Digital Signature on Web Form Transaction

by

Mr. Sithisak Chotepornsiri

A Final Report of the Six-Credit Course
IC 6998 E-Commerce Practicum

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Internet and E-Commerce Technology
Assumption University

November 2002

159433

46

Digital Signature on Web Form Transaction

by
Mr. Sithisak Choteporнsiri

A Final Report of the Six-Credit Course
IC 6998 E-Commerce Practicum

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Internet and E-Commerce Technology
Assumption University

November 2002

Project Title	Digital Signature on Web Form Transaction
Name	Mr. Sithisak Choteppornsiri
Project Advisor	Rear Admiral Prasart Sribhadung
Academic Year	November 2002

The Graduate School of Assumption University has approved this final report of the six-credit course, IC 6998 E-Commerce Practicum, submitted in partial fulfillment of the requirements for the degree of Master of Science in Internet and E-Commerce Technology.

Approval Committee:

[illegible]
(Rear Admiral Prasart Sribhadung)
Advisor

[illegible]
(Prof.Dr. Srisakdi Charmonman)
Chairman

[illegible]
(Dr. Ketchayong Skowratananont)
Member

[illegible]
(Assoc.Prof. Somchai Thayarnyong)
MUA Representative

November 2002

ABSTRACT

This project proposes the enhancement and implementation on the security of an E-commerce site, https://www.smtmobile.com, an online cyber mobile store. The purchasing order transaction via web form is digitally signed by customers and consequently verified for its integrity by the webmaster.

The project creates an online web signed-form model to demonstrate how online web signed-form works and compares the valid versus invalid verification procedure output. This online web signed-form model consists of two main parts. First, a shopping cart is designed by ASP program and run on the IIS 5.0 Windows XP Professional web server. Second, signing and verifying module is operated by namely signform object.

In this project, the customer's certificate issued by the certificate authority (CA) is used to sign the data of the web form before it was sent to the web server. Moreover, the webmaster can verify the integrity of customer's signed data so that the customer cannot repudiate the transaction. The signing and verifying process is operated at the browser of client's side by running the especial ActiveX component object. The object is written in VC++ language and supported the CryptoAPI library by Microsoft platform SDK.

This is a pilot project in Thailand to implement the usage of digital signature via web form. Furthermore, the project also demonstrates the difference of the plain data approach and the signed data approach when the data is supposedly tampered.

ACKNOWLEDGEMENTS

I am indebted to the following people for without them, this project would not have been possible.

I wish to express sincere gratitude to my advisor Rear Admiral Prasart Sribhadung. His patience, guidance and constant encouragement has led me to this completion. I would like to take this opportunity to thank the entire faculty that taught me during my Master of Science in Internet and E-commerce Technology. The knowledge that I acquired from them indeed helped make this a successful report.

Finally, I would like to thank my parents, brother and friends for their support and patience throughout the project.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Background of the Project

Internet Technology has opened up many exciting possibilities for organizing and running a business that is transforming organizations and the use of information systems in everyday life. It creates platform for buying and selling goods or services and for driving important business processes inside the firm as "Electronic Commerce." Along with bringing many new benefits and opportunities, the internet has provided customers for convenience, time saving, and etc. At the same time, vendors can take more advantages from cost saving, comfortable, paperless, and so on.

Among the new challenges and more benefits of the Internet coming up with many online businesses, there, however, are many differences between commerce in the real world and in the cyber world such as business documents, contract, signature, and others. In order to attempt making the online stores like as physical stores, trusting the system as well as adopting electronic documents have become important factors to create reliability and reduce obstacles on electronic commerce.

1.1 Objectives of the Project

(1) Enable to apply the knowledge in the field of security of the Master of Science in Internet and Electronic commerce Technology into the Electronic commerce Practicum Project.

(2) To support the Electronic commerce transaction and digital signature Act.

(3) Save time and money by eliminating paperwork and paper-based signatures.

(4) To prevent a transaction from being denied or challenged.

(5) Verifies the integrity of customer-submitted data as it is received, ensuring it has not been altered.

(6) Verifies the validity of the signer's digital certificate, providing real-time user authentication.

(7) Archives received information before processing for easy retrieval and validation.

1.2 Scopes of the Project

(1) Conduct online business transaction to be digitally sign through web form.

(2) The online transaction form on electronic business is loaded from the server side of vendors. People can reach to the form at anyplace, anytime, and any where around the world.

(3) Only fill out data in each field are digitally signed.

(4) User's certificate including his/her own private key is kept on the client side. Two storage types of certificate can be alternatively recognized and utilized. First, certificate is retrieved from MS Outlook Express in case of using own PC machine and already having existing user's certificate. At another, file in PKCS#12 format will be loaded from any portable media.

(5) The supported browser is MS Internet Explorer.

1.3 Traditional Business Process Overview

Before the days of computers, most businesses have been either physical or directly related to personnel such as communication channel, business documents and contract, security, written signature, and so on. Business communication channels fall into face to face and post. Both parties can identify and create reliable for each others in making business transaction.

In aspect of documents, contracts and customers database, they are kept in type of paper-based. These documents have a legal effect and can be used as evidence in the courts in case of dispute occurred. In addition, security system can be easily maintained because companies only look out for losing in documents from unauthorized and outside people.

For written signature, it is used to provide both legal and business assurances through the following:

Signifying intent: The essence of a signature is to identify the signer and signify that the signer intended to carry out what was stipulated in the signed document.

Authentication and approval: A signature authenticates a document by linking the signer with the signed document. A signature may also express the signer's approval or authorization of the document and what it contains, as well as his or her intent that it has legal effect. The signature provides evidence that the signer really did something and actually saw and approved a particular document at the time of signing.

Security: Signatures are often used as a means of security. For instance, to a limited degree, the signature on a check is a form of security because drafting unauthorized check often requires forging a signature. A signature on a document often imparts a sense clarity and finality to the transaction and may lessen the subsequent need to inquire beyond the faced of a document.

Ceremony: The act of signing warns or puts the signer on notice that he or she may be making a legally binding commitment. The signature will show that a meaningful act occurred when the person deliberate over the document, become aware of its significance, and memorialize the document's finality.

According to the above, we notice that most of businesses operating processes are directly related to personnel and face to face.

1.4 Obstacles on Cyber Shop System

When business is conducted over a computer network, as in electronic commerce, business communication and operating processes don't depend on personnel and physical way onwards. Both parties can make an online transaction in different place at the same time by employing network infrastructure. Under this situation, both the customer and the vendor have difficulty proving their identity to each other with certainty, particularly during a first transaction.

Many questions are doubtfully asked comparing with commerce in real world. How does the buyer securely transmit sensitive information to the seller? How does the seller know that this is a legitimate purchase order? How do both parties know that a nefarious third-party hasn't copied and/or altered the transaction information?

These questions, and others, describe the problems affecting commercial transactions over the Internet, or any public network. Customers or clients need to be sure that:

(1) They are communicating with the correct server.

(2) What they send is delivered unmodified.

(3) They can prove that they sent the message.

(4) Only the intended receiver can read the message.

(5) Delivery is guaranteed.

However, vendors or servers, on the other side of the commerce transaction, need to be sure that:

(1) They are communicating with the right client.

(2) The content of the received message is correct.

(3) The identity of the author is unmistakable.

(4) Only the author could have written the message.

(5) They acknowledge receipt of the message.

From all of the concerns listed above, they indicate that electronic commerce requires security application and trust system between vendors and customers.

1.5 Digital Signature and CA Applied to Web Form Transaction

As aforementioned on obstacles affecting commercial transaction, the concepts of basic security services are specified to cover solving problems of cyber world commerce. These four basic security services are integrity, confidentiality, identification and authentication. They may be necessary in a particular application.

Confidentiality services restrict access to sensitive data to only those individuals whom are authorized to view the data. Confidentiality measures prevent the unauthorized disclosure of information to unauthorized individuals or processes.

Integrity services address the unauthorized or accidental modification of data. This includes data insertion, deletion, and modification. To ensure data integrity, a system must be able to detect unauthorized data modification. The goal is for the receiver of the data to verify that the data have not been altered.

Identification and authentication services establish the validity of a transmission or message, and its originator. The goal is for the receiver of the data to be able to determine its origin.

Non-repudiation services prevent an individual from denying that previous actions have been preformed. The goal is to ensure that the recipient of the data is assured of the sender's identity.

All four basic security services listed above are recognized to develop some security application. Digital signature and can be applied by using some combination of cryptographic methods. But cryptography by itself is not enough; we still have the issue of trust as it applies to sharing and managing the keys we needs for cryptography to work. The solution to this problem is to use certificates.

Digital signatures are frequently regarded as the electronic equivalent of hand-written signatures. However, whereas a hand-written signature is additional to the message being sent (e.g. at the end of letter), intrinsic to the signatory and should be recognizably the same on all messages, a digital signature is message dependent. In fact, a digital signature is data that depends on the message and a secret parameter known only to the signatory.

The objectives of digital signatures are to enable the recipient to prove the identity of the sender and guarantee the integrity of the data being transferred. A digital signature must be easy to create, easy to verify, and difficult to forge.

A digital signature is usually represented as a string of bits. The signature may be appended to a message or the message may form an integral part of it. Thus, by its very nature, it differs from a written signature. Typically, a digital signature is produced by a machine or computer program. The signer merely provides some input to the process, and it is this input that determines what particular pattern of bits makes up the digital signature. The aim is that no other person should be able to produce the same pattern of bits, which means that they must not know the signer's input. Thus it is the use of the signer's input which identifies him or her. The signer's input may be some information that only the signer knows or a physical characteristic such as fingerprint.

Since it is the use of the signer's input, which is the identifier, if the input is secret information then the signature only identifies the signer as someone can obtain that information, then the person will be able to impersonate the signer.

If someone wants to impersonate you, then they can either obtain the use of your secret key, so they will produced the same signature value as you, or they can try to convince other users that their public key belongs to you. In the latter case they will not be forging hour signature, since the number they produce will be different from that produced by your secret key. Nevertheless, the number that they produce will be accepted by the public as coming from you because the public believes that their public key is owned by you.

In order to prevent the first type of attack, we need strong algorithms and strong physical protection over secret keys. In order to prevent the second type of attack we need to introduce the concept of signature certificates, together with a complete supporting infrastructure: the so-called Public Key Infrastructure (PKI). A central concept of PKI is that of a Certification Authority (CA).

A certificate authority is a trusted third-party organization or company that issues digital certificates used to create digital signatures and public-private key pairs. The role of the CA in this process is to guarantee that the individual granted the unique certificate is, in fact, who he or she claims to be. It is vital to the reliability of the entire process.

According to all of the above, coming up to the project, we employ mechanisms of digital signature and certification authority (CA) to apply with electronic transaction form on the Web in cyber world commerce. This provides both of customers and vendors making online transaction trustingly and reliably including wipe out some obstacles of security and trust system on electronic commerce.

II. PRINCIPLE OF DIGITAL SIGNATURE

This chapter widely provides a technical overview of principle of digital signature. We will discuss specific mechanism, to describe how they are used in applications and investigate the problems that arise when implementing these mechanisms in a business environment.

It is perhaps helpful to first place digital signature mechanisms within the broad scope of information security. Information security is concerned with providing assurances over confidentiality, integrity and availability. Physical techniques have traditionally been sufficient to provide information security. However, today more and more data is being stored and communicated in digital form. Such data has very different characteristics than data stored on paper. Physical techniques do not appear capable of providing effective security for data stored in digital form. Cryptography provides an alternative or additional means of providing information security.

Cryptographic schemes provide assurances about the security of data by transforming the data itself. Therefore, cryptographic schemes are more versatile than traditional physical security mechanisms because their operation is independent of the medium on which the data is stored and independent of the physical protection provided for the data. They appear well suited to ensuring the security of data stored in electronic, or digital form.

Digital signature schemes are examples of cryptographic schemes. These schemes were proposed in 1976 by two Stanford researchers, Whitfield Diffie and Martin Hellman. Digital signatures are designed to provide aspects of assurance such as data origin authentication, data integrity, non-repudiation. Traditionally these services were supplied to data stored on paper by appending a handwritten signature to the document.

Digital signature schemes are therefore analogous to the process of forming and checking handwritten signatures, hence the nomenclature. An implementation of a digital signature scheme is known as a digital signature mechanism.

2.1 Digital Signature Mechanisms

2.1.1 Digital Signature Scheme

The basic principle of a digital signature scheme is that a user has a secret key (SK) which is known only to them, and it is accepted that use of the value SK identifies that user. Corresponding to each SK, is a public key (PK), which can be given to everyone, and can be used to confirm that SK has been used. However, possession of PK does not enable the computation of SK.

The technical specification of a digital signature scheme involves the description of three computational procedures or algorithms. These are key generation procedure, signing procedure, and verifying procedure. For key generation procedure, the key generation procedure is employed to generate the keys that are used by the signing procedure. Each time it is used the procedure generates a key pair consisting of a signature key and the corresponding verification key. It is important to note that the key generation procedure uses a random number generator and will generate a different pair each time it is used.

Because in applications the signing key is kept secret, SK is often known as the secret key. Similarly, because in applications of most signature schemes, the verification key is distributed to all users who want to verify signatures, PK is often known as the public key.

The next, signing procedure, the signing procedure transforms the data to produce a signature. Each time it is used the procedure takes as input a signature key generated using the key generation procedure and data form some predetermined data space. The signing procedure outputs a signature. In practice both messages and their signatures will usually be bit strings.

If the signing procedure of a digital signature scheme is probabilistic, meaning that each message may have a variety of valid signatures, then it is called a probabilistic signature scheme. If the signing procedure of a digital signature scheme is deterministic, meaning that each message has only one valid signature, then it is called a deterministic signature scheme.

The last, verifying procedure, users who receive signed messages need to be able to check that the signature appended to the message is correct, in the sense that it is a value which would be produced if the signing procedure was applied to the received message using the sender's signing key. The verifying procedure takes as input the received message and signature together with the public key of the purported sender and then either accepts or rejects the signature. If the verifying procedure outputs 'Accept,' then the message is accepted as valid; otherwise it is rejected as invalid.

The procedures are summarized in the following diagram, Figure 2.1.

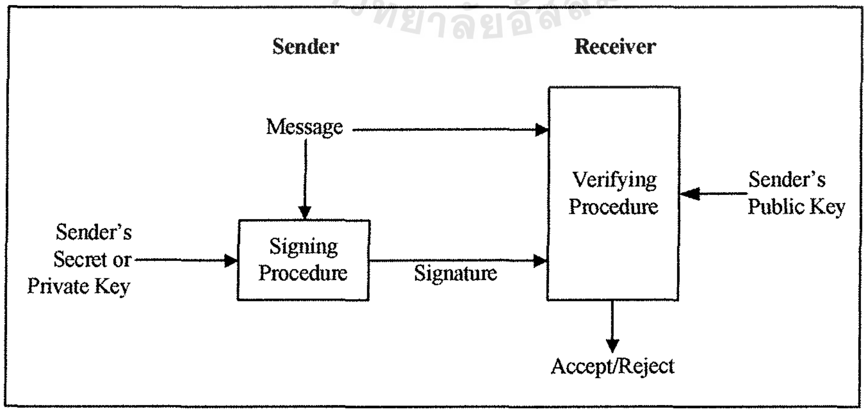


Figure 2.1. Key Generation, Signing and Verifying Procedures.

2.1.2 Properties for Providing Security Services

In cryptography it is traditional to describe schemes operated by two users named Alice and Bob. In what follows, assume that Alice is signing the message and is sending it to Bob to verify.

The first and most obvious requirement is that the three procedures involved must be *efficient*. What efficient means in practice will vary from application to application, but broadly speaking it means that Alice and Bob must be able to operate the key generation procedure, the signing procedure and the verifying procedure in a reasonable amount of time and at a reasonable cost.

The second requirement is that the underlying signature scheme is *well-defined*. A well-defined signature scheme is one in which the verifying transformation does indeed accept that the signed messages that Alice herself generates are valid. Clearly this requirement is necessary for the effective operation of the signature mechanism; otherwise Bob will not accept that Alice's genuine signed messages are valid.

The final requirement is that the signature mechanism is *secure*. Just as with handwritten signatures, digital signatures are secure if forgery is hard. However, forgery in the digital case differs from forgery in the handwritten case. Since digital data is represented by binary strings and is easily duplicated, it is not sufficient for Alice to simply append a fixed string to each message - in this case an adversary could forge Alice's signature simply by copying the string and appending it to any message. Instead digital signatures must be message dependent and only Alice should be able to produce valid signed messages. This means it must be infeasible for anyone apart from Alice to produce a signed message that a receiver will accept as coming from her.

We can now demonstrate why digital signature mechanisms which possess the three properties described above, provide the services of data origin authentication, data integrity and non-repudiation. Suppose that Bob has received a message *M* with a signature $\sum$ that he believes has come from Alice. If Bob is convinced that he has the correct value of Alice's public key then he can use that value to verify that $\sum$ is indeed Alice's signature for the message *M*. This gives him the assurance that *M* originated from Alice and has not been altered, since it is infeasible to forge valid signed messages. Data origin authentication and data integrity are therefore provided. Furthermore, if Alice later denies signing *M*, Bob can present authenticated $(M, \sum)$ to a third-party. The third-party verifies $(M, \sum)$ using Alice's public key and concludes that Alice must have signed *M* herself because it is infeasible to forge singed messages. Thus theoretically non-repudiation is also provided. Of course, in real life, non-repudiation can only be provided within an established legal framework. Since all the information used by Bob to check the signature is public, if Alice later denies signing the message, then any third-party will be able to perform the same calculations as Bob to check the validity of Alice's signature.

2.1.3 Types of Digital Signature Mechanisms

A number of criteria can be used to classify digital signature schemes: symmetric versus asymmetric schemes, true versus arbitrated schemes, schemes with or without message recovery and one-time schemes. The relationship between signing or secret keys (SK) and verifying or public keys (PK) determines whether a scheme is symmetric or asymmetric. If SK and PK are the same, or if it is easy to derive SK from PK, then the digital signature scheme is symmetric. If it is hard to derive SK from PK, then the digital signature scheme is asymmetric.

The way verifying keys are distributed during the operation of a scheme determines whether it is a true digital signature scheme or an arbitrated digital signature scheme. As one example, if Alice's verifying key PK_{Alice} is given directly to Bob so that he can use it to verify Alice's signatures himself, then the scheme is a true digital signature scheme. If PK_{Alice} is given to a trusted third-party or arbiter and this arbiter verifies signatures on behalf of Bob, then the scheme is an arbitrated digital signature scheme.

2.1.4 Asymmetric Digital Signature Schemes

The relationship between keys in a pair in asymmetric signature schemes, namely that SK is hard to derive from PK, means that the schemes are used precisely as described earlier.

To enable the operation of the scheme, Alice securely publishes her verification key PK_{Alice} so that it can be retrieved by users who want to verify her signatures. Because SK_{Alice} is hard to derive from PK_{Alice} it does not matter that an adversary may also gain knowledge of PK_{Alice} during this process. Note however, that the authenticity of the published copy of PK_{Alice} must be ensured.

Because PK_{Alice} can be published in this way, Bob is able to verify Alice's signatures himself. There is no apparent need for an arbiter. Arbitrated asymmetric signature schemes offer no added utility, and so we will only discuss true asymmetric signatures.

2.1.5 Building an Asymmetric Signature Scheme

A number of constituents are combined to form asymmetric signature schemes. One essential constituent in any asymmetric signature scheme is a mathematical function known as a one-way function. Broadly speaking, this function underpins the curious relationship between signing keys and verifying keys; namely that PK accepts as valid signatures formed using SK even though it is hard to derive SK from PK. Other constituents common to many or all asymmetric signature schemes are cryptographic hash functions, redundancy functions, random number generators and pseudorandom number generators.

The security of an asymmetric signature scheme relies on each of these constituents functioning effectively. We will therefore describe the requirements of each constituent in more detail. They entail the following:

Hash functions: Hash functions map bit strings of arbitrary length onto bit strings of a fixed but short length. Hash functions are used in asymmetric schemes when signatures of long messages are required. Instead of breaking the long message into short sections and signing each section individually, the long message will be hashed and the resulting hashed value is then signed.

One-way functions: Loosely speaking, one-way functions are easy to compute in one direction, but hard to compute in the other. Unfortunately, no mathematical functions have been proven to be one-way functions. The best that can be done is to use functions believed to be one-way functions.

Redundancy functions: Redundancy functions are applied to bit strings (usually messages). They are used primarily to format messages before they are transformed by the mathematical one-way functions described above. Redundancy functions must be both secure and efficient. The security requirement for redundancy functions is usually to embed the message space sparsely within the range of the mathematical function and to hide the algebraic structure of the underlying mathematical function.

Random number generators: Random number generators are used to produce unpredictable bit strings. They are essential to the provision of secure key generation procedures. Many signing procedures also require random number generation. Random number generators must also be both secure and efficient. As we have already mentioned, for proper security, random number generators must be unpredictable; that is an adversary must be unable to predict the bit strings produced. A wide variety of random number generators are used in practice.

Pseudorandom number generators: Pseudorandom number generators take as inputs a truly random bit string and output a substantially longer unpredictable bit string. Because random number generators are not particularly efficient, they are often employed in combination with pseudorandom number generators to improve the efficiency of the production of unpredictable bit strings.

Pseudorandom number generators must again be both efficient and secure. From the point of view of an entity who knows the random seed input to the pseudorandom number generator, the output is entirely predictable. The security requirement for pseudorandom number generators is that their output is unpredictable from the pint of view of an adversary who does not know the random seed. This should remain true even if the adversary sees a portion of the output of the pseudorandom number generator it should still be hard to predict the rest of the output.

As we have already mentioned, asymmetric signature schemes are formed from combinations of these constituents. The security and efficiency of the schemes is dependent upon the constituents.

