SECURE ELECTRONIC ONLINE PAYMENT SYSTEM USING DUAL SIGNATURE

SU SU SHWE

M.C.Tech.

FEBRUARY, 2016

SECURE ELECTRONIC ONLINE PAYMENT SYSTEM USING DUAL SIGNATURE

By

SU SU SHWE

B.C.Tech. (Hons:)

Dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Technology (M.C.Tech.)

of the

Computer University, Mandalay

FEBRUARY, 2016

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to **Dr.Win Aye,** Reactor, Computer University (Mandalay), for her kind permission to prepare this thesis.

And then, I would like to describe my sincere thank to **Dr.Thi Thi Soe**, Associate Professor, Software Department, Computer University (Mandalay),for her helpful recommendations and suggestions.

It could not have been possible to complete this thesis without help from many resources. I would like to express my deep appreciation to my supervisor, **Dr. Zarni Sann**, Associated Professor, Computer University (Mandalay), for her close supervision, helpful advice, encouragement, numerous invaluable suggestions and comments in my thesis.

Especially, I would like to thank my teacher, **U Thaung Kyaw,** Associate Professor, Head of the English Department, Computer University, (Mandalay), for his wise advice and editing of my thesis from the English language point of view.

I also thank my parents and all the teachers who have enabled me to obtain this Master of Computer Technology degree.

Finally, I would like to express grateful thanks to all the friends and many colleagues for their contribution to the completion of this thesis.

ABSTRACT

Payment protocol backed by Mastercard and Visa and virtually all the major players in the e-commerce industry facilitate secure transmission of credit card payment information over the Internet and other networks. An open industry standard, Secure Electronic Transaction (SET) blocks out credit card details, denying merchants access to a consumer's credit card information and thus preventing merchants, hackers and electronic thieves from accessing this information. SET enables merchants to verify that buyers are who they claim to be, and it protects buyers by transferring credit information directly to the card issuer for authorization and billing.

The secure electronic transaction (SET) protocol is used to facilitate the secure transmission of consumer credit card information via electronic avenues, such as the Internet. When enabling these kinds of services, protection of an individual privacy, computer security. There are several technologies available for these growing markets saving companies for marketing storage and logistics cost. The SET is attended for data integrity, authentication and availability. In this system, MD5 used for data integrity, RSA and RC5 also used for authentication. This system is implemented with C# programming language.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The computer network security is one of the most important parts in the field of Computer Information Technology. Nowadays, more and more sensitive information is stored on computer and transmitted over the Internet, and the information is also needed to ensure security and safety. The need to protect the integrity and privacy information is essential to communication channel. When messages transmit from one secure place to another, the key is needed to encrypt or decrypt the message. The key secure is also important in cryptographic system to prevent the hacking. So, the security technique is increasingly important.

Cryptography is the art of creating and using cryptosystems. Cryptanalysis is the art of breaking cryptosystems: seeing through the disguise even when the users are not supposed to be able to. Cryptology is the study of both cryptography and cryptanalysis [1]. One objective of cryptography is to solve the confidentiality problem.

Cryptography concerns ways in which the meaning of messages may be concealed so that only certain people can understand them, and methods of ensuring that the content of messages remains unaltered. In general, cryptography can provide privacy, authenticate that a message has not changed in transit, implicitly authenticate the sender and potentially, cryptography can hide *secrets,* either from others, or during communication.

From technological perspective there are issues like network solution, security and data communication standards graphical user interfaces, multimedia technology, data security related like Internet payments and banking. Development of a mobile phone, PDA's and

roaming technology enables usage of the ecommerce services independent of location.

E-commerce Security is a part of the Information Security framework and is specifically applied to the components that affect e-commerce that include Computer Security, Data security and other wider realms of the Information Security framework[5]. E-commerce security has its own particular nuances and is one of the highest visible security components that affect the end user through their daily payment interaction with business.

In this system, symmetric keys cryptosystem or secret key cryptosystem, one key is used to encrypt and decrypt the data. Strength of symmetric key cryptosystem depends on the size of key used. The MD5 message-digest algorithm takes an input message of arbitrary length and produces a 128-bit hash value of the message. The security of session key distribution based on RSA by using public key encryption scheme to apply public key cipher and pseudorandom number generator to achieve the secure distribution of the session key[9].

Moreover, in order to avoid eavesdropper's interception, public key cryptosystem will be used during the transmission of data while implementing key exchange protocol. This system will provide not only ensuring the secure transmission of section key, but also the certainty of the authentication. So, this system can efficiently support the modern popular peer to peer (P2P) applications to implement distributing explicit session keys with guaranteed or verifiable authenticity.

1.1 Objective of Thesis

The objectives are:

- To meet confidentiality of Information
- To know the integrity of data
- To ensure the use of the good security
- To ensure the integrity of payment instructions for goods and services, order data;
- To authenticate both the cardholder and the merchant

1.2 Motivation of the System

Today, privacy and security are a major concern for electronic technologies. E-commerce shares security concerns with other technologies in the field. Privacy concerns have been found, revealing a lack of trust in a variety of contexts, including commerce, electronic health records, e-recruitment technology and social networking, and this has directly influenced users. Security is one of the principal and continuing concerns that restrict customers and organizations engaging with ecommerce. Web e-commerce applications that handle payments (online banking, electronic transactions or using debit cards, credit cards, PayPal or other tokens) have more compliance issues, are at increased risk from being targeted than other websites and there are greater consequences if there is data loss or alteration. Online shopping having certain steps to buy a product with safe and secure.

1.3 Organization of Thesis

This book has (5) chapters. Chapter1 describes the introduction of the system. Chapter 2 states the theory background such as cryptography, security service, security requirements and attack. Chapter 3 explains the details of symmetric algorithm (RC5), Asymmetric algorithm (RSA) associated with man-in-middle attack. Message digest function MD5 and application area. Chapter 4 deals with the design and detailed implementation of the system. Chapter 5 concludes the thesis with suggestion for limitation and further extension.

CHAPTER 2
THEORETICAL BACKGROUND

2.1 Cryptography

The word cryptography comes from Greek word "kryptos" which means "hidden" while "graphia" stands for "writing". Cryptography is the science of protecting data, which provides methods of converting data into unreadable form, so that the data cannot be accessed for unauthorized use.

Cryptography is one of the technological means to provide security to data being transmitted on information and communication systems. It is especially useful in the cases of financial and personal data, irrespective of the fact that the data is being transmitted over a medium or is stored on a storage device. Because of the development of electronic commerce, cryptographic techniques are extremely critical to the development and use of defense information systems and communications networks [1].

The meaning of Cryptography has now become an industry standard for providing information security, trust, controlling access to resources, and electronic transactions. Its use is no longer limited to just securing sensitive military information. In fact, cryptography is now recognized as one of the major components of the security policy of an organization.

Cryptographic Process: the original message is called a plaintext. The disguised message is called a cipher text. Encryption means any procedure to convert plaintext into cipher text. Decryption means any procedure to convert cipher text into plaintext. Generally, all cryptographic processes have four basic parts:

1. Plaintext -This is the original message or data that is fed into the algorithm as input.

2. Cipher text -Cipher text is the encrypted plaintext that is transmitted to the receiver.

3. Cryptographic Algorithm -Mathematical formula is used to scramble the plaintext to yield cipher text. Converting plaintext to ciphertext using the cryptographic algorithm is called encryption and converting ciphertext back to plaintext using the same cryptographic algorithm is called decryption.

4. Key -A mathematical value that determines a plaintext message is encrypted and a cipher text message is decrypted. The key is the only way to decipher the scrambled information [6].

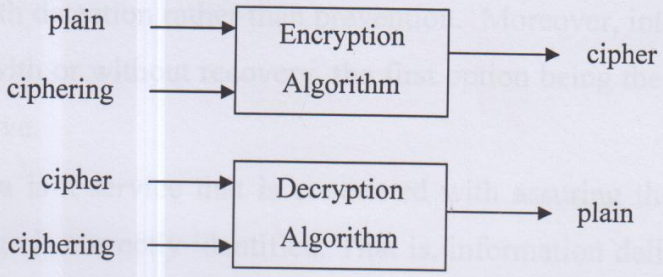

Figure 2.1 Encryption/Decryption Block Diagram

2.2 *Security Service*

Cryptography involves the study of mathematical techniques that allow the practitioner to achieve or provide the following objectives or services [4].

- **Confidentiality** is a service used to keep the content of information accessible to only those authorized to have it. This service includes both protections of all user data transmitted between two points over a period of time as well as protection of traffic flow from analysis.
- **Integrity** is a service that requires that computer system assets and transmitted information be capable of modification only by authorized users. Modification includes writing, changing the status, deleting, creating, and the delaying or replaying of transmitted messages. It is important to point out that integrity relates to active attacks and therefore, it is concerned with detection rather than prevention. Moreover, integrity can be provided with or without recovery, the first option being the more attractive alternative.
- **Authentication** is a service that is concerned with assuring that the origin of a message is correctly identified. That is, information delivered over a channel should be authenticated as to the origin, date of origin, data content, time sent, etc. For these reasons, this service is subdivided into two major classes: entity authentication and data origin authentication. Notice that the second class of authentication implicitly provides data integrity.
- **Non-repudiation** is a service which prevents both the sender and the receiver of a transmission from denying previous commitments or actions. These security services are provided by using cryptographic algorithms.

2.3 Types of Cryptographic Algorithms

There are several ways of classifying cryptographic algorithms based on the number of keys that are employed for encryption and decryption, and further defined by their application and use [3,5]. The three types of algorithms are:

1. Secret Key Cryptography (SKC): It uses a single key for both encryption and decryption
2. Public Key Cryptography (PKC): It uses one key for encryption and another for decryption
3. Hash Functions: It is used for a mathematical transformation to irreversibly "encrypt" information

2.3.1 Secret Key Cryptography

Secret key cryptography involves the use of one key for both encryption and decryption. The secret key or symmetric key technique further classifies itself into Block ciphers and Stream ciphers. Symmetric key cryptography is the method where the plaintext is converted to cipher text based on the unique key and the function used. The main problem with symmetric key algorithms is that the sender and the receiver have to agree on a common key. A secure channel is also required between the sender and the receiver to exchange the secret key [9].

