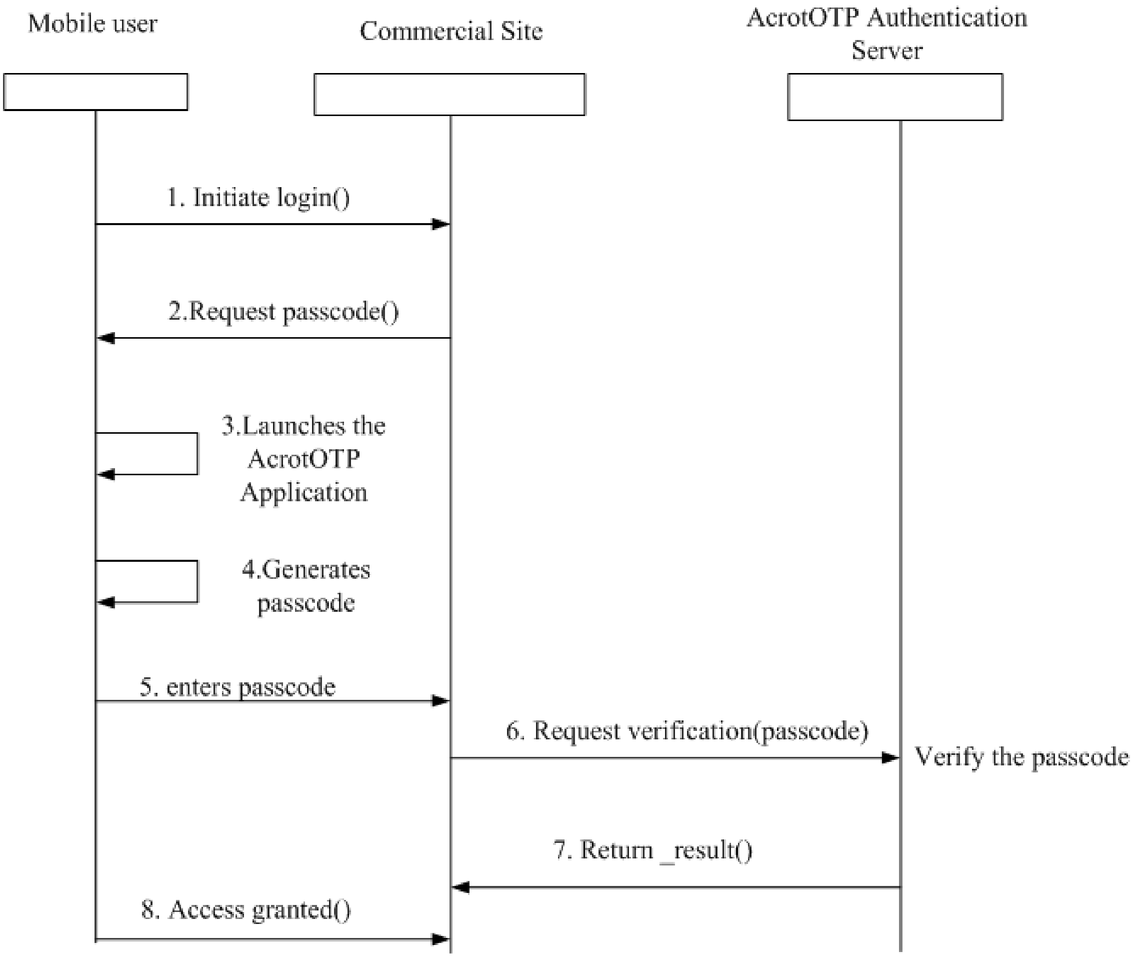


Figure 4: AcrotOTP authentication process flow

User initiates login process with a commercial site. The site prompts user to enter a passcode. User launches AcrotOTP application on his mobile phone and subsequently generates a passcode with the application by entering his pin code. User enters the generated passcode into the commercial site and the entered passcode is verified with the AcrotOTP authentication server. The user is either granted or denied access based on the results returned from the authentication server.

4.1.4. Accumulate Mobile everywhere

The Accumulate [19] Mobile Everywhere (ME) authentication system makes use of the mobile device as hardware token. This is more secure than hardware token used with online banking because the mobile device is always in possession of the customer. In order for such a system to be successful, it makes use of two different authentication parties. The ME mobile client application generates the OTT and the ME transaction server works as an authentication server that verifies the validity of the OTT entered by the user. This system ensures that the OTT can only be used once to validate a particular authentication session. A strong 2048 bits RSA encryption is used to ensure that the OTT is not compromised during the exchanges.

How it works

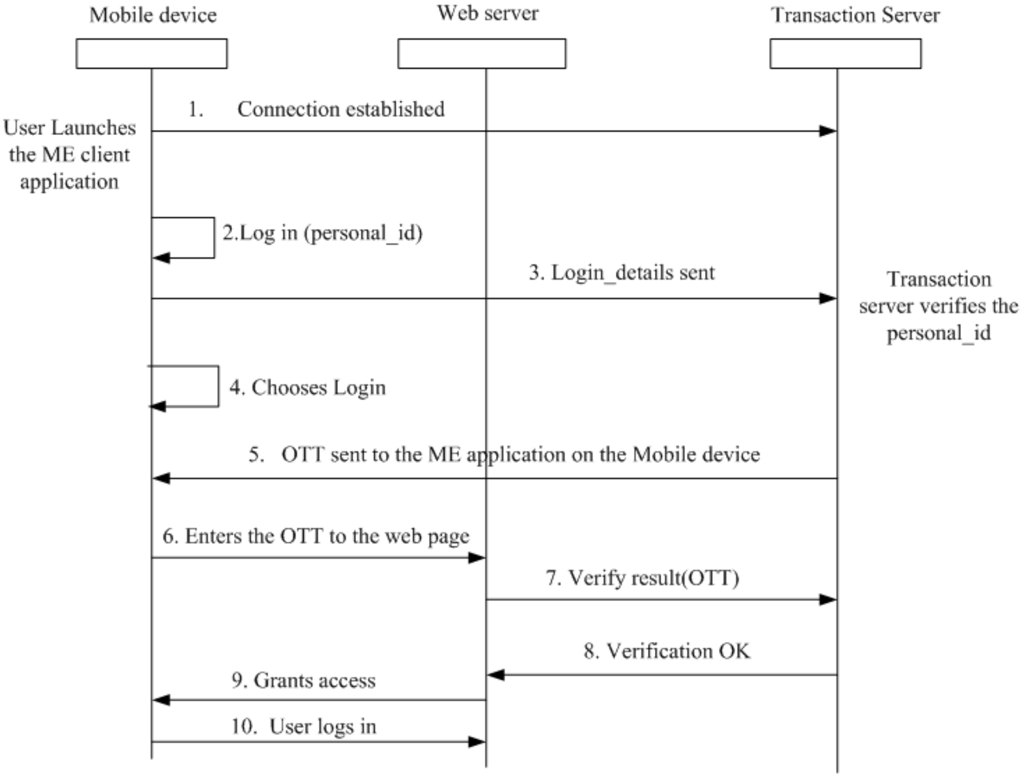

Mobile device
Web server
Transaction Server
User Launches the ME client application
1. Connection established
2.Log in (personal_id)
3. Login_details sent
Transaction server verifies the personal_id
4. Chooses Login
5. OTT sent to the ME application on the Mobile device
6. Enters the OTT to the web page
7. Verify result(OTT)
8. Verification OK
9. Grants access
10. User logs in

Figure 5: Accumulate ME authentication process flow

A user accesses his web account by launching the Accumulate ME client application. Once the application is launched, a connection is automatically established between the application and the Accumulate Transaction server. The user then logs into the ME client application using his personal identification number. The client’s login detail is sent to the transaction server for verification. If the details are found by the server to be authentic, the user is allowed to select “choose login” option on the Accumulate ME client application. A one-time ticket (OTT) is sent directly from the Transaction server to the ME client application. This enables the user to then continue the financial transaction by entering his OTT into the commercial web page. The webpage server verifies the OTT entered with the Accumulate Transaction server. Finally, the user is allowed to log in if the returned result is OK.

4.1.5. Authentify Out-of-Band

This system [20] provides a multi-level authentication mechanism which includes "something the user has”, “something the user is" and "something the user knows". Out-of-band method requires the customers to make a call to confirm a transaction. This ensures that a transaction can be terminated if any fraudulent activity is discovered during the process. This prevents Man-in-the browser attack which is an attack that is always targeted against the verification process of a financial transaction.The Authentify OOB system also makes use of two separate communication channels; one channel through the Internet and another through the mobile network.

How it works

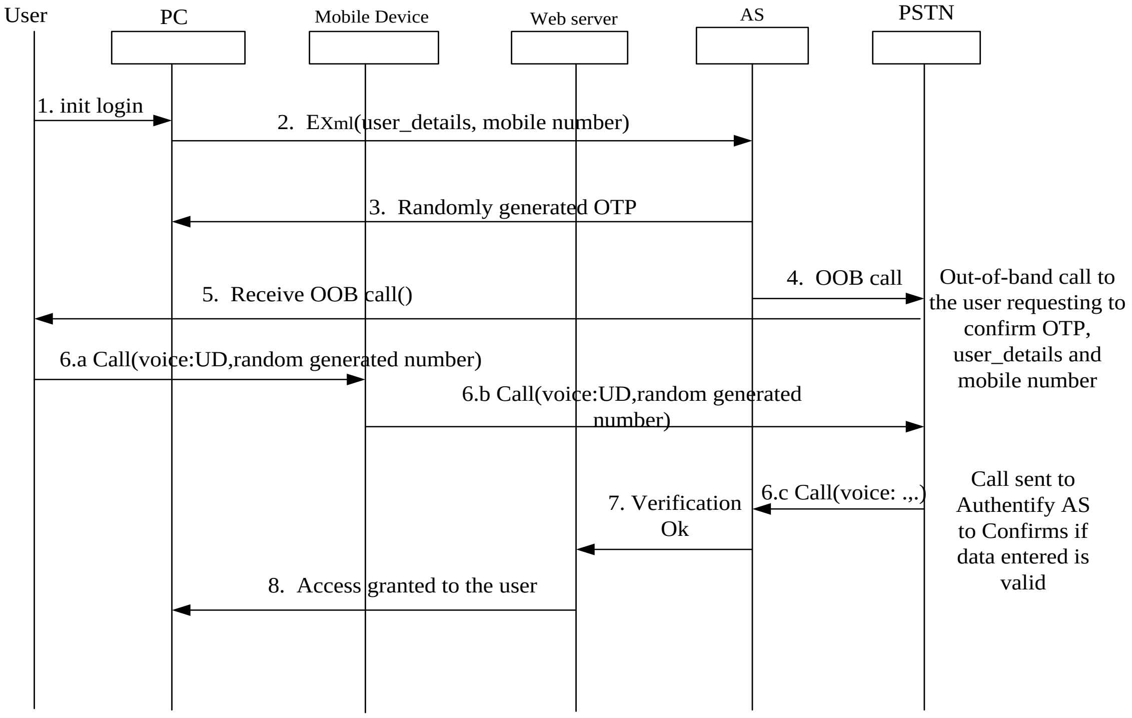

User
PC
Mobile Device
Web server
AS
PSTN
1. init login
2. EXml(user_details, mobile number)
3. Randomly generated OTP
4. OOB call
Out-of-band call to the user requesting to confirm OTP, user_details and mobile number
5. Receive OOB call()
6.a Call(voice:UD,random generated number)
6.b Call(voice:UD,random generated number)
6.c Call(voice: .,.)
7. Verification Ok
Call sent to Authentify AS to Confirms if data entered is valid
8. Access granted to the user

Figure 6: Authentify authentication process flow

The User uses the PC to begin an authentication session. The account details and mobile number of the user are sent from the corporate network to the Authentify authentication server in an XML encrypted format. The Authentify authentication server sends a randomly generated number which is displayed automatically on the user pc screen. At the same time, Authentify authentication server makes an out-of-band call to the user over the Public Switched Telephone Network (PSTN). The user orally confirms his details (randomly generated number) through his unique voice over the PSTN to the Authentify authentication server. The Authentify authentication server sends an XML message to the web server confirming the identity of the individual. The web server grants access.