2.2 Digital Signature Applications

Digital signatures are used to provide some or all of data integrity, data origin authentication and non-repudiation for messages or data. They are also used as a means of establishing one's identity to another entity. The presentation of a digital certificate does not establish the identity of the person producing that certificate. Instead it establishes the identity of the owner of the secret key corresponding to the public key listed in the certificate. It is important to remember that certificates may be copied without limitation and distributed widely. In this topic, a few specific applications of digital signatures are examined.

2.2.1 Secure Distribution of Symmetric Keys

Because of advantages in processing requirements and performance, symmetric key systems are the system of choice for information encryption and other cryptographic services. One of the most common applications of digital signature is authentication of the sender and protection of the integrity of key distribution messages for symmetric keys.

When the key distribution authority (the sending user or application) wishes to distribute a symmetric key it:

(1) Generates the symmetric key

(2) Forms the key into a key distribution message, containing the key and, optionally, other information (the identity of the key, its defined use, etc.)

(3) Sends the message, encrypted using the recipient's public key (so only the recipient can decrypt it), and digitally signed using the sender's secret key (so the recipient is sure that the message came from the claimed user and was not changed in transmission).

On receipt of the message, the recipient verifies the signature using the sender's public key which he or she obtains from a certificate included with the message or accessed form a directory. The receiver then decrypts the message using its own secret key and extracts the symmetric key.

The recipient usually obtains the sender's public key it requires to perform this procedure from the sender's certificate. The certificate used to verify the sender's public key will normally be signed by the CA next higher in the hierarchy. This CA's signature is verified using the CA's public key obtained from a certificate signed by the CA next highest in the hierarchy. This verification process continues until the chain is completed up to the Root CA. The Root CA's public key is accepted as being valid by means outside the public key infrastructure.

The symmetric key generation, encryption and decryption is often performed by a processor housed in a tamper resistant enclosure. The above protocol is illustrated in Figure 2.2.

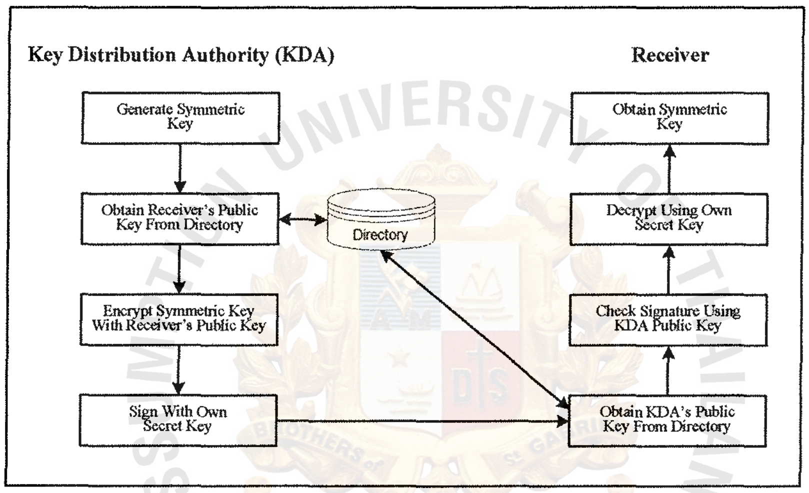

Key Distribution Authority (KDA)
Receiver
Generate Symmetric Key
Obtain Receiver's Public Key From Directory
Encrypt Symmetric Key With Receiver's Public Key
Sign With Own Secret Key
Directory
Obtain Symmetric Key
Decrypt Using Own Secret Key
Check Signature Using KDA Public Key
Obtain KDA's Public Key From Directory

Figure 2.2. Digital Signature Application.

In discussing the distribution of symmetric keys we have used a public key algorithm to provide both confidentiality and digital signatures. For the former, the sender uses the public key of the receiver and the receiver then uses their secret key to reverse the process. For digital signatures, the sender uses their own secret key and the receiver then uses the sender's public key to check that the secret key is used. The order in which the keys are used for each mechanism is important and is illustrated in Figure 2.3.

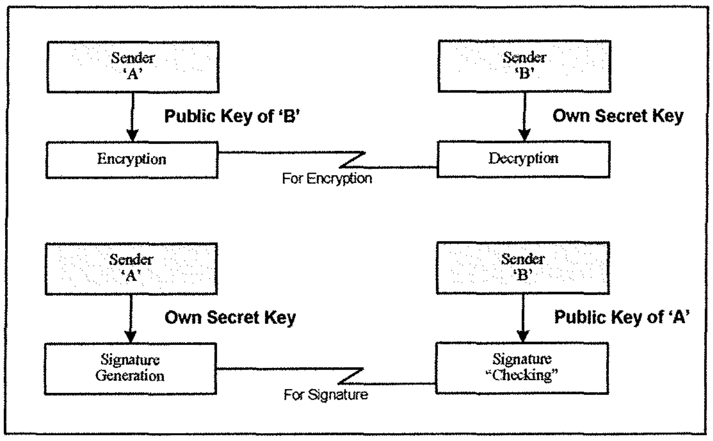


Figure 2.3. Key Order Illustration.

2.2.2 Secure Browsing – SSL

Secure Sockets Layer (SSL) is a protocol designed to extend the transmission Control Protocol (TCP) to establish a secret session key for use by a cipher in order to achieve integrity and optional confidentiality of the TCP connection between two computers. Its dominant use is to provide a secure channel between a Web browser and a server to which it connects. Such a secure channel is used in practice for business applications including home baking, payment by credit card and extranet applications where one organization allows controlled access of their web site by others using the Internet as a communications medium.

SSL connections are originated by the requester (usually a client) and the authenticity of the responder (usually the server) is achieved by the use of certificates and digital signatures during the setup of an SSL connection and before any user data has been exchanged. Also during the setup of the connection, a server may optionally request the client to authenticate itself. SSL requires the existence of a public key infrastructure (PKI) for an SSL participant to verify the authenticity of the credentials. The data format used to encode the credentials and the digital signature of the

Certification Authority (CA) which make up a certificate, is taken from International Telecommunications Union (ITU) standard X.509. The data format of the digital signature used during the SSL protocol is taken from RSA Public Key Cryptography Standards (PKCS) for encoding digital signatures. To verify the correctness of the server's certificate, the client requires the CA's signature verification key (pubic key). For SSL enabled Web browsers this is usually distributed in the form of a CA self-signed certificate as part of the program. Other information such as certificate expiration date is also included.

To achieve authenticity, SSL requires some recognizable credentials to be included in the certificate sent from the server to client and optionally form the client to the server. The CA specifies the form of the credentials and these may be different for server certificates and client certificates. The credentials of a server include, in addition to the full text of the ISP company name providing the service, the Internet fully qualified domain name (FQDN) of the server.

2.3 Security and Control Issues

Most systems employ a Public Key Infrastructure involving Certification Authorities to prevent impersonation attacks. The next, we discuss why Certificate Authorities are needed and look at how they might operate.

2.3.1 The Need for Certification Authorities

In order to provide users with the ability to obtain authentic copies of other users' public keys most systems rely on Certificates and Certification Authorities.

A certification Authority (CA) is a trusted entity who issues certificates which bind an entity E to its public key value PK_E. Each digital certificate which consists of a message containing E's identity or in some applications some other attribute of E, and a public key value, is digitally signed by the CA. Anyone who is in possession of E's certificate and the CA's public key can now obtain assurance of the authenticity of E's public key by verifying the CA's signature on the certificate.

This solution to the public key distribution problem is attractive for a number of reasons. If for instance, there is a community of users who all trust a common CA, then their overhead is minimized because the distribution of all users' authentic public keys is reduced to the distribution of one CA's authentic public key. However, users are totally reliant on the integrity of the CA and must trust the CA to sign only bona fide certificates. Provided this trust is justified, they can have confidence in the identity of the owners of public keys and avoid impersonation attacks.

The current expansion of the use of public-key cryptography and, in particular, digital signatures, is leading to a need for CAs. However the role of a CA is complicated by the difficulties inherent in the operation of a CA. These difficulties stem from the stringent requirements involved. As a primary requirement, a CA will need to generate and distribute certificates securely and efficiently. As a secondary requirement, in some applications a CA will also need to be seen to act securely. For example, when certifying signature keys, a CA may become entangled in repudiation disputes and face liability if ti is unable to demonstrate the security and validity of its actions. Added to these concerns is the additional requirement of a commercial CA to make money. This is not straightforward because the person gaining most from a certificate is the user who uses it to authenticate a public key whereas most of the expense is incurred by the CAs.

2.3.2 Operation of Certification Authorities

Now, we are considering the kind of processing required during the certification process so that impersonation attacks can be avoided.

The following diagram, Figure 2.4, is a very simplistic model of the certification process. It assumes that users generate their own key set which, of course, is only one of the options available.

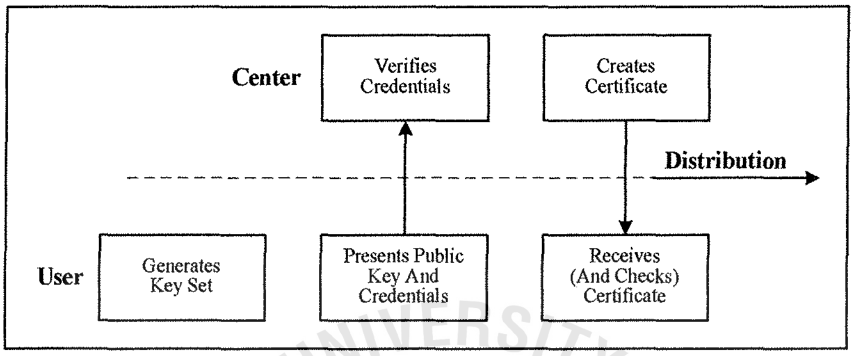


Figure 2.4. Certification Process.

The most fundamental task of a CA is to produce certificates. In a typical application, this process will involve the following events:

(1) The key pair of the CA is generated.

(2) The key pair of an entity *E* is generated.

(3) A certificate for an entity *E* is requested.

(4) *E*'s identity is verified.

(5) *E*'s key pair is validated.

(6) The CA produces *E*'s certificate.

(7) *E* checks that the certificate is correct.

Although it is clear that each of these events should take place, it is not at all obvious where they will occur in practice. Different applications are likely to require the tasks to be distributed in different ways. Often optimal security and control will be sacrificed and the practical distribution will differ from the ideal theoretical distribution.

Generation of the CA's Key Pair

A key pair must be generated for the CA if it is going to sign certificates. The key pair of the CA represents a highly attractive target for attack, since knowledge of the private key of the CA allows the forgery of any certificate and hence impersonation of any user. Protecting the security of the generation of this key pair is therefore vitally important.

Precautions taken are likely to include: the use of a tamper-resistant module to generate the key, the use of a highly secure generation method, the use of highly secure (large) parameters and possibly the distribution of 'shares' of the private key among several modules so that certificates cannot be created by any one device.

Generation of Entity's (E) Key Pair

A key pair must be generated for *E*. Although not as vulnerable to attack as a CA's key pair, compromise of an entity's key pair represents a seious security breach in most applications. A secure generation method should therefore be used to produce such entity's key pair.

In theory it is desirable that only *E* itself ever has possession of its won secret key. This is particularly pertinent in this case of signature keys; otherwise in the event of a repudiation dispute, *E*'s defense can sample claim that the contested signature was produced by the other party who has accessed its secret signing key. Whatever the use of the key pair, if it is generated by another party, *E* will have to trust the key generator no to compromise the key and the key will be particularly vulnerable during its transportation from the key generator to *E*. Therefore, it is generally accepted good practice for each entity to generate its won key pair.

However, from a practical standpoint, it is often unrealistic to expect that all entities will have the capability to generate their own keys. Since each entity is required to register public keys securely with a CA, and CAs are likely to have a degree of technical sophistication, entities may tend to rely on the CA. One obvious solution is for the CA to generate *E*'s key pair, and hand over the secret key to *E* during the certification process.

In order to partially eliminate the security concerns that this solution raises, the CA (or any other trusted entity) could generate *E*'s key pair in a tamper-resistant hardware Security Module that is configured to output keys only to the owner's storage device.

Certificate Request

In some system, *E* (or some third-party like a registration authority) will approach the CA and request that a certificate is issued to *E*. In this case, the medium used for the request will affect efficiency. The obvious options are either a paper-based request or an electronic request. Of the two, paper-based requests are likely to be more cumbersome for the CA to handle.

If it is necessary for the CA to maintain a record of the request as evidence, then some sort of proof of authenticity of the request will be required. Presumably a signature (written or digital) attached to the request will be needed. Note however, that if *E* is later required to acknowledge acceptance of the certificate, then it may be more pertinent of the CA to maintain evidence that *E* has accepted its certificates as valid.

Identity Verification

Clearly the CA needs some evidence that PK_E is *E*'s public key before it agrees to issue a certificate to *E*. Furthermore there may be a security requirement for this evidence to be presented to a third-party. The theoretical model here is simple: *E* provides the CA with a passport or some other form of identification as well as PK_E. If *E* already possesses a certified signature key, then the CA may instead accept a signature produced using this certified key testifying to the binding between E and PK_E. However, this may present the CA with an unacceptable workload and it therefore may be necessary to consider using a third-party known as a Registration Authority (RA) to testify to the binding between E and PK_E. One additional advantage of this approach is that the RA may be a trusted authority with which E has an established trust relationship. Another may be that RAs are run on a local basis so that E does not have to travel so far to register.

However, if the RA and the CA are two different entities, this creates the problem of transferring evidence of E's registration from the RA to the CA. Again either paper-based or digital routines may be used. Since RAs are designed to remove the need for the CA to heavily rely on paper-based routines, digital routines are more attractive.

Therefore, a natural solution is the use of registration certificates. These are certificates signed by the RA rather than the CA, which testify solely to the binding between E and PK_E. Only the CA verifies a registration certificate, and if the certificate is correct, the CA then goes ahead and issues E a full-blown certificate. Of course, now we have a new problem: who generates the RA's key? Clearly these keys are important. In fact they have the same level of importance as the keys of a CA.

A further difficulty associated with separating the roles of RA and CA is that disputes may arise between the two trusted entities in the event of a fake certificate appearing. Therefore, the CA will probably store registration certificates and produce them a evidence if such a dispute does arise. The threat of a security breach during the transition from RA to CA means that it is often preferable to combine the roles of CA and RA.

Key Validation

Ideally the CA (or RA) should check that PK_E really is E's public key. This involves checking that E knows SK_E corresponding to PK_E and checking that PK_E is valid. These checks have the dual purpose of protecting both E and the CA. They prevent E from mistakenly certifying the wrong key and they guard the CA against liability for certifying an incorrect or invalid key. One way of checking that E knows SK_E is to have E simply show SK_E to the CA. An alternative solution is for E to demonstrate knowledge of SK_E by signing some challenge data of the CA's choice using SK_E.

Certification Production

Once the CA is convinced that PK_E is E's public key, and that E wants to have a certificate testifying to this binding, the CA goes ahead and produces the certificate. Of course, the CA should be required to sign the certificates securely and so sets of rules for the secure production of certificates are continually emerging. In many applications it will also be desirable to produce certificates of as standard form. One emerging specification can be found in the standard X.509.

Distribution of CA Keys and Cross Certification

Typically, the CA's public key will be given to E at the same time that E gets a certificate. However, in systems where more than one CA is operation, E may be also want access to the public keys of the other CAs. In practice it will be impractical if not impossible for E to visit all the other CAs to collect their public keys in person. Instead E's CA may cross certify the public keys of the other CAs. In this approach, each CA obtains the public keys of the other CAs and gives copies of these other keys to E, either by handing them over in person or by issuing special 'authority certificates' binding each CA to its public key. A related approach is to implement a hierarchy of CAs. Here high-level CAs certify low-level CAs. Now all *E* needs to verify any entity's certificate is the public key of the highest level CA, which is frequently called the root CA, and a trail of authority certificates leading to the entity's certificate. These approaches are illustrated in Figure. Here cross certification enables any user below CA_1 to check the certificate of any user below CA_2 and vice versa.

The process of cross certification raises two concerns for the CA. Firstly, *E*'s certificate can now be checked by users of other CAs. This means that the CA's potential liability is increased. Provided the CA is issuing certificates securely, this should not be a problem, but disputes in other domains may interpret the CA's responsibilities differently. The CA may choose to limit potential problems of this kind by stating in the certification policy which entities are allowed to rely on *E*'s certificate, and to what extent they may rely on it. If unwarranted reliance is placed on the certificate, then the CA may be able to deny responsibility.

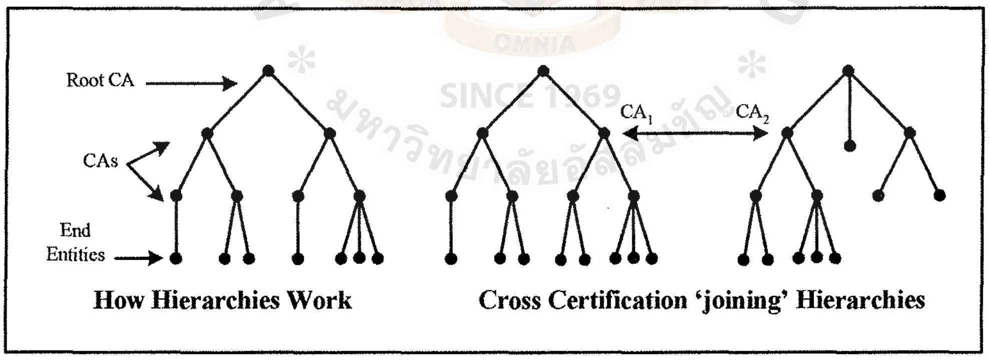


Figure 2.5. Distribution of CA Keys and Cross Certification.

Secondly, the CA can face liability for a certificate issued by another CA because it has implicitly supported the validity of the certificate by cross certifying. To prevent this problem, the CA may wish to carefully evaluate the CAs it cross certifies. The investigation procedure may include inspection of the other CA's certificate policies and certification practice statement. However, if the other CA's policies have not been independently audited, the CA must still trust the other CA about the accuracy of its policies.

Alternatively, the CA may wish to include a statement in the certificate policy of each authority certificate it issues regarding the limit on the level of trust that it determines should be placed on certificates issued by the other CA. If this level of trust is exceeded, then the CA may be able to deny responsibility. Entities relying on long chains of certificates must check the certificate policy in each certificate before judging whether its reliance on the relevant public key is justified. The process of cross certification may also raise similar concerns for *E*. *E* will need to be confident that each CA in the chain is reliable. Otherwise *E* may end up relying on worthless certificates.

Revocation

In most systems, it is reasonable to expect that the secret keys of users could be compromised. Additionally, there will probably be other reasons for wishing to invalidate an existing certificate. Therefore it is necessary to have a mechanism in place so that users can revoke their certificates.

Since the CA is a trusted entity and is responsible for producing certificates, the burden for maintaining this revocation mechanism is likely to fall on the CA. In some circumstances this is advantageous because it also allows the CA to revoke certificates easily if its own key is compromised, or if an entity does not acknowledge receipt of its certificate. Both phases of the revocation process represent a security concern: the request phase during which *E* asks for his certificate to revoked, and the notification phase during which entities are informed of the revocation. The revocation request mechanism will differ from system to system depending on the potential cost of revoking a perfectly good certificate and on the potential cost of delaying a genuine revocation request.

2.3.3 Certification Life Cycle

According to operation of CA, Figure 2.6, summaries the discussion of the certification process, also known as the certification life cycle.

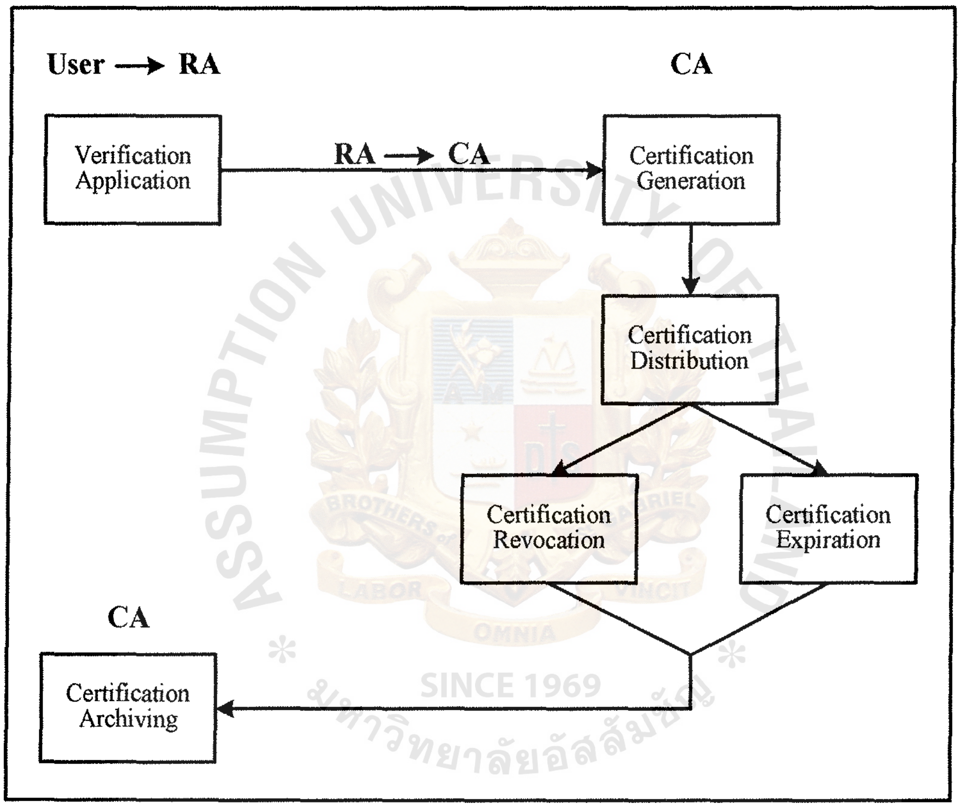


Figure 2.6. Certification Process.