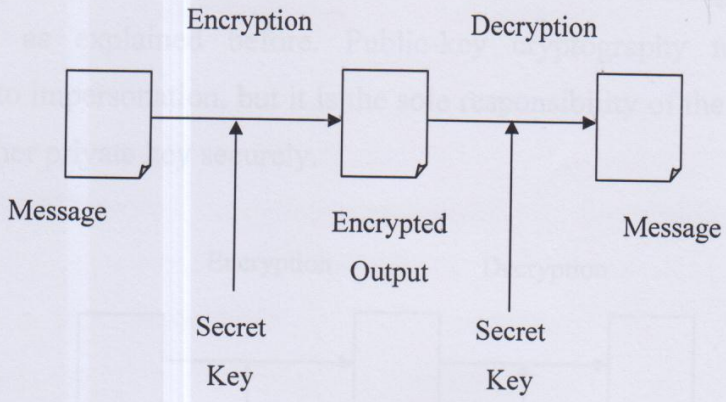

Figure 2.2 Scenario for Symmetric Key Cryptography

2.3.2 Public Key Cryptography

Public key cryptography involves the use of two keys, one for encryption and the other for decryption. Public-key cryptography uses two kinds of functions and two different keys. The keys are terms as the private key and public key. The public key is the one which is kept 'visible', (i.e.), commonly transmitted over networks, etc. and the other one which is kept 'secret' the private key, which is never revealed to anybody.

In public key cryptography, the data that is encrypted with the public key can only be decrypted with the corresponding private key. Conversely, data encrypted with the private key can only be decrypted with the corresponding public key. Due to this asymmetry, public key cryptography is known as asymmetric cryptography.

Public-key algorithms are slower compared to the speed of symmetric key (secret-key) algorithms. Also, public-key algorithms are prone to attack than the secret-key algorithms. Another main of public-key algorithms is to produce digital signatures, which plays the lead role in identifying the origin of the message[9]. The digital signature thus

provides the way to authenticate which is known as message non-repudiation as explained before. Public-key cryptography may be vulnerable to impersonation, but it is the sole responsibility of the user to protect his/her private key securely.

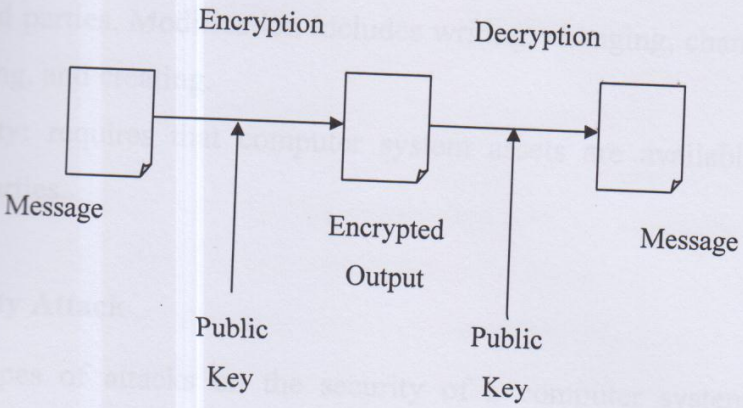

Figure 2.3 Scenario for Asymmetric Key Cryptography

2.4 Security Requirements and Attacks

Network security field consists of three security requirements and four types of attack. Four types of attack are divided into two main groups. These groups are passive attack and active attack.

2.4.1 Security Requirements

The requirements of information security within an organization have undergone two major changes in the last several decades. With the introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident; this is especially the case for a shared system, such as a time-sharing system, and the need is even more acute for systems that can be accessed over a public telephone or data network. The generic name for the collection of

tools designed to protect data and to thwart hackers is computer security. Computer and network security address three requirements[3].

(a)**Secrecy:** requires that the information in a computer system only be accessible for reading by authorized parties.

(b)Integrity: requires that computer system assets can be modified only by authorized parties. Modification includes writing, changing, changing status, deleting, and creating.

(c)Availability: requires that computer system assets are available to authorized parties.

2.4.2 Security Attack

The types of attacks on the security of a computer system or network are best characterized by viewing the function of the computer system as providing information. In general, there is a flow of information from a source, such as a file or a region of main memory, to a destination, such as another file or a user. Four general categories of attack are the following as shown in figure (2.4).

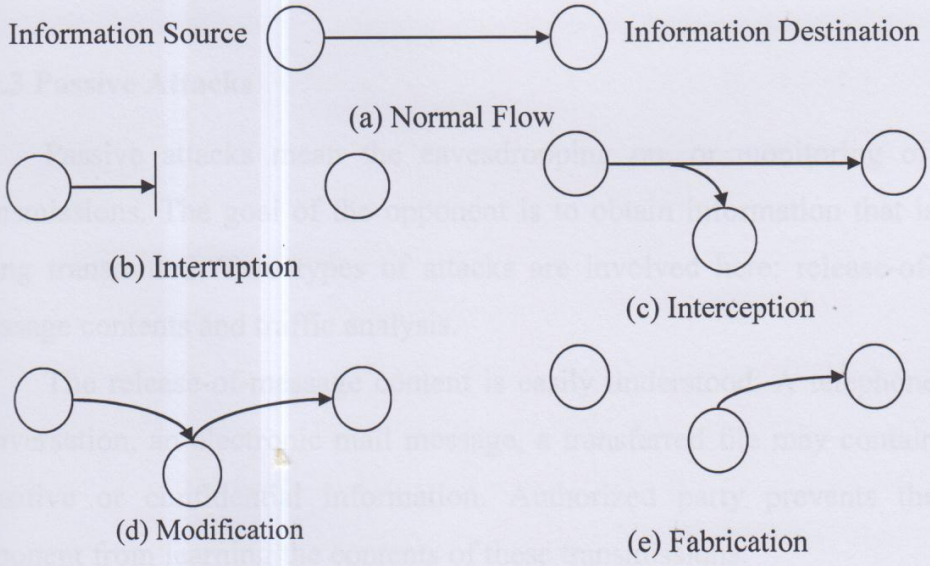

Figure 2.4 General Categories of Attack

- **Interruption:** An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability. Examples include destruction of a piece of hardware, such as a hard disk, the cutting of a communication line, or the disabling of the file management system.
- **Interception:** An unauthorized party gains access to an asset. This is an attack on confidentiality. The unauthorized party could be a person, a program, or a computer. Examples include wiretapping to capture data in a network, and the illicit copying of files or programs.
- **Modification:** An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity. Examples include changing values in a data file, altering a program so that it performs differently, and modifying the content of messages being transmitted in a network.
- **Fabrication:** An unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity. Examples include the insertion of spurious messages in a network or the addition of records to a file.

2.4.3 Passive Attacks

Passive attacks mean the eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of attacks are involved here: release-of-message contents and traffic analysis.

The release-of-message content is easily understood. A telephone conversation, an electronic mail message, a transferred file may contain sensitive or confidential information. Authorized party prevents the opponent from learning the contents of these transmissions.

- **Interruption:** An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability. Examples include destruction of a piece of hardware, such as a hard disk, the cutting of a communication line, or the disabling of the file management system.
- **Interception:** An unauthorized party gains access to an asset. This is an attack on confidentiality. The unauthorized party could be a person, a program, or a computer. Examples include wiretapping to capture data in a network, and the illicit copying of files or programs.
- **Modification:** An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity. Examples include changing values in a data file, altering a program so that it performs differently, and modifying the content of messages being transmitted in a network.
- **Fabrication:** An unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity. Examples include the insertion of spurious messages in a network or the addition of records to a file.

2.4.3 Passive Attacks

Passive attacks mean the eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of attacks are involved here: release-of-message contents and traffic analysis.

The release-of-message content is easily understood. A telephone conversation, an electronic mail message, a transferred file may contain sensitive or confidential information. Authorized party prevents the opponent from learning the contents of these transmissions.

Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect. The denial of service prevents or inhibits the normal use or management of communications facilities. This attack may have a specific target. An entity may suppress all messages directed to a particular destination. Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks absolutely, as to do so would require physical protection of all communications facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them. A useful categorization of these attacks is in terms of passive attacks and active attacks (Figure 2.5).

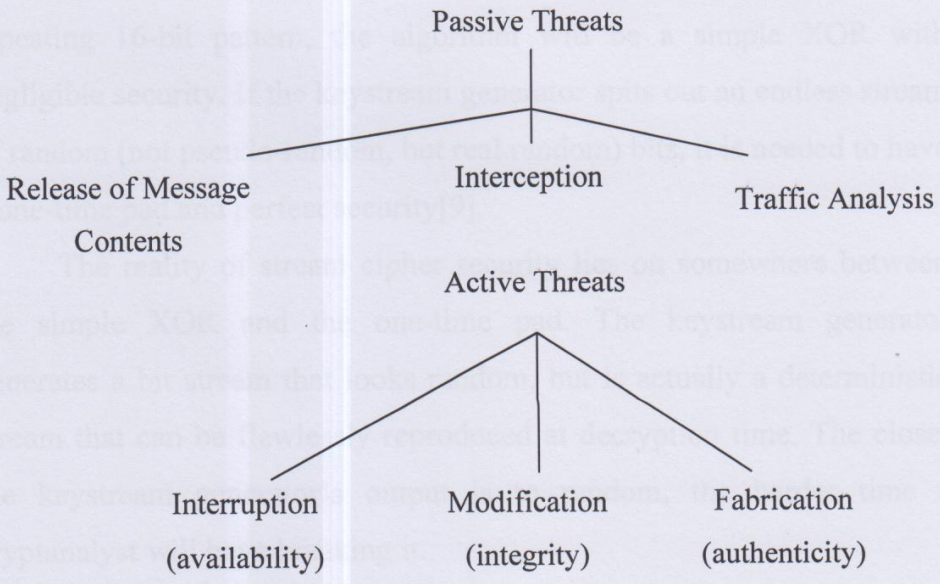

Figure 2.5: Active and Passive Network Security Threats

2.5 Stream Ciphers

Stream ciphers convert plaintext to ciphertext 1 bit at a time. A keystream generator (sometimes called a running-key generator) outputs a stream of bits: $k_1, k_2, k_3, \ldots, k_i$. This keystream (sometimes called a running key) is XORed with a stream of plaintext bits, $p_1, p_2, p_3, \ldots, p_i$, to produce the stream of ciphertext bits.

$c_i = p_i \oplus k_i$

At the decryption end, the ciphertext bits are XORed with an identical keystream to recover the plaintext bits.

$p_i = c_i \oplus k_i$

Since

$p_i \oplus k_i \oplus k_i = p_i$

The system's security depends entirely on the insides of the keystream generator. If the keystream generator outputs an endless stream of zeros, the ciphertext will equal the plaintext and the whole operation will be worthless. If the keystream generator spits out a repeating 16-bit pattern, the algorithm will be a simple XOR with negligible security. If the keystream generator spits out an endless stream of random (not pseudo-random, but real random) bits, it is needed to have a one-time pad and perfect security[9].

The reality of stream cipher security lies on somewhere between the simple XOR and the one-time pad. The keystream generator generates a bit stream that looks random, but is actually a deterministic stream that can be flawlessly reproduced at decryption time. The closer the keystream generator's output is to random, the harder time a cryptanalyst will have breaking it.