4.1.6. Summary

	Multiple Channel	Multiple-factor authentication	No Additional Hardware	Platform Independent	Security Critical Domain
PayPal			√	√	√
Web SEAL SSO			√	√	√
AcrotOTP System		√	√		√
Accumu-late ME		√	√		√
Authentify	√	√	√	√	√

Figure 7: Properties of authentication systems

The table above summarizes a number of key properties which are considered necessary for a platform independent authentication method. The authentication method is intended for use in systems that require adequate level of security and therefore requires different layers of security to be put in place to ensure data integrity, confidentiality and availability. AcrotOTP, Accumulate ME, Authentify and various other products in the market adopt multiple-factors as a way of enforcing multiple layers of security. A system which is platform dependent like AcrotOTP and Accumulate ME need some amount of cryptographic encryption to ensure that sensitive data stored on the platforms are not compromised.

4.2. Payment Systems

While conducting literature review on related studies (see chapter 2.6), we came across a widely used platform independent way of making payment electronically by using online payment service providers such as PayPal [21], Google checkout [22] and Authorize.net [23].

4.2.1. PayPal Payment System

This system gives customers the financial capacity to make purchases from online stores. The customer has two options (1) creating an account with PayPal which is funded through his credit card or by his bank account (2) Inputting his credit card details during the course of payment without having to register for an account. When payments are successfully carried out, the amount of the transaction is transferred into the merchant account of the seller. Online stores that make use of PayPal to receive payments, do not have to implement compulsory requirements (see chapter 6.2.1) set by the credit card payment industry because these industry requirements are already implement by PayPal. This saves the online store time, money and resources needed to implement these requirements themselves.

How it works

PayPal works with a number of online commerce shops today. The predominant one today is e-bay. When a customer on e-bay wins a particular auction for goods he has bided online, on paying for the goods, he is directed to a web page where he is given options to make a payment through a number of payment methods (e.g. Visa, Master Card or by PayPal). When the mobile user selects the PayPal payment option, he is redirected to a PayPal page. The user confirms the payment transaction and PayPal will then transfer the money from the user's bank account to the merchant e-bay account. After that, a confirmation email or SMS is sent to both the seller and the buyer.

4.2.2. Localized payments systems

"Localized payment systems" refer to all localized payment systems which implement the compulsory requirements set by the credit card industry. These systems are quite common among big online stores who have the resources to implement all the compulsory requirements. These localized payment systems process, store and transmit customer's card information without redirecting customers to third party online payment service providers like PayPal, Google Checkout etc. Examples of such organizations include Power VoIP [24].

How it works

The implementation of a localized payment system is unique to an organization. A high-level description of how it works involves the receipt of transaction details from interested buyers. The transaction information is processed by transmitting it via backend functions to a chosen gateway. The gateway handles the actual processing of the transaction, and notifies the merchant when payments are completed.

5. Threat modeling

Over more than 200 mobile device threats have been reported in the last decade. All reported threats have been seen to be highly dependent on the type of mobile device. For example the SymbianOS has reported more viruses than any other brand. It is believed that, the number of threats would grow exponentially as long as mobile devices become more sophisticated. That is, as long as the speed, technological advancement and commerce demand for transactions to be carried out on mobile devices increases, the stakes for attacking mobile devices and their applications will also increase. Some of the mobile device manufacturers like Apple have put in place security measures to control what software gets installed on their platform. Although such measures are currently in place, hackers have been able to jailbreak [25] such devices opening the platform to new vulnerabilities. The goal of this chapter is to define a threat model that can be used to verify the security of m-commerce based authentication methods. In order to accomplish this, we first identified various properties from which threats can be classified. These include behavior, environment, and families/invariants. We then identified various mobile threats that exist today by going through several antivirus threat databases and other security sources. With these on hand, we were able to come up with a threat model for m-commerce applications and how to mitigate them.

5.1. Mobile Threats

An ordinary computer threat like a computer virus is not in any way different from the mobile virus, in that they exhibit the same behavior such as propagating from one vulnerable device into another. The difference lies in its adaption, that is viruses on mobile devices are specially adapted for the cellular environment whereas the later it is adapted for networks of connected PCs. When a threat is discovered in a mobile device, security experts classify the threat based on three main categories. These include:

1. Behavior
2. Environment (operating system)
3. Families and Invariant.

All threats have a particular behavior by which they can be identified or grouped. These are viruses, Trojans, worms and spyware. A virus or Trojan or worm or spyware is like any normal computer program. So, when a program is seen to duplicate itself from one device to another, then such a program based on its behavior is categorized as a worm or virus. If it is seen to be stealing certain information from the device to some remote server then it may be categorized as a Trojan. Security experts will further investigate the kind of environment that the threat operates in. SymbianOS has reported a lot more threats than any other mobile platform. Why? It has been reported by Nokia that hackers are able to bypass the security platform thus allowing users to execute unsigned code. This gives users or hackers the access to execute unsigned code on files and areas of the SymbianOS that they initially had no access to. Finally, the threats are then grouped into a particular family. For example, Trojan.SimbianOS.Skuller [26] has 31 different invariants or complex forms. iPhone Ikee has two different invariants that are Ikee-A and Ikee-B. A more detailed description of these and more threats based on their behavior is given in the following subsections.

5.1.1. Mobile Viruses

They are similar to computer viruses with the ability to spread from one infected device to another by propagating through Bluetooth or SMS.

This was the first form of threats that were faced by computers. Examples include Elk Cloner from 1982 [27], Brain Computer Virus [28]. They existed even before the Internet became the main source of communication. Back then, viruses were distributed mainly by installing them manually on the target computer. Viruses became an attractive form of attack for adversary when the Internet became widely used. This was due to the ease of virus infection, which was mostly via email attachment, video streaming etc. Today, viruses are becoming less common amongst malicious ware writers (both professional cyber criminals and normal hackers) due to the efficiency of antivirus protection (e.g. Symantec, MacAfee etc.). More stealthy and resilient form of attacks are widely used and preferred over normal viruses today. Examples of these threats include Trojans and Worms. Based on this trend, computer viruses are less common today, and even more so in mobile devices. Example of one of the few viruses on mobile phones is a proof of concept virus called WinCE.Duts [29].

WinCE.Duts
This malicious program was released in 2004. It infects mobile devices that run under Windows CE. WinCE.Duts is an Advanced Risc Machine (ARM) processor program that runs with a total size of 1520 bytes [29]. This malicious program when ran displays the following message “Dear User, am I allowed to spread”. If the user agrees to install it, then the virus infects all non-infected executable files that are stored in the root folder. The virus writes itself to the ending of the files and establishes an entry point at the beginning of the file. Although it has no payload, the intended purpose was to demonstrate that it was possible to infect mobile devices with viruses.

5.1.2. Mobile Worms

Mobile Worms are self-replicating programs that execute independently and travel across the mobile network.

Commwarrior
Commwarrior [30] is a worm that was discovered in 2005 and originated from Russia. Its targets SymbianS60v2 and it is propagated through bluetooth and MMS. Propagation through MMS medium occurs by sending an infected SIS file via MMS messages to other devices. The devices become infected on opening the attached copy of the file. The problem with this file is that it may be named differently and this makes it apparently difficult for a normal user to know whether the mms message received is a SIS file or not. In the case of propagation through the bluetooth medium, the worm uses the bluetooth of the infected device to search for victims within the devices’ bluetooth range. It then sends the infected SIS files to any device it successfully pairs to.

iPhone IKEE-B

This was discovered in 2009 two weeks after the emergence of IKEE-A [31] botnet. Just like the IKEE-A botnet, it converts the jail broken iPhone into a self propagating worm and infects other iPhone devices by exploiting a vulnerability in ssh[32]. That is, it propagates by scanning specific IP address ranges for SSH services (An example port 22/TCP) and attempts to connect to those services as root by using the default password *apline* [32]. IKEE-B [33] botnet differs from the IKEE-A botnet, because it contains a command and control logic that enables the infected iPhones to be controlled by a master botnet. Each infected iPhone becomes a bot which is programmed by a command and control(C&C) logic server. The server controls logic and redirects infected phones to new C&C remote servers every 5 minutes. IKEE-client creates a unique ID, enabling the Command and control logic to send custom logic to individual bot clients. The botmaster is able to upload and execute shell commands on all infected iPhone botclients.

Megoro

Megoro [34] infects SymbianOS mobile devices and was discovered in 2010. It propagates from one infected phone to another be sending links to a SISX executable file (via SMS) to contacts on the infected phone. An automatic download occurs once the link is visited, thereby infecting the devices with the worm. Megoro just like any other worm can replace files and carry out malicious functions based on its payload.

5.1.3. Mobile Trojans

A Trojan (sometimes called Trojan horse) is a malicious program that masquerades as a harmless legitimate program. The program initially appears to carry out useful services before it is installed by unsuspecting users. It however also contains disguised malicious functions that can harm the mobile phone. The malicious function can be used to log user key strokes, carry out the modification or deletion of important files and install remote software on the mobile phone.

Zitmo

Zeus in the Mobile (Zitmo) [35] is a Trojan horse that intercepts SMS messages (e.g. OTP) that banks send to customers during an online banking transaction. The aim of this attack is to circumvent the confidentiality behind two-factor SMS authentication and approve financial transaction without the knowledge of the customer. The Trojan when installed on the mobile devices monitors incoming SMS by using the SMS stack for its own benefit without the user knowing of its presence. The Trojan uses cross-site scripting to gather information about the mobile user such as mobile number and model (e.g. BlackBerry, Symbian phone). It then sends an SMS containing an executable link to the appropriate (based on the mobile model) malicious Zeus program (e.g. BlackBerry jar, Symbian Packages for SymbianOS). On infection, the Trojan installs a database where it stores all the information it steals. It later sends the stolen information via SMS to the adversary based on its configuration.

Geinimi

Geinimi [36] is embedded into a series of pirated Android applications that causes them to behave as Trojans. It was discovered in late 2010. This malicious software possesses the ability of storing and sending personal information from the victim's device to remote servers. Malicious capabilities of Gemini include: SMS monitoring, harvest and send device data, silently download files, launch browsers with predefined urls etc. An Android device gets infected with the Trojan when a user installs a pirated Gemini infected applications from a third party repository. To ensure that communication between the Gemini code and the remote servers are kept confidential, the adversary encrypts the communication using a weak DES cipher.

JiFake

JiFake [37] is a Trojan that was discovered in 2010 and affects J2ME OS based mobile devices. This malware appears to create a backdoor for other viruses and worms affecting the mobile device. JiFake monitors the victim's online activities and steals personal information such as credit card details .The stolen data is then sent to the adversary's remote servers. Mobile devices get infected with JiFake when unsuspecting users download them unknowingly from compromised websites.

5.1.4. Mobile Spyware

Mobile spyware are spy programs that perform certain unauthorized functions without the consent of the mobile users. It can be used to listen to every call, view SMS messages or perform a stealthy monitoring of the users activities.

Cell Phone Recon

Cell Phone Recon [38] is a mobile spyware that was discovered in 2010. It infects all mobile platforms except the IPhone. Once installed on the device, the user finds it extremely difficult to know that such malicious program is installed because it provides no application icon. The software performs all forms of monitoring (e.g. View text messages and View HTML email content including embedded images, etc) and in addition provides hackers with an administrator website from which to carry out the monitoring process. So far, four different variants have been discovered.

Trusters Spy Phone

This is another spyware [39] that was discovered in early 2010. It affects a number of Operating systems which includes SymbianOS, Windows Mobile and BlackBerry. Once installed on the victim's mobile device, the adversary is able to monitor the mobile communication of the victim. This threat can only be installed when the adversary has physical access to the device. This spyware application can be remotely controlled via SMS, forward and record incoming SMS messages, listen to surrounding audio and listen to both sides of a conversation.