2.3.4 Timestamping Service

A timestamping service is trusted to testify to the existence of some data at a certain time. The data may consist of, for example, digital signatures or laboratory notes to support patent claims. Typically, the timestamper will append the time to the data, and then sign the resulting string. To check the stamp, anyone can now read the time and verify the appended signature. Timestamping digital signatures can help prevent repudiation disputes. First, timestamps can help to testify to the freshness of a signature. Second, they can reveal that a signature was created before the corresponding key was compromised.

III. WEB SIGNED-FORM WORK FLOW

In the previous chapter, we have provided an overview of the principle of digital signature. The chapter has discussed the mechanism and working process of digital signature. The illustration depicts how they can apply it in the application. In addition, security system is mentioned to be benefit for commerce in cyber world.

Now, this chapter will apply the knowledge of digital signature that has already been discussed in the previous chapter with electronic transaction in the cyber commerce. The chapter illustrates how web signed-form can be worked. All working process will be indicated to step of making electronic transaction via web signed-form.

3.1 Overview Process of Web Signed-Form Tasks

This topic provides overview the process of all tasks in order to easily understand whole components. The illustrated process will start from selecting product for ordering until signing and verifying process. Figure 3.1 indicates overview in all tasks.

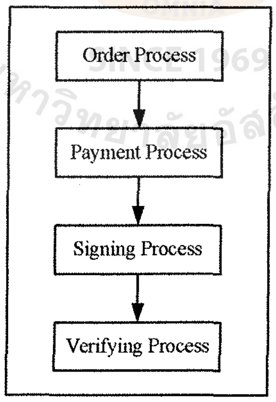


Figure 3.1. Overview the Process of Web Signed-Form Tasks.

The illustration depicts whole working process starting with order process. The web site of http://www.smtmobile.com is created under trademark of SproakMakerTelecom in order to employ demonstration for this project. The site launches for selling mobile phone as online mobile phone shop. This demonstrated web page will display all the tasks as if it is running online now.

When customers come into the web site, they will look for their interested product. After they satisfy their shopping, they ready to make an order. Now, they are coming to the process of payment. Web signed-form is requested from the server side. Process of signing is proceeded and submitted to server side. Server side will receive order list including a signature. Then, verifying process will be implemented. The valid verifying process will return output of data of order list.

The above illustration widely indicates all working processes. All working processes consist of working process of shopping cart, order transaction in client side, verifying transaction in server side, as well as signing module and verifying module. More details of each working process will be discussed later in this chapter.

3.2 Working Process of Shopping Cart

In this section, more details in working process of shopping cart will be explained to indicate how the cart is designed. The shopping cart model in Figure 3.2 points to entire working process of program before digital signature will be signed.

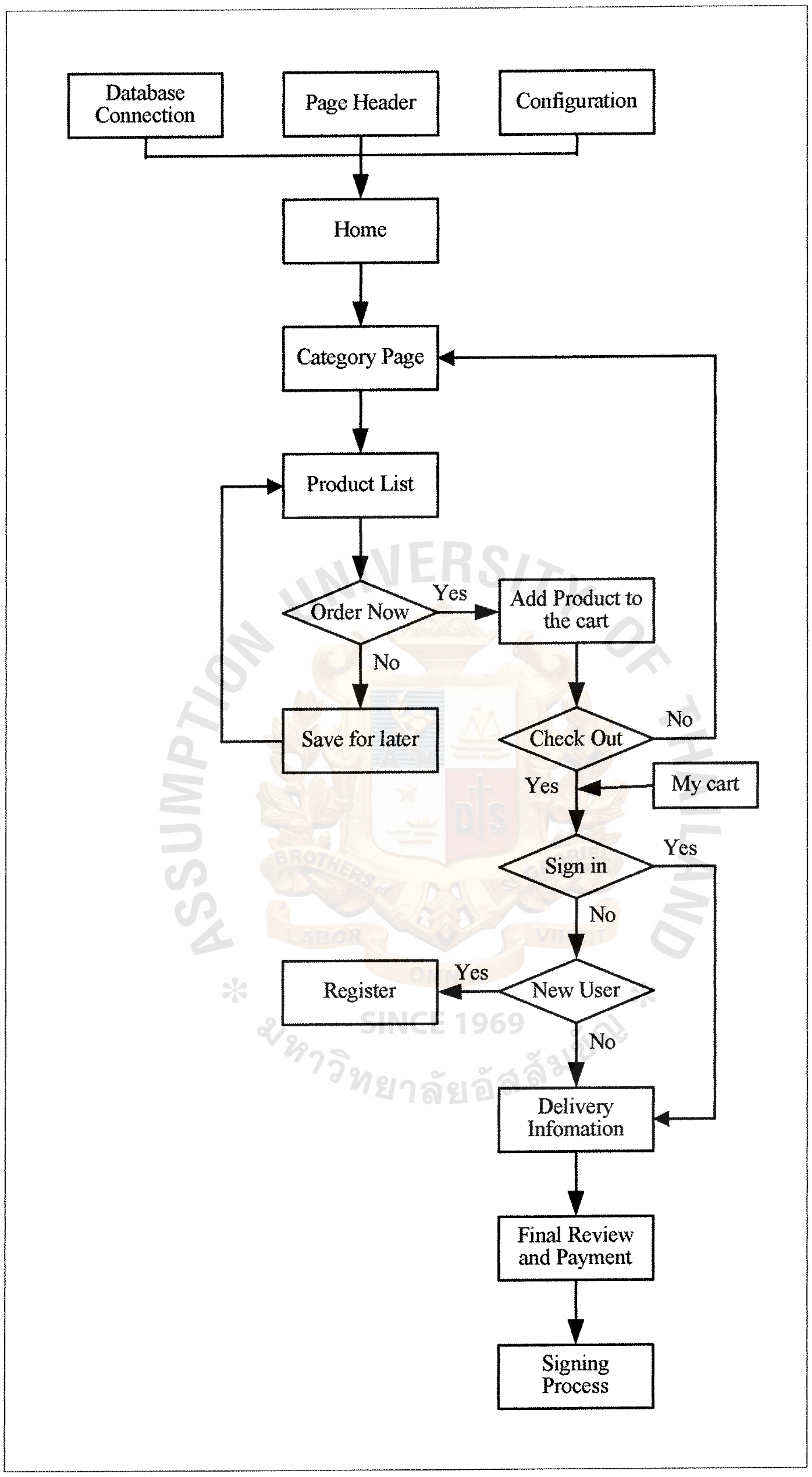


Figure 3.2. Working Process of Shopping Cart.

In Figure 3.2, we start by creating a file of database connection, menu bar, and some configurations. These files are included in all working processes of shopping cart. Take a look at the database connection file, this file has functions to connect with a MS-Access database via ODBC driver. The data is called from database and show on the browser.

Focus on a file of displaying menu bar, on menu bar displays number of product(s) in shopping cart and various links of product categories, review products in the cart, favorite list, order details, and sign in. Links of review product in the cart, favorite list, and order details will appear on a menu bar in case of having product in the cart, favorite list, and order list respectively. Normally, menu bar is shown in all pages on the browser. To make its easy and convenient, a file is separately created and added to the other working process file.

Some often used configurations are collected into the configuration file in order to set and update variables easily such as currency symbol, store name, tax rate, and etc. These variables appear almost every page on browser. We are not necessary to change these variables in every page.

The first page of the cart mainly shows featured products and categories menu. Products which are assigned to featured products are randomly displayed in each loading. They can be directly added into the cart from this section. For categories menu, categories name show on the menu as a links which navigate to product lists in theirs categories.

When a link of product categories on menu bar is clicked from home page, it will link to product categories file. Entering into this page, it firstly checks whether product category is selected or not. The error page will be loaded in case of no action on product categories links. In product categories page, links of products list and image in each categories are shown on the page. They will lead to more details of products in the product page.

In the product page, a list of products are displayed. It is necessary to firstly check requested record of the product before entering into this page. If there is no requested record of the product, the error page will be returned instead. However, if the record of product is true; details of product will be shown such as price and specification information. The record of products can be decided in two ways. Both ways fall into saving for later and adding to the cart respectively.

If selected products are not ordered now, they may be saved into the "Wish List" in order to wait for order at the next time. The link of "My wish list" will appear on the menu bar after products are added into the "Wish List." In "Wish List," the lists show product ID, items, and price. The items in the lists can be deleted until empty. When the list is empty, warning message will appear on the page. However, products in the list can continue to order by clicking a link in each item. They will turn back to product page. Then, ordering process can be done.

In case of ordering now, the selected products are added into the shopping bag and wait for checking out. When products are added, session will be created in the cart and keeps values of added products. The links of "My cart" will appear on the menu bar. Reviewing and updating order can be made by clicking this link. In the cart, it displays item, quantity, price, total amount, and grand total. Tax rate is included in the grand total. Deleting items in the cart can be processed by calling from a function. The cart can be repeatedly updated until satisfy through categories and product page.

Now, order process is satisfied and ready to go to the payment process. The process firstly checks sign-in status. If users have already signed-in before, the process will move to checking delivery information. Whereas if sign-in is not made, the sign-in page will be loaded. Then, they continue to check user status. In case of old users, login and password can be singed immediately. Then, the system will verify the login e-mail address and password. If sign-in process is invalid, this process will be required again. After verifying valid sign-in process, it will move to the process of checking delivery information. Registration process will be skipped. The link of "My Profile" will appear on the menu bar. Users can update personal information through this link.

However, the new users are required to register. A registered page will be opened to fill out new user's information. While submitting personal information, the working process checks for completeness of input in each field. If any fields are invalid, warning message will be returned and will require to be filled out again.

Then, the working process comes into checking delivery information. Both old and new users are required to check their delivery information. In case of name and address in personal information and delivery information is different; the working process will require registering delivery information. Then, they have verified delivery information. If delivery information is invalid, registration of delivery information will be returned again.

After finishing checking sing-in and user status as well as personal and delivery information, the system will inspect ordered product in the cart. If there are no products in the cart or session timed out, review page will be returned with warning message. Then, the working process will be back to beginning stage in order to make an order. On the other hand, if ordered product lists are found in the cart, final review page will be loaded to display final review of order, billing and delivery, and payment information.

Every section in final review page can be finally deleted or updated before submitting order. In final review of order, this section finally reviews order lists. It shows details of product items, quantity, price, and total like as displaying in "My cart." Users can add or delete order lists by repeating order process again. For the section of review of billing and delivery information, this part shows the details of name and address of billing information and delivery information. Both parts of information can be updated by clicking edit button. The last section in final review page, credit card information must be filled out in every field. Credit card information consists of card type, card number, and expiry date. These information will be verified during placing the order.

Now, we are moving to the highlight of this project. When users are ready to place the order, working process of digital signature will start from this point. Order data will be submitted and digitally signed to vendor. More details of signing and verifying data will be discussed later in this chapter.

3.3 Working Process of Data Signing

As in the previous section, all commerce transactions are operated via web browser. The client hereby is web browser and the server is web server. To make a purchasing order, the customer's web browser, called the client, sends the **GET** HTTP request to get the form from the web server. The web server sends back the HTTP response for the form being requested by the client. The customer, in general, fills out the form and submits the data to the web server by **POST** HTTP request. However, this project requires the working process of data signing before the data being submitted and sent out.

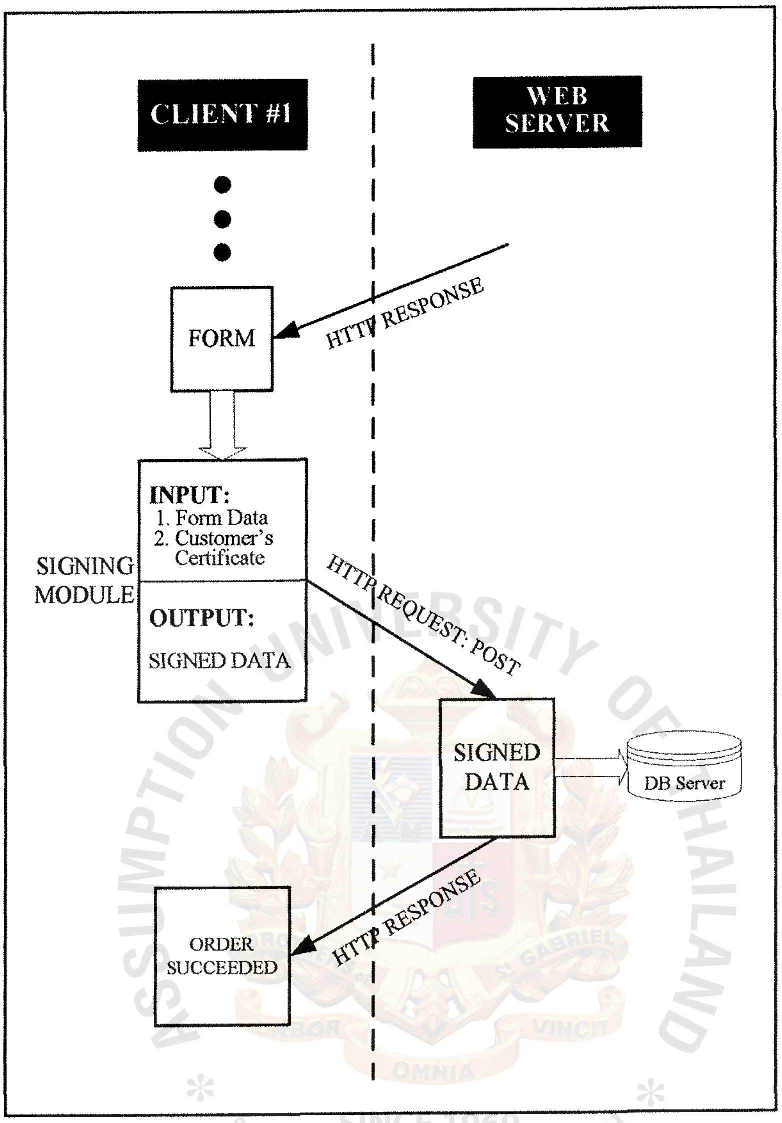


Figure 3.3. Working Process of Data Signing.

The signing module will be depicted in detail in section 3.5, which is responsible for this functionality. An ActiveX object namely **signform** must be dynamically loaded and run from the web page. Also, it was written in C++ language by using Microsoft Visual C++ program and Microsoft Platform SDK.

The **signform** object mainly consists of two methods, **SignMessage()** and **VerifyMessage()**. This section describes only the **SignMessage()** method and mentions the **VerifyMessage()** in section 3.4.

In **signform** object, there are three corresponding properties. In VC++ object-oriented programming, it should be mentioned that **function** is called **method**, and **variable** is called **property**. The three object properties are described as follows:

*the 1st property : **plainMsg***

This property refers to all data in the form to be signed.

*the 2nd property : **signerName***

This property refers to the signer's common name or e-mail address, for example, "Sithisak Choteporsiri".

*the 3rd property : **cryptMsg***

This property refers to the data being signed.

The **SignMessage()** method has no any argument. The input is set via a property called ***plainMsg*** which is the plain data to be signed. This method returns the boolean output whether the data can be successfully signed or not.

The **SignMessage()** method processes with the following steps:

(1) pop up the dialog box that lists all existing certificates in the personal "MY" certificate store and lets the signer to choose his/her own certificate

(2) set the appropriate parameters for the **CryptSignMessage** function (This function is provided by cryptoAPI of Microsoft Platform SDK)

(3) call the **CryptSignMessage** function to sign the input data, **plainMsg**

(4) the **CryptSignMessage** function returns the signed data in BLOB format

(5) encode the BLOB with base-64 encoding (encode the BLOB to be the user-friendly text)

(6) return the output after having base-64 encoding

In order to sign the data before submitting, the web page must load the ActiveX object, **signform**. With JavaScript, each field of data in the form are treated, concatenated and kept in a single string, **plainMsg**. The **plainMsg** is only one property set as the input of **SignMessage()** method.

After the output is returned from the signing module, it must be submitted via the HTML form by a field which the name is **cryptMsg** and the type is **hidden**. It should be noticed that the name of HTML output field is identical to the name of the object's output because of the ease to remember and refer. Now the signed data is sent to the web server and kept in the database.

The code in JavaScript language is shown as follows:

```
signform.plainMsg = "This is the data to be signed.";
form.cryptMsg.value = signform.SignMessage();
```

Suppose that the HTML form name is **form**.

3.4 Working Process of Data Verifying

The previous section describes how the customer signs his/her purchasing order data in web form and submits the signed data to the web server. Consequently, the webmaster accesses to the web page to check out the order that the customer has already made. The purchasing orders of each customer are listed. Once webmaster clicks to view the data of each purchasing order, the data is retrieved from the database at the web server and also processed by the **VerifyMessage()** method at the web browser.

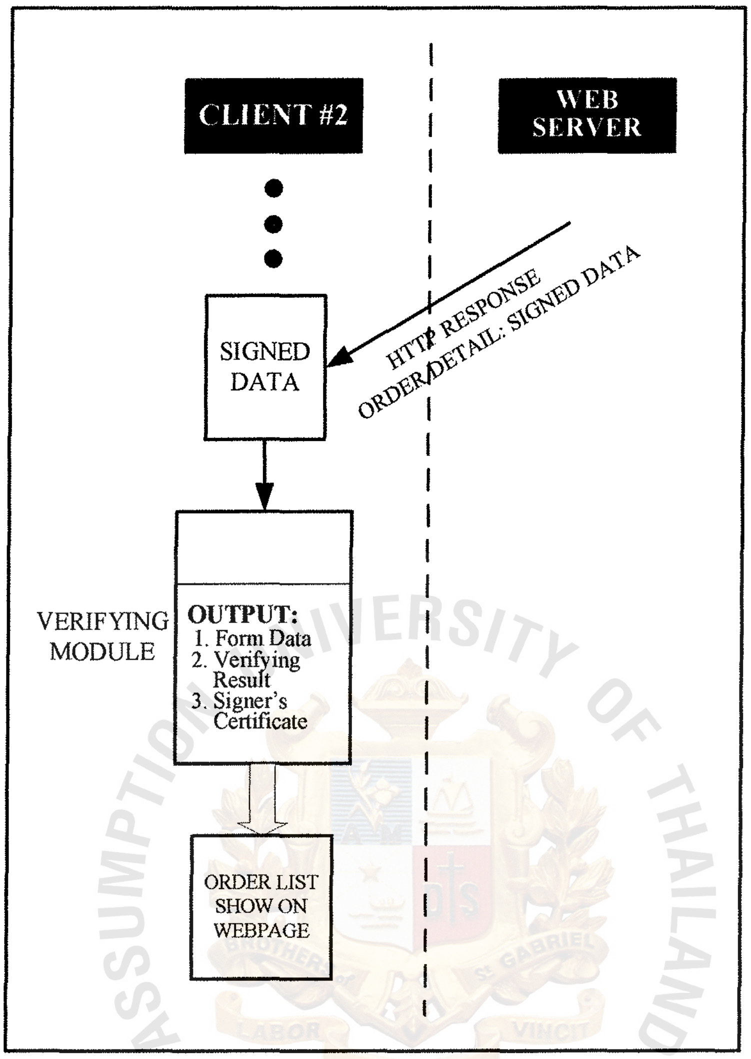


Figure 3.4. Working Process of Data Verifying.

The data retrieved from the database is signed by the customer, webmaster has to verify the customer's digital signature in order to ensure the integrity of data and the identity of customer. The **VerifyMessage()** method in **signform** object is written to verify the customer's purchasing order data.

The **VerifyMessage()** method has no any argument. The input is set via the property, **cryptMsg** where **cryptMsg** is the data being signed. This function sets the output value of two properties. The first output is **plainMsg** that refers to the plain data before it was signed by the customer. The other output is **signerName** that refers to the signer name. This method returns the boolean output whether the signed data can be verified or not.

The **VerifyMessage()** method processes with the following steps:

(1) set the appropriate parameters for the **CryptVerifyMessageSignature** function (This function is provided by cryptoAPI of Microsoft Platform SDK)

(2) call the **CryptVerifyMessageSignature** function to verify the input signed data, **cryptMsg**

(3) the **CryptVerifyMessageSignature** function returns two output variables of the plain data before being signed and the signer name

(4) set the **plainMsg** property to the plain data before being signed and set the **signerName** property to the signer name

(5) the method returns the result of verifying (true or false).

When webmaster queries the customer's purchasing order via web page, that web page must load the **signform** ActiveX object. Also, the signed data must be queried from the database at the server side. The object can be set the **cryptMsg** property by JavaScript language as shown in the following:

```
signform.cryptMsg = "<% = cryptMsg %>";
signform.VerifyMessage();
```

It should be mentioned that the code, '<% = cryptmsg %>', means the signed data string represented by the variable in ASP programming, namedly *cryptMsg*.

3.5 Signing Module

In the previous topic, we had discussed the working process of order transactions by customers in client side and verifying transaction by webmaster in server side via web page. Now, we will explore more details of the signing module, whereas verifying module will be discussed at the next topic.

3.5.1 Creating a Signed Data

Signed data consists of content of any type and encrypted message hashes of the content by zero or more signers The resulting hash can confirm that the original message has not been modified since signing and that particular persons or entities signed the data.

In the Figure 3.5, the chart depicts the tasks that must be accomplished to create a signed message. The steps are listed following the illustration.