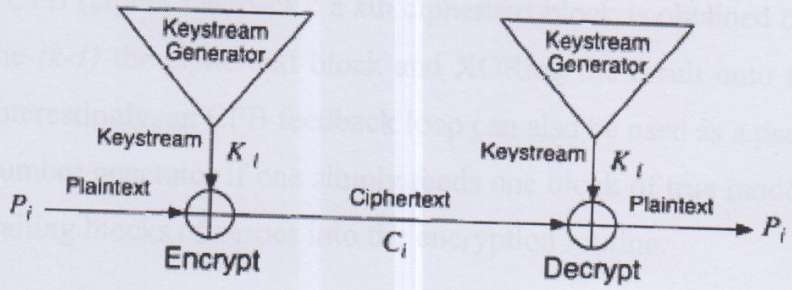

Figure 2.6 Stream Cipher

2.6 Block Ciphers

The block method divides a large data set into blocks (based on predefined size or the key size), encrypts each block separately and finally combines blocks to produce encrypted data. Block algorithm operate on plaintext in groups of bytes, call blocks[9].

Many commonly used ciphers are block ciphers. Block ciphers transform a fixed-size block of data (usually 64 bits) into another fixed-size block (possibly *64* bits wide again) using a function selected by the key. If key, input block and output block have all n bits, a block cipher basically defines a one-to-one mapping from n-bit integers to permutations of n-bit integers. If the same block is encrypted twice with the same key, the resulting ciphertext blocks are also the same (this *mode* of encryption is called electronic code book, or ECB). This information could be useful for an attacker. To cause identical plaintext blocks being encrypted to different ciphertext blocks, two standard modes are commonly used:

• CBC (cipher block chaining): a ciphertext block is obtained by first XORing the plaintext block with the previous ciphertext block, and encrypting the resulting value. This way leading blocks influence all trailing blocks, which increases the number of plaintext bits, one ciphertext bit depends on, but also leads to synchronization problems if one block is lost.

• CFB (cipher feedback): a *k*th ciphertext block is obtained by encrypting the *(k-1)* the ciphertext block and XORing the result onto the plaintext. Interestingly, an CFB feedback loop can also be used as a pseudo-random number generator if one simply feeds one block of true random data with trailing blocks of zeroes into the encryption routine.

2.7 Symmetric Cryptosystem

Symmetric (private key) cryptosystem is one in which both Alice and Bob share a common secret key **K** and both encryption and decryption depend on this key. Formally, it can define such a cryptosystem as a quintuple $<\mathbf{M},\mathbf{K},\mathbf{C},\mathbf{e}(\cdot,\cdot),\mathbf{d}(\cdot,\cdot)>$, where **M** is the message space, the set of all possible messages, **K** is the key space, the set of all possible keys, and **C** is the cryptogram space, the set of all possible cryptograms. Then $\mathbf{e}:\mathbf{M}\times\mathbf{K}\rightarrow\mathbf{C}$, is the encryption function and $\mathbf{d}:\mathbf{C}\times\mathbf{K}\rightarrow\mathbf{M}$, is the decryption function. To ensure that cryptograms can be decrypted they must satisfy the fundamental identity $\mathbf{d}(\mathbf{e}(\mathbf{m},\mathbf{k}),\mathbf{k})=\mathbf{m}$, for all $\mathbf{m}\in\mathbf{M}$ and $\mathbf{k}\in\mathbf{K}$. This identity implies that there must be at least as many cryptograms as messages [1,9]. There are various symmetric key algorithms such as RC2, RC4, RC5 and RC6 information. RC2 is vulnerable to a related-key attack using 234 chosen plaintexts (Kelsey et al., 1997). RC4 has become part of some commonly used encryption protocols and standards, including WEP and WPA for wireless cards and TLS. RC6 proper has a block size of 128 bits and supports key sizes of 128, 192, and 256 bits, but, like RC5, it may be parameterized to support a wide variety of word-lengths, key sizes, and number of rounds. RC6 is very similar to RC5 in structure, using data-dependent rotations, modular addition, and XOR operation.

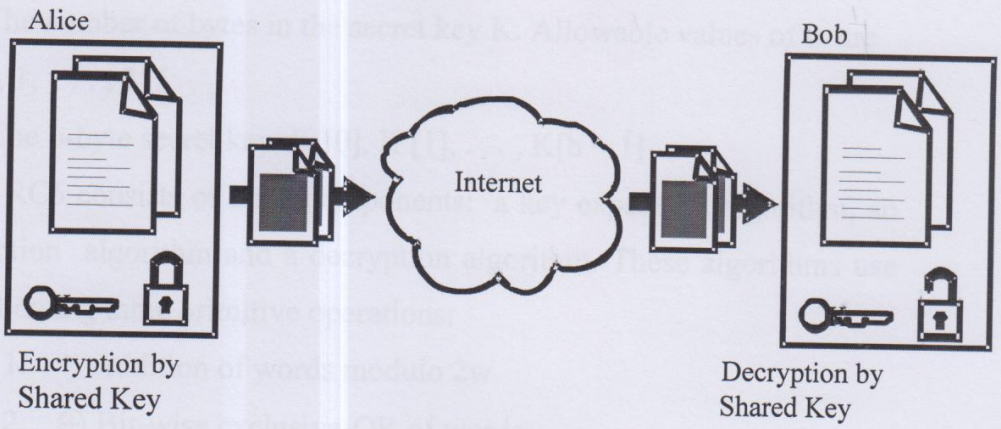

Figure 2.7 Simplified Model of Symmetric Cryptosystem

2.7.1 Description of RC5

RC5 is a symmetric block cipher designed to be suitable for both software and hardware implementation. It is a parameterised algorithm, with a variable block size, a variable number of rounds and a variable-length key. This provides the opportunity for great flexibility in both performance characteristics and the level of security[9].

A particular RC5 algorithm is designated as RC5-w/r/b. The number of bits in a word, w, is a parameter of RC5. Different choices of this parameter result in different RC5 algorithms. RC5 is iterative in structure, with a variable number of rounds. The number of rounds, r , is a second parameter of RC5.

RC5 uses a variable-length secret key. The key length b (in bytes) is a third parameter of RC5. These parameters are summarised as follows:

w: The word size, in bits. The standard value is 32bits; allowable values are 16, 32 and 64. RC5 encrypts two-word blocks so that the plaintext and ciphertext blocks are each 2w bits long.

r : The number of rounds. Allowable values of r are 0, 1, . . . , 255. Also, the expanded key table S contains t = 2 (r + 1) words.

b: The number of bytes in the secret key K. Allowable values of b are 0, 1, . . . , 255.

K : The b-byte secret key; K [0], K [1], . . . , K[b − 1]

RC5 consists of three components: a key expansion algorithm, an encryption algorithm and a decryption algorithm. These algorithms use the following three primitive operations:

1. + Addition of words modulo 2w
2. ⊕ Bit-wise exclusive-OR of words
3. <<< Rotation symbol: the rotation of x to the left by y bits is denoted by x <<< y.

One design feature of RC5 is its simplicity, which makes RC5 easy to implement. Another feature of RC5 is its heavy use of data-dependent rotations in encryption; this feature is very useful in preventing both differential or linear cryptanalysis.

2.7.2 Key Expansion

The key-expansion algorithm expands the user's key K to fill the expanded key table S, so that S resembles an array of $t = 2(r + 1)$ random binary words determined by K. It uses two word-size magic constants Pw and Qw defined for arbitrary w as shown below:

$Pw = Odd\ ((e - 2)2w)$

$Qw = Odd\ ((\varphi - 1)2w)$

where

$e = 2.71828\ldots$ (base of natural logarithms)

$\varphi = (1 + \sqrt{5})/2 = 1.61803\ldots$ (golden ratio)

Odd(x) is the odd integer nearest to x.

2.7.3 Encryption

The input block to RC5 consists of two w-bit words given in two registers, A and B. The output is also placed in the registers A and B[5,9]. Recall that RC5 uses an expanded key able, S[0, 1, . . . , t − 1], consisting of t = 2(r + 1) words.

The key-expansion algorithm initializes S from the user's given secret key parameter K. However, the S table in RC5 encryption is not like an S-box used by DES. The encryption algorithm is given in the pseudocode as shown below:

A = A + S[0];

B = B + S[1];

for i = 1 to r do

A = ((A ⊕ B) <<< B) + S[2i];

B = ((B ⊕ A) <<< A) + S[2i + 1];

The output is in the registers A and B.

2.7.4 Decryption

RC5 decryption is given in the pseudocode as shown below.

For i = r down to 1 do

B = ((B − S[2i + 1]) >>> A) ⊕ A

A = ((A − S[2i]) >>> B) ⊕ B

B = B − S[1]

A = A − S[0]

CHAPTER 3
SECURE ELECTRONIC ONLINE PAYMENT SYSTEM

Security is very important in online shopping sites. Nowadays, a huge amount is being purchased on the internet because it is easier and more convenient. Almost anything can be bought such as music, toys clothing, cars, food and even porn. Even though some of these purchases are illegal we will be focusing on all the items you can buy legally on the internet. Some of the popular websites are eBay, iTunes, Amazon, HMV, Mercantila, dell, Best Buy and much more.

3.1 Asymmetric Cryptosystem

The concept of asymmetric (public key) cryptography was introduced in 1976 by Whitfield Diffie and Martin Hellman in order to solve the key management problem. In their concept, each person gets a pair of keys, one called the *public key* (KU) and the other called the *private key* (KR). Each person's public key is published while the private key is kept secret. The need for the sender and receiver to share secret information is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. No longer is it necessary to trust some communications channel to be secure against eavesdropping or betrayal.

The only requirement is that public keys are associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory). Anyone can send a confidential message by just using public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient. Furthermore,

public-key cryptography can be used not only for privacy (*encryption*), but also for authentication (*digital signatures*) [9, 10].

When Alice (A) wishes to send a secret message to Bob (B), she looks up Bob's public key KU_B in a public key directory, uses it to encrypt the message M, to receive ciphertext $C = E_{KU_B}(M)$ and sends it off. Bob then uses his private key KR_B to decrypt to receive the message $M = E_{KR_B}(C)$. No one listening in can decrypt the message without knowing private key KR. Anyone can send an encrypted message to Bob but only Bob can read it. Clearly, one requirement is that no one can figure out the private key from the corresponding public key.