NeoCall

NeoCall [40] was discovered in the late 2009 and affects SymbianOS and Windows Mobile platforms. It is a spyware which when installed on the target mobile device, enables the adversary to remotely issue SMS commands to retrieve requested data. It performs all the function as the Trusters Spy phone and in addition performs the following functions: localization of GPRS coordinates of the device to enable the adversary pinpoint the location of the victim, retrieve SIM card details etc. Just like the Trusters Spy Phone, the adversary needs to gain physical access to the device in order to install the NeoCall spyware.

5.2. Threat Model

Whenever a security analyst is called upon to evaluate the security of a system, he first would have to create a threat model of the application he intends to evaluate or design. This is necessary to enable him to fully assess the possible threats that may occur by looking at the system from the attacker's point of view. We will describe a threat model in the next section in order to identify the kind of threats that can be faced by the authentication system. In order to successfully create a threat model for an m-commerce application, one would first need to understand the target system. We did this by first identifying the system assets, system users and vulnerable points. We weren't able to come up with possible attacks for the threat model based on this information and from the various mobile threats that we have identified to exist today (see 5.1). We have also investigated and documented possible ways of mitigating attacks identified in the threat model.

5.2.1. Assets to be protected

In order for users to successfully prove who they are during an authentication process, they will have to provide certain authentication data and in some cases personal data. The security of these data must not be compromised during the authentication process or via vulnerabilities in the authentication method.

User data
The confidentiality, integrity and availability of user data must be assured at all times. Examples of such data include the names, telephone number, address, credit card details etc.

Authentication data
A system verifies a user during authentication by requesting for some secret information (OTP, password, pin, etc.) which only the user knows. For that reason, these authentication data must be securely protected from getting into the hands of any other person.

5.2.2. Users

A system cannot be fully evaluated unless a clear picture of all those that will be using the system is available. In a secure system, security privileges, access and data are made available only to a certain group of users while it is blocked for others. Therefore, a good understanding of the different users that can and will interact with a system must be taken into consideration when designing secure systems.

Legitimate users
A legitimate user is the person who is the rightful owner of an asset, or that has exclusive access to certain system privileges. An example of a legitimate user is the owner of a credit card used in a financial transaction.

Adversaries
This is the person who intentionally tries to acquire assets which does not belong to him, or maliciously tries to gain system privileges which he is not entitled to.

Administrators

Administrators are persons who have been legally mandated by the organization to handle day to day running of the m-commerce system. Tasks carried out by administrators include system modification, account deletion and so on.

5.2.3. Vulnerable points

Inputs and output points are avenues in which users or data enters or leaves the system's trusted network. These entry points are vulnerable to attacks because they serve as the only way the attacker can have access to the system resources.

Communication channel

Smartphones provide access to applications via several communication channels. These channels serve as entry and exit points to m-commerce applications and are vulnerable to different types of attacks. Examples of communication channels on Smartphones include short message service (SMS), Bluetooth, http etc.

Web browsers

Most m-commerce applications reside on remote servers and are accessed via web browsers such as opera mini, safari and so on. Thus, vulnerabilities in these browsers will also affect the security of the m-commerce application.

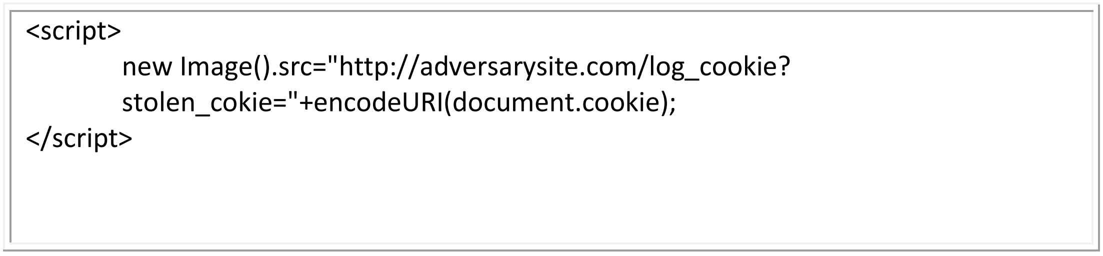
Mobile phone OS

In most cases, an application implemented on a mobile phone is dependent on the mobile phone operating system (OS) for communications with the system processes and hardware. Popular OS include Android, iPhone, Symbian etc. A security hole in any of these operating systems could be used as an entry point into attacking the m-commerce application that resides on them. An example of such a case is when an adversary gains administrative rights to an OS. He can for instance configure the security settings of the default Internet browser to allow connections to unsecure websites.

5.2.4. Attacks

Cross site scripting (XSS)

This attack has been demonstrated to be possible on mobile phones as illustrated in *Zitmo* (5.1.3). Cross site scripting involves a process where an adversary attacks an m-commerce website by embedding malicious html, CSS, JavaScript or VBScript via various vulnerable points, in order to steal data from unsuspecting visitors [41]. For example, a dynamic website that allows user to enter comments on its sites (such as Facebook) can be XSS attacked.



```
<script>
        new Image().src="http://adversarysite.com/log_cookie?
        stolen_cokie="+encodeURI(document.cookie);
</script>
```

Figure 8: Sample XXS attack

An adversary can attach the malicious script (see Figure above) to a comment entered on a popular site. If the site happens to be vulnerable to XSS, then cookies of visitors (which may contain usernames, passwords) will be sent to the adversary site on adversarysite.com. This will happen to every person that visits the page were the comment and script resides.

Eavesdropping Attack

Eavesdropping [42, 43] creates the opportunity for adversaries to listen to or possibly extract personal details and information of their victims. Eavesdropping can be carried out through a number of ways. One way is by installing a spyware on the system (see 5.1.4). Another way is by using a network sniffer on the network to capture and reassemble packet as they are transmitted across the network.

Replay Attack

This is a form of attack [44] in which the attacker intercepts a valid data during transmission and maliciously replays it. For example, a user is given a session token after successful authentication. An attacker can then intercept this session token and replay it to gain access to the restricted account session at a later time.

Smishing and Vhishing Attacks

Smishing [45] is similar to Phishing attacks in that the attacker sends SMS messages to a legitimate user claiming to be an established entity in an attempt to obtain user information and details. This form of attack is difficult to discover especially when the URL is well crafted by the adversary. Smishing attacks almost always contain messages that require you to carry out an “immediate action”. Sometimes, such actions may lead to victims revealing their personal details like credit card information and so on. Whereas Smishing makes use of the SMS, Vhishing on the other hand makes use of voice to carry out phishing attacks. For example, an adversary sends an email requesting the victim to make a call. On calling the number, a voice response system is activated requesting the victims’ financial details. The deceptive nature of Vhishing lies in the underlying fact that victims are deceived into thinking that they are dealing with their legitimate banks or other recognized organizations.

Man in the Browser Attacks

MITB [46] is an attack that is aimed at intercepting streams of data that is sent over a secure communication channel between the customer’s web browser and the mobile store. The adversary or hacker would normally embed a Trojan in the customer’s web browser so that whenever the customer visits online banking sites, the mobile threat is activated to modify of the data entered. Using html injection, the Trojan displays fake web pages to the victim so that the victim gives away his transaction credentials. The stealthy nature of this attack makes it difficult to detect since any activity carried out by the Trojan seems to be coming from the user’s browser. For example, a MITB

Trojan embedded in the browser can change messages received from the mobile store before displayed by the user's browser.

DOS Attack
Denial of Service [47] is an attack where an adversary attempts to deprive the victim of services that he requires. Mobile worms are mostly responsible for causing DoS in mobile devices. For example, a DoS attack can occur when an adversary continually requests for Bluetooth pairing with a victim's device. The result may be that the victim's device becomes unresponsive due to packet flooding from the adversary. DoS can also occur when the victims Bluetooth stack tries to allocate resources to the adversary's requests that never complete the handshaking protocol. This eventually leads to exhaustion of Bluetooth stack memory.

5.3. Mitigating threats

This sections aims to cover some known and proven ways of mitigating the attacks described in the threat model (see 5.2.4). These attacks include XSS, Eavesdropping, Smishing/Vhishing, DoS, MITB and Replay attacks.

5.3.1. Cross site scripting (XSS)

XSS can be mitigated in different ways, but the two most common and effective ways are filtering and escaping [48].

Filtering
An XSS attack is accomplished by embedding malicious html, CSS, JavaScript or VBScript through some form of external input. The most cost-effective way to mitigate these malicious input is by passing all input through a filter, where all potentially dangerous keywords like <script>, JavaScript commands, html markups, etc. are removed. Many XSS mitigating libraries exist to aid developers to build XSS secure sites. These include Microsoft Web Protection Library [49], HTML Purifier [50] etc. This method might have some drawbacks, in that the filter cannot tell the difference between non-malicious and malicious keyword usage. Thus, a non-malicious user like me including keywords (e.g. <script>) in my text will have it removed by the site filters. One way to solve this issue is by using escaping techniques.

Escaping
Filtering replaces potential malicious keywords with whitespace, escaping instead makes the keyword to be interpreted as pure data and nothing else. This way, keywords like <script> alert ("don't forget to sign out") </ script>, <b> look here <\b> etc. can be displayed but will be interpreted by the browser as text for display purpose only. Just like filtering, there are also various libraries that can be used for escaping keywords in different languages. Below are some examples of escaping dynamic keywords.

HTML Markup	Escaped HTML Markup
<h1></h1>	&lt;h1&gt;&lt;/h1&gt;
<b></b>	&lt;b&gt;&lt;/b&gt;
<br />	&lt;br /&gt;
<html><\html>	&lt;html&gt;&lt;\html&gt;
<body><\body>	&lt;body&gt;&lt;\body&gt;
<script><\script>	&lt;script&gt;&lt;\script&gt;

Table 2: Escaping dynamic keywords

5.3.2. Eavesdropping Attack

One way to prevent application-level eavesdropping where Trojans/Spyware are maliciously or intentionally installed on mobile devices is by scanning all files for malware before installation. Another way is by encrypting all sensitive data; this ensures the confidentiality of data even when the adversary eavesdrops on them. Encryption should also be adopted to mitigate network eavesdropping because it makes the reassembling of stolen packets unusable to the adversary.

5.3.3. Replay Attacks

Mere encryption cannot be used to mitigate replay attacks since the adversary can simple replay the encrypted data (e.g. authentication data, login session etc.) without decrypting it. In order to make replay attacks impossible, it is necessary to introduce some random data to each transmission. Several mechanisms can be used or combined to accomplish this [43].

One Time Passwords
One-time passwords, as the name implies, help create randomness of data. Each transmitted data will always be unique.

Session token
During a login session, a random session token can be sent from a server to the user attempting to login. The user carries out some form of computation using the session token and his password (e.g. appending and hashing them) and sends the result to the server. The server performs the same computation as the user and verifies if the results match. The method is able to counter replay attacks by sending a different session token for every login session.

Time Stamping
Computing session tokens with time stamps makes it even more difficult for replay attacks to be carried out. This is because the time stamp of the login session will no longer be valid due to the time difference between when the login session was first used and when replayed by the adversary.

5.3.4. Smishing and Vhishing Attacks

It is necessary to prevent Smishing and Vhishing Attacks since its occurrence may affect the image and the potential customer base of a company. Customers who feel threatened about the privacy of their data may deter in using such services for fear of being duped.

Blocking
In some Smishing attacks, users are tricked into opening links that redirect them to Smishing websites, were they become vulnerable to information theft. Organizations can protect themselves against this threat by proactively blocking redirection to Smishing sites from their network. This function exists in major browsers such as Internet explorer, Mozilla etc. and should be enabled to help prevent such attacks.

Educate customers
It is necessary that companies educate their customers about possible social engineering techniques that could be used by adversaries to steal their data. For instance, companies can set security policies such as “We will never request user authentication details (e.g. passwords, username) through email or phone”, and inform their customers about these policies. This method can therefore prove useful against mitigating Vhishing attacks.

Data access restriction
For instance, adversaries may target employees of corporate organizations with the aim of stealing customer information. The leakage of company sensitive data via employees can be limited or avoided by applying restrictions on the amount of customer sensitive information they have access to.