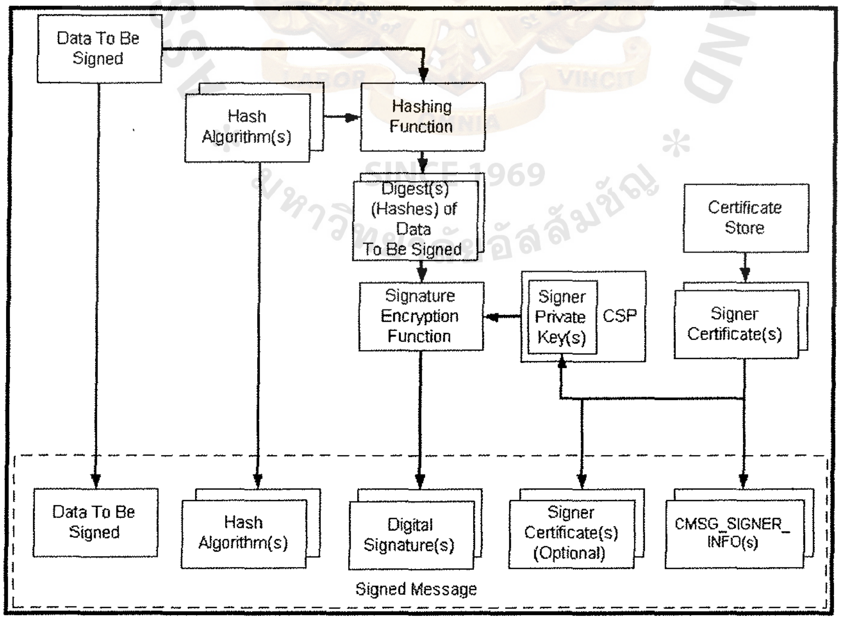


Figure 3.5. Signing a Message.

The general process for encoding signed data is as follows:

(1) The data is created, and a pointer to it is retrieved.

(2) A certificate store is opened that contains the signer's certificate.

(3) The private key for the certificate is retrieved. There are two properties that must be set on the certificate before using it. Firstly, it is used to tie a certificate to a particular CSP, and within that CSP, to a particular private key container. Secondly, it is used to indicate which hashing algorithm is to be used when a digest operation is called for. These need only be set once.

(4) A certificate's property determines the hash algorithm.

(5) A hash of the data is created by sending the data through the hashing function.

(6) The signature is created by encrypting the hash using the private key, obtained through a property on the certificate.

(7) The following data is included in the finished, signed message:

    (a) The original data to be signed

    (b) The hash algorithms

    (c) The signatures

    (d) The signer info structures, which includes the signer identifier (certificate issuer and serial number)

    (e) The signer's certificate (optional)

3.5.2 Procedure for Signing Data

A Single function, CryptSignMessage, performs all of the tasks listed in **Creating a Signed Message**. However, initialization of structures and other data is still necessary. The following illustration shows the relationship between those function parameters that point to structures or arrays and their initialized data. The illustration shows only the function parameters and structure members that are derived from other structures or functions. The rest of the parameters are straightforward initializations.

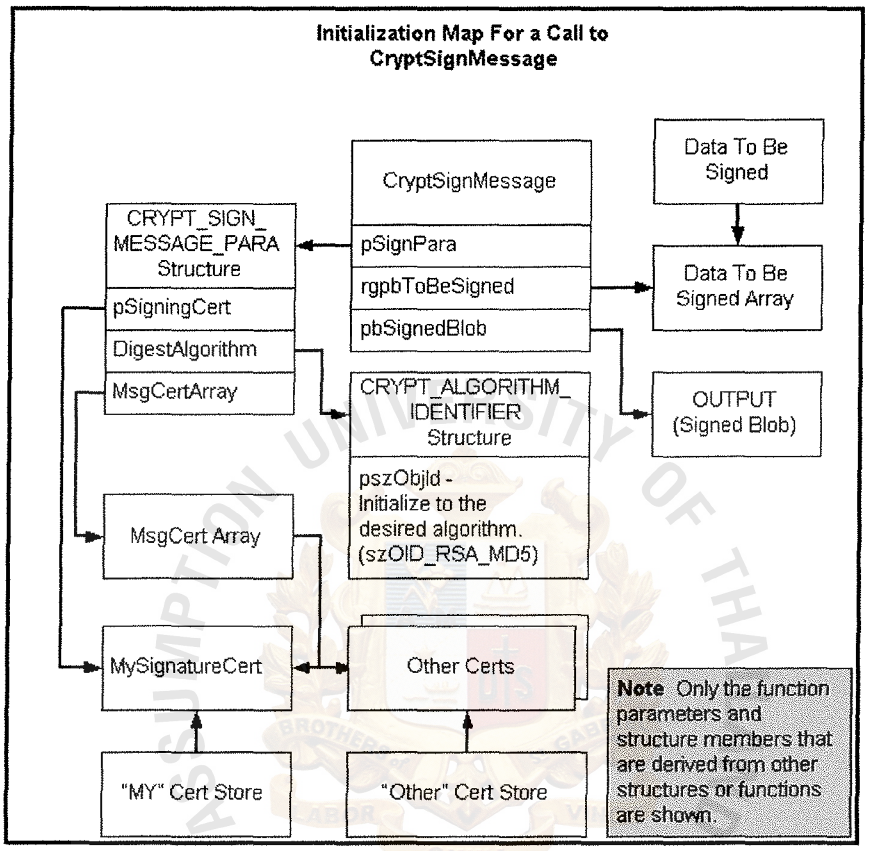

Initialization Map For a Call to CryptSignMessage
CryptSignMessage
pSignPara
rgpbToBeSigned
pbSignedBlob
Data To Be Signed
Data To Be Signed Array
OUTPUT (Signed Blob)
CRYPT_SIGN_MESSAGE_PARA Structure
pSigningCert
DigestAlgorithm
MsgCertArray
CRYPT_ALGORITHM_IDENTIFIER Structure
pszObjId - Initialize to the desired algorithm. (szOID_RSA_MD5)
MsgCert Array
MySignatureCert
Other Certs
"MY" Cert Store
"Other" Cert Store
Note Only the function parameters and structure members that are derived from other structures or functions are shown.

Figure 3.6. Initialization Map for a Call to CryptSignMessage.

To sign data using CryptSignMessage

(1) Get a pointer to the data that is to be signed.

(2) Assign the pointer to the data to index zero of a "data to be signed" array.

(3) Get a handle to the cryptographic provider.

(4) Open a certificate store that contains the signer's certificate.

(5) Get an address to the signer's certificate.

(6) Assign the address of the certificate to the zero index of the *MsgCert* array.

(7) Assign the addresses of any other certificates to be included with the message to the *MsgCert* array.

(8) Initialize the **CRYPT_ALGORITHM_IDENTIFIER** structure, initializing the **pszObjId** member to the desired hash algorithm and the other members as appropriate.

(9) Initial the **CRYPT_SIGN_MESSAGE_PARA** structure, initializing the **pSigningCert** member to the address of the signer's certificate, the MsgCert array member to the address of the signer's and other's certificate, the **HashAlgorithm** member to the address of the **CRYPT_ALGORITHM_IDENTIFIER** structure, and the other members as appropriate.

(10) Call the **CryptSignMessage** function, passing the **CRYPT_SIGN_MESSAGE_PARA** structure for the *pSignPara* parameter, the address of the "data to be signed" array for the *rgpbToBeSigned* parameter, an address for the *pbSignedBlob* output parameter, and values for the other parameters as appropriate.

3.6 Verifying Module

According to the previous part, we already know the working process of signing module. In this topic, we will discuss more details of verifying a signed message and decoding Signed Data.

3.6.1 Verifying a Signed Message

These steps verify the signature of signed data. The following illustration depicts the individual tasks that must be accomplished, as shown in the list that follows it.

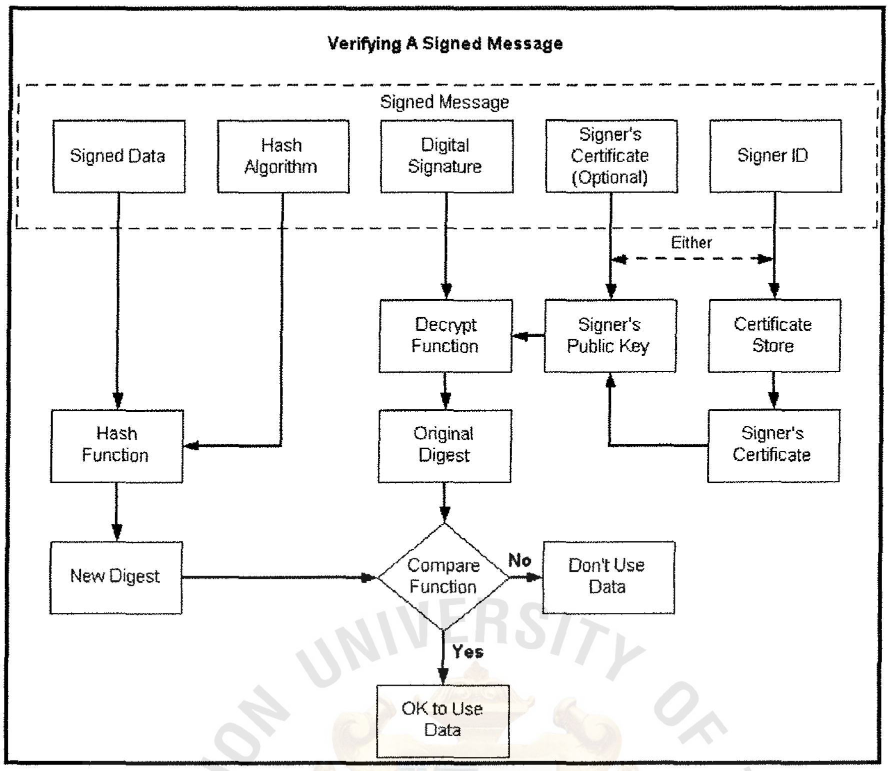


Figure 3.7. Verifying a Signed Message.

To verify the signature of a signed message

(1) Get a pointer to the signed message.

(2) Open a certificate store.

(3) Using the signer ID contained in the message,, get the sender's certificate and get a handle to its public key. As an alternative to steps 2 and 3, you can use the certificate contained in the message to retrieve the signer's public key.

(4) Using the signer's public key, decrypt the digital signature, producing the original digest of the data in the message.

(5) Using the hash algorithm contained in the message, hash the data contained in the message, yielding a new digest.

(6) Compare the digest retrieved from the message with the new digest just created.

(7) If the two digests match, the signature is verified. This means that the private key that was used to sign the data matches the public key just used to decrypt the signature, and that the data has not changed since the data was signed. If the two digests do not match, the signature is not verified, and either the private/public keys do not match, or the data has been changed since the data was singed, or both.

A single function, **CryptVerifyMessageSignature**, can be used to verify a signature, as shown in the following procedure. **To verify a signed message**

(1) Get a pointer to the signed message.

(2) Get the size of the signed message.

(3) Get a handle on a cryptographic provider.

(4) Initialize the **CRYPT_VERIFY_MESSAGE_PARA** structure.

(5) Call **CryptVerifyMessageSignature** to verify the signature.

3.6.2 Decoding Signed Data

The following general process decodes a signed data type. **To decode a signed message**

(1) Get a pointer to the encoded BLOB

(2) Call **CryptMsgOpenToDecode**, passing the necessary arguments.

(3) Call **CryptMsgUpdate** once, passing in the handle retrieved in step 2 and a pointer to the data that is to be decoded. This causes the appropriate actions to be taken on the message, depending on the message type.

(4) Call **CryptMsgGetParam**, passing in the handle retrieved in step 2 and the appropriate parameter types to access the decoded data.

The following general process verifies the signature of a decoded, signed message. **To verify the signature of a decoded, signed message**

(1) Call **CryptMsgGetParam**, passing in the message handle and CMSG_SIGNER_CERT_INFO_PARAM to get the signer's **CERT_INFO** from the message.

(2) Call **CertOpenStore** to open a temporary store that is initialized with the certificates from the message.

(3) Call **CertGetSubjectCertificateFromStore** to get the signer's **CERT_INFO** from the certificates included in the message.

(4) Call **CryptMsgControl**, passing in CMSG_CTRL_VERIFY_SIGNATURE to verify the signatures.

(5) Call **CryptMsgClose** to close the message.

The result of these procedures is that the signature is verified and a pointer is retrieved to the decoded message content obtained in step 4 of the procedure for decoding a signed message.

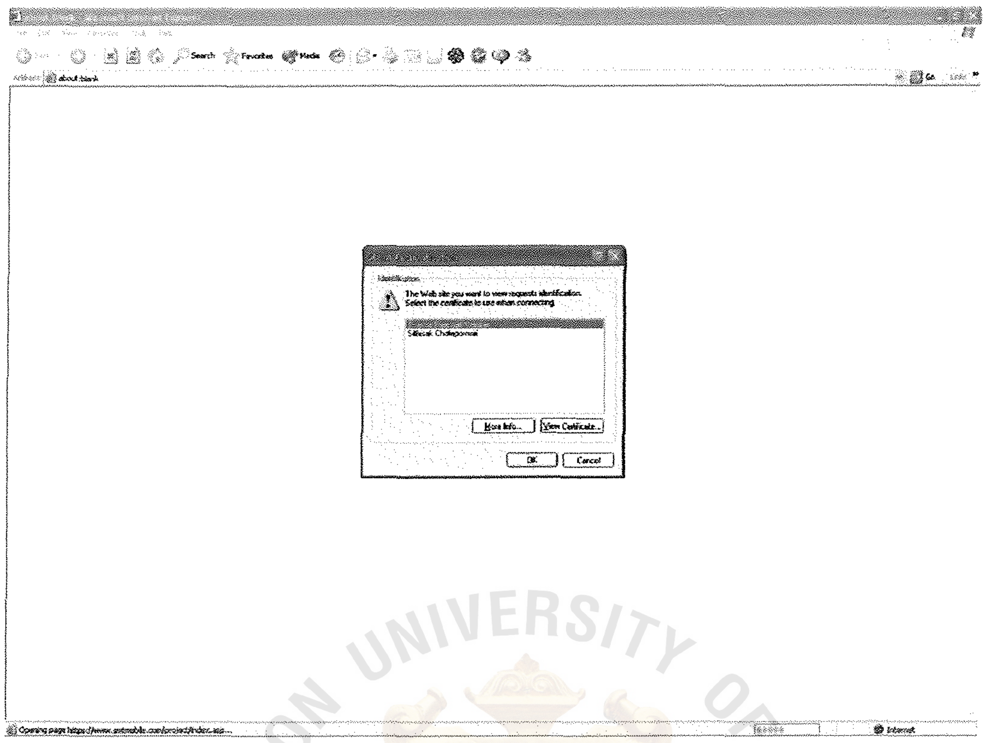
IV. DEMONSTRATION OF WEB SIGNED-FORM MODEL

In the previous chapter, we had illustrated the entire working process of the tasks. The chapter has explained how web signed-form can be worked. Now, this chapter will demonstrate all work in the model. It will show the steps of making an electronic transaction with digital signature through the web page. Furthermore, there is comparison of valid versus invalid verifying procedure outputs.

4.1 Working Steps of Web Signed-Form and Valid Verification

To easily understand, the demonstration is mainly classified to four sections. First, it shows the steps of authentication when the customer comes into the web site. Second, the steps of making purchasing orders are demonstrated after the customers are authenticated to enter the web site. Third, this section depicts the steps of sending submitted data with digital signature. Last, this step concerns to the webmaster who is allowed to view the customers' transaction. The steps display how the webmaster can verify signed customers' transaction.

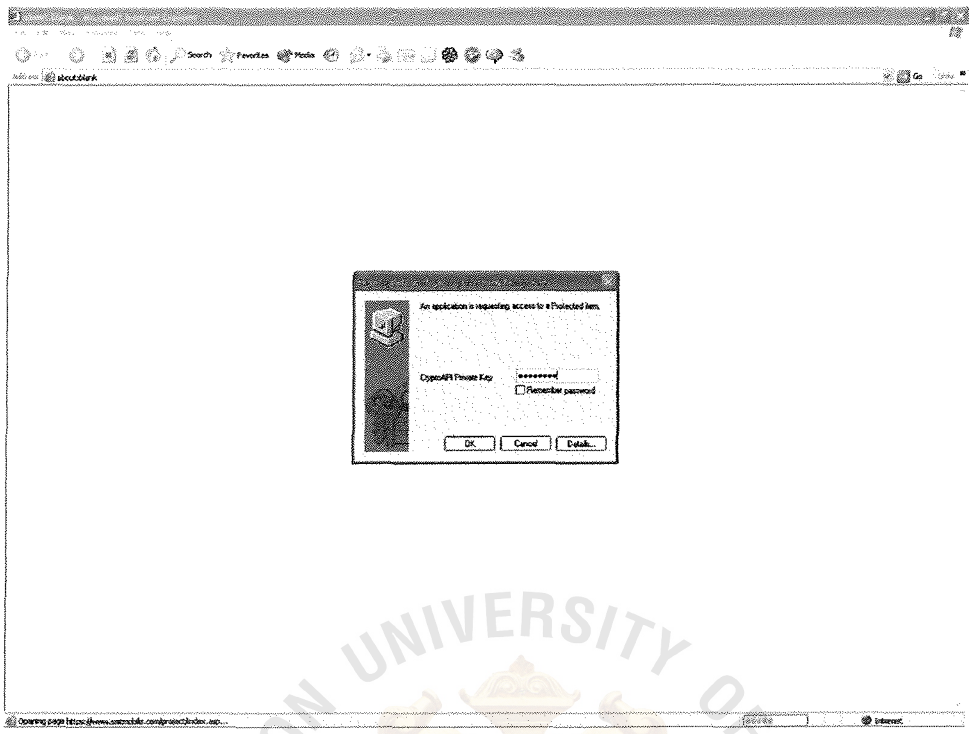
Now, the first section is demonstrating the steps of authentication. Before customers entering into the web site, the customers are required to register for personal certificate with certificate service provider (CSP) and install their certificate into the machine. The personal certificate can be launched to display by MS Internet Explorer or MS Outlook Express.



The Web site you want to view requests identification. Select the certificate to use when connecting.

Figure 4.1.   Client Certificate Requirement.

In Figure 4.1, when the customer enters into the web site of https://www.smtmobile.com/project/index.asp, the customer is required to authenticate for entering into the web site. The certificate store will be opened to select their personal certificate if there are many personal certificates in the machine. On the other hand, if there is only one personal certificate in the machine, the process will skip to fill out CrytoAPI Private Key Password. The customer is not necessary to select personal certificate from certificate store.

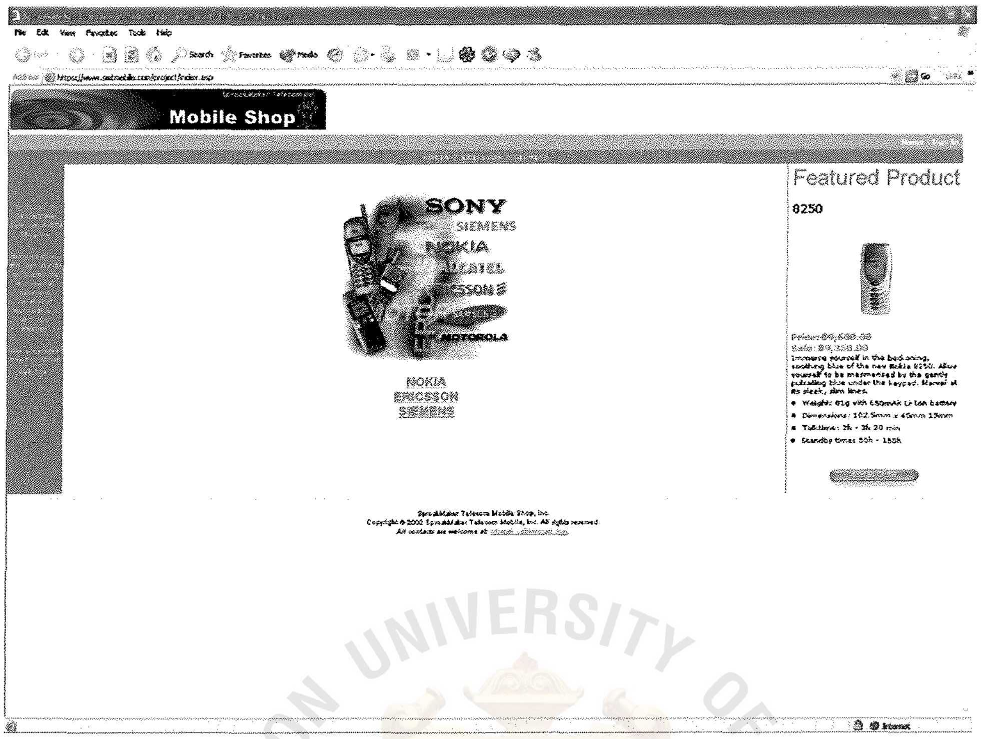

An application is requesting access to a Protected item.

CryptoAPI Private Key ••••••••

☐ Remember password

Figure 4.2. CrypToAPI Private Key Password for Authentication.

From the previous step, when the customer clicks "OK" button on the dialogue box of Client Authentication, the dialogue box of Signing data with your private exchange key will be pop up in case of high security as shown in Figure 4.2. The customer is required to fill out the password of private key. In case of low security, the customer is authenticated to access into the web site without requiring private key password after selecting personal certificate in certificate store.



~~Price: ฿9,680.00~~
Sale: ฿9,350.00
Immerse yourself in the beckoning, soothing blue of the new Nokia 8250. Allow yourself to be mesmerised by the gently pulsating blue under the keypad. Marvel at its sleek, slim lines.

- Weight: 81g with 650mAh Li-Ion battery
- Dimensions: 102.5mm x 45mm 19mm
- Talktime: 2h - 3h 20 min
- Standby time: 50h - 150h

SpreadMaker Telecom Mobile Shop, Inc.
Copyright © 2000 SpreadMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at [illegible]

Figure 4.3. The Home Page of the Web Site.

The system determines validation of the personal or client certificate for authentication. If the client certificate is invalid, the customer is not allowed to access the web site. On the other hand, if validation of client certificate is determined, the customer has authorized to access the web site. Then, the home page of web site will be launched as shown in Figure 4.3. The customer can freely enjoy shopping the product on the web site. The authentication section has ended at this point.