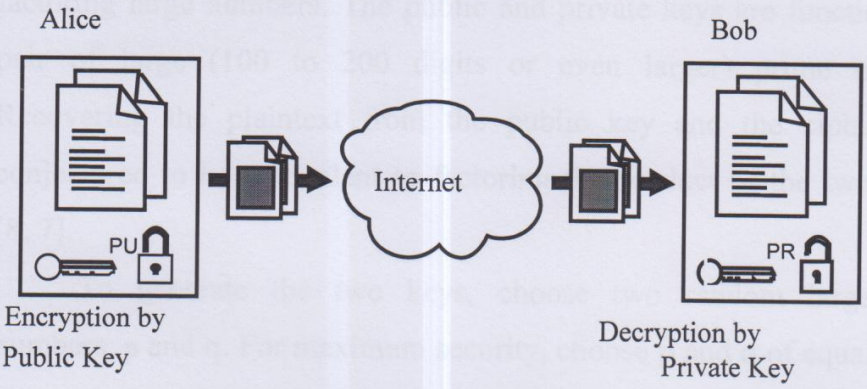

Figure 3.1 Simplified Model of Asymmetric Cryptosystem

According to the observation of public key cryptography, most of them are unsuitable for encryption of large amount of message content due to their low speed. However, public key cryptography fulfills an extremely important role in the overall design and operation of secure computer networks because it leads to superior protocols for managing and distributing secret session keys that can subsequently be used for the symmetric encryption of large amount of message.

Public-key cryptography can be used with secret-key cryptography to get the best of both worlds. For encryption, the best solution is to

combine public- and secret-key systems in order to get both the security advantages of public-key systems and the speed advantages of secret-key systems. The public-key system can be used to encrypt a secret key which is used to encrypt the bulk of a file or message. Such a protocol is called a *digital envelope* [1].

3.2 RSA Public Key Cryptosystem

RSA is a public-key cryptosystem that offers both encryption and digital signatures (authentication). Ron Rivest, Adi Shamir, and Leonard Adleman developed RSA in 1977; RSA stands for the first letter in each of its inventors' last names. RSA gets its security from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large (100 to 200 digits or even larger) prime numbers. Recovering the plaintext from the public key and the ciphertext is conjectured to be equivalent to factoring the product of the two primes [8, 7].

To generate the two keys, choose two random large prime numbers, **p** and **q**. For maximum security, choose **p** and **q** of equal length. Compute the product: $\mathbf{n} = \mathbf{pq}$. Then randomly choose the encryption key, **e**, such that **e** and $(\mathbf{p} - 1)(\mathbf{q} - 1)$ are *relatively prime*. Finally, use the extended Euclidean algorithm to compute the decryption key, **d**, such that; $ed = 1 \bmod (p-1)(q-1)$. In other words, $\mathbf{d} = \mathbf{e}^{-1} \bmod ((\mathbf{p}-1)(\mathbf{q}-1))$. Note that **d** and **n** are also *relatively prime*. The numbers **e** and **n** are the public keys; the number **d** is the private key. The two prime's **p** and **q** are no longer needed. They should be discarded, but never revealed.

To encrypt a message **M**, first divide it into numerical blocks $\mathbf{m}_i$ smaller than **n**. The encrypted message, **C**, will be made up of similarly sized message blocks, $\mathbf{c}_i$ of about the same length. The encryption

formula is simply $\mathbf{c_i} = \mathbf{m_i^e}$ *mod* $\mathbf{n}$. To decrypt a message, take each encrypted block $\mathbf{c_i}$ and compute $\mathbf{m_i} = \mathbf{c_i^d}$ *mod* $\mathbf{n}$. The following steps are the example for RSA key generation

Select primes: **p** = 17 and **q** = 11

Compute **n** = **p.q** = 17 x 11 = 187

Compute **ø(n)** = (**p**–1) (**q**-1) = 16 x 10 = 160

Select **e**: **gcd(e**, 160) = 1; choose **e** = 7

Determine **d**: **de** = 1 *mod* 160 and **d** < 160 Value is **d** = 23.

(Since 23 x 7 = 161= 10 x 160 + 1)

Publish public key **KU** = {7, 187}

Keep secret private key **KR** = {23, 187}

and here is the example for RSA encryption and decryption steps;

given message **M** = 88 (note: 88<187)

encryption: **C** = 88^7 mod 187 = 11

decryption: **M** = 11^{23} mod 187 = 88.

3.3 Man-in-the-Middle Attack on RSA Cryptosystem

The opponent not only can listen to messages between Alice and Bob but can also modify messages, delete messages, and generate totally new ones. The opponent can imitate Bob when talking to Alice and imitate Alice when talking to Bob. Here is how the attack works:

(1) Alice sends Bob his public key. The opponent intercepts this key and sends Bob his own public key.

(2) Bob sends Alice his public key. The opponent intercepts this key and sends Alice his own public key.

(3) When Alice sends a message to Bob, encrypted in Bob's public key, the opponent intercepts it. Since the message is really encrypted with his own public key, he decrypts it with his private key, re-encrypts it with Bob's public key, and sends it on to Bob.

(4) When Bob sends a message to Alice, encrypted in Alice's public key, the opponent intercepts it. Since the message is really encrypted with his own public key, he decrypts it with his private key, re-encrypts it with Alice's public key, and sends it on to Alice.

Even if Alice's and Bob's public keys are stored on a database, this attack will work. The opponent can intercept Alice's database inquiry and substitute his own public key for Bob's. He can do the same to Bob and substitute his own public key for Alice's. Man-in-the-middle attack works because Alice and Bob have no way to verify that they are talking to each other. Assuming the opponent doesn't cause any noticeable network delays, the two of them have no idea that someone sitting between them is reading all of their supposedly secret communications.

3.4 Security of RSA Cryptosystem

This subsection discusses various security issues related to RSA encryption. Various attacks which have been studied in the literature are presented, as well as appropriate measures to counteract these threats.

3.4.1 Relation to Factoring

The task faced by a passive adversary is that of recovering plaintext **m** from the corresponding ciphertext **c**, given the public information (**n**, **e**) of the intended receiver A. This is called the RSA problem (RSAP). There is no efficient algorithm known for this problem. One possible approach which an adversary could employ to solving the RSA problem is to first factor **n**, and then compute **φ** and **d**. Once **d** is obtained, the adversary can decrypt any ciphertext intended for A.

On the other hand, if an adversary could somehow compute **d**, then it could subsequently factor **n** efficiently as follows. First, note that since $\mathbf{ed} \equiv 1 \pmod{\varphi}$, there is an integer **k** such that $\mathbf{ed} - 1 = \mathbf{k}\varphi$. Hence, $\mathbf{a}^{\mathbf{ed}-1} \equiv 1 \pmod{\mathbf{n}}$ for all $\mathbf{a} \in \mathbf{Z}^*_{\mathbf{n}}$. Let $\mathbf{ed} - 1 = 2^{\mathbf{s}}\,\mathbf{t}$, where **t** is an odd integer. Then, it can be shown that $a^{2^{s-1}t} \neq \pm 1 (\mathbf{mod} n)$ for at least half of all $\mathbf{a} \in \mathbf{Z}^*_{\mathbf{n}}$; if a and i are such integers then $\gcd\left(a^{2^{s-1}t} - 1, n\right)$ is a non-trivial factor of **n**. Thus, the adversary simply needs to repeatedly select random $\mathbf{a} \in \mathbf{Z}^*_{\mathbf{n}}$ and compute $\gcd\left(a^{2^{s-1}t} - 1, n\right)$; the expected number of trials before a non-trivial factor of n is obtained is 2.

Fact: The problem of computing the RSA decryption exponent d from the public key (**n**, **e**), and the problem of factoring **n**, are computationally equivalent.

When generating RSA keys, it is imperative that the primes **p** and **q** be selected in such a way that factoring $\mathbf{n} = \mathbf{pq}$ is computationally infeasible.

3.5 MD5 Message-Digest Algorithm

The MD5 message-digest algorithm was developed by Ronald Rivest at MIT in 1992. This algorithm takes a input message of arbitrary length and produces a 128-bit hash value of the message. The input message is processed in 512-bit blocks which can be divided into 16 32-bit subblocks. The message digest is a set of four 32-bit blocks, which concatenate to form a single 128-bit hash code. MD5 (1992) is an improved version of MD4, but is slightly slower than MD4 (1990).

The following steps are carried out to compute the message digest of the input message.

3.5.1 Append Padding Bits

The message is padded so that its length (in bits) is congruent to 448 modulo 512. That is, the padded message is just 64 bits short of being a multiple of 512. This padding is formed by appending a single ' 1' bit to the end of the message, and then ' 0' bits are appended as needed such that the length (in bits) of the padded message becomes congruent to 448 (= 512 − 64), modulo 512.

3.5.2 Append Length

A 64-bit representation of the original message length is appended to the result of the previous step. If the original length is greater than 264, then only the low-order 64 bits of the length are used for appending two 32-bit words.

The length of the resulting message is an exact multiple of 512 bits. Equivalently, this message has a length that is an exact multiple of 16

(32-bit) words. Let M[0 . . . N − 1] denote the word of the resulting message, with N an integer multiple of 16.

3.5.3 Initialise MD Buffer

A four-word buffer represents four 32-bit registers (A, B, C and D). This 128-bit buffer is used to compute the message digest.

3.5.4 Define Four Auxiliary Functions (F, G, H, I)

F, G, H and I are four basic MD5 functions. Each of these four nonlinear functions takes three 32-bit words as input and produces one 32-bit word as output. They are, one for each round, expressed as:

$F(X, Y, Z) = (X\ Y) + (X\ Z)$

$G(X, Y, Z) = (X\ Z) + (Y\ Z)$

$H(X, Y, Z) = X \oplus Y \oplus Z$

$I(X, Y, Z) = Y \oplus (X + Z)$

where X Y denotes the bitwise AND of X and Y; X + Y denotes the bitwise OR of X and Y; X denotes the bitwise complement of X, i.e. NOT(X); and X ⊕ Y denotes the bitwise XOR of X and Y.

These four nonlinear functions are designed in such a way that if the bits of X, Y and Z are independent and unbiased, then at each bit position the function F acts as a conditional: if X then Y else Z. The functions G, H and I are similar to the function F in that they act in 'bitwise parallel' to their product from the bits of X, Y and Z. Notice that the function H is the bitwise XOR function of its inputs.

3.5.5 FF, GG, HH and II Transformations for Rounds 1, 2, 3 and 4

If M[k], $0 \leq k \leq 15$, denotes the kth sub-block of the message, and <<< s represents a left shift s bits, the four operations are defined as follows:

FF(a, b, c, d, M[k], s, i) : a = b + ((a + F(b, c, d) + M[k] + T[i] <<< s)

GG(a, b, c, d, M[k], s, i) : a = b + ((a + G(b, c, d) + M[k] + T[i] <<< s)

HH(a, b, c, d, M[k], s, i) : a = b + ((a + H(b, c, d) + M[k] + T[i] <<< s)

II(a, b, c, d, M[k], s, i) : a = b + ((a + I(b, c, d) + M[k] + T[i] <<< s)

3.5.6 Computation of Four Rounds (64 Steps)

Each round consists of 16 operations. Each operation perform a nonlinear function on three of A, B, C and D.