5.3.5. MITB attack

One of the effective ways to protect the victim’s browsers or financial details against this attack is employing an out-of-band transaction verification. This form of transaction verification ensures that the customer verifies his transaction to the host organization (say the bank) by making use of another channel of communication than the web browser. This channel could in particular be used to make a phone call where the voice and what he knows about the transaction are verified in order for the transaction process to proceed.

5.3.6. DoS attack

One way to perform DoS attacks is to exhaust the mobile phone’s resources, for example by flooding the Bluetooth mobile device with paring requests, which might lead to dropping of legitimate pairing requests if there is not enough resources to accept all incoming requests. This can be prevented by turning off Bluetooth pairing when not needed to prevent attacks from being launched at idle mobile phones. Another way is to allocate enough resources to handle large amounts of request. This however, is not practical due to the limitations on computing power and small memory of mobile phones.

6. Proposed method

The aim of this study was to propose a secure platform-independent authentication and payment method for m-commerce applications. To achieve this, we reviewed previous studies that were conducted in the area of authentication and payment (Chapter 2), as well as reviewing what underlining methods, current existing authenticating systems use (Chapter 4). We have found several plausible solutions to the problem, of which we propose the use of an OSP authentication method and two payment approaches.

6.1. OSP

OTP via SMS-Password authentication (OSP) method involves the use of the multiple authentication factors and communication channels for authentication. When using the OSP, a user provides his username and password, the OSP system sends a one-time password (OTP) to the user's mobile device, the user completes the process by retrieving and providing the retrieved OTP.

6.1.1. Requirements

a. The authentication method implementation should be platform independent. One implementation should be deployable on different Smartphones (e.g. iPhone, BlackBerry, Android phone etc.) with little or no modifications.
b. The authentication system must use at least two factors for authentication. For example, something the user knows (e.g. OTP, token) and something he has (e.g. a Smartphone, telephone number). An intruder, with access to only one factor, will not be able to authenticate himself as a legitimate user.
c. In cases where one-time passwords (OTP) are used, secure cryptographically generated OTPs are required.
d. All data been exchanged between an authenticated person/device and the verifier must be encrypted during the entire authentication process.
e. In the case where passwords are used, users must be required to choose strong passwords.
f. All communication between all involved systems must be encrypted with a strong form of encryption
g. OTPs can only be used once
h. OTPs must have a limited lifetime

6.1.2. Design

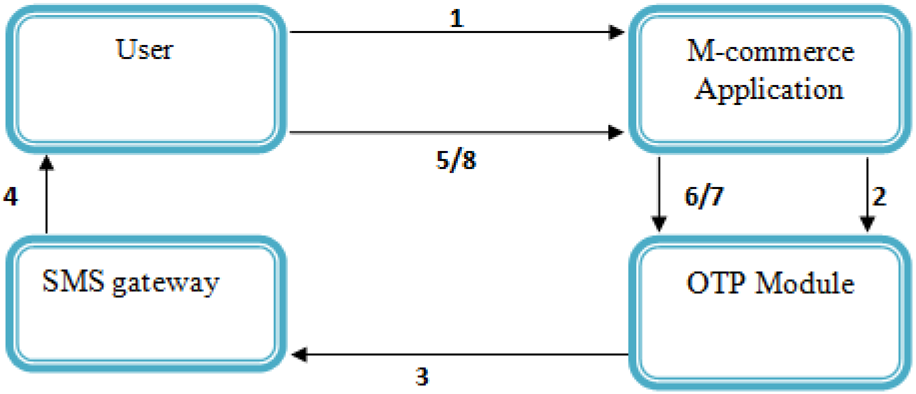


Figure 9: OSP authentication process

1. The user signs in with a username and password.
2. M-commerce application requests for OTP generation from the OTP Module
3. OTP module generates an OTP and sends it to the SMS gateway
4. SMS gateway sends the received OTP to the user via SMS messaging
5. The user enters OTP into M-commerce application
6. M-commerce application verifies the received OTP with the OTP module
7. OTP module verifies OTP and relays back the result to M-commerce application
8. M-commerce application denies or grants access to the user based on result from OTP module.

6.1.3. Pros and cons

Pros

a. If an intruder gets unauthorized access to the mobile device (something you have), he still cannot successfully authenticate himself as the legitimate owner of the phone, since the application requires knowledge of the username and password (something you know)

b. Likewise, if an intruder gets unauthorized access to the username and password, he still cannot successfully authenticate himself as the legitimate user without retrieving the OTP from the mobile phone.

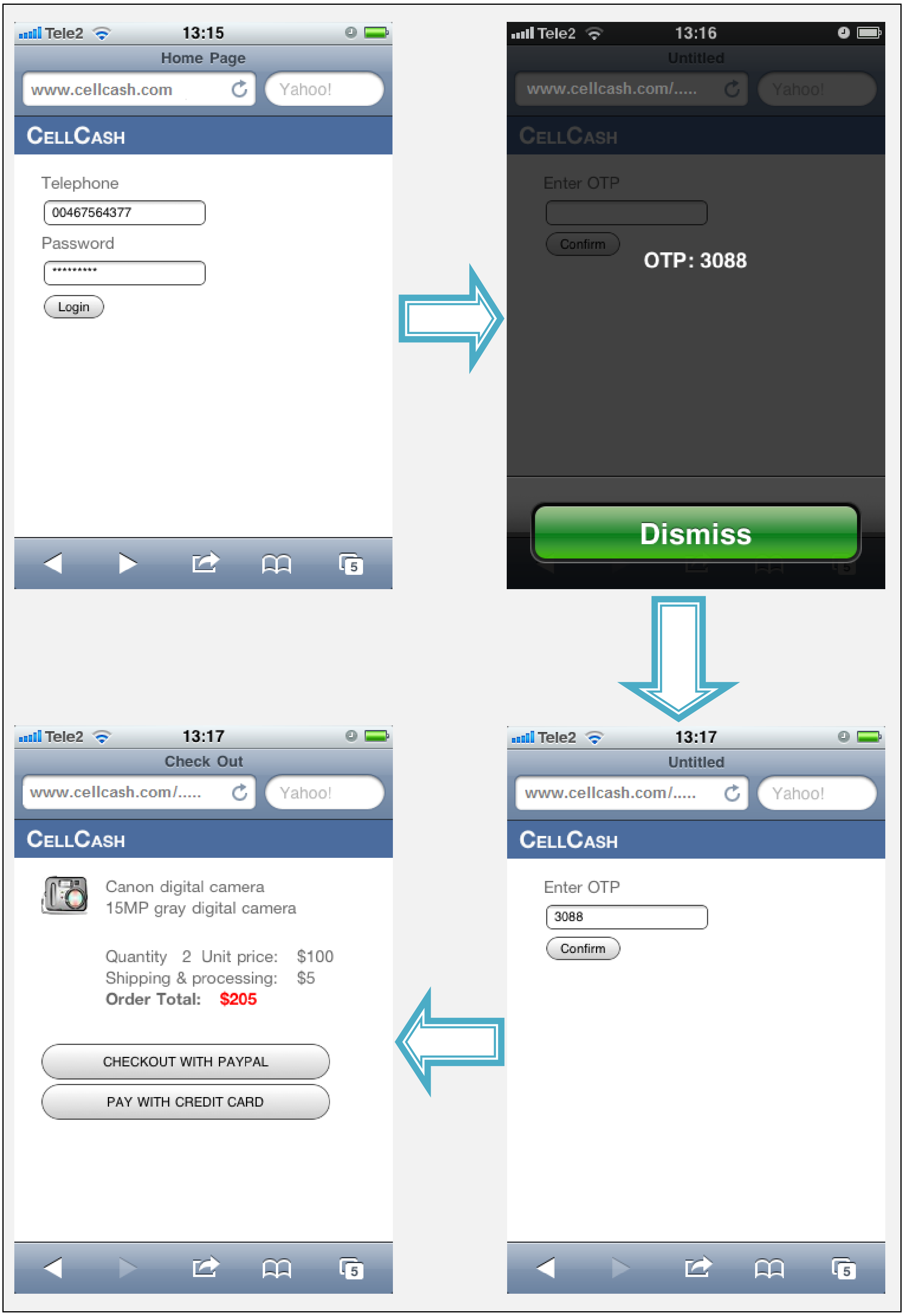
c. An authentication request by an intruder using a compromised username and password will be immediately detected by the legitimate user, since an OTP will be sent to his phone.

Cons

a. Latency of SMS messages is not constant. Thus, there might be some performance issues.

b. The cost of sending SMS messages may be too high to be used for m-commerce applications that don’t require high level of security.

6.1.4. Prototype



Tele2 13:15

Home Page

www.cellcash.com

CELLCASH

Telephone

00467564377

Password

****

Tele2 13:16

Untitled

www.cellcash.com/.....

CELLCASH

Enter OTP

OTP: 3088

Tele2 13:17

Untitled

www.cellcash.com/.....

CELLCASH

Enter OTP

3088

Tele2 13:17

Check Out

www.cellcash.com/.....

CELLCASH

Canon digital camera
15MP gray digital camera

Quantity 2 Unit price: $100
Shipping & processing: $5
Order Total: $205

Figure 10: OSP authentication prototype

The prototype is implemented as a solution for an m-commerce online application/store that needs to authenticate buyers before they can make purchases. The prototype is based on a client-server architecture with a very slim client. It was implemented in ASP.NET and targeted at .NET framework 4. The solution uses a third party SMS gateway provider called CellSynt [51] for sending OTPs to buyers' mobile phones. All the OSP logic concerning cryptographic secure OTP generation, SMS sending and so on are located on the server side. The figure above shows a live run of the OSP solution prototype implementation. The scenario is an m-commerce store that sells electronics such as laptops, cameras etc. The system authenticates the buyer by requesting for is telephone number and password (first authentication factor). On receipt of a valid telephone number, the system generates an OTP and sends it to the registered buyer's phone (second authenticating factor). The user retrieves and enters the OTP into the application and is granted access to purchase the item after successful verification of the OTP. The prototype has been tested on an iPhone and an Android OS with no change to the code. It is important to note that the prototype only implements certain parts of the OSP method for simplicity. Thus a completed version should implement and meet all requirements described in 6.1 such as encrypted data communication [51].

6.2. Secure payment approaches

Payment processing is complicated and involves several parties and communications between them. In this section, we will look at different methods of adding financial transaction capabilities to m-commerce applications. An evaluation (based on security and other factors) of these methods was also carried out and described in chapter 7.

6.2.1. Understanding how payment works

Making a payment with a credit card takes a few seconds. However, the amount of work that goes on at the background cannot be easily explained within a few minutes. A payment process involves several cooperating parties, several exchanges, cash exchange and so on. It also involves following the payment card industry data security standard (PCIDSS). Table 3 shows the PCIDSS[52] security requirements and figure 11 shows the data to be protected, the parties involved and a scenario of what happens from when a payer conducts a transaction with his or her credit card until the merchant receives the transferred amount.