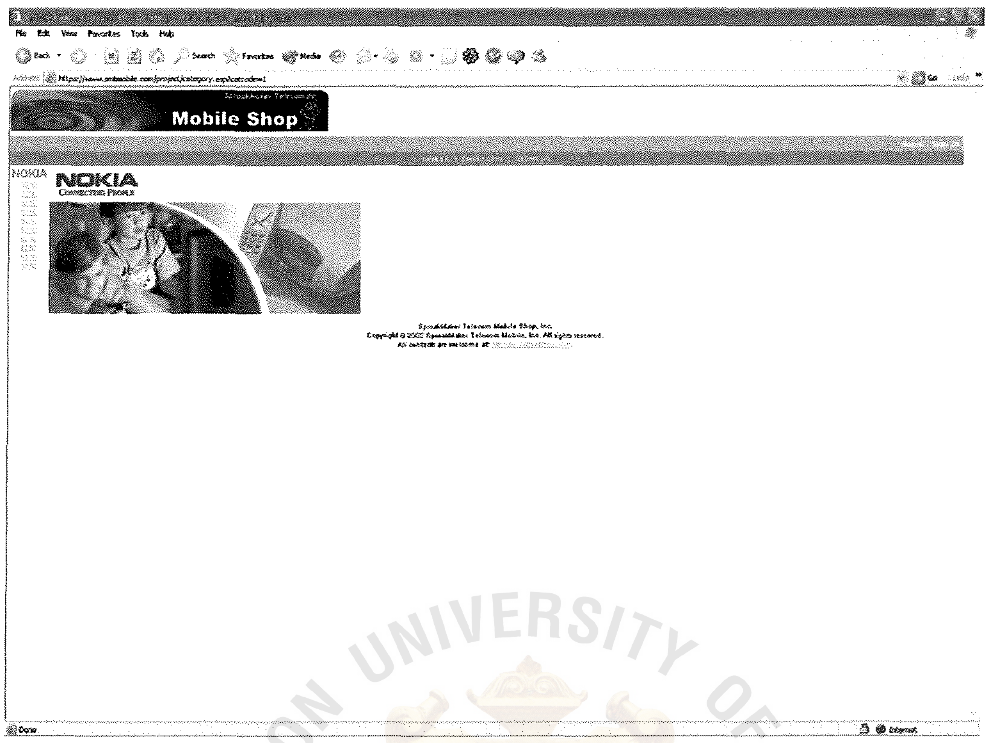
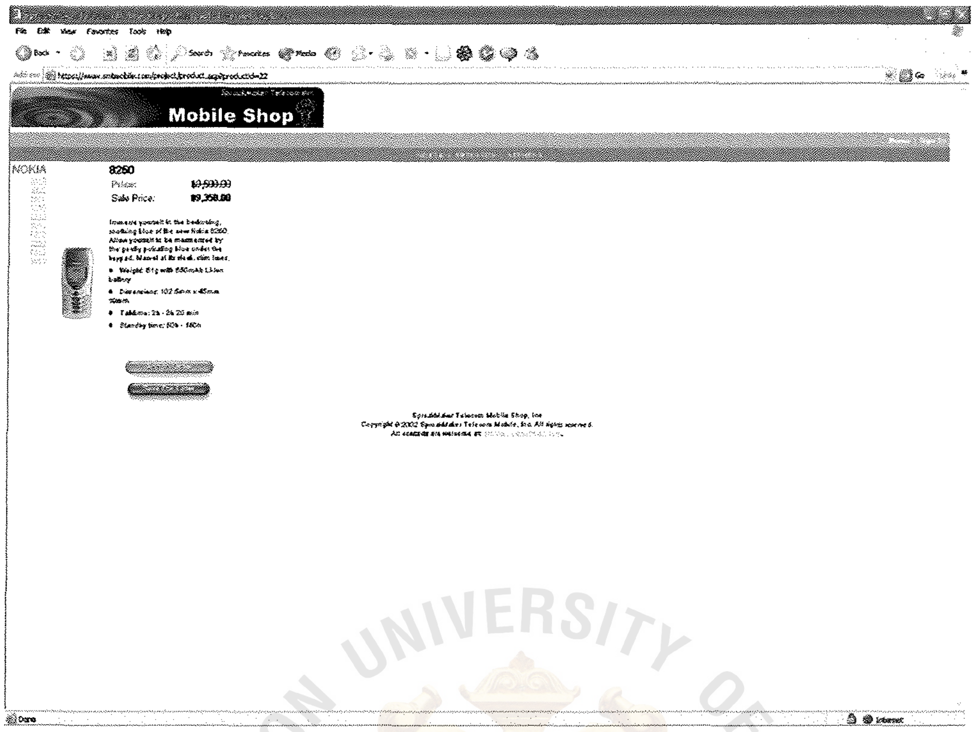


Figure 4.4. Product Categories Page.

In this step, the second section, making purchasing orders process is demonstrated how it works. The customers can select ordering the products from each category that is shown in Figure 4.4. The links of category names appear on menu bar. Customers can switch the categories by clicking the links on menu bar. Product name in each category will be listed on the left hand side of the page. The links of product lists will lead to the product page that shows more details of each product.


File Edit View Favorites Tools Help

https://www.smtmobile.com/project/product.asp?productid=22

Mobile Shop

NOKIA

8260

Price: ~~฿9,599.00~~

Sale Price: ฿9,350.00

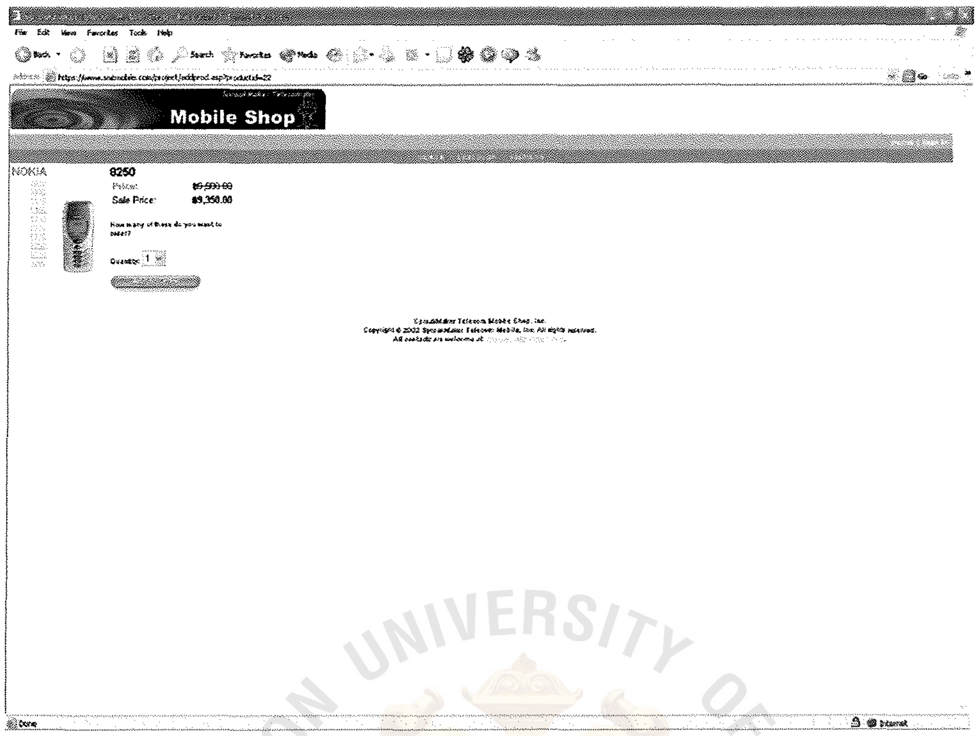
Immerse yourself in the bedazzling, soothing blue of the new Nokia 8260. Allow yourself to be mesmerized by the gently pulsating blue under the keypad. Marvel at its sleek, slim lines.

- Weight: 81g with 650mAh Li-Ion battery
- Dimensions: 102.5mm x 45mm x 20mm
- Talktime: 2h - 2h 20 min
- Standby time: 50h - 180h

Spraddeler Telecom Mobile Shop, Inc
Copyright © 2002 Spraddeler Telecom Mobile, Inc. All rights reserved.
All contents are welcome at [illegible]

Figure 4.5. Product Details Page.

The Figure 4.5 shows the product detail page. After customers already choose the product categories, the product detail page will be displayed to show on the page such as qualification, price, and etc. Then, customers can decide into two ways. Firstly, they can make an order by clicking the link of “Add to Order.” Secondly, they would not like to order now. They can save for ordering in the later. In this case, we will show only ordering process. When the customer would like to add their favorite product into the cart, they can click the image button of “Add to Order.” Then, they will move to purchasing order quantities page.


File Edit View Favorites Tools Help
Back Search Favorites Media
https://www.snbmobile.com/project/addprod.asp?productid=22
Mobile Shop
NOKIA
8250

Retail: ~~฿9,590.00~~
Sale Price: ฿9,350.00

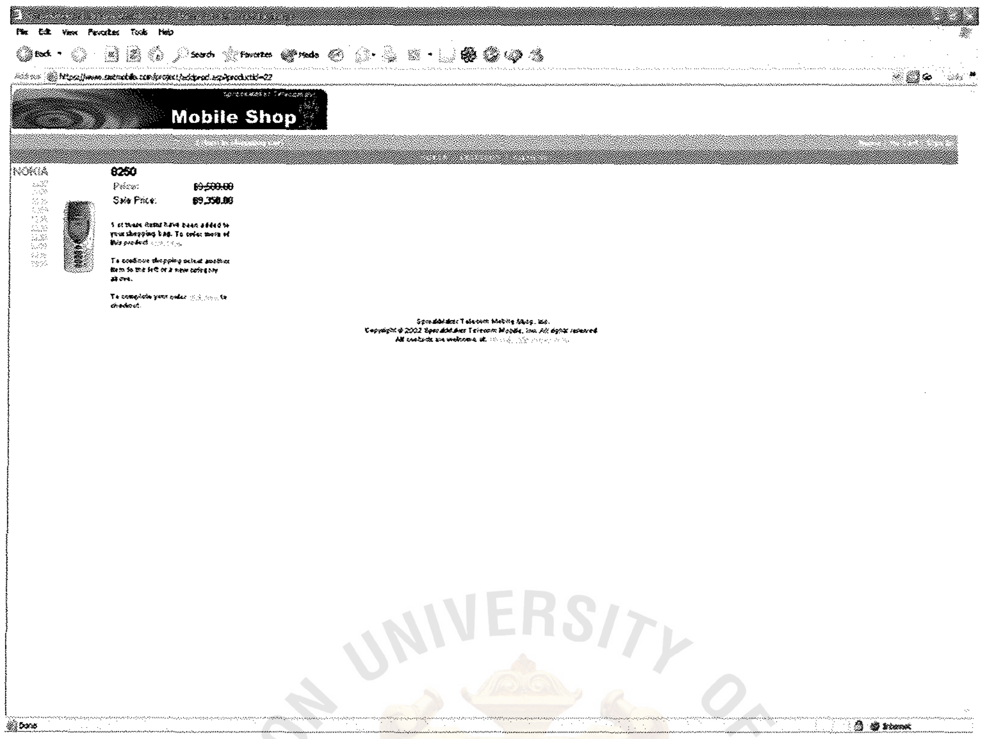
How many of these do you want to order?

Quantity: 1

SpeakMaker Telecom Mobile Shop, Inc.
Copyright © 2002 SpeakMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at

Figure 4.6. Purchasing Order Quantities Page.

The Figure 4.6 depicts purchasing order quantities page. The customer can select a number of purchasing order quantities on the product they would like to add into the cart. The possible minimum purchasing order quantity is one item, whereas nine items are the possible maximum purchasing order quantities. Then, the customer can click image button of "Add to Order" in order to add their favorite product into the cart until they satisfy. In this step, the products that added into the cart have not been stored into the MS Access database. They have been still kept in the temporary session.



https://www.smtmobile.com/project/addprod.asp?productid=27

8250

Price: ~~฿9,500.00~~

Sale Price: ฿9,350.00

1 of these items have been added to your shopping bag. To order more of this product click here.

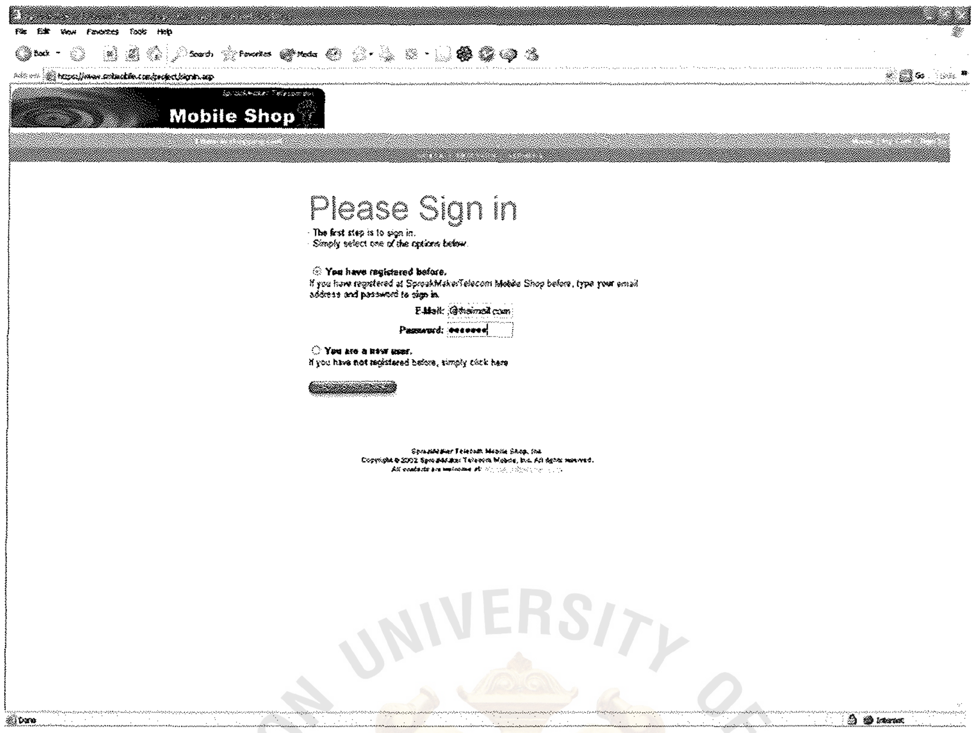
To continue shopping select another item to the left or a new category above.

To complete your order click here to checkout.

SpreadMaker Telecom Mobile Shop, Inc.
Copyright © 2002 SpreadMaker Telecom Mobile, Inc. All rights reserved
All contacts are welcome at: [illegible]

Figure 4.7. The Page Showing Ready to Make a Payment.

After the customer click "Add to Order" button from the previous step, they are leaded to the page as shown in Figure 4.7. There are three decision choices for customer. First, the customer can add more quantities of the product. Second, the customer can continue shopping by selecting another item at the left menu or new categories above. The last, the customer complete their purchasing order and ready to make a payment. The customer can click the provided image link of "check out" in order to lead them to payment process. The section of sending submitted data with digital signature is starting at this point.



Please Sign in

· The first step is to sign in.
· Simply select one of the options below.

[✓ You have registered before.]
If you have registered at SpeakMakerTelecom Mobile Shop before, type your email address and password to sign in.

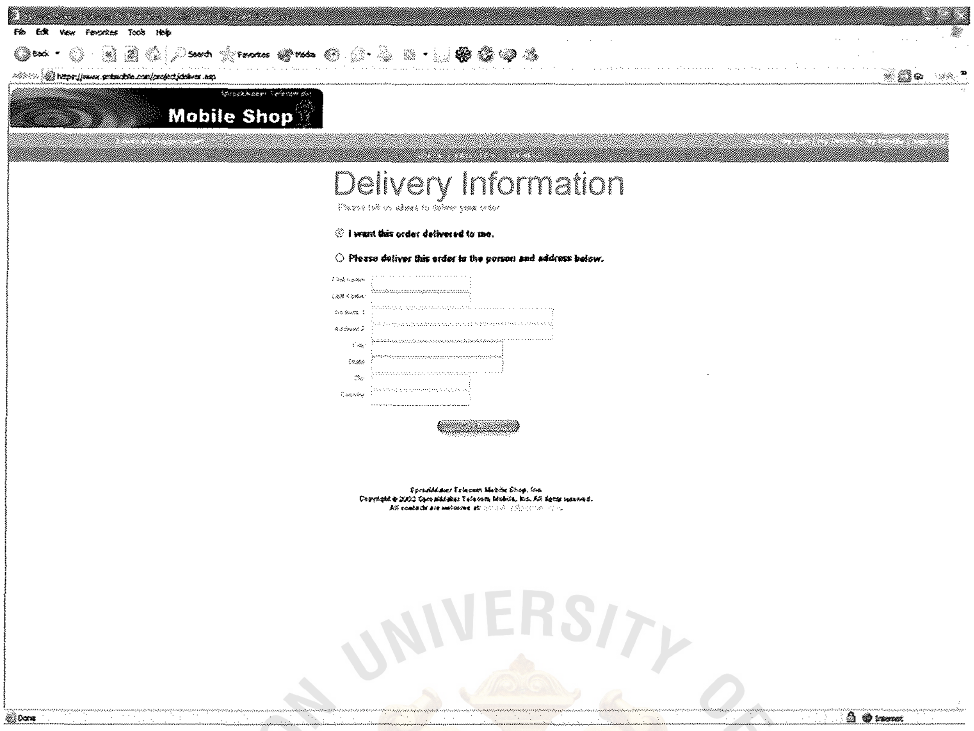
E-Mail: @thaimail.com
Password: •••••••

○ You are a new user.
If you have not registered before, simply click here

SpeakMaker Telecom Mobile Shop, Inc.
Copyright © 2002 SpeakMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at:

Figure 4.8. Sign-in as a Member Page.

From the previous step, when the customer clicks the link of "check out", they are required to check for signing in as a member. The sign in page as shown in Figure 4.8 will be loaded. The customer is required to fill out their e-mail address and password as a registered member before. If the customers are not members of the online store, they are required to register as a new member. After the customer has already signed in, they click the image button of "Sign up for 1 Click" to check delivery information.


File Edit View Favorites Tools Help
Back Search Favorites Media
https://www.smbmobile.com/project/deliver.asp
Mobile Shop
Done
Internet

Delivery Information

Please tell us where to deliver your order

[✓ I want this order delivered to me.]

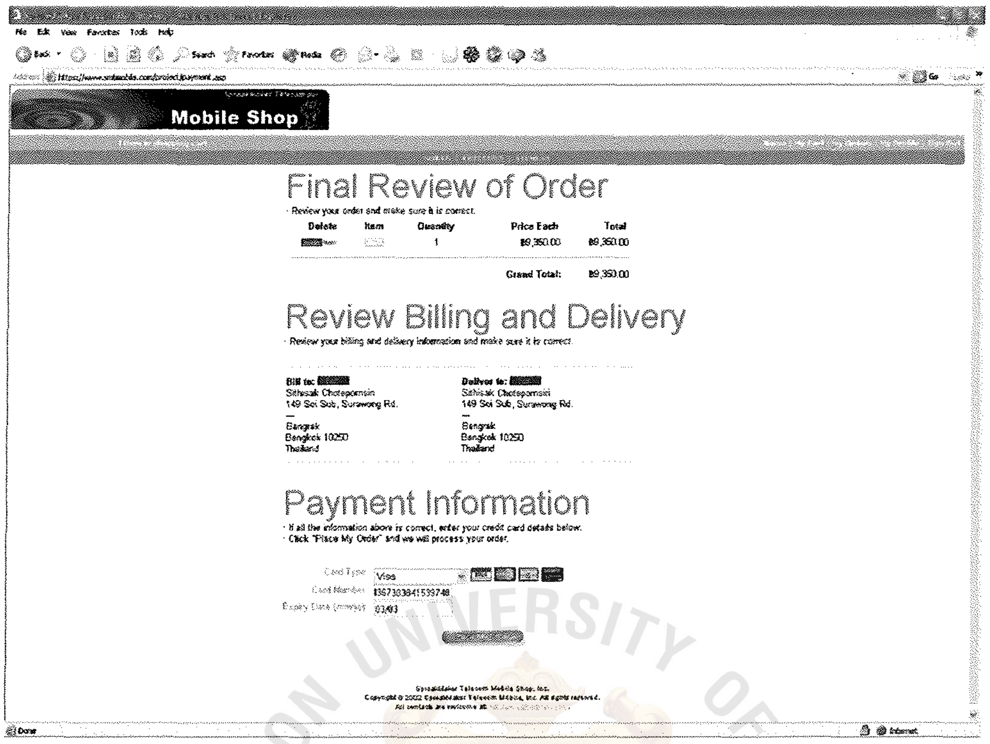
Please deliver this order to the person and address below.

First Name	
Last Name	
Address 1	
Address 2	
City	
State	
Zip	
Country	

SpreadMaker Telecom Mobile Shop, Inc.
Copyright © 2003 SpreadMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at: [illegible]

Figure 4.9. Delivery Information Page.

The Figure 4.9 shows the delivery information page after customers already sign-in. If the customers would like to change their delivery address information, they can fill out the new address into the fields. This is flexible for customers to change their delivery information. On the other hand, if the delivery and billing address information are the same, they don't take any actions into the fields. Then, they will move to the final review of the order page.



Final Review of Order

· Review your order and make sure it is correct.

Delete	Item	Quantity	Price Each	Total
		1	฿9,350.00	฿9,350.00
			Grand Total:	฿9,350.00

Review Billing and Delivery

· Review your billing and delivery information and make sure it is correct.