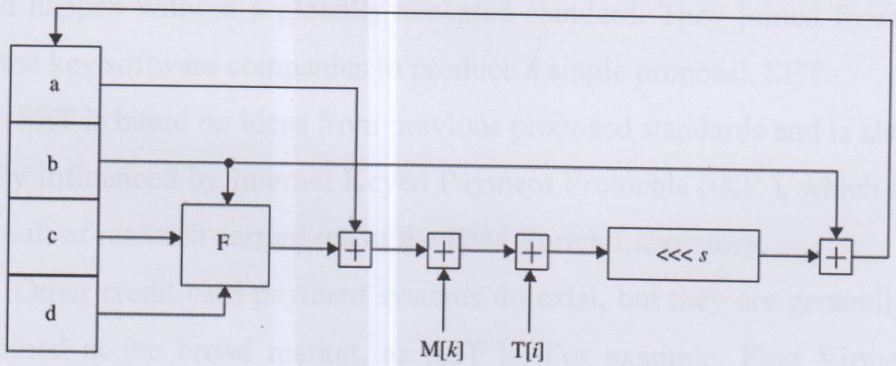

Figure 3.2 Basic MD5 Operation

3.6 Web Payment Systems

SET is the electronic payment system designed for the World Wide Web. It is, however, emerging as the only significant standard for credit card transactions.

Banks and financial institutions have had networks for electronic payment processing for many years. These networks connect highly secure, trusted computer systems, using dedicated links and powerful cryptographic hardware. A number of international standards exist to define the protocol for messages exchanged over the network.

The challenge for Internet credit card processing lies in producing a scheme that can provide adequate protection at a reasonable cost without compromising trust in any of the existing systems.

During 1995, various financial organizations and technology companies formed a number of alliances aimed at producing standards for credit card payment. This was a confusing time, with a number of competing standards and consortia. The technical community would probably still be arguing the merits of one solution or another, but the two largest credit card companies, Visa and Mastercard, realized that nothing would happen without a globally accepted standard. They joined forces with the key software companies to produce a single proposal, SET.

SET is based on ideas from previous proposed standards and is also heavily influenced by Internet Keyed Payment Protocols (iKP), which is the result of research carried out at the IBM Zurich Laboratory.

Other credit card payment systems do exist, but they are generally not aimed at the broad market, as SET is. For example, First Virtual Internet Payments System (FVIPS), operated by First Virtual Holdings Inc. is a scheme by which the prospective buyer registers credit card details with First Virtual and receives a personal identification number (PIN). The buyer can then use the PIN in place of a card number at any merchant that has an account with First Virtual.

Payment details must be confirmed by e-mail before any purchase is completed. Although this scheme has been successful, it is limited due to the requirement for both buyer and seller to be affiliated with the same

service. SET more closely follows the model of normal credit card payments, in which the only relationship between the organization that issues the card and the one that processes the purchase is that they subscribe to the same clearing network.

3.7 Secure Electronic Transaction

SET exists to allow businesses to receive payment for goods and services in a safe, reliable and consistent manner. The major requirements are

- Authentication

Authentication is ensured by the use of digital signatures.

- Confidentiality

Confidentiality is ensured by the use of message encryption.

- Integrity

Integrity is ensured by the use of digital signatures.

3.7.1 SET Format General

SET uses combinations of cryptographic techniques to address the business specification of a secure transaction. SET defines a series of message exchanges for the payment processes and for certificate management.

- Symmetric key (or bulk) encryption
- Public key encryption
- Secure hash (or digest) functions
- Dual Signature

Within the SET protocols there is a situation where the cardholder communicates with both the merchant and payment gateway in a single message. The message contains an order section, with details of the

products/services to be purchased, plus a payment section. The payment instruction will be used by the acquirer and the order by the merchant, but the messages are both sent together. This means that the message packaging must:

1. Prevent the merchant from seeing the payment instruction
2. Prevent the acquirer from seeing the order instruction
3. Link the two parts of the message, so that they can only be used as a pair

In this case, SET uses a technique called dual signature. When the order and payment instruction is sent by the cardholder, the merchant will be able to see only the order instruction, and the acquirer only the payment instruction. The merchant will not see the cardholders account information. In a SET transaction, the transfer of money and offer are linked allowing the money to be transferred to the merchant only if the cardholder accepts the offer.

In Secure Electronic Transaction (SET), various encryption algorithms are used such as RC5 (Data Encryption Standard) and RSA algorithm. Data Encryption Standard (RC5) is used to encrypt online transactions. This encryption technique was not much secure and can be easily cracked using modern software embedded hardware.

Secure Electronic Transaction (SET) permitted communicating parties to identify and authenticate each other in hidden manner and exchange sensitive information securely. The main advantage of SET is that all communication will be taken place in hidden manner.

In SET, the merchant cannot access the customer sensitive credit card information. Such strong protection is provided for the benefits of customers as well as credit/debit card companies to avoid any type of financial frauds.

In SET, the merchant cannot access the customer sensitive credit card information. Such strong protection is provided for the benefits of customers as well as credit/debit card companies to avoid any type of financial frauds.

3.7.2 SET Roles

SET defines a number of different roles, some that are intuitively obvious, others that are more confusing:

- **Cardholder**

  This is you or me, the average person in the street with a wallet full of credit cards.

- **Merchant**

  A person or organization that has goods or services to sell to the cardholder.

- **Payment Gateway**

  This is a function provided by or on behalf of an acquirer. (It is more accurately called the acquirer payment gateway.) The payment gateway interfaces between SET and the existing bankcard association networks for authorization and capture functions. To put it another way; the payment gateway acts as a proxy for the bankcard network functions.

3.7.3 SET Relationship

Many of the inter-role relationships are implicit in the definitions of the roles themselves (above). SET can divide the relationships into three types:

1. Contractual relationships: These represent agreements in law between the different parties to provide services and accept

responsibilities. They have nothing to do with SET directly, except that SET assumes the relationships already exist.

2. Administrative relationships: These are relationships required to set up the SET environment before any payments can flow.

3. Operational relationships: These are the short-term relationships that take place when a payment happens. These are all defined by SET protocol flows.

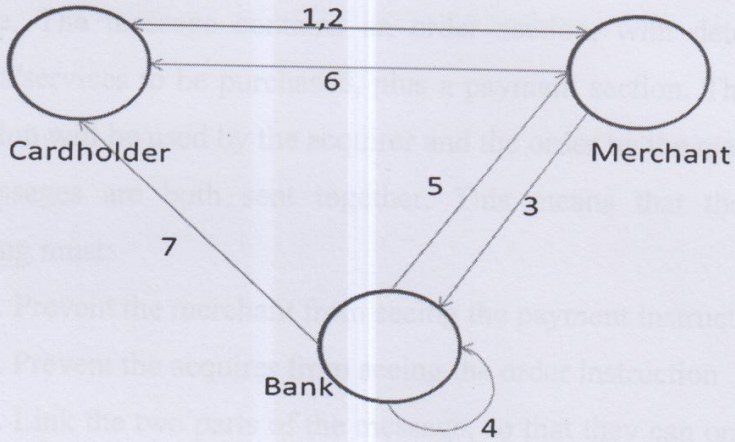

Figure 3.3 Basic SET Relationships

1. Customer browser and decides purchase
2. SET sent Order and payment information
3. Merchant forward payment information to bank
4. Bank Check and authorization payment
5. Merchant capture transaction
6. Merchant responds the order
7. Bank send card information

3.8 Key Technologies of SET

The key technologies of the secure electronic transcription are

- Confidentiality of information: RC5
- Integrity of data: MD5 hash codes
- Cardholder account authentication: RSA signatures

- Merchant authentication: RSA signatures
- Privacy: separation of order and payment information using dual signatures.

3.8.1 Dual Signature

Within the SET protocols, there is a situation where the cardholder communicates with both the merchant and payment gateway in a single message. The message contains an order section, with details of the products/services to be purchased, plus a payment section. The payment instruction will be used by the acquirer and the order by the merchant, but the messages are both sent together. This means that the message packaging must:

1. Prevent the merchant from seeing the payment instruction
2. Prevent the acquirer from seeing the order instruction
3. Link the two parts of the message, so that they can only be used as a pair

In this case, SET uses a technique called dual signature. When the order and payment instruction is sent by the cardholder, the merchant will be able to see only the order instruction, and the acquirer only the payment instruction. The merchant will not see the cardholders account information. In a SET transaction, the transfer of money and offer are linked allowing the money to be transferred to the merchant only if the cardholder accepts the offer.

- Concept: Link Two Messages Intended for Two Different Receivers:
  - Order Information (OI): Customer to Merchant
  - Payment Information (PI): Customer to Bank

- Goal: Limit Information to A "Need-to-Know" Basis:
  - Merchant does not need credit card number.
  - Bank does not need details of customer order.
  - Afford the customer extra protection in terms of privacy by keeping these items separate.
- This link is needed to prove that payment is intended for this order and not some other one.
- The operation for dual signature is as follows:
  - Take the hash (MD5) of the payment and order information.
  - These two hash values are concatenated [H(PI) || H(OI)] and then the result is hashed.
  - Customer encrypts the final hash with a private key creating the dual signature.

$$DS = E_{KRC} [H(H(PI) || H(OI))]$$

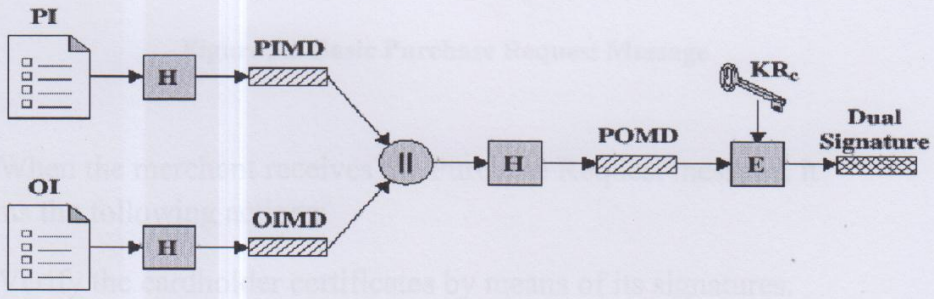

Figure 3.4 Basic Dual Signatures

Cardholder creates purchase request form by using Dual signature, Symmetric key such as RC5 and Asymmetric key (RSA). And then cardholder sends the purchase request message to the merchant[9].