Secure payment data and requirements

In order for merchants to be PCI compliant, they must carefully implement the 12 requirements stated in the Payment Card Industry Data Security Standard (PCI DSS). Figure 11 shows the cardholder and authentication data that must be properly protected. Following that are the 12 requirements that each m-commerce business must implement. These requirements help to secure the m-commerce service network, protect cardholder data, maintain a vulnerability management program, implement strong access control measures, regularly monitor and test the m-commerce service network and maintain information security within the m-commerce organization

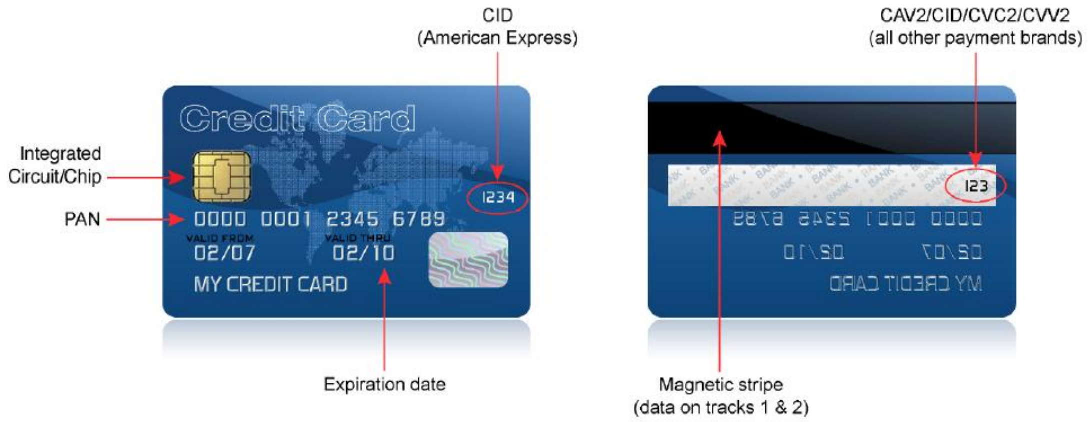


Figure 11: Card data to be protected [53]

How to build and maintain a secure network	
Requirement 1	The cardholder and authentication data which is been stored, processed or transmitted within the m-commerce private network should be protected from outside untrusted networks. This should be done by setting up a firewall with secure configuration rules required to protect the private network.
Requirement 2	Organizations should not retain or use vendor-supplied default authentication parameters (e.g. passwords) or other settings. This reduces the risk of compromising the systems through the default passwords or settings, which might be known outside the organization.
How to protect cardholder data	
Requirement 3	The integrity, confidentiality of stored cardholder data should be protected at all times. This should be carried out by using cryptographically strong methods and keys for data encryption, hashing, masking and so on. This ensures that in situations where cardholder data are maliciously accessed, the intruder will still not be able to understand the content of the data without the right encryption keys. Organizations should also avoid storing unnecessary cardholder data except in cases where specifically needed.
Requirement 4	Transmission of cardholder data within an open domain such as the Internet must always be encrypted. A public domain is susceptible to various forms of malicious attacks such as man-in-the-middle, interception etc. Thus, encrypting cardholder data gives assurance that only the intended recipient is able to use the data.
How to maintain a vulnerability management program	
Requirement 5	Malicious software such as viruses, Trojans and worms are continuously evolving into different variants that target certain application vulnerabilities. Organizations must handle these new threats by installing and regularly updating antivirus programs.
Requirement 6	Develop secure systems and ensure that their security is always maintained at all times. This can be done by being up to date with security patches which eliminate new vulnerabilities that are discovered in the system.
How to implement strong access control measures	
Requirement 7	Access to cardholder data by employees in an organization must solely be granted on a need to know bases. Access must only be granted to the least amount of data required for an assignment.

Requirement 8	Unique identification numbers must be allocated to all employees with computer access. This will ensure accountability and easy tracing of defrauding employees, by going through their usage history.
Requirement 9	Restrictions must be placed on access to physical cardholder data or critical systems. For example, temporary employees such as contract workers should not been given access to critical systems. This limits the danger of unauthorized persons removing sensitive hardware, such as hard disk containing sensitive cardholder data.
How to regularly monitor and test networks	
Requirement 10	Mechanisms to prevent, detect, and minimize the impact of data compromise must be implemented in organizations that process, store and transmit cardholder data. These mechanisms include the ability to track and monitor all access within the local network and cardholder data. This ability to track and monitor helps in easy detection of unusual behavior within the network and thus helps to prevent or minimize the impact of data that might arise from such behaviors.
Requirement 11	All security systems and processes must be regularly tested in order to be proactive in detecting vulnerabilities that might exist in the system. This puts you a step ahead of attackers who themselves are actively looking form vulnerabilities to exploit in your system.
How to maintain an information security policy	
Requirement 12	The organization must maintain an information security policy that contains security rules, best practices and regulations guiding how employees work. Employees should be aware of these policies and regularly educated on them.

Table 3: PCI-DSS requirement

Parties

***Merchant bank*:** A merchant bank is the financial organization that provided the m-commerce firm (merchant) with a merchant account. The merchant bank receives credit card transactions and pays the required amount into the m-commerce firm merchant account. It does the deposition of the money into the merchant account even before the said amount is transferred to it from the cardholder's issuing bank (see figure 12 below).

***Issuing bank*:** The issuing bank (e.g. SwedBank and SEB) is the financial organization that provides credit cards to people for conducting purchases and other transactions. In most cases, one can only acquire credit cards from banks where they have an account. Exceptions to this are American Express and Discover card associations which issue cards directly to individuals.

***Card associations (Brands)*:** Card associations or sometime referred to as brands are the establishments such as MasterCard, Visa, American Express etc. They regulate and standardize the card payment industries by working together with different financial and governmental stake holders. They do this by making rules on security requirements and interchange rate of financial transactions.

***Merchant processor*:** Merchant processors are the link between the m-commerce firm (merchant) and the rest of the parties involved in a financial transaction process. The merchant bank has access to the entire card brand associated with the merchant. It handles sending authentication request to the appropriate card issuing organization. It also forwards approved financial transaction processing request to the merchant banks on behalf of the merchant (see figure 9)

Payment process

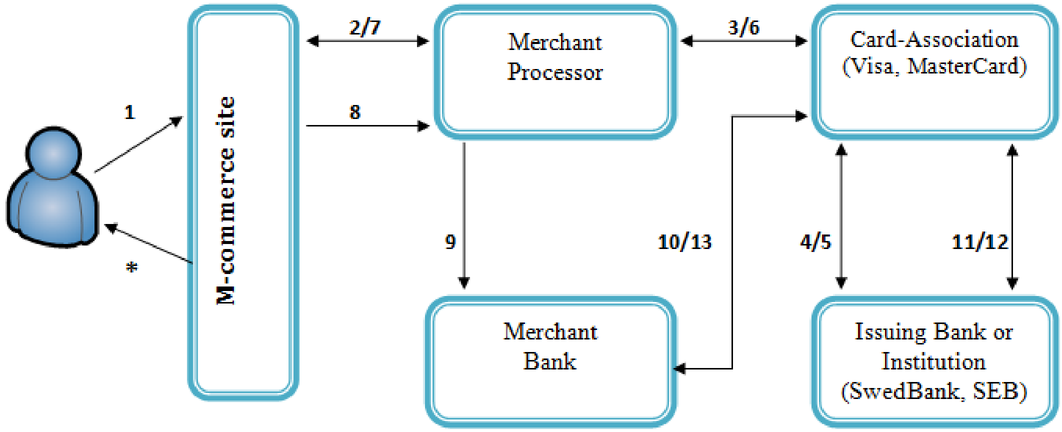

1
*
M-commerce site
2/7
8
Merchant Processor
3/6
Card-Association (Visa, MasterCard)
9
10/13
4/5
11/12
Merchant Bank
Issuing Bank or Institution (SwedBank, SEB)

Figure 12: Payment process

Message	Description
1	The buyer chooses to pay for an item at an m-commerce store using a credit card (e.g. visa, MasterCard etc.). Inputs credit card details
2	m-commerce application request authentication by sending card information and cost of product (transaction details) to merchant processor
3	Merchant processor detects card brand (visa, MasterCard etc.) based on cards first 6 digits, and sends an authorization request to the identified card association
4	Card association identifies the (based on internal database) bank that issued the card, and sends the authorization request to the issuing bank
5	The issuing bank will either approve or decline the transaction based on their set criteria. The decision is sent back to the card association.
6	The card association sends the decision back to the merchant processor
7	The merchant processor relays the decision back to the m-commerce store
8	In the case where the transaction was approved, the m-commerce application sends it to the merchant processor for processing
9	The merchant processor sends the transaction to a merchant bank where the m-commerce store has an account
10	The merchant bank pays the amount involved in the transaction into the m-commerce store account. It then requests an equivalent amount from the identified (see message 3) card association
11	The card association re-sends the request for payment to the bank that issued the card.
12	The issuing bank deducts the requested amount from the buyers bank account tied to the credit card. It then transfers the amount to the card association
13	The card association transfers the money to the merchant bank
*	The m-commerce store returns transaction result (completed, pending, denied etc.) to the buyer. This can take place any time after message 7, and based on whether the transaction was authorized or not

Figure 13: Details of the entire payment process

6.2.2. PCI compliant approach

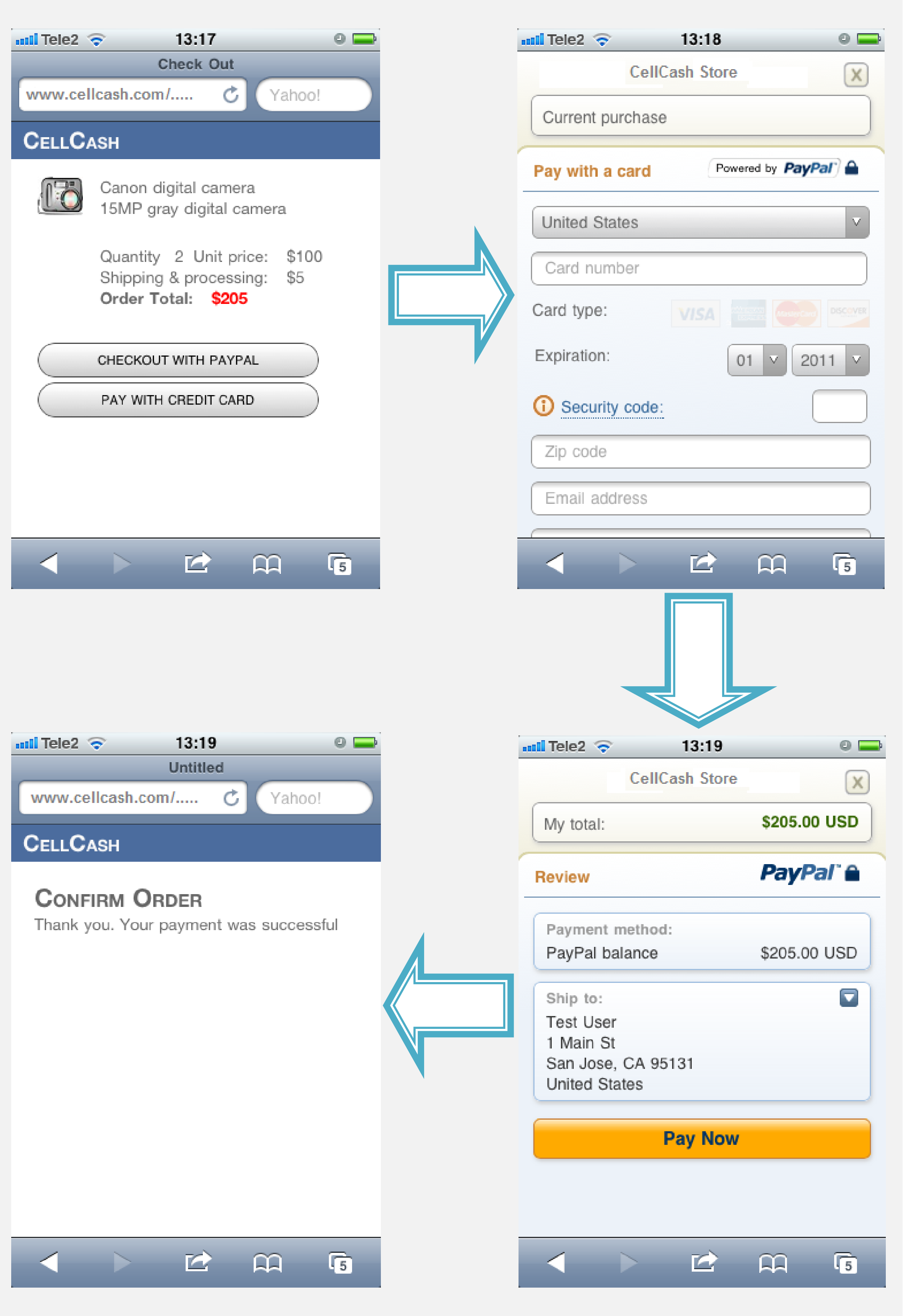
In order for merchants to store, process or transmit payment cardholder data, they must implement the Payment Card Industry Data Security Standard (PCI DSS). This standard includes 12 best practices (requirements) for security management, policies, procedures, network architecture, software design and other critical protective measures. In the PCI compliant approach, the merchants carry out the processing, transmission and storage of cardholder data. Thus, it is essential for any m-commerce application using this approach to be PCI compliant.