Bill to:	Deliver to:
Sithisak Chotepornsin	Sithisak Chotepornsin
149 Soi Sub, Surawong Rd.	149 Soi Sub, Surawong Rd.
—	—
Bangrak	Bangrak
Bangkok 10250	Bangkok 10250
Thailand	Thailand

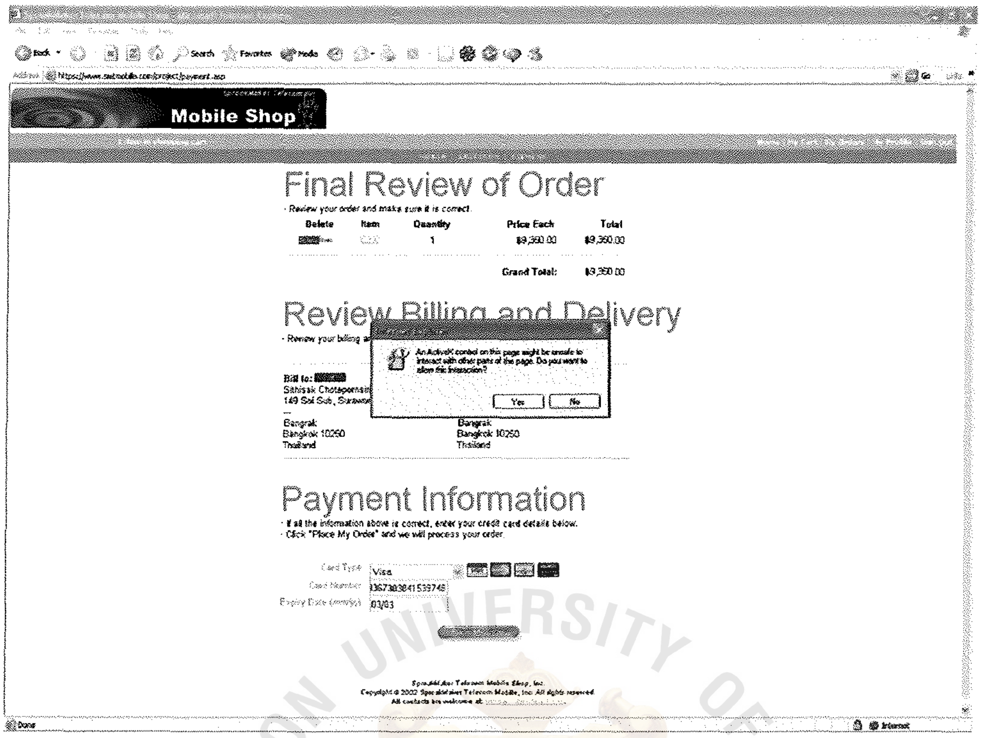
Payment Information

· If all the information above is correct, enter your credit card details below.
· Click "Place My Order" and we will process your order.

Card Type	Visa
Card Number	1357303841539748
Expiry Date (mm/yy)	03/03

Figure 4.10. Order Confirmation Page.

The Figure 4.10 shows the final review and payment page. The customers can finally check their order, and billing and delivery address. They can update both parts from this page. Then, payment information must be filled out before submitting the signed data to server side. Customers have to fill out their credit card information. If every part in this page is determined to be satisfied, then they can proceed to submit the purchasing order transaction with digital signature.


Mobile Shop

Final Review of Order

· Review your order and make sure it is correct.

Delete	Item	Quantity	Price Each	Total
		1	฿9,350.00	฿9,350.00
			Grand Total:	฿9,350.00

Review Billing and Delivery

· Review your billing a

An ActiveX control on this page might be unsafe to interact with other parts of the page. Do you want to allow this interaction?

Yes No

Bill to:
Sithisak Chotepornsi
149 Soi Sub, Surawo
—
Bangrak
Bangkok 10250
Thailand

Bangrak
Bangkok 10250
Thailand

Payment Information

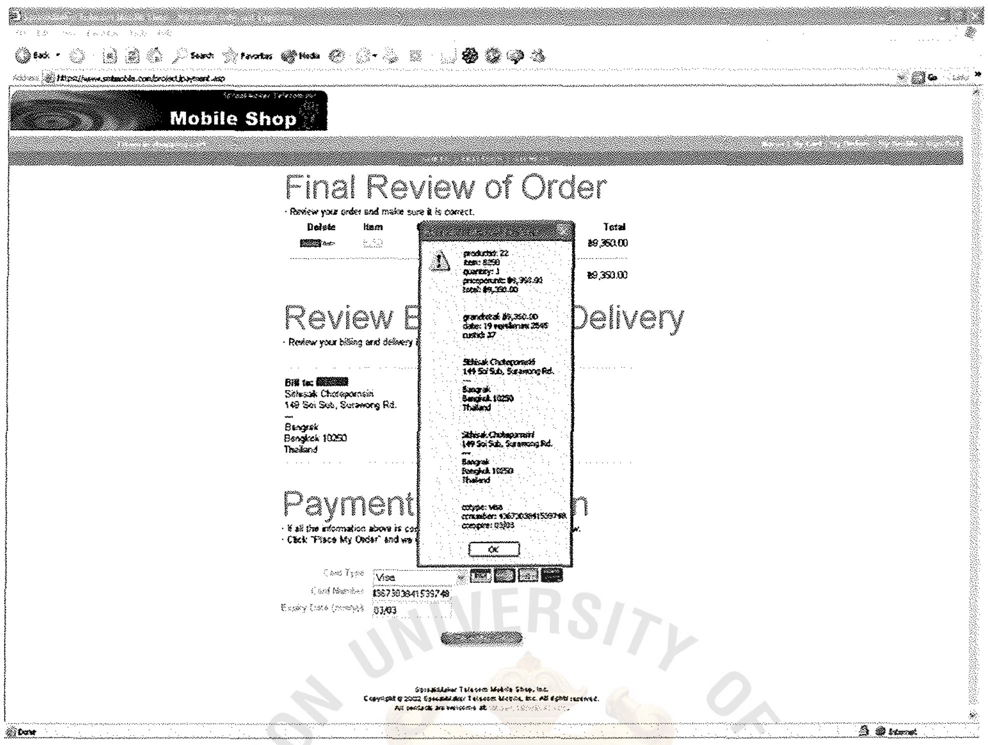
· If all the information above is correct, enter your credit card details below.
· Click "Place My Order" and we will process your order.

Card Type: Visa
Card Number: 3673838841539748
Expiry Date (mm/yy): 03/03

Specialist Telecom Mobile Shop, Inc.
Copyright © 2002 Specialist Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at

Figure 4.11. Alert Dialogue Box of ActiveX Control.

Now, the purchasing order transaction is ready to submit with digital signature. The customer can click the image button of "Place My Order" to proceed signing module before sending signed message to server side. The purchasing order transaction data to be signed will be combined as a string in the variable. The JavaScript command is used to call ActiveX Object written by Visual C++. Then, the alert page that warns employing the ActiveX Object will be pop up as shown in Figure 4.11. The signing module is proceeding in next step.


Mobile Shop

Final Review of Order

· Review your order and make sure it is correct.

Delete	Item	Total
[redacted]	[illegible]	฿9,350.00
		฿9,350.00

Review B… Delivery

· Review your billing and delivery …

Bill to: [redacted]
Sithisak Chotepornsin
149 Soi Sub, Surawong Rd.
—
Bangrak
Bangkok 10250
Thailand

Payment …

· If all the information above is co…
· Click "Place My Order" and we …

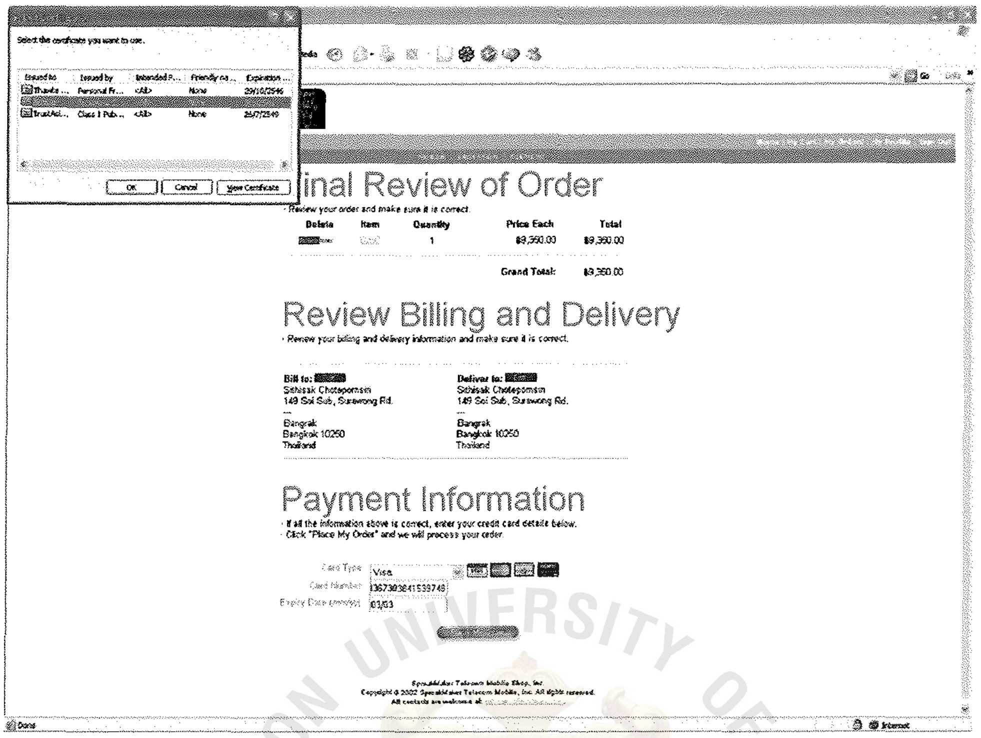
Card Type: Visa
Card Number: 4367303841539748
Expiry Date (mm/yy): 03/03

```
productid: 22
item: 8250
quantity: 1
priceperunit: ฿9,350.00
total: ฿9,350.00

grandtotal: ฿9,350.00
date: 19 กรกฎาคม 2545
custid: 37

Sithisak Chotepornsin
149 Soi Sub, Surawong Rd.
—
Bangrak
Bangkok 10250
Thailand

Sithisak Chotepornsin
149 Soi Sub, Surawong Rd.
—
Bangrak
Bangkok 10250
Thailand

cctype: visa
ccnumber: 4367303841539748
ccexpire: 03/03
```

Copyright © 2002 Speedmaker Telecom Mobile, Inc. All rights reserved.

Figure 4.12. Alert Dialogue Box of Data Confirmation.

After the customer clicks "Yes" button in the Figure 4.11, the alert page will be pop up to confirm the data to be signed as shown in Figure 4.12. The filled out data to be signed can be classified to five parts. The first part shows the data of product ID., item, quantity, unit price, and total, whereas, the data of grand total, date of order, and customer ID. are displayed in the second part. The billing and delivery address are shown in the third and forth part respectively. Lastly, payment information such as credit card type, credit card number, and expire date is displayed.



Select the certificate you want to use.

Issued to	Issued by	Intended P...	Friendly na...	Expiration ...
Thawte ...	Personal Fr...	<All>	None	29/10/2546
TrustAsi...	Class 1 Pub...	<All>	None	25/7/2549

inal Review of Order

· Review your order and make sure it is correct.

Delete	Item	Quantity	Price Each	Total
		1	฿9,360.00	฿9,360.00
			Grand Total:	฿9,360.00

Review Billing and Delivery

· Renew your billing and delivery information and make sure it is correct.

Bill to:	Deliver to:
Sithisak Chotepornsiri	Sithisak Chotepornsiri
149 Soi Sub, Surawong Rd.	149 Soi Sub, Surawong Rd.
---	---
Bangrak	Bangrak
Bangkok 10250	Bangkok 10250
Thailand	Thailand

Payment Information

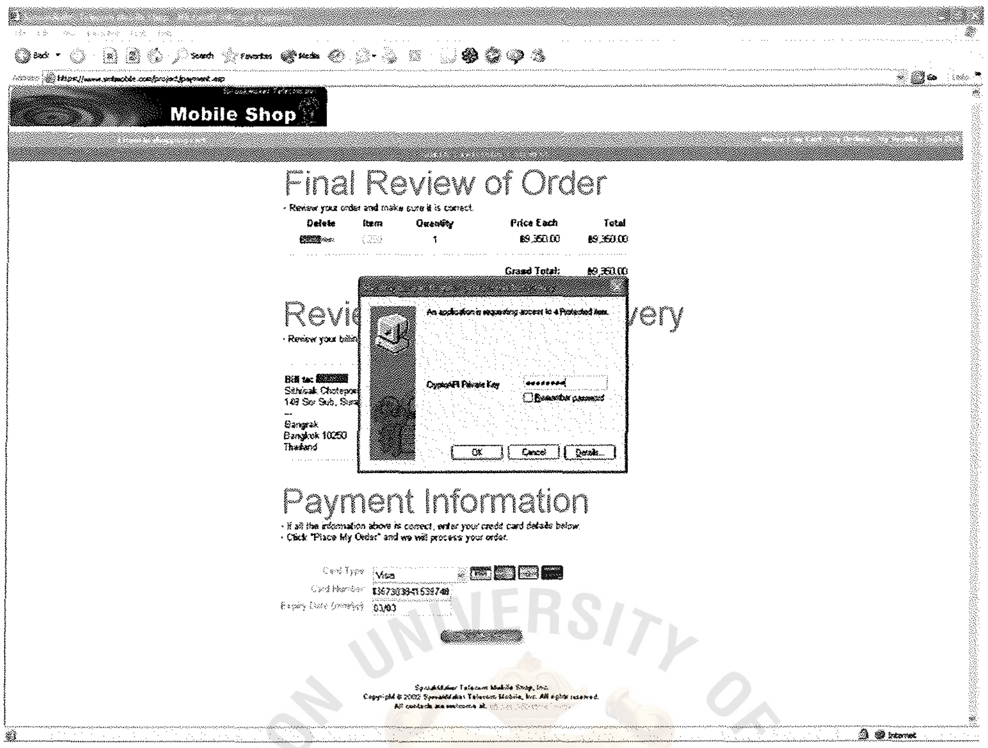
· If all the information above is correct, enter your credit card details below.
· Click "Place My Order" and we will process your order.

Card Type: Visa
Card Number: 367303841539748
Expiry Date (mm/yy): 03/03

SpeakMaker Telecom Mobile Shop, Inc.
Copyright © 2002 SpeakMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at:

Figure 4.13. Opening Certificate Store.

From the previous step, when the customer clicks "Yes" button on the alert page, they are coming to step of selecting personal certificate. In this step, the certificate store is opened and the customer can select their personal certificate from the certificate store. Only one personal certificate is allowed. In this case, in the Figure 4.13, the second personal certificate issued to "Sithisak Chotepornsiri" is selected. The customer can view more details of the certificate by clicking "View Certificate" button.



Final Review of Order

· Review your order and make sure it is correct.

Delete	Item	Quantity	Price Each	Total
[illegible]	[illegible]	1	฿9,350.00	฿9,350.00
			Grand Total:	฿9,350.00

Revie… …very

· Review your billin…

Bill to: [illegible]
Sithisak Chotepo…
149 Soi Sub, Sura…
--
Bangrak
Bangkok 10250
Thailand

An application is requesting access to a Protected item.

CryptoAPI Private Key ********

☐ Remember password

OK Cancel Details...

Payment Information

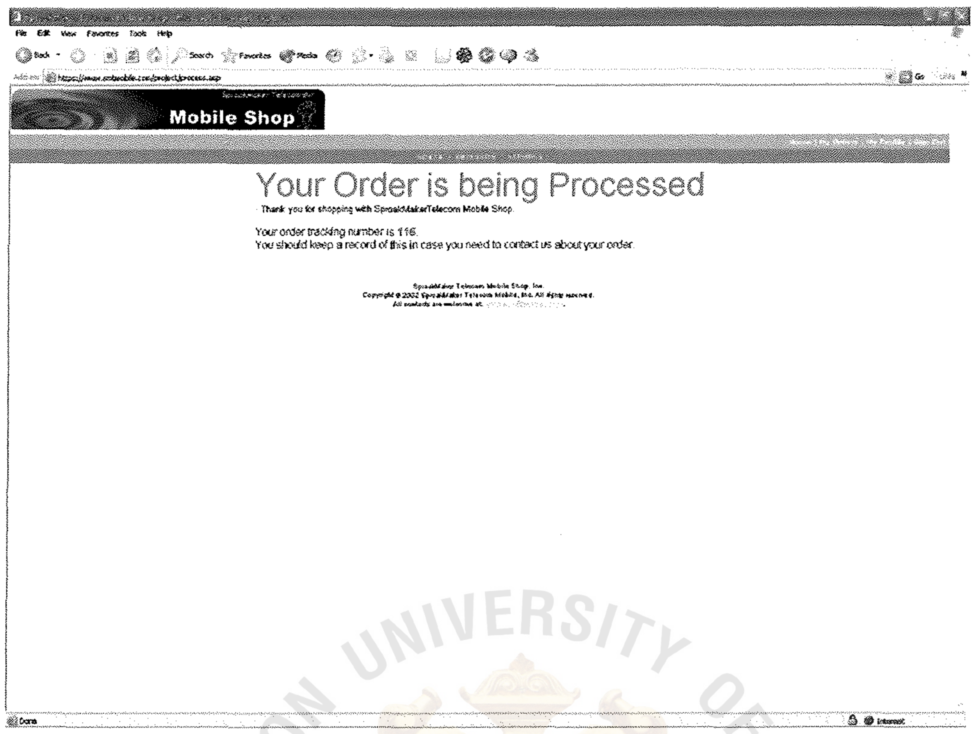
· If all the information above is correct, enter your credit card details below.
· Click "Place My Order" and we will process your order.

Card Type: Visa
Card Number: 4367303841539749
Expiry Date (mm/yy): 03/03

SpeakMaker Telecom Mobile Shop, Inc.
Copyright © 2002 SpeakMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at: [illegible]

Figure 4.14. Private Key Password of Personal Certificate.

After the customer already selects their personal certificate, the dialogue box of "Signing data with your private exchange key" will be pop up in case of high security as shown in Figure 4.14. The customer is required to fill out the password of private key. In case of low security, the password of private key is not required and this step is skipped to the next one.



Your Order is being Processed

Thank you for shopping with SpeakMakerTelecom Mobile Shop.

Your order tracking number is 116.
You should keep a record of this in case you need to contact us about your order.

SpeakMaker Telecom Mobile Shop, Inc.
Copyright © 2002 SpeakMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at.

Figure 4.15. Successful Purchasing Order Page.

Now, the purchasing order transaction is digitally signed and submitted to the web server. The signed data will be stored in the MS Access database in the table of "orders." Then, the page indicates that the purchasing order transaction is successful will be returned together with order tracking number as shown in Figure 4.15. In this case, the order tracking number is #116. The order tracking number should be remembered beneficially for demonstration of verifying and comparing valid versus invalid verification.

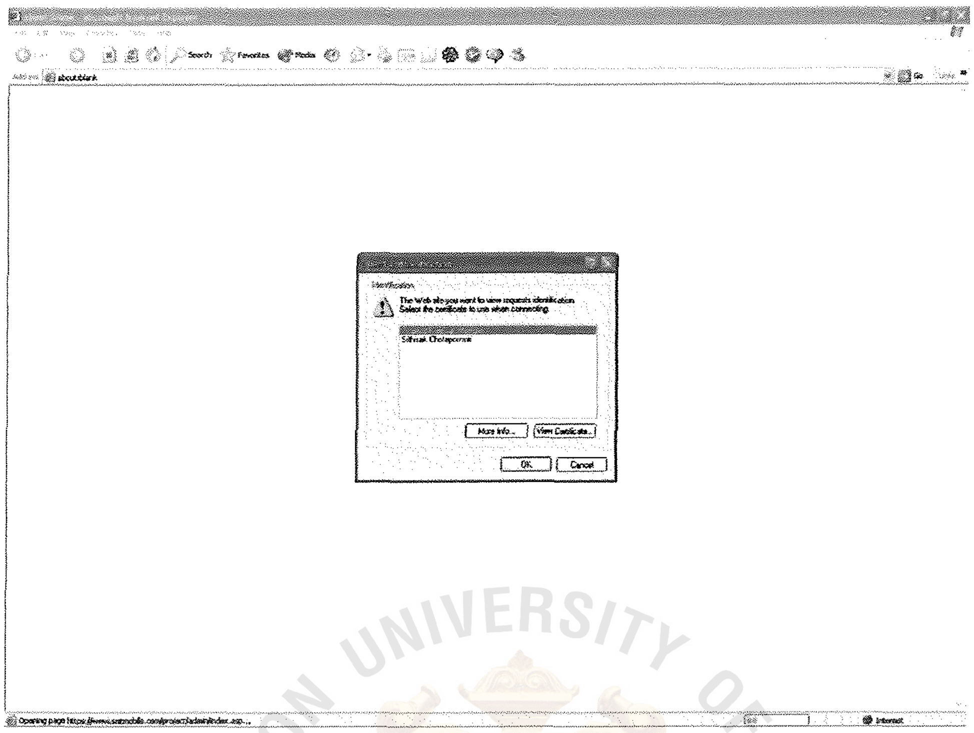
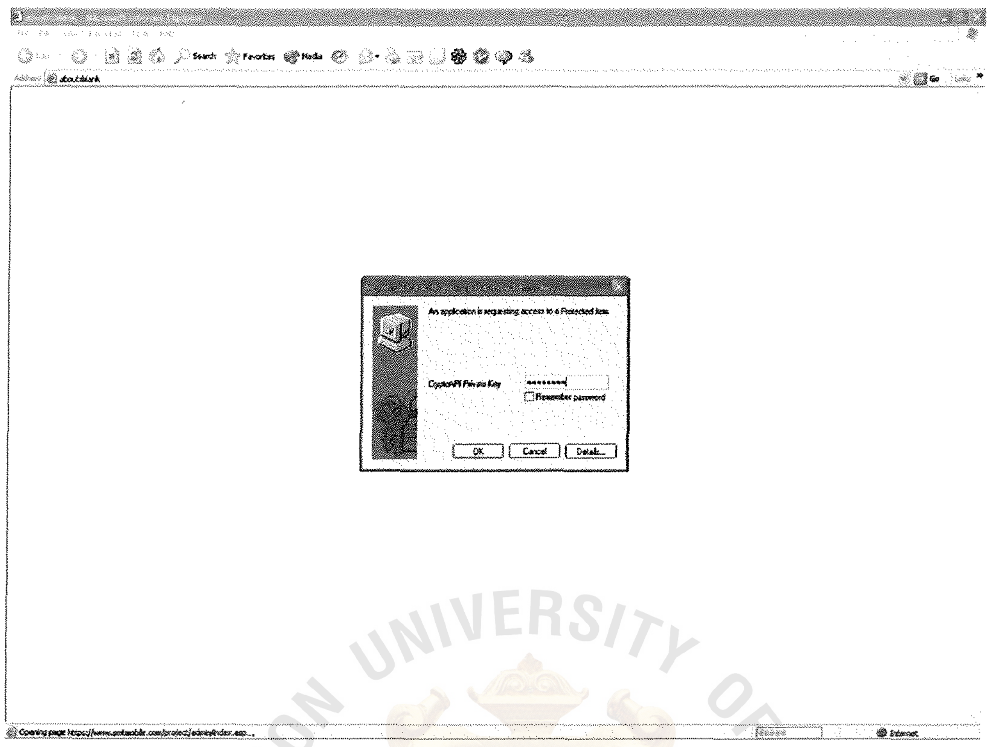

Identification
The Web site you want to view requests identification.
Select the certificate to use when connecting.
Sithisak Cholapoonrat
More Info...
View Certificate...
OK
Cancel

Figure 4.16. Client Certificate Requirement for Entering to Admin Page.