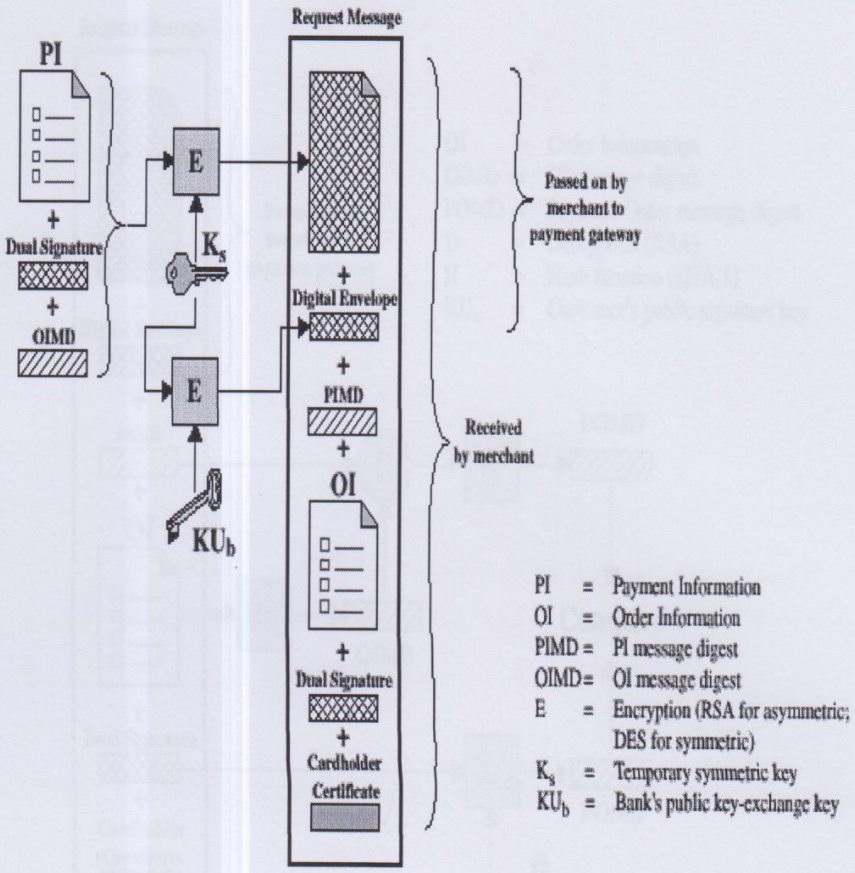

Figure 3.5 Basic Purchase Request Message

When the merchant receives the Purchase Request message, it performs the following actions:

- Verify the cardholder certificates by means of its signatures.
- Verifies the dual signature using the customer's public key signature
- Processes the order and forwards the payment information to the payment gateway for authorization.
- Sends a purchase response to the cardholder.

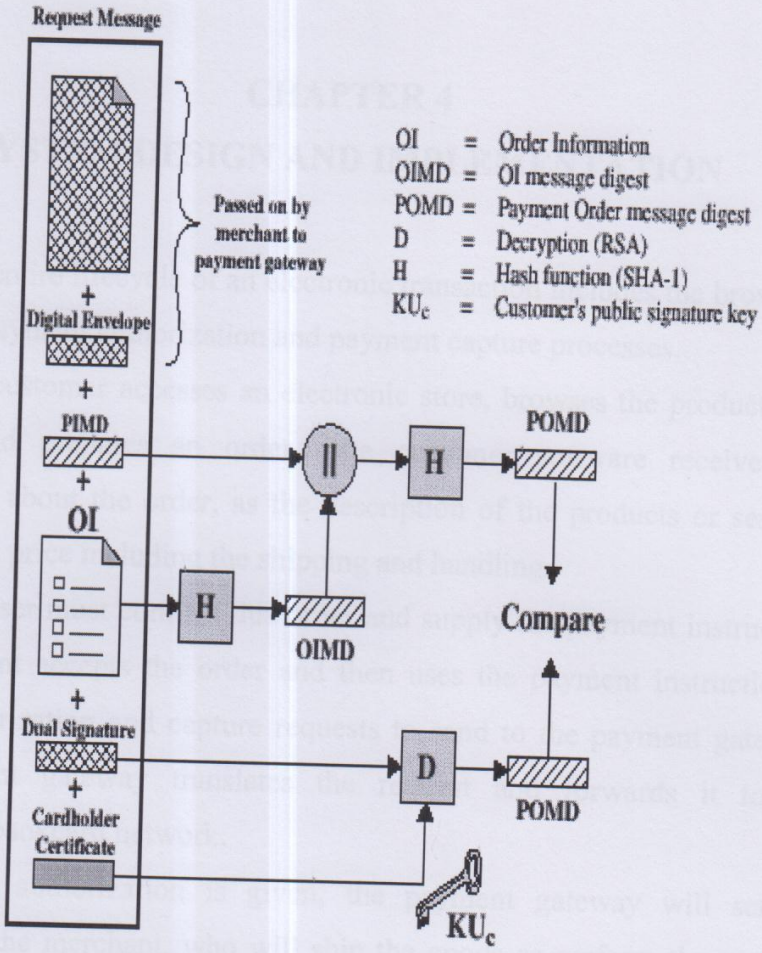

Figure 3.6 Merchant Verifies Customer Purchase Request

The payment gateway performs the following tasks:

- Verify All Certificates
- Decrypt Authorization Block Digital Envelope to Obtain Symmetric Key and Decrypt Block
- Verify Merchant Signature on Authorization Block
- Decrypt Payment Block Digital Envelope to Obtain Symmetric Key and Decrypt Block
- Verify Dual Signature on Payment Block
- Verify Received *Transaction ID received from* Merchant Matches PI Received from Customer

CHAPTER 4
SYSTEM DESIGN AND IMPLEMENTATION

The entire lifecycle of an electronic transaction includes the browsing, purchase, payment authorization and payment capture processes.

The customer accesses an electronic store, browses the products and services and prepares an order. The customer software receives the information about the order, as the description of the products or services and the total price including the shipping and handling.

The user must confirm this order and supply the payment instruction. The merchant accepts the order and then uses the payment instruction to create authorization and capture requests to send to the payment gateway. The payment gateway translates the request and forwards it to the appropriate bankcard network.

If the authorization is given, the payment gateway will send a response to the merchant, who will ship the goods or perform the services indicated in the order.

The merchant uses the capture request to initiate the payment from the payment gateway. As in the case of authorization, the payment gateway translates the capture request into a funds transfer to the merchants account.

The whole process has three parts that are

1. Customer side
2. Merchant side
3. Bank side

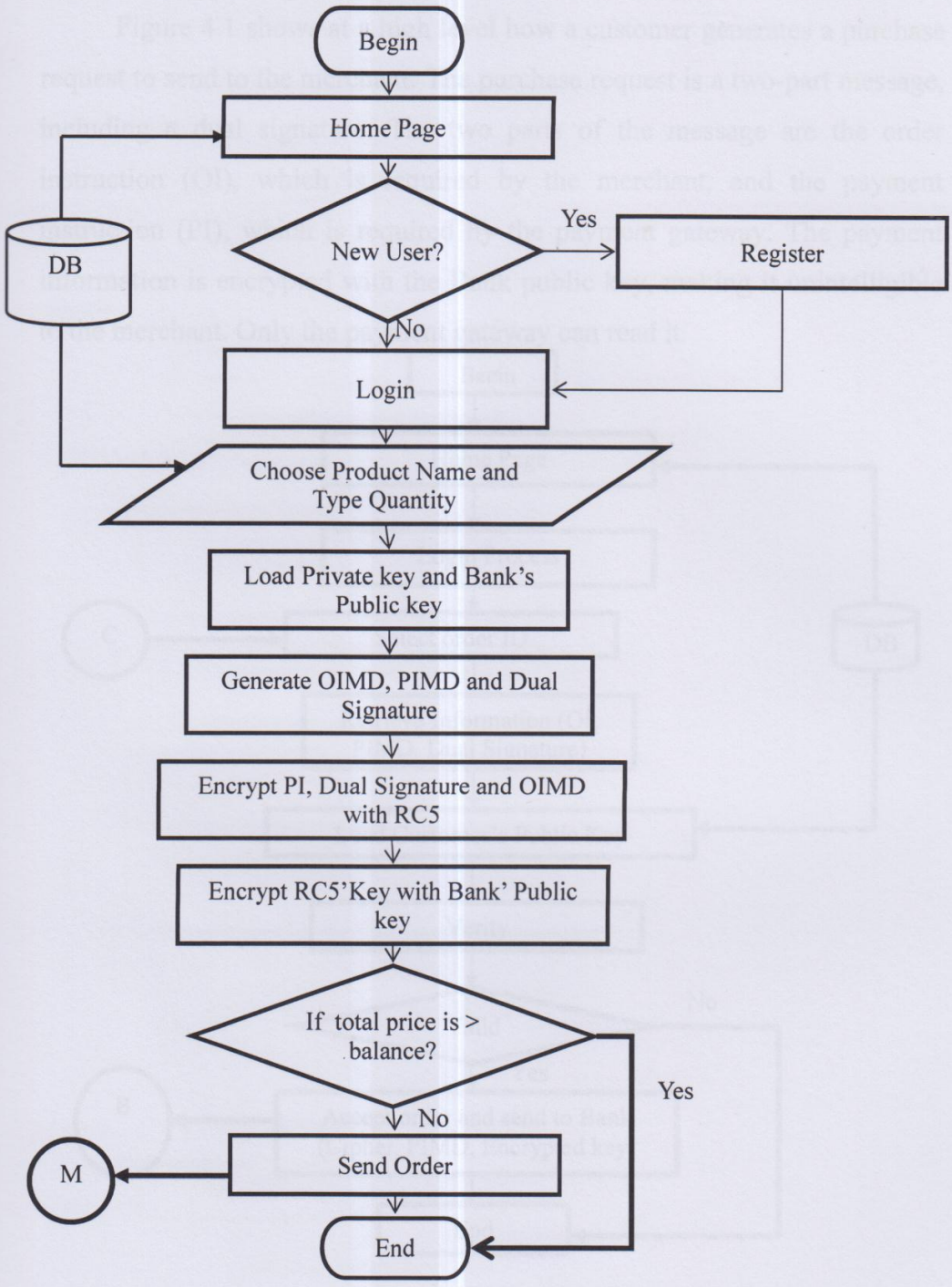

Figure 4.1 Customer Process

Figure 4.1 shows at a high level how a customer generates a purchase request to send to the merchant. The purchase request is a two-part message, including a dual signature. The two parts of the message are the order instruction (OI), which is required by the merchant, and the payment instruction (PI), which is required by the payment gateway. The payment information is encrypted with the Bank public key, making it unintelligible to the merchant. Only the payment gateway can read it.