How it works
Chapter 6.1.2 illustrates how the PCI compliant approach can be carried out. Another possibility is to have a gateway provider serves as a middle man between the m-commerce store and the remaining parties in the payment process.
"Payment Gateways connect the Merchant to the bank or Processor that is acting as the front-end connection to the Card Associations". "They are called Gateways because they take many inputs from a variety of different applications and route those inputs to the appropriate bank or Processor" [54].
The choice of method to use is left to m-commerce store. However, both variants require that the m-commerce store is PCI complaint in order to be able to store, process or transmit cardholder data to other parties. Making use of a gateway also offers another way (using third party approach) of carrying out financial transaction in m-commerce applications. This approach is described in the nest section.

6.2.3. Third party approach

This approach involves using the services of third party payment gateways to facilitate credit card payments. Examples of companies that offer such solutions include PayPal and Google.

How it works
The third party gateway provides the m-commerce site an abstraction of the payment process. It does this by providing the m-commerce site with a merchant account that handles the security of transaction information (e.g. credit card numbers), implements PCI compliance requirements and facilitates communication to the card association and card issuing banks. It also offers quick and easy use of its services by providing APIs which can be accessed via secure https requests from the merchants store. Integrating such third party payment capabilities to an m-commerce web-site only takes few hours and average programming skills. This makes it a preferred choice for small and medium m-commerce businesses, which lack highly skilled IT staff and resources. The picture below shows our prototype implementation which was done using PayPal [55] as a third party payment gateway.



Canon digital camera
15MP gray digital camera

Quantity 2 Unit price: $100
Shipping & processing: $5
Order Total: $205

Current purchase

Pay with a card — Powered by PayPal

United States
Card number
Card type:
Expiration: 01 2011
Security code:
Zip code
Email address

My total: $205.00 USD

Review — PayPal

Payment method:
PayPal balance $205.00 USD

Ship to:
Test User
1 Main St
San Jose, CA 95131
United States

CONFIRM ORDER
Thank you. Your payment was successful

Figure 14: Third party approach

The prototype is implemented as a solution for an m-commerce online application/store that needs to accept payment from customers. The prototype is based on client-server architecture with a very slim client. It was implemented in ASP.NET and targeted at .NET framework 4. The solution uses a third party payment gateway provider called PayPal for accepting card payments. The figure above shows a live run of the “Third party approach” payment solution. The scenario in figure 14 above is an m-commerce store that sells electronics such as laptops, cameras and a buyer who wishes to purchase a camera. The user is first authenticated (see chapter 6.4), and then redirected to PayPal based on the payment option chosen. He pays for the camera by entering required card details and is redirected back to the m-commerce store after completion of the payment. The prototype has been tested on an iPhone and Android OS with no change to the code. It is important to note that the prototype only implements one-way authentication (HTTPS) between the m-commerce store and PayPal for simplicity. Thus a completed version should implement a secure two-way authenticated HTTPS connection between the m-commerce store and PayPal.

7. Evaluation

An exploratory methodology (see chapter 3) which involved data collection, analysis, validation and evaluation was adopted in this study. This chapter describes the evaluation carried out of the OSP authentication method described in chapter 6.1.The evaluation involves an analysis of how well the method mitigates attacks from the threat model presented in chapter 5. Also described is the evaluation of the PCI Compliant and Third Party payment approach presented in chapter 6.2.2 and 6.2.3.

7.1. OSP authentication method evaluation

The various attacks identified in the threat model include XSS, Eavesdropping, Replay, Man in the Browser, DoS, Vhishing and Smishing attacks. Below is the security analysis of each attack in relation to the OSP authentication method.

7.1.1. XSS attack

An XSS attack can be used to insert malicious scripts or commands into m-commerce applications in order to steal user data. In the case of the OSP authentication method, the security of the method depends on the inability of an adversary to compromise authentication data (username, password, OTP). Thus, it is required that XSS mitigating techniques (see threat section) should be implemented in any m-commerce applications that use the OSP authentication method. However the OSP method is by itself secure against XSS attacks using OTP. For example, even if an adversary is successful in compromising an m-commerce site and is able to steal a legitimate username and password, as illustrated in the figure below, he will still not be able to be authenticated using the stolen credentials. This is assured based on the two-factor authentication employed by the OSP method. The adversary cannot log in since he is not in possession of the generated OTP and the legitimate user will find out that his account is compromised when he receives the SMS.

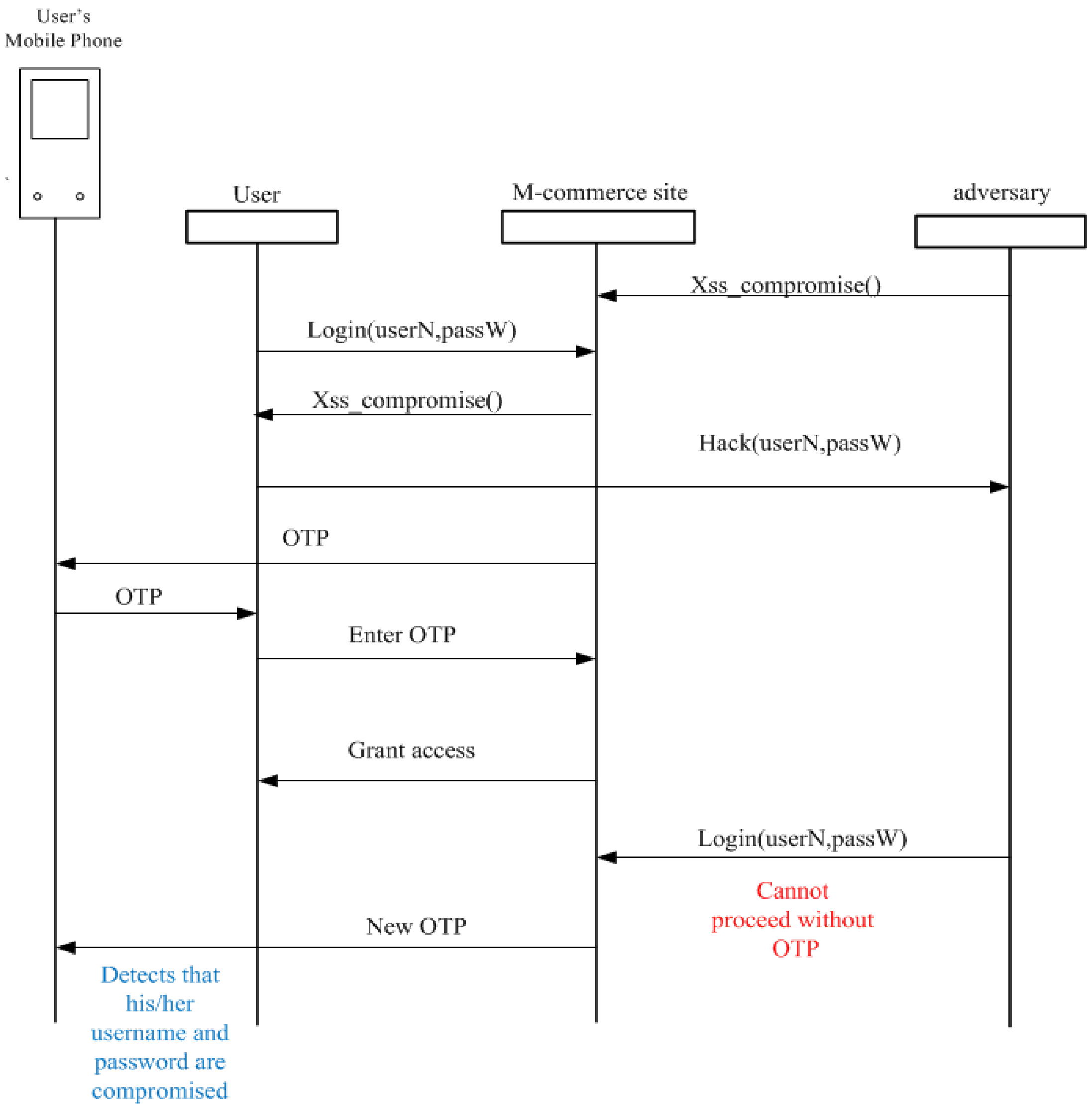


Figure 15: Mitigating XSS attack

7.1.2. Eavesdropping Attack

Eavesdropping attacks as explained in chapter 5.2.4, can either take place at the application or network level. A successfully installed eavesdropping spyware can steal usernames and passwords via key loggers and even retrieve or access stored one-time passwords. If this is allowed to happen, then the security of the OSP method which depends on the secrecy of passwords and OTPs is at risk. However, the OSP method was designed with preventive measures which eliminate the possibility of this to occur. One of the countermeasures is to use a security feature provided by the CellSynt SMS gateway. This feature allows sending the OTP as a flash message which is only displayed and not stored in the user's inbox. The main aim of this is to eliminate the possibility of storing retrieving the SMS from the inbox of the mobile phone. This makes it impossible for spyware (which have the capabilities of stealing stored SMS) to retrieve the sensitive OTP. However, a drawback with using SMS based flash messages is that the implementation might vary depending on the mobile phone model, operator and other factors. Therefore, individual tests should be carried out to see how secure the different implementations of flash messages are against spywares. Another measure which has been implemented to mitigate eavesdropping is to ensure that all communications involving authentication and afterwards are encrypted (using https). This makes is impossible for adversaries to use packet sniffers to intercept and reassemble meaningful data.

7.1.3. Replay Attacks

This attack involves the interception of data between two or more parties and the malicious use of that data at a later time. Interception can occur either in real-time during active communication between the involved parties, or by obtaining communicated data at a later time from either of the concerned parties. There are several known techniques for mitigating replay attacks (chapter 5.3.3). One such approach is the use of OTP which is a key feature of the OSP authentication method. During the process of authentication with the OSP, an OTP is sent from the m-commerce site to the customer's mobile phone. The customer then authenticates himself by sending the OTP back to the m-commerce site. The interception of the OTP by an adversary to be used later in a replay attack will be unsuccessful since an OTP is only valid once and also only for a short period of time.

7.1.4. Smishing Attacks

A Smishing attack (see chapter 5.2.4) uses SMS messaging in deceiving unsuspecting users to disclose sensitive information. This form of attack relies heavily on social engineering techniques. For example, an adversary can send a text message to a certain bank customer claiming to be an employee and requesting for account details (e.g. credit card number). The adversary can then illegally use the credit card information to make unauthorized purchases.

The Smishing Attack which is a socially oriented attack cannot be solely mitigated by a technical authentication method such as the OSP. Therefore, social engineering mitigating techniques must be adopted at the organizational level. For example, security policies can be put in place which aims to educate customers and employees about the dangers and ways of avoiding Smishing Attacks (see chapter 5.3.4).

7.1.5. Vhishing Attacks

In the case of Vhishing attacks, the OSP is not designed to make use of voice recognition as the third factor needed for authentication. So an adversary will not be successful in using voice messaging system to try and deceive customers since the entire process of authentication is well known to them. Customers should to be educated about not giving their credit card details to any person or voice automated machine requesting them. In a nutshell, OSP alone is not enough to mitigate both Smishing and Vhishing and based on that, extra mitigating techniques must be adopted.

7.1.6. Man in the Browser Attack

Man in The Browser (MITB) attacks were specifically designed to beat strong multiple factor authenticating methods such as the OSP. This claim is supported by statistics which show that MITB attacks have been more frequent in countries (Germany, the Nether-lands, Spain, etc.) where two-factor authentication are used for online banking. The reason why the MITB have been successful against two-factor authentication is that it does not attack the authentication process itself. It rather allows a bank application to successfully authenticate a user, and then attacks the subsequent transactions that occur between the user and the bank. This, combined with its stealthy nature makes it impossible for the OSP and other multiple factor authenticating methods to protect against it without additional security measures. One of such additional security measure is out Out-of-band communication which aims to verify user transactions (not authentication) through other means and channels.