In the previous section, the signed data is sent to the web server. In this section, the verification of signed data is demonstrated. The webmaster has authorized to view the customers' transaction through admin page. In Figure 4.16, when the webmaster accesses into the admin page of https://www.smtmobile.com/project/admin/index.asp, the webmaster is required to authenticate for entering into the admin page.

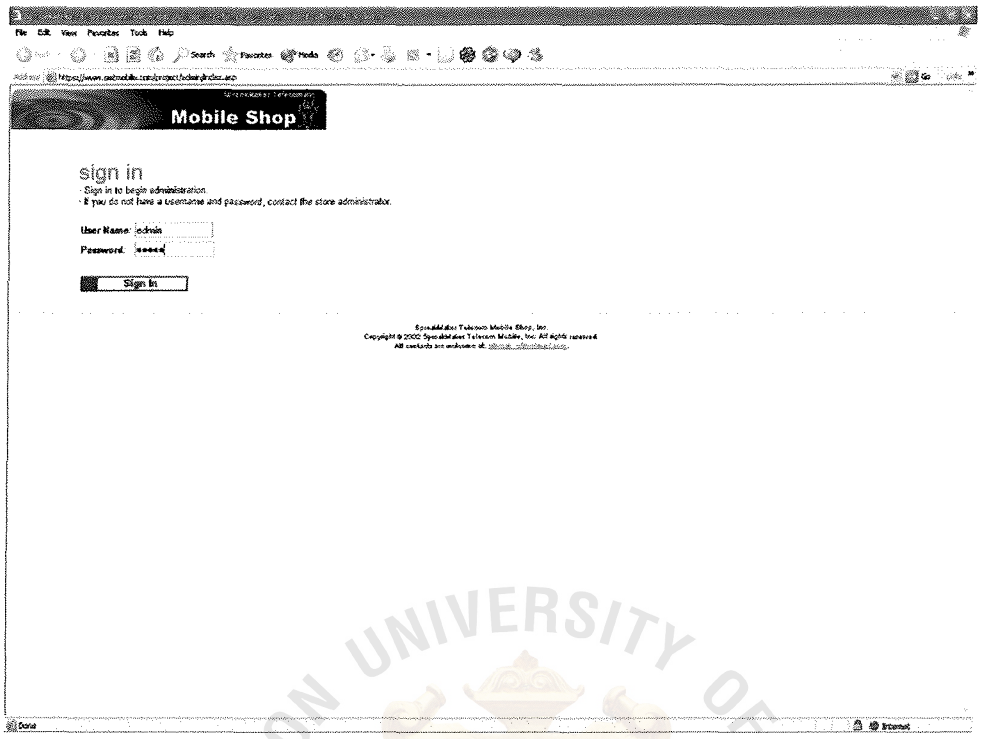
The certificate store will be opened to select their personal certificate if there are many personal certificates in the machine. On the other hand, if there is only one personal certificate in the machine, the process will skip to fill out CrytoAPI Private Key Password. The customer is not necessary to select personal certificate from certificate store.


An application is requesting access to a Protected item.

CryptoAPI Private Key ********

Figure 4.17. Private Key Password for Entering Admin Page.

From the previous step, when the webmaster clicks "OK" button on the dialogue box of Client Authentication, the dialogue box of Signing data with your private exchange key will be generated in case of high security as shown in Figure 4.17. The webmaster is required to fill out the password of private key. In case of low security, the webmaster is authenticated to access into the admin page without requiring private key password after selecting personal certificate in certificate store.



sign in

· Sign in to begin administration.
· If you do not have a username and password, contact the store administrator.

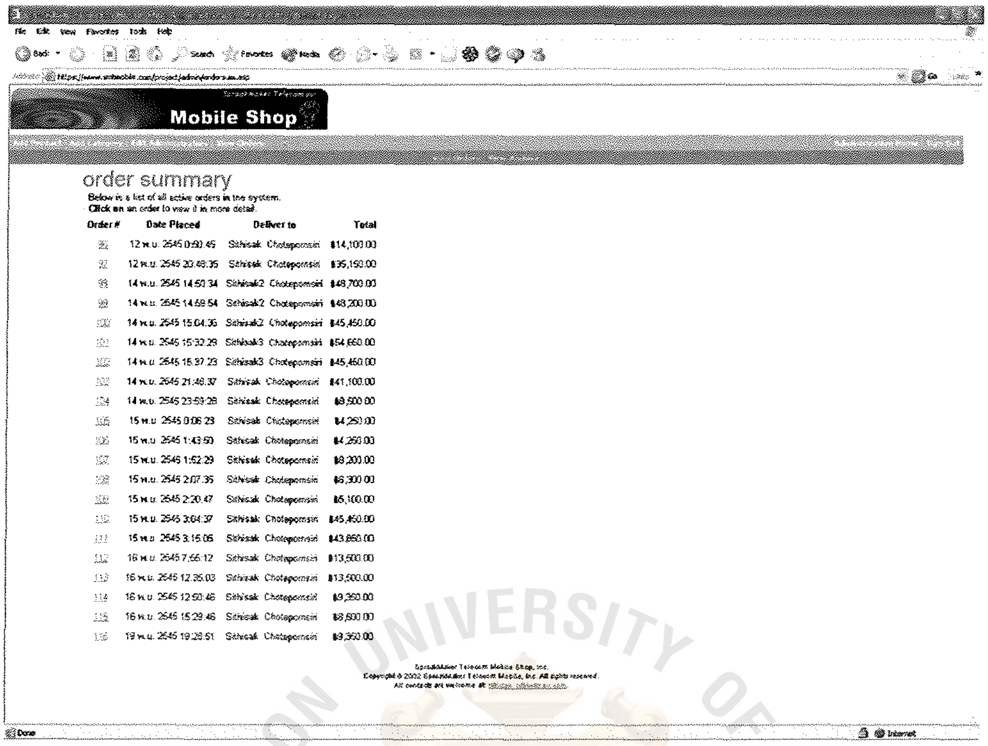
User Name: admin
Password: *****

Sign In

Specialdistributor Telecom Mobile Shop, Inc.
Copyright © 2002 Specialdistributor Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at: [illegible]

Figure 4.18. Webmaster Sign in Page.

After determining authentication for accessing the admin page, if the webmaster has authorized to enter into the admin page, the webmaster sign in page will be launched as shown in Figure 4.18. The webmaster is required to fill out username and password as an administrator status for administration. Then, they can access to view the customers' transaction.



Mobile Shop

order summary

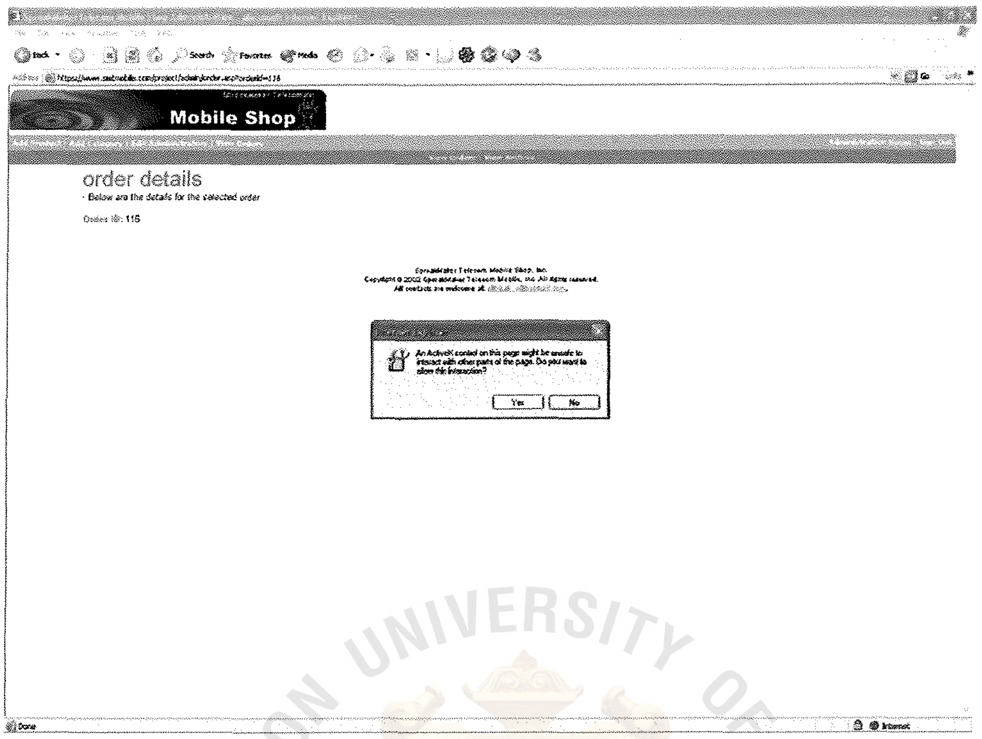
Below is a list of all active orders in the system.
Click on an order to view it in more detail.

Order #	Date Placed	Deliver to	Total
[illegible]	12 พ.ย. 2545 0:50:45	Sithisak Chotepornsiri	฿14,100.00
[illegible]	12 พ.ย. 2545 20:48:35	Sithisak Chotepornsiri	฿35,150.00
98	14 พ.ย. 2545 14:50:34	Sithisak2 Chotepornsiri	฿48,700.00
99	14 พ.ย. 2545 14:59:54	Sithisak2 Chotepornsiri	฿48,200.00
100	14 พ.ย. 2545 15:04:36	Sithisak2 Chotepornsiri	฿45,450.00
101	14 พ.ย. 2545 15:32:29	Sithisak3 Chotepornsiri	฿54,660.00
102	14 พ.ย. 2545 15:37:23	Sithisak3 Chotepornsiri	฿45,450.00
103	14 พ.ย. 2545 21:48:37	Sithisak Chotepornsiri	฿41,100.00
104	14 พ.ย. 2545 23:59:28	Sithisak Chotepornsiri	฿8,500.00
105	15 พ.ย. 2545 0:06:23	Sithisak Chotepornsiri	฿4,250.00
106	15 พ.ย. 2545 1:43:50	Sithisak Chotepornsiri	฿4,250.00
107	15 พ.ย. 2545 1:52:29	Sithisak Chotepornsiri	฿8,200.00
108	15 พ.ย. 2545 2:07:35	Sithisak Chotepornsiri	฿6,300.00
109	15 พ.ย. 2545 2:20:47	Sithisak Chotepornsiri	฿5,100.00
110	15 พ.ย. 2545 3:04:37	Sithisak Chotepornsiri	฿45,450.00
111	15 พ.ย. 2545 3:15:05	Sithisak Chotepornsiri	฿43,850.00
112	16 พ.ย. 2545 7:56:12	Sithisak Chotepornsiri	฿13,500.00
113	16 พ.ย. 2545 12:35:03	Sithisak Chotepornsiri	฿13,500.00
114	16 พ.ย. 2545 12:50:46	Sithisak Chotepornsiri	฿9,350.00
115	16 พ.ย. 2545 15:29:46	Sithisak Chotepornsiri	฿8,600.00
116	19 พ.ย. 2545 19:28:51	Sithisak Chotepornsiri	฿9,350.00

Speedmaker Telecom Mobile Shop, Inc.
Copyright © 2002 Speedmaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at [illegible]

Figure 4.19. Order Transaction Summary Page.

The Figure 4.19 depicts the customers' order transaction summary page. The orders lists will be retrieved from web server sorted by order ID. Each transaction consists of order tracking number, date placed, name delivered, and grand total. The webmaster can view in more details of customers' order transaction by clicking the link of order tracking number in each list. Then, the verification of signed data will begin at this point.


Mobile Shop

order details

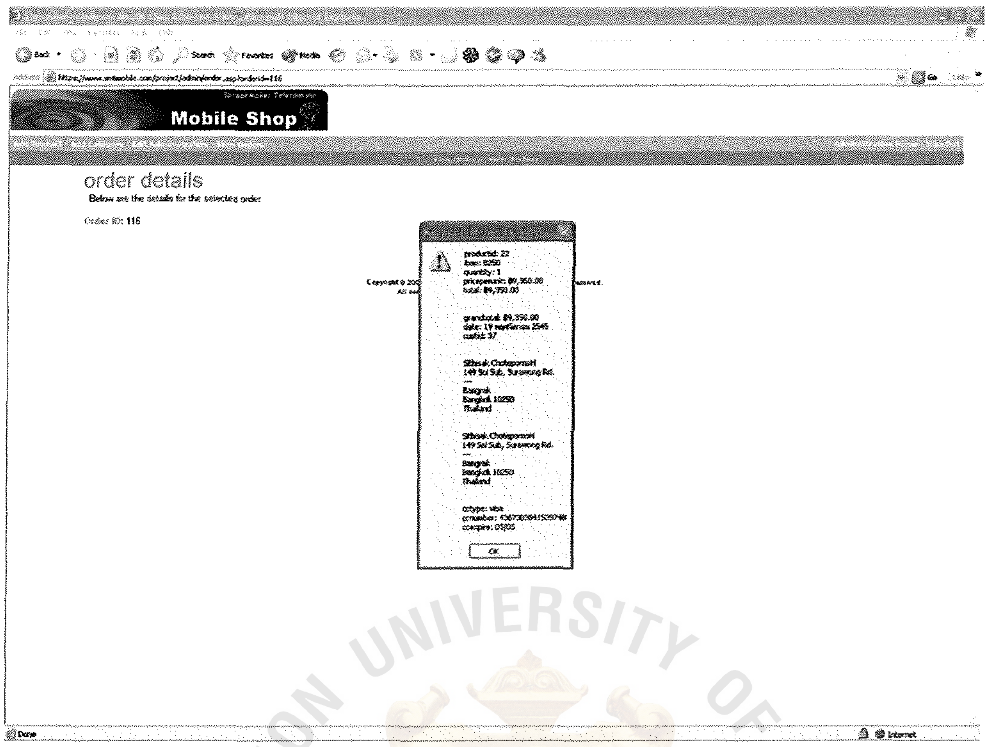
· Below are the details for the selected order

Order ID: 116

An ActiveX control on this page might be unsafe to interact with other parts of the page. Do you want to allow this interaction?

Figure 4.20. Alert Dialogue Box of ActiveX Control for Verification.

From the previous step, the webmaster clicks the link of order tracking number to view more details of customers' transaction. Then, he arrives to the page of details of customers' transaction. In this page, the JavaScript command is used to call ActiveX Object written by Visual C++ for verification. Then, the alert page that warns employing the ActiveX Object will be pop up as shown in Figure 4.20. The verifying module is proceeding in this step.



Mobile Shop

order details

Below are the details for the selected order

Order ID: 116

productid: 22
item: 8250
quantity: 1
priceperunit: ฿9,350.00
total: ฿9,350.00

grandtotal: ฿9,350.00
date: 19 [illegible] 2545
custid: 37

Sithisak Chotepornsiri
149 Soi Sub, Surawong Rd.

Bangrak
Bangkok 10250
Thailand

Sithisak Chotepornsiri
149 Soi Sub, Surawong Rd.

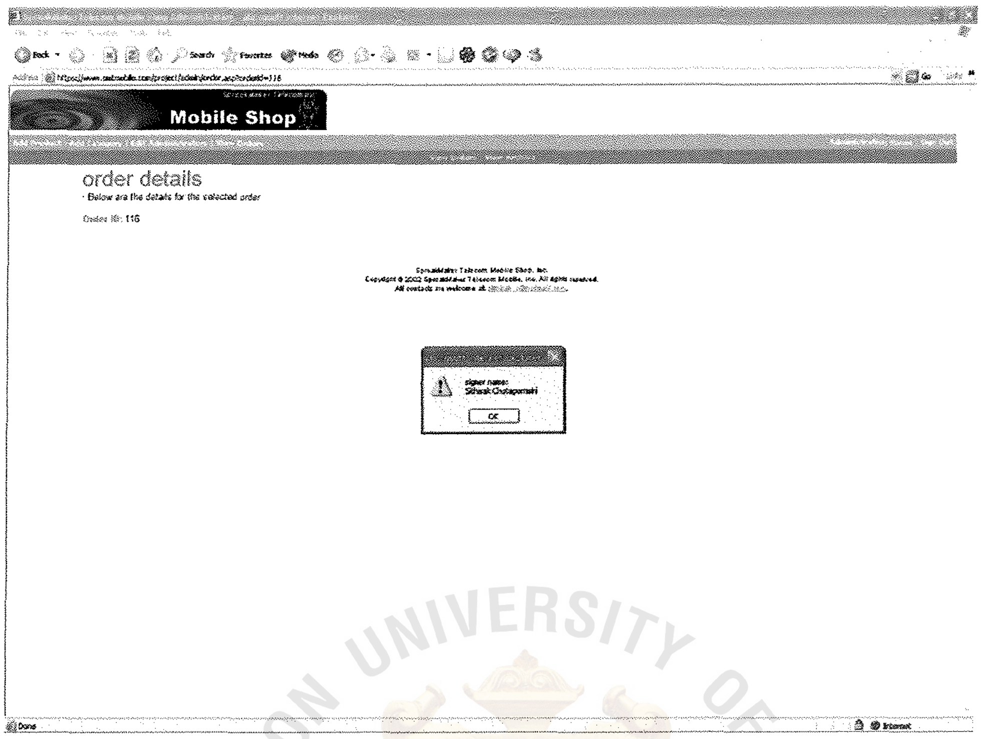
Bangrak
Bangkok 10250
Thailand

cctype: visa
ccnumber: 4567300941505798
ccexpire: 05/05

OK

Figure 4.21. Alert Dialogue Box of Plain Text to be Verified.

After the customer clicks "Yes" button in the Figure 4.20, the alert page will be pop up to display the data to be verified as shown in Figure 4.21. The filled out data to be verified can be classified to five parts. The first part shows the data of product ID., item, quantity, unit price, and total, whereas, the data of grand total, date of order, and customer ID. are displayed in the second part. The billing and delivery address are shown in the third and forth part respectively. Lastly, payment information such as credit card type, credit card number, and expire date is displayed. We can notice that the plain text after verification that shown in the alert dialogue box are same as the plain text to be signed in the Figure 4.12.


Mobile Shop

order details

· Below are the details for the selected order

Order ID: 116

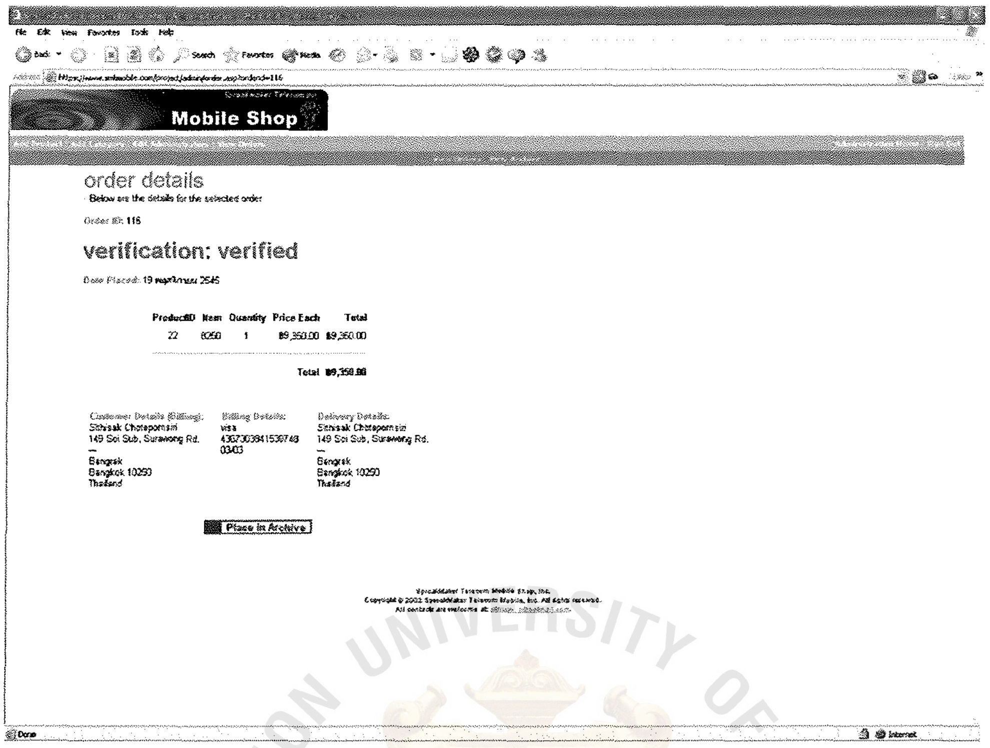
SpecialMaker Telecom Mobile Shop, Inc.
Copyright © 2002 SpecialMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at [illegible]

signer name:
Sithasak Chotapornsiri

OK

Figure 4.22. Dialogue Box Showing Signer Name.