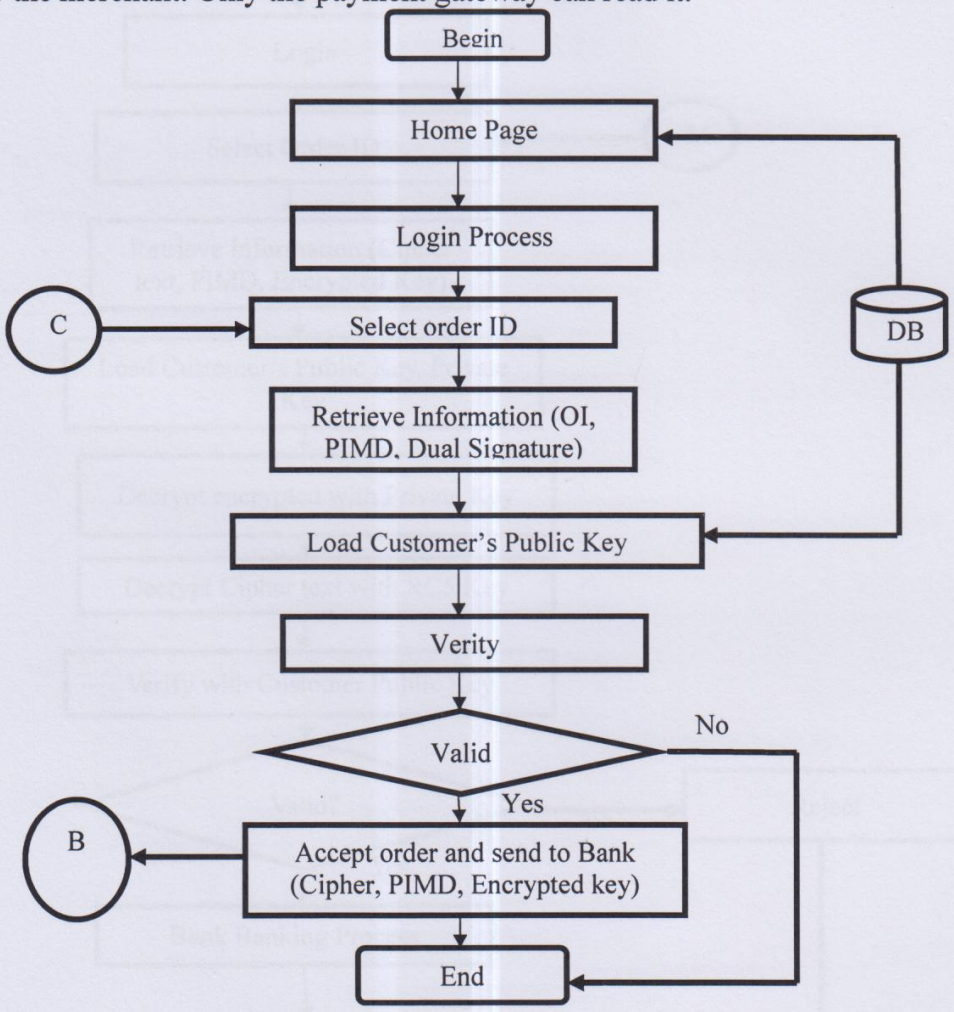

Figure 4.2 Merchant Process

Figure 4.2 shows the merchant process .The merchant processes the order request and forwards the encrypted payment message and payment digital envelope to the payment gateway for payment authorization. The merchant creates the response message and then send response message to customer.

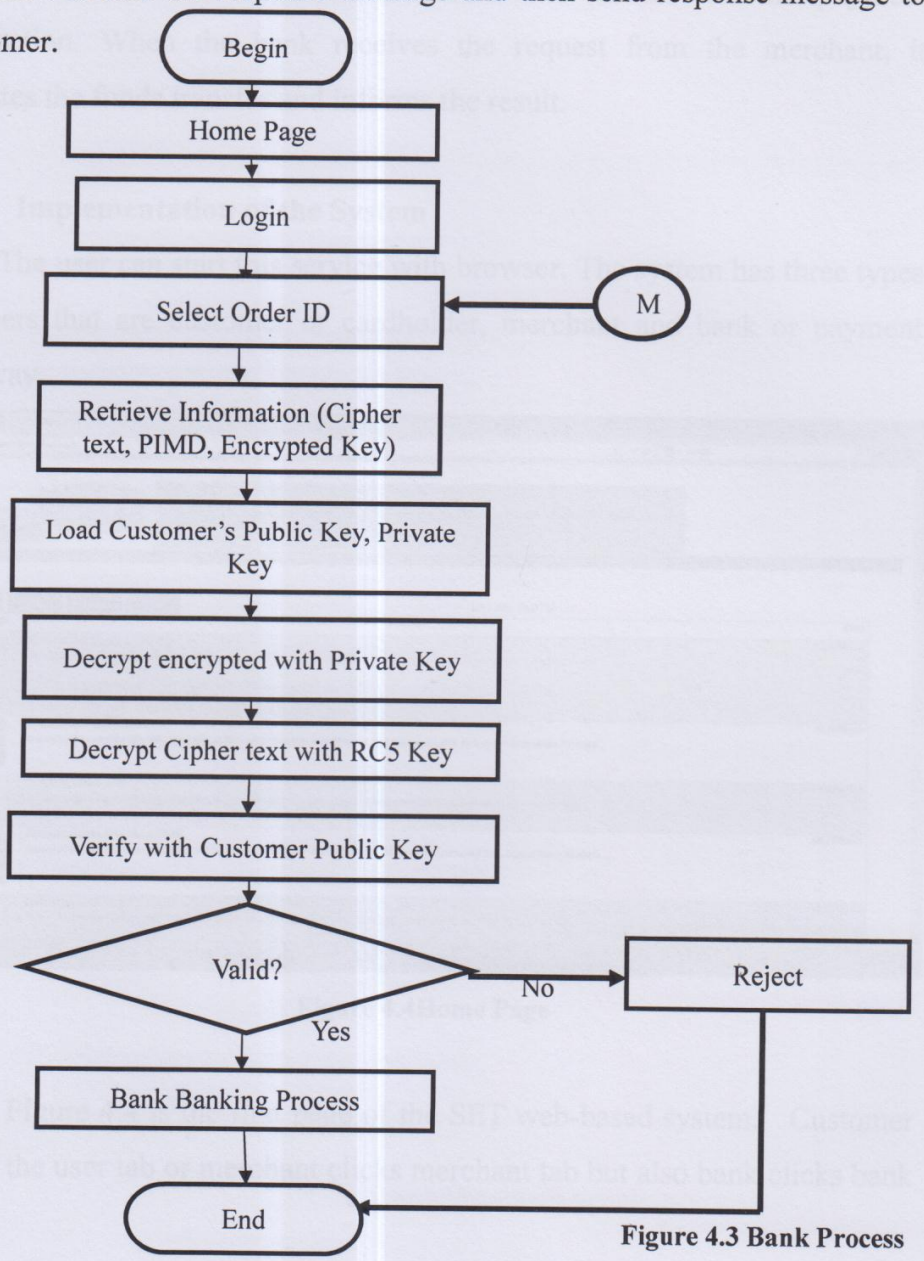

Figure 4.3 Bank Process

Figure 4.3 shows the banking process. The bank receives the authorization message and verifies the merchant and customer. The Bank also verifies the integrity of the transaction by checking the transaction ID received from the merchant matches the ID sent with the customer payment instruction. When the bank receives the request from the merchant, it initiates the funds transfer and informs the result.

4.1 Implementation of the System

The user can start this service with browser. The system has three types of users that are customer or cardholder, merchant and bank or payment gateway.

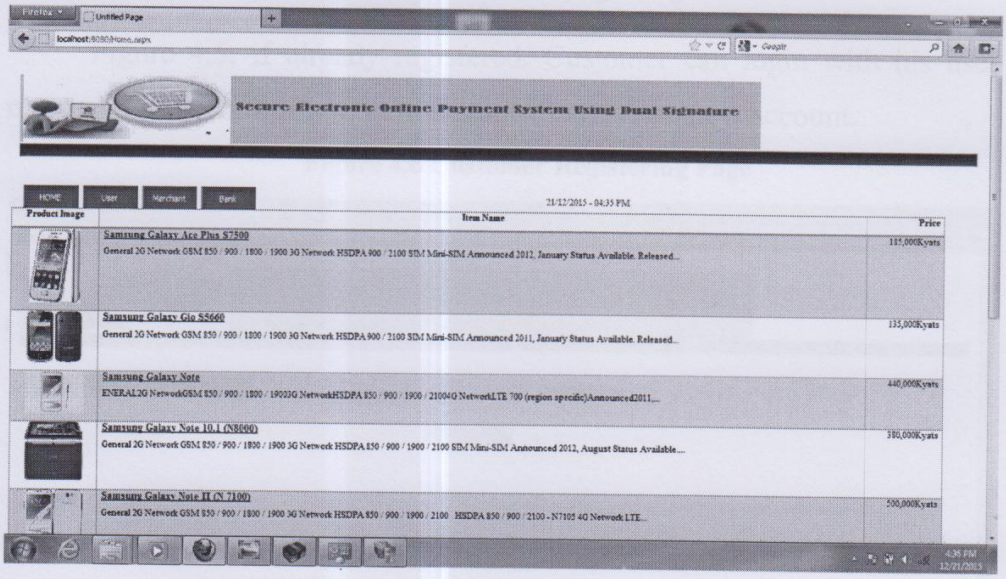

Figure 4.4Home Page

Figure 4.4 is the first page of the SET web-based system. Customer clicks the user tab or merchant clicks merchant tab but also bank clicks bank tab.

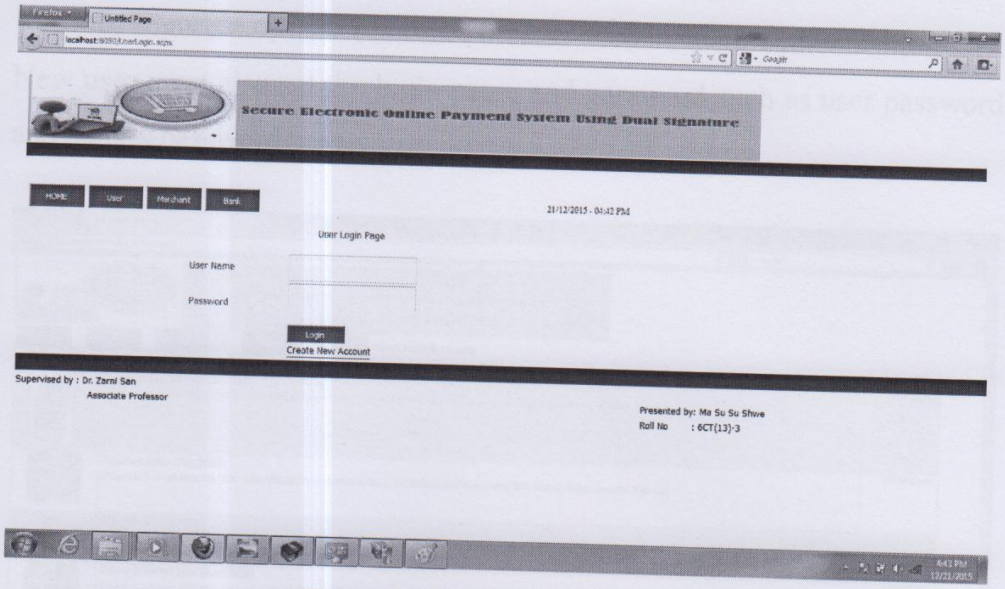

Figure 4.5 Customer Login Page

Figure 4.5, if already registered. Customer can login with his user name and password. If not, new customer can create new account.