7.1.7. DoS Attack

Denial of Service (DoS) attacks are outside the scope of the OSP authentication method, and is another attack type which is known to be difficult to protect against. One of the basic reasons for this is the fact that DoS attacks are mostly done by carrying out legitimate operations in extremely large proportions (for example exhausting CPU time or network or memory resources). One solution to this problem could be planning for adequate resources to accommodate this extremely large amount of operations. This will definitely solve the problem but, such extreme amount of operations may never happen in the life time of the system, which means a mere waste of resources. Thus, an organization will have to make a tradeoff between security (preventing DoS attacks) and cost (providing only required resources) when designing their system.

7.1.8. Summary

Threats category	Threats	Attacks	Mitigation	OSP
Virus	• WinCE/Duts	N/A	N/A	N/A
Worms	• Commwarrior • iPhone-Ikee-B • Megoro	XSS	Filtering, Escaping	√
		DoS	Increase resources	X
Trojans	• Zitmo • Geinimi • JiFake	Smishing	Blocking, Educating, Access restriction	X
		MITB	OOB	X
Spyware	• Cell phone • Trusters • NeoCall	Eavesdropping	Scanning, Encryption	√
		Replay	OTP, Time stamping, MAC, Session token	√

Figure 16: OSP security overview (√ - Mitigation possible, X – Mitigation not possible)

7.2. Payment approach evaluation

The two approaches presented in chapter 6 are currently the most widely used forms of integrating payment capabilities into m-commerce sites. This is due to the suitability to various businesses and their large expanding user base. What approach is adopted depends on the size, time to market, security policies, category etc. of the m-commerce site. We have evaluated the two approaches based on their security and other important properties with a view of helping prospective adopters choose an appropriate approach. It is also important to note that the economy, technology, threats and payment methods are continuous evolving.

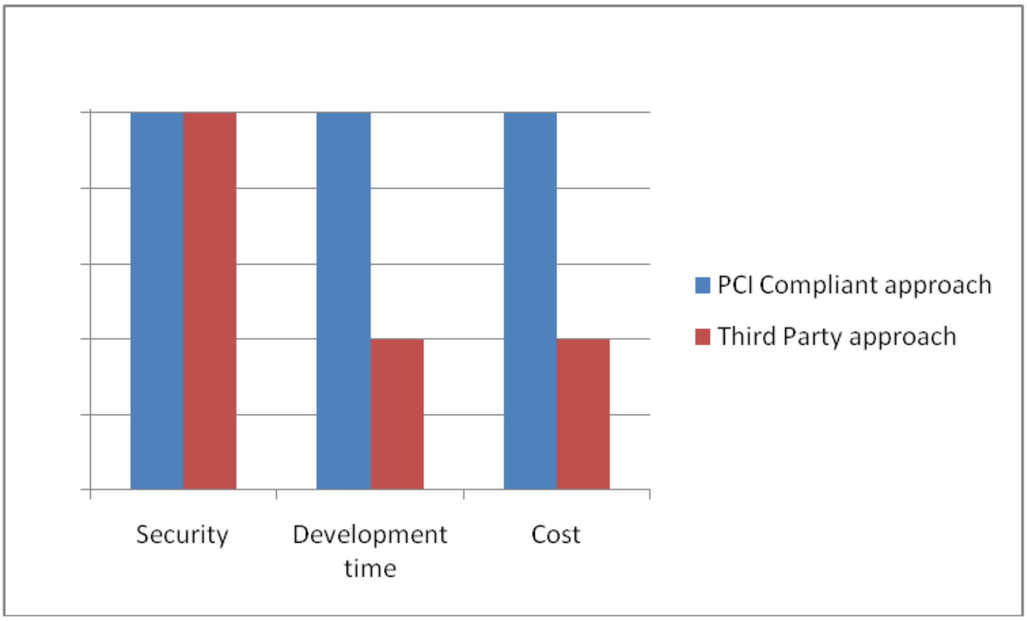


Figure 17: PCI compliant vs. Third party Approaches

7.2.1. Security

The security of the two different approaches above was evaluated based on the security requirements contained in the PCIDSS. PCIDSS was selected since it is a standard focused on the past and constantly evolving threats faced by payment systems and organizations. These requirements are also legally binding for any organization who wishes to conduct financial transactions electronically. Both approaches when adopted as instructed in this study can be said to meet the PCIDSS.

The PCI compliant approach as the name implies involves the implementation and compliance with the entire requirement stated in the PCIDSS documentation (see chapter 6.2.1). The third party approach also aims to be PCI compliant, but does this by using the services of third party payment sites (payment gateways) that are already PCI compliant. One gray area in the use these approaches is how well they are implemented. Implementing the PCI compliant approach requires a decent amount of security knowledge and constant assessment to ensure that all parts of the requirement are fulfilled. Therefore, evaluation should be an ongoing process that should be carried out periodically. This also applies to third party approaches, but the responsibility to ensure that the payment gateway is always PCI compliant, lies with the third party site. In the long run, the third party approach will be easier to implement and maintain. However, both approaches are equally secure if the implementations are done following the requirements to specifications.

7.2.2. Development Time

One important factor in the development of IT systems is time to market. Organizations are always trying to release their products as soon as possible, in order to gain an edge over competitors. This makes the development time of such products critical in the attainment of this goal. Adaptation of the PCI Compliant Approach is a lengthy process in comparison to the Third Party Approach. An m-commerce site hoping to integrate payment capabilities into their site within a short period of time is better off using the Third Party Approach. This conclusion is only valid for organizations implementing the payment systems for the first time where the expertise and competence of implementing the PCI requirements are not yet present. In cases where the organizations have used the PCI Compliant Approach a number of times on different systems, then the development time will the same or shorter than using the Third Party Approach.

7.2.3. Cost

It might be obvious at this stage what approach will be more expensive to implement, but it is still important to discuss the cost of the two approaches. IT security firms such as Emagined Security [56] and Fortrex Technologies [57]. have given three main categories for analyzing PCI compliance cost [58]. These include upgrading security infrastructure such as firewalls and antivirus software, undergoing an assessment of the organizations compliance with the PCI DSS and sustaining compliance with the PCI DSS. The cost of security infrastructure and sustaining compliance might vary to a large degree from one firm to another based on the difference in size and other factors. However, a report by Ponemon Institute in March 2010 showed the cost of assessment for large organizations to be in the range of $100,000 to $500,000 per annum [59]. All this show that there is a considerable cost involved in the PCI Compliance Approach. In the Third Party Approach, the cost of been PCI Compliant is negligible since the responsibility to be PCI Compliant resides with the third party gateway provider. The makes the Third Party approach a preferred choice when cost is an issue for the organizations requiring a payment solution.

8. Conclusion

The technology improvement and wide spread use of mobile phones has led to the growth of m-commerce. M-commerce applications are used in social networking, online stores, and financial applications. The work conducted in this study involved (1) proposing and implementing a suitable secure platform-independent authentication method and (2) implementation of a prototype platform-independent payment approach for m-commerce applications. The prototype implementation has shown that the proposed authentication method and third party payment approach can be implemented within a short time and with as little as two developers. This makes the authentication and payment methods appropriate for small to large m-commerce businesses. These goals were accomplished by investigating the following problems areas:

- ***What are the security threats that are currently faced by m-commerce systems?***

During our study, we discovered a number of security threats (see chapter 5) that are faced in the mobile industry. They include Viruses, Trojans, Worms and Spyware. These threats are continuously evolving into different variants in order to circumvent current protective security measures. The impact of these threats ranges from mere inconveniences like unwanted messages to severe implications such as identity theft, financial losses and national security breaches. It is not possible to have a threat-free m-commerce industry due to the tradeoffs that must be taken into consideration. Tradeoffs include compromises between security and performance or usability or costs. Based on the findings of the study, we have come to the conclusion that technology alone cannot be used to mitigate these threats. All practical solutions must include a security policy that covers and combines technology and social engineering best practices.

- ***What are the necessary security requirements that must be met by a platform-independent authentication and payment system?***

Authentication and payment systems must provide a high level of security due to the sensitive functions they perform. Studies carried out by researchers have stressed the need for a strong method of authentication due to the failures of weak form of authentications such as password systems (See chapter 2). In order to achieve a strong authentication method, it is necessary to identify and understand what needs to be protected, possible attacks, how to protect vulnerable points and a ways to detect attacks. These methods can also be made more resilient to attacks by incorporating multiple authentication factors and communication channels. Most importantly, the limitations of the authentication method used must be clearly understood. In the case of the proposed OSP authentication method, security of the method only involves authentication verification but not transaction verification. This is one of the reasons why it was only able to mitigate some of the threats identified in this study (See chapter 7.1.5), while other mechanisms (e.g. education, access restriction etc.) were required to provide mitigation against the remaining threats.

Requirements for payments systems are straightforward based on our findings. These have been properly documented in the PCI DSS document (See chapter 6.2). However, the challenges to organizations are the time it takes, cost involved and ensuring proper implementation of the requirements contained in the document. With that in mind, we recommend that small startup organizations will be better off starting with the third party approach. They can later change to the PCI compliant approach, when they are matured enough to adequately tackle the challenges stated above.

- ***What are the current authentication methods/solutions available?***

Based on the data collection and analysis carried out in this study, we were able to identify three major authentication methods. These methods include single factor, single sign-on and multiple factor authentication methods. It is no secret today the low level of security that single factor (password) authentication methods provide. Despite these shortcomings, this method is still widely in use today. Single sign-on involves the process where a user only needs to authenticate himself to a central third party, in order to be authenticated to other service providers. This authentication method provides ease of use to users, but it can also be argued that this is at the expense of security. If the communication between the user and the central authentication systems becomes compromised, then the authentication process with other service providers will be at risk. Multiple factor authentication (chapter 2.2 and 2.3) aims to solve the various problems associated with single factor methods by adding another layer of security to the authentication process. The method involves the use of something the user knows (e.g. OTP) and something the user has which improves the strength of the authentication process.

The methods can be found in use in several existing solutions or products today. These include AcrotOTP (e.g. Multiple factor), Accumulate Mobile (e.g. Multiple factor), Authentify (e.g. Multiple factor), WebSEAL single sign-on (e.g. Single Sign-on) and PayPal (e.g. Single factor). These solutions also incorporate other security mechanisms such as multiple channels, encryption, and certificates. These additional security mechanisms are required to cover other security vulnerabilities that the methods do not protect against.

- ***What are the current payment methods/solutions available?***

One of the aims of this thesis is to investigate and propose a suitable platform independent payment method for m-commerce application. During data collection, we came across several payment methods and solutions that could be adopted for the purpose above. It became obvious at an early stage of our research that the online payment service provider was what we needed to achieve a platform independent payment method. We came to this conclusion for two reasons (1) it supports a platform independent solution. (2) It is suitable for large and small m-commerce organizations based on its support for micro and large payment possibilities.

We were able to identify two payment solutions that could be used to provide a platform independent payment solution. These are the PCI compliant approach and the third party approach (chapter 6.2). These two solutions were evaluated with respect to security, development time and cost (chapter 7.2). The choice of what approach should be taken is depends on which of the above properties the organizations prioritizes. However, we recommend that organizations should aim to switch to a PCI compliant approach only when they become mature enough to undergo the compliance process. The reason for this recommendation is that implementing the security best practices contained in the PCI DSS (chapter 6.2), does not only improve the security of the payment process but also of the entire organization.