The Figure 4.22 depicts the alert dialogue box that shows the signer name. After the webmaster clicks "Yes" button in the Figure 4.21, the signer name from personal certificate will appear on the alert dialogue box. This step indicates to prove the identification. The customer can not repudiate making this transaction. Moreover, it can be also used as evidence in the courts.



Mobile Shop

order details

Below are the details for the selected order

Order ID: 116

verification: verified

Date Placed: 19 พฤศจิกายน 2545

ProductID	Item	Quantity	Price Each	Total
22	8250	1	฿9,350.00	฿9,350.00
			Total	฿9,350.00

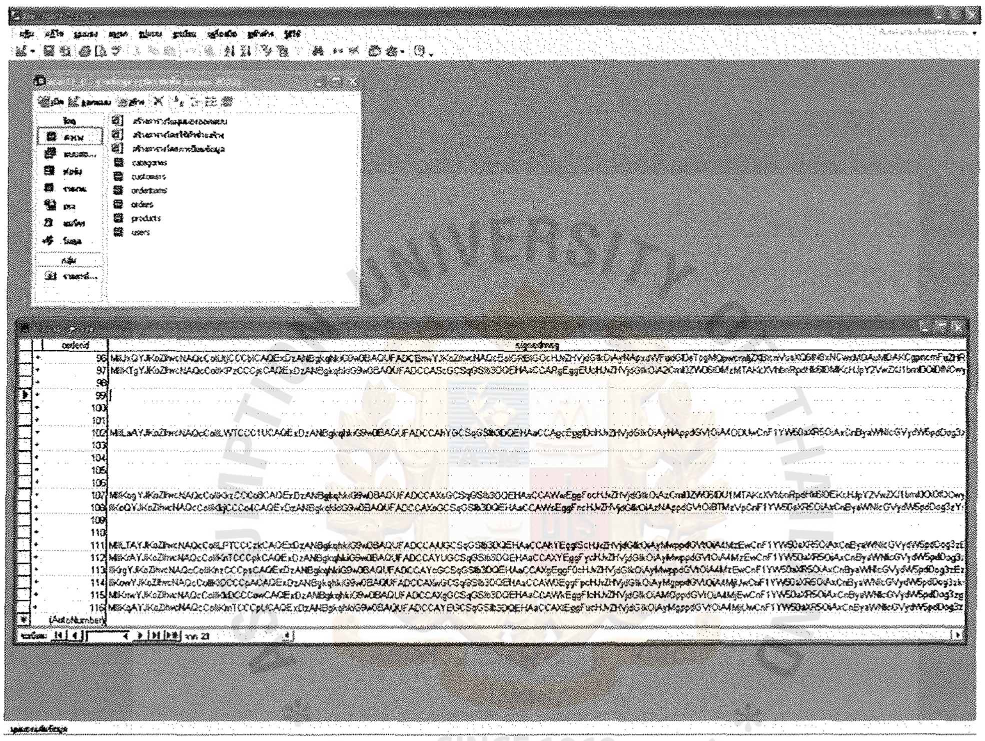
Customer Details (Billing):	Billing Details:	Delivery Details:
Sithisak Chotepornsiri	visa	Sithisak Chotepornsiri
149 Soi Sub, Surawong Rd.	4367303841539743	149 Soi Sub, Surawong Rd.
—	03/03	—
Bangrak		Bangrak
Bangkok 10250		Bangkok 10250
Thailand		Thailand

Figure 4.23. Successful Verifying the Signed Data.

The Figure 4.23 depicts the details of customers' transaction that are successfully verified. In this case, the order tracking number #116 is verified as a valid verification. Then, the details of order transaction will be displayed on the page and appeared the message of successful verifying. As a result, this order transaction is reliable. The webmaster can insist the integrity of this order transaction that has not been altered.

4.2 Demonstrating of Invalid Verifying Procedure Output

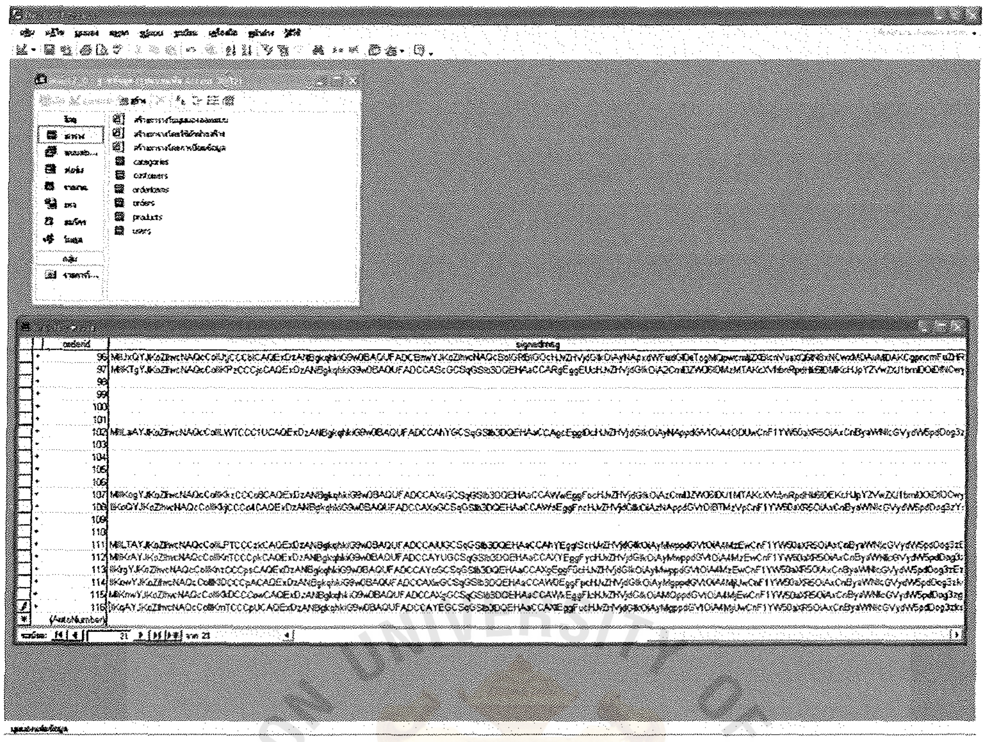
In the previous section, we had demonstrated the working of web signed-form and performing of valid verifying output. Now, the demonstrating of invalid verifying output is being explored in case of altered signed-data.



orderid	signedmsg
96	[illegible]
97	[illegible]
98	
99	
100	
101	
102	[illegible]
103	
104	
105	
106	
107	[illegible]
108	[illegible]
109	
110	
111	[illegible]
112	[illegible]
113	[illegible]
114	[illegible]
115	[illegible]
116	[illegible]
(AutoNumber)	

Figure 4.24. Original Signed Data Stored in Database.

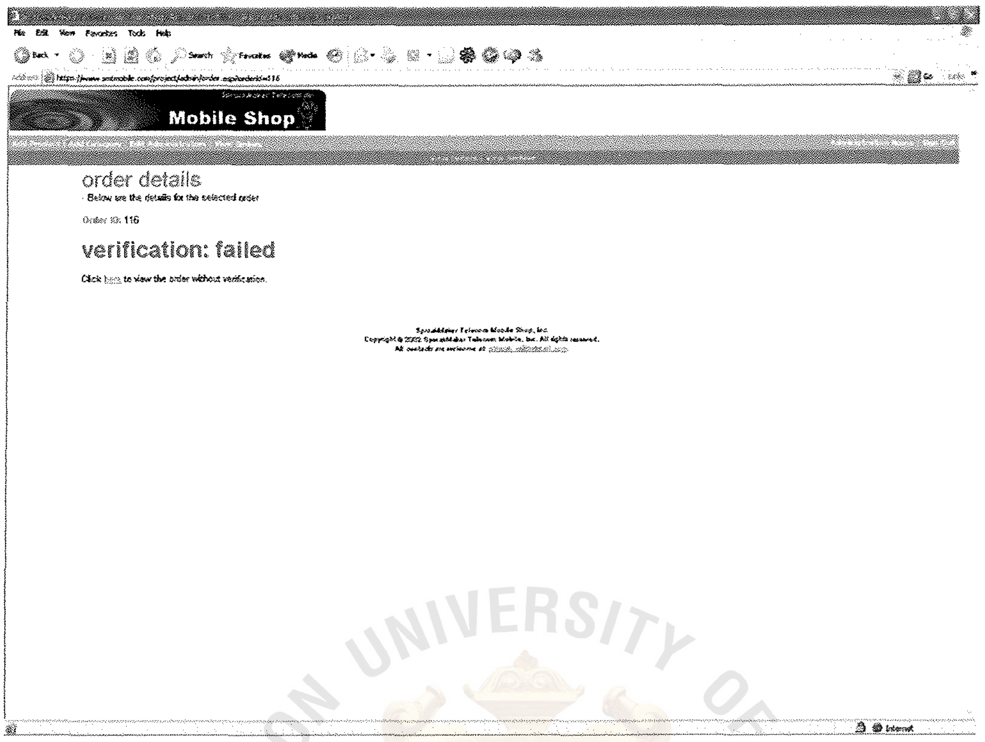
When the customer submitted the signed data to the web server, the signed data had been stored in the MS Access database. The format of signed data that is kept in the table of order is encoded with base64 encoding as friendly text. In Figure 4.24, the order tracking number #116 is focused to alter the signed data.



orderid	signedmsg
96	[illegible]
97	[illegible]
98	
99	
100	
101	
102	[illegible]
103	
104	
105	
106	
107	[illegible]
108	[illegible]
109	
110	
111	[illegible]
112	[illegible]
113	[illegible]
114	[illegible]
115	[illegible]
116	[illegible]
(AutoNumber)	

Figure 4.25. Signed Data in Database Altered.

The signed data that is altered should be invalid data. In Figure 4.25, the signed data of order tracking number #116 is altered or changed. Suppose we change or alter the original singed data by eliminating the first character. The Figure 4.24 and Figure 4.25 are different. We notice that the character "M" is cut. Then, we save the new change in the database.


Mobile Shop

order details

· Below are the details for the selected order

Order ID: 116

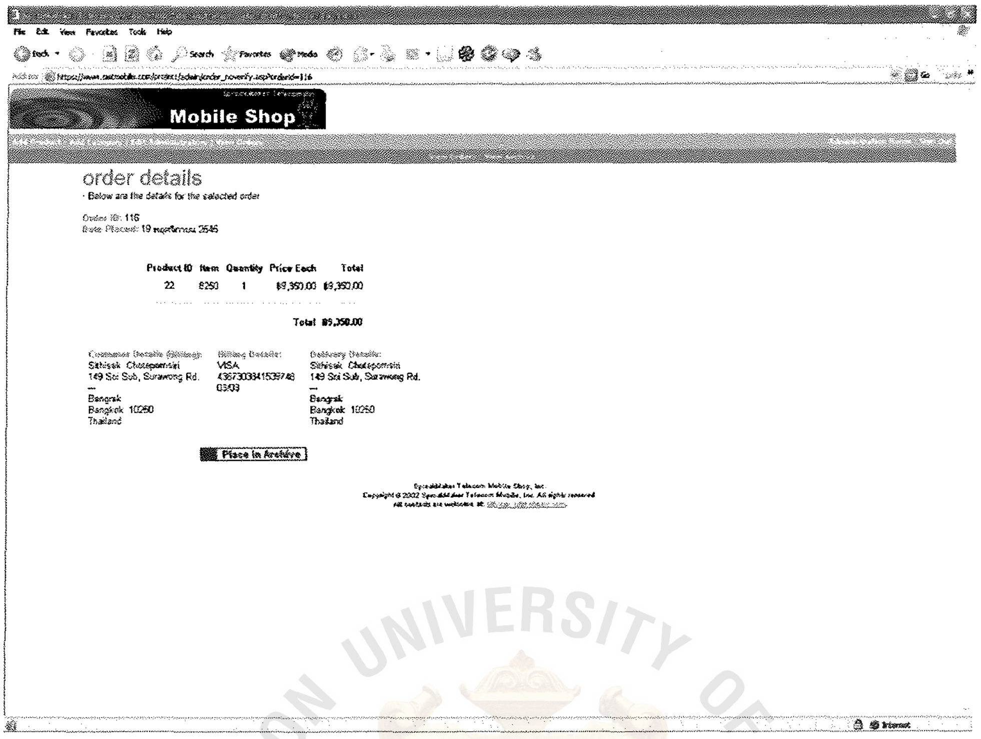
verification: failed

Click here to view the order without verification.

SpeakMaker Telecom Mobile Shop, Inc.
Copyright © 2002 SpeakMaker Telecom Mobile, Inc. All rights reserved.
All contacts are welcome at [illegible]

Figure 4.26. Invalid Verification Page.

Now, we are back to view order transaction summary on the admin page again as shown in Figure 4.19. Then, we click the link of order tracking number #116 to view more details of the transaction. The alert dialogue box of ActiveX Control will be pop up as shown in Figure 4.20. The ActiveX Object is called to verify the signed data. In this case, the verification is invalid and returns the page of unsuccessful verifying as shown in Figure 4.26. The details of order transaction will not appear on the page. However, the webmaster can view the order transaction without verification by clicking provided link.


File Edit View Favorites Tools Help
Back
Search
Favorites
Media
https://www.cmtmobile.com/project/admin/order_noverify.asp?orderid=116
Mobile Shop
Place in Archive
Internet

order details

· Below are the details for the selected order

Order ID: 116
Date Placed: 19 พฤศจิกายน 2546

Product ID	Item	Quantity	Price Each	Total
22	8250	1	฿9,350.00	฿9,350.00

Total ฿9,350.00

Customer Details (Billing):	Billing Details:	Delivery Details:
Sithisak Chotepornsiri	VISA	Sithisak Chotepornsiri
149 Soi Sub, Surawong Rd.	4367303841539748	149 Soi Sub, Surawong Rd.
---	03/03	---
Bangrak		Bangrak
Bangkok 10250		Bangkok 10250
Thailand		Thailand

Spreadtrum Telecom Mobile Shop, Inc.
Copyright © 2002 Spreadtrum Telecom Mobile, Inc. All rights reserved
All contacts are welcome at: [illegible]

Figure 4.27. Viewing Order Details without Verification.

After the verification is invalid, the details of order transaction can be viewed without verification as shown in Figure 4.27. However, this order transaction is unreliable. It indicates that the transaction is altered from the original. As a result, this transaction can be discarded.

In conclusions, a demonstration of online mobile shop is indicated to show how web signed-form can work and manifest valid and invalid verification procedure output. This demonstration will make it easy to understand after working process has been explained in the previous chapter.

V. CONCLUSIONS AND RECOMMENDATIONS

5.1 Evaluation

According to this project, some security knowledge of digital signature and certificate are applied to online store in electronic commerce system. They allow web forms to be digitally signed, enabling online business communications and transactions to be certified and authenticated. The model of this project can abate some problems of commerce in cyber world between both parties such as identity, reliability, integrity, as well as action repudiation.

The model of this project will conduct to enhance approach of making online transaction through web browser and new vision of commerce in cyber world. The signed data is more reliable in the system of electronic commerce. The certificate of the signer insists to the person who makes the online transaction as identity. As a result, any persons who make online transaction can not repudiate their action. This can abate the problem of loss of cost and time for webmaster to check many junk transactions. In addition, the customers can convince the integrity of data that is sent to webmaster has not been altered.

However, there are a few limitations in the model of this project. The storage of customer's certificate including own private key are kept on individual PC machine and can be retrieved from MS Outlook Express. Because of the online store can be accessed at any place around the world, so the private key will be exported to file in PKCS#12 format and load to any PC machine they use. This is inconvenient for customers to carry the diskette that contain private key file to everywhere. Unlike online store, MS Outlook Express is entered only to individual PC machine, so there is no problem for e-mail. The other limitation is about browser to display the web page. The supported browser in this project is MS Internet Explorer only. This is because the ActiveX component object is supported the **CryptoAPI** library by Microsoft platform SDK.

5.2 Conclusions

The project proposed to enhance and implement on the security of E-commerce system. The security system becomes important obstacles for development cyber world commerce such as identity, reliability, non-repudiation, integrity, and so on. In order to abate some problems, the digital signature and certificate are applied for this project. Web signed-form is created for customers to make transaction and signing with their digital signature. Online transaction will be digitally signed by customers and verified by webmaster.

All working process of web signed-form is explored in order to know and understand how the process does it work. The process starts from customers coming to the web site and make transaction until finishing to the process of webmaster verifying signed data. The customers can load the web page from web server and make their transaction until ready to submit the order. They are required for signing digital signature. The order data is digitally signed by the process of signing module and keep into the database.

The signed data can be loaded to view the order from the web server by webmaster. Working process of verifying is proceeded in this step. Then, the order detail is returned with the result of verifying. All these working process can be easily understood by demonstration step by step in the chapter 4.

5.3 Recommendations

This project can continue to develop for suitable in the future use. According to limitation of the storage of certificate including with private key, now, customers have to import their private key in the format of PKCS#12 from diskette to PC machine every time in case of the customers access into the web page from any available PC machine that they are not own. This makes inconvenient for customers to make a transaction. In the near future, if smart card technology is popular, it can be applied into this project. The customer's private key can be kept into the smart card. It will be flexible for usage more than the present. In addition, this project is developed only sign function in the present. It can be developed for creating functions of encryption or sign and encryption in the future.

APPENDIX A

GLOSSARY

GLOSSARY

Certificate store

Typically, a permanent storage where certificates, certificate revocation lists (CRLs), and certificate trust lists (CTLs) are stored. It is possible, however, to create and open a certificate store solely in memory when working with certificates that do not need to be put in permanent storage. The certificate store is central to much of the certificate functionality in CryptoAPI.

Certificate trust list (CTL)

A predefined list of items that have been signed by a trusted entity. A CTL can be anything, such as a list of hashes of certificates, or a list of file names. All the items in the list are authenticated (approved) by the signing entity.

Cryptographic service provider (CSP)

An independent software module that actually performs cryptography algorithms for authentication, encoding, and encryption.

Digital signature

A digital string that is bundled with the message or transmitted separately. They are used to authenticate messages. A valid digital signature confirms that the message has not been tampered with, and may also identify the entity who signed the message. Signing a message does not alter the message itself.

Hash

A fixed-size result obtained by applying a mathematical function (the *hashing algorithm*) to an arbitrary amount of data. (Also known as "message digest.")

Hash object

An object used to hash messages or session keys. The hash object is created by a call to **CryptCreateHash**. The definition of the object is defined by the CSP specified in the call.

Hashing algorithm

An algorithm used to produce a hash value of some piece of data, such as a message or session key. Typical hashing algorithms include MD2, MD4, MD5, and SHA-1.

Hashing functions

A set of functions used to create and destroy hash objects, get or set the parameters of a hash object, and hash data and session keys.

Key container

A part of the key database that contains all the key pairs (exchange and signature key pairs) belonging to a specific user. Each container has a unique name that is used when calling **CryptAcquireContext** to get a handle to the container.

Private key

The secret half of a key pair used in a public key algorithm. Private keys are typically used to encrypt a symmetric session key, digitally sign a message, or decrypt a message that has been encrypted with the corresponding public key.

Public key

A cryptographic key typically used when decrypting a session key or a digital signature. The public key can also be used to encrypt a message, guaranteeing that only the person with the corresponding private key can decrypt the message.

Signed data

A data content type defined by PKCS #7. This data type consists of any type of content plus encrypted message hashes (digests) of the content for zero or more signers. The resulting hash(es) can be used to confirm who signed the message. These hashes also confirm that the original message has not been modified since the message was signed.

BIBLIOGRAPHY

English References

1. Atreya, Mohan, et al. Digital Signatures. New York: McGaw-Hill, c2002.

2. Laudon, Kenneth C. and Jane P. Laudon. Management Information Systems. New Jersy: Printice-Hall International, c2000.

3. Ryan, Russell and Teri Bidwell. Hack Proofing Your E-commerce. Rockland: Syngress Publishing, c2001.

4. Stallings, William. Cryptography and Network Security. New Jersey: Printice-Hall International, c1999.

5. Turban, Efraim, et al. Electronic Commerce a Managerial Perspective. New Jersy: Printice-Hall International, c2000.

Thai References

1. ธวัชชัย สุริยะทองธรรม, ธาริน สิทธิธรรมชารี, และ ประชา พฤกษ์ประเสริฐ. สร้างเว็บเพจอย่างไร้ขีดจำกัด ASP. กรุงเทพมหานคร: บริษัท ซัคเซส มีเดีย จำกัด, 2544.

2. นันทนี แขวงโสภา. อินไซท์ Access XP 2002. กรุงเทพมหานคร: บริษัท โปรวิชั่น จำกัด, 2544.

3. ยุทธนา ลีลาศวัฒนกุล. คู่มือการเขียนโปรแกรมและใช้งาน Visual C++ 6.0. นนทบุรี: สำนักพิมพ์อินโฟเพรส, 2544.

4. วราภรณ์ โกวิทวรางกูร. ระบบฐานข้อมูลและการออกแบบ. กรุงเทพมหานคร: สำนักพิมพ์ พิทักษ์อักษร, 2544.