8.1. Current and future work

We have proposed the use of OSP as a way of achieving a secure platform independent authentication method for m-commerce applications. The security of this method has been evaluated by investigating how well it mitigates known attacks. These attacks include XSS, DoS, Smishing, MITB, Eavesdropping and Replay Attacks. The method successfully mitigates XSS, Eavesdropping but was not able to do eliminate Smishing, DoS and MITB attacks. The reason for this is that OSP aims to protect the process of verifying that a user is who he claims to be, while the three unmitigated attacks are directed at other areas outside the authentication process. Therefore, other mechanisms such as encryption, Out of Band calls are needed in order to protect against these attacks (chapter 7.1.5). To conclude, organizations should endeavor to combine the OSP (which contains security features such as multiple channels and factors, OTP etc.) with these other security mechanisms in order achieve greater security of their assets.

Future research can also focus on extending the OSP to include these mechanisms in order to protect against the three unmitigated attacks. More work should also be carried out to investigate how well the OSP authentication method mitigates attacks that were not covered in this study.

Reference

[1] Wen-Chen Hu, Jyh-haw Yeh, Hung-Jen Yang and Chung-wei Lee, "Mobile handheld devices for mobile commerce," Mehdi Khosrow-Pour, editor, *Encyclopedia of E-Commerce, E-Government and Mobile Commerce,* pp. 792-798, Idea Group Publishing, 2006

[2] Wikipedia. “Mobile Commerce”, modified in 2006 , cited on 2011 Mar 2nd , Available at: http://en.wikipedia.org/wiki/Mobile_commerce#History

[3] Sverker Brundin, “A Giant industry Grows”, created on 2011 Feb 15th, cited on 2011 Apr 2, Computer Sweden, Available at: http://www.idg.se/2.1085/1.368601/en-jatteindustri-vaxer-fram

[4] Aloul, F., Zahidi, S. & El-Hajj, W. (2009) “Two factor authentication in mobile devices.” in IEEE/ACS International conference on Computer Systems and Applications 2009, pp 641 - 644.

[5] Niu Ying, Zhao Yao & Zou Hua (2009) “The study of multi-level authentication-based single sign-on system”, in 2nd IEEE international conference on Broadband Network & Multimedia Technology, 2009 pp. 448 - 452.

[6] Do van Thanh, Jorstad, I., Jonvik, T. & Do van Thuan (2009), “Strong authentication with mobile phone as security token”, in 6th IEEE International conference on Mobile Adhoc and Sensors Systems, pp. 777 - 782.

[7] Soleymani, B. & Maheswaran, M. (2009), “Social Authentication Protocol for Mobile Phones”, in International Conference on Computer Science and Engineering 2009, pp. 436 - 441.

[8] Gianluigi Me Strangio, M.A. (2005) “EC-PAY- An Efficient and Secure ECC-based Wireless Local Payment Scheme”, in Third International Conference on Information Technology and Applications 2005, pp. 442 - 447.

[9] M. Jakobsson and S. Wetzel (2001), “Security weaknesses in Bluetooth”. Topics in Cryptology-CT-RSA 2001, LNCS 2020: pp176–191.

[10] P. Lekkas and R. Nichols. Wireless Security-Models, Threats and Solutions. McGraw Hill Telecom Professional, 2002.

[11] Smith, A.D. (2008) “Online payment service providers and customer relationship management’, in Int. J. Electronic Finance, Vol. 2, No. 3, pp.257–283.

[12] F-Secure Labs , created in 1988 ,cited on 2011 Feb 24nd ,Available at: http://www.f-secure.com/sv/web/home_se/home

[13] Symantec ,created in 1995,cited on 2011 Feb 24nd.Available at: http://www.symantec.com/index.jsp

[14] Mcafee Labs, created in 2003, cited on 2011 May 2nd, Available at: http://www.mcafee.com/us/

[15] John W Creswell, 2008, “Research Design: Qualitative, Quantitative, and Mixed Methods Approaches”, Sage Publications Inc.

[16] PayPal Account login, created in 1999, cited on 2011 May 3rd, PayPal, Available at: https://www.paypal.com/uk/cgi-bin/webscr?cmd=_login-run&dispatch=5885d80a13c0db1f8e263663d3faee8d422be6d275c375afb284863ba74d6cdc.

[17] Rick Wu, Rebecca Chen, Andrew Tsai & Deepak Kaul. “WebSEAL Single Sign-On with Telecom WAP 2.0/GPRS/3G gateways” ,created in 2007, cited on 2011 May 3, IBM developers works, Available at: http://www.ibm.com/developerworks/tivoli/library/t-ssotele/index.htm

[18] “AcrotOTP Secure One-Time-Pass code (OTP) Generator, Authenticate with your mobile phone”, cited on 2011 May 4th , Acrot CA technologies company, Available at: http://www.arcot.com/products/arcototp/

[19] “Accumulate Secure authentication Fact sheet” ,cited 2011 May 4th, Accumulate AB, Available at: http://www.accumulategroup.com/webb/pdf/FactSheetAccumulateAuthenticationSolution.pdf

[20] “Out-Of-Band Multi factor authentication .”, cited on 2011 May 6th , Authentify Inc , Available at : http://www.authentify.com

[21] “PayPal express checkout”, cited 2011 May 9th , PayPal X developer network, Available at : https://www.x.com/community/ppx/ec

[22] Google Checkout , created in 2011 ,cited on 2011 May 9th, Google, Available at: http://www.google.com/checkout/gettingstarted.html

[23] Authorize.net., created in 1996, cited on 2011 May 10th,Available at: http://www.authorize.net/

[24] Power VoIP, created in 2005, cited on the 2011 May 10th Available from: http://www.powervoip.com/en/index.html

[25] David Jurick, Adam Stolarz & Damien Stolarz,(2009) , “iPhone Hacks”, Published by make books, an imprint of Maker Media, April 2009.

[26] Symantec, “Trojan.SimbianOS.Skuller”, discovered on 2005 May 10th,cited on 12th,Available at: http://www.symantec.com/security_response/writeup.jsp?docid=2005-112115-0533-99

[27] The Sydney Morning Herald, “ The First Virus hatched as a practical joke”, created on 2007 Sept 3rd, cited on 2011 May 11th , Available at: http://www.smh.com.au/articles/2007/09/01/1188671795625.html

[28] Infoniac.com ,“List of Computer Viruses developed in 1980”, created on 2009 Sept 9th, cited on 2011 May 11th, Available at: http://www.infoniac.com/hi-tech/list-of-computer-viruses-developed-in-1980s.html

[29] Symantec , “ WinCE.Duts”, :: , cited on 2011 May 12th , Available at : http://www.symantec.com/security_response/writeup.jsp?docid=2004-071710-2120-99&tabid=2

[30] F-secure Labs, “Commwarrior”,:: , cited on 2011 May 11th, Available at : http://www.f-secure.com/v-descs/commwarrior.shtml

[31] F-secure Labs, “iPhone Ikee-A” , :: , cited on 2011 May 12th,Available at : http://www.f-secure.com/v-descs/worm_iPhoneos_ikee.shtml

[32] Phillip Porras, Hassen Saidi & Vinod Yegneswaran, “ Analysis of the Ikee-B (Duh) iPhone botnet ” , a SRI International Technical Report, created on 2009 Dec 21st, cited on 2011 may 12th, Available at : http://mtc.sri.com/iPhone/

[33] Symantec ,“iPhone Ikee-B”, ::, Nov 22, cited on 2011, May 12th, http://www.symantec.com/security_response/writeup.jsp?docid=2009-112217-4458-99&tabid=2

[34] Juniper Global Threat Center , “Megoro”, created on 2010 Aug 10, cited on 2011, May 12th , Available at http://globalthreatcenter.com/?p=1898

[35] McAfee, “ Zitmo”, created on 2010 ,Oct 13 , cited on 2011 ,May 13th, Available at: http://home.mcafee.com/VirusInfo/VirusProfile.aspx?key=290717

[36] McAfee , “Geinimi”,created on 2010, Dec 31st, cited on 2011, May 13th, Available at : http://globalthreatcenter.com/?p=2056

[37] McAfee , “JiFake”,created on 2010 Oct 3rd, cited on 2011, May 13th , Available at :http://globalthreatcenter.com/?p=1875

[38] Juniper Global Threat Center, “ Cell phone Recon”, created on 2010 ,Sept 28th, cited on 2011 May 14th, Available at : http://globalthreatcenter.com/?p=1944

[39] Juniper Global Threat Center, “Trusters Spy Phone”, created on 2010,April 20th ,cited on May 14th , Available at :http://globalthreatcenter.com/?p=1865

[40] Juniper Global Threat Center , “NeoCall”, created on 2010 Sept 20th , cited on 2011, May 14th,Available at: http://globalthreatcenter.com/?p=1861

[41] Acunetix Web Application Security, “Cross-site scripting (XSS) Attack”, ::, cited on 2011 May 15th,
Available at: http://www.acunetix.com/websitesecurity/cross-site-scripting.htm

[42] Yi-Bing Lin & Meng-Hsun Tsai (2007), “Eavesdropping through Mobile Phone”, IEEE Transaction on Vehicular Transaction, 2007”, vol. 56, no. 6, pp 3596-3597.

[43] David Kim & Michael G. Solomon (2010), “Fundamentals of Information System Security”, published by Jones &Bartlett Learning, LLC, pp 104.

[44] Emmett Dulaney, “CompTIA Security + Deluxe Study Guide”, John Wiley and Sons, 2008, pp 62.

[45] RSA Security Inc, a Security Division under EMC, “Phishing ,Vishing and Smishing: Old Threats present new Threats”, created in 2009, cited on 2011, May 16th, Available at:
http://www.rsa.com/products/consumer/whitepapers/10538_PSV_WP_1109.pdf

[46] RSA Security Inc, a Security Division under EMC, “Making Sense of Man in the browser Attacks, Threats Analysis and Mitigation of Financial Institution”, Created in 2010, cited on 2011, May 17th, Available at:
http://viewer.media.bitpipe.com/1039183786_34/1295277188_16/MITB_WP_0510-RSA.pdf

[47] Emmett Dulaney, “CompTIA Security + Deluxe Study Guide”, John Wiley and Sons, 2008, pp 56.

[48] Pullicino Jeremy, “Preventing XSS Attacks”, created on 2011,March 22, cited on 2011 May 15th, Acunetix Web Application Security ,Available at: http://www.acunetix.com/websitesecurity/cross-site-scripting.htm

[49] “Microsoft Web Protection Library”, created on 2010 may 12th, cited on 2010,May 20th, Available at: http://wpl.codeplex.com/

[50] “HTML Purifier”, cited on 2011,May 20th, http://htmlpurifier.org/

[51] CellSynt mobile services, “ Integrating with CellSynt SMS gateway via HTTP interface (technical documentation)”, :: , cited on 2011, May 20th, Available at : http://www.cellsynt.com/sv/

[52] PCI Security Standards Council, “PCI SSC Data Security Standard Overview” , :: , Cited on 2011, May 20th,
Available at: https://www.pcisecuritystandards.org/security_standards/index.php

[53] PCI Security Standards council(2008 version 1.1), "Understanding the intent of the Requirements" , cited on , Available at : https://www.pcisecuritystandards.org/pdfs/navigating_pci_dss_v1-1.pdf,

[54] Shift 4 Secure Payment Processing, "Credit Card 101", cited on 2011, May 20th, Available at : http://www.shift4.com/players.cfm

[55] PayPal Mobile Express Checkout ,cited on 2011,May 20th, Available at :https://www.paypal.com/se

[56] Emagined Security, Cited on 2011 May 23rd, Available at :http://www.emagined.com/

[57] Fortrex Technologies, Cited on 2011 May 23rd, Available at : http://www.fortrex.com

[58] PCI DSS Compliance Blog, "Cost of PCI Compliance" , created on 2009,Feb 2nd, cited on May 23rd, Available at: http://blog.elementps.com/element_payment_solutions/2009/02/pci-compliance-costs.html

[59] Ponemon Institute, "PCI DSS Trends 2010: QSA Insights Report Recommendations and guidance for achieving compliance from Qualified Security Assessors", created on 2010, March. ,cited on May 23rd, Available at : http://www.ponemon.org/local/upload/fckjail/generalcontent/18/file/PCI%20DSS%20Trends%20-%20QSA%20Insights%20010310.pdf