Figure 4.6 Customer Registering Page

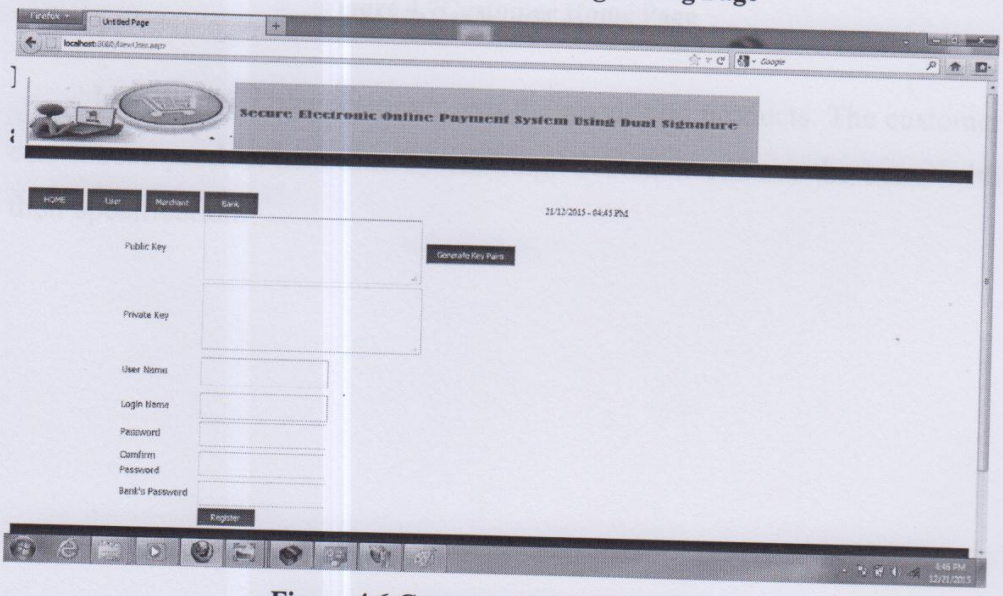

Figure 4.6 Customer Registering Page

In Figure 4.6, new customer creates account with authentication key. New user must also set the login name and password such as user password and bank's password.

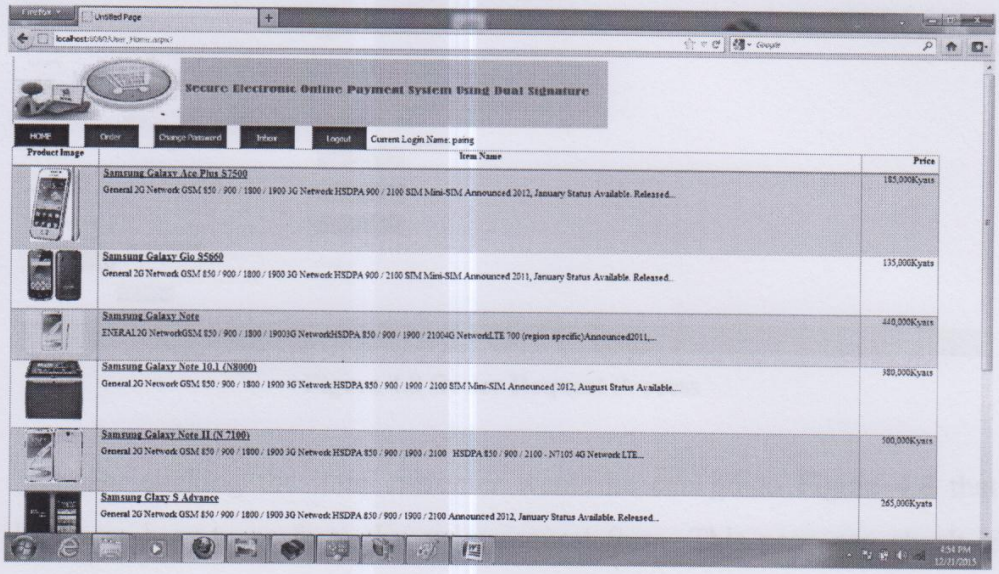

Figure 4.7Customer Home Page

In Figure 4.7, the customer sees the goods and products. The customer can choose product he wants by clicking. Customer can see the items with their specification.

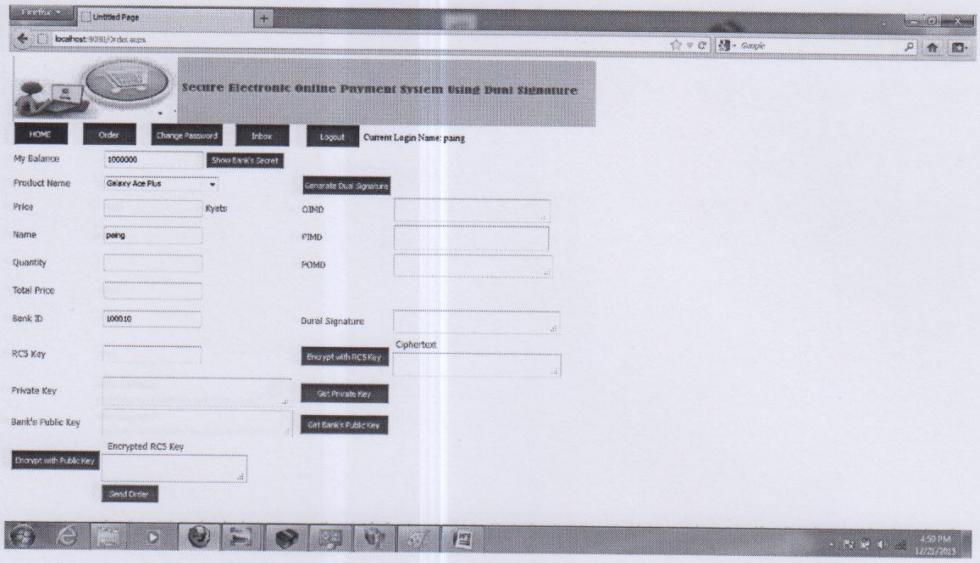

Figure 4.8 Order Request Process

By clicking the item customer wants he can go to Figure 4.8 that illustrates how to perform the order request form. This page can check if balance is enough or not automatically. If there is no enough balance, you can request bank admin.

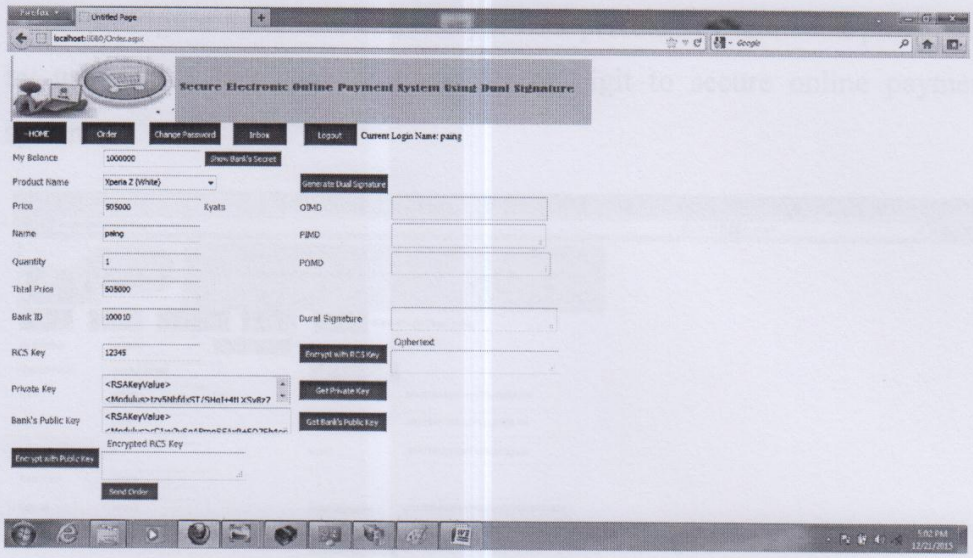

Figure 4.9 Key Expension Process

In Figure 4.9, customer chooses product name, total price and quality. Customer can define RC5 key like character or digit number. Customer can get private key and bank's public key.

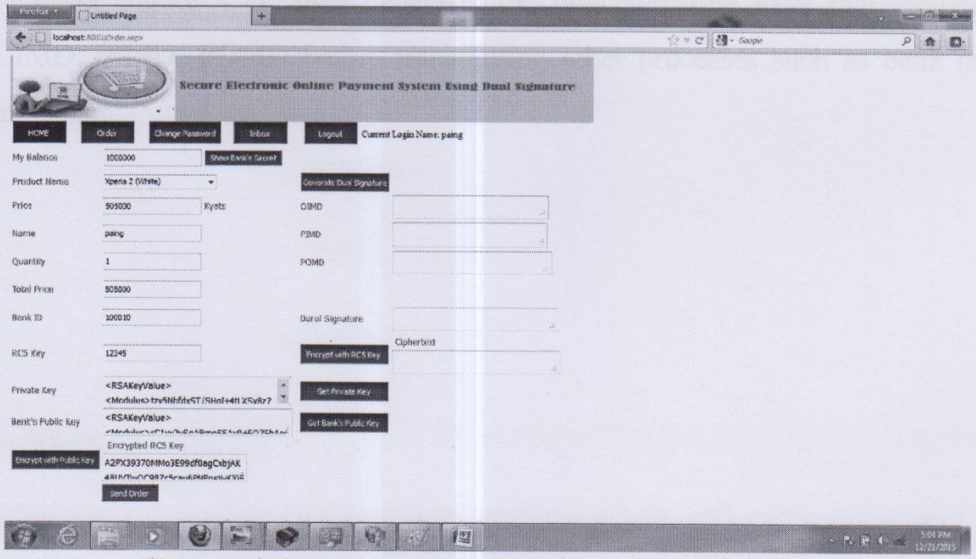

Figure4.10 Create Encrypted Key (Encrypt with RC5key).

In Figure 4.10, user can encrypt user private key and bank public key by using RC5 key such as character or digit to secure online payment system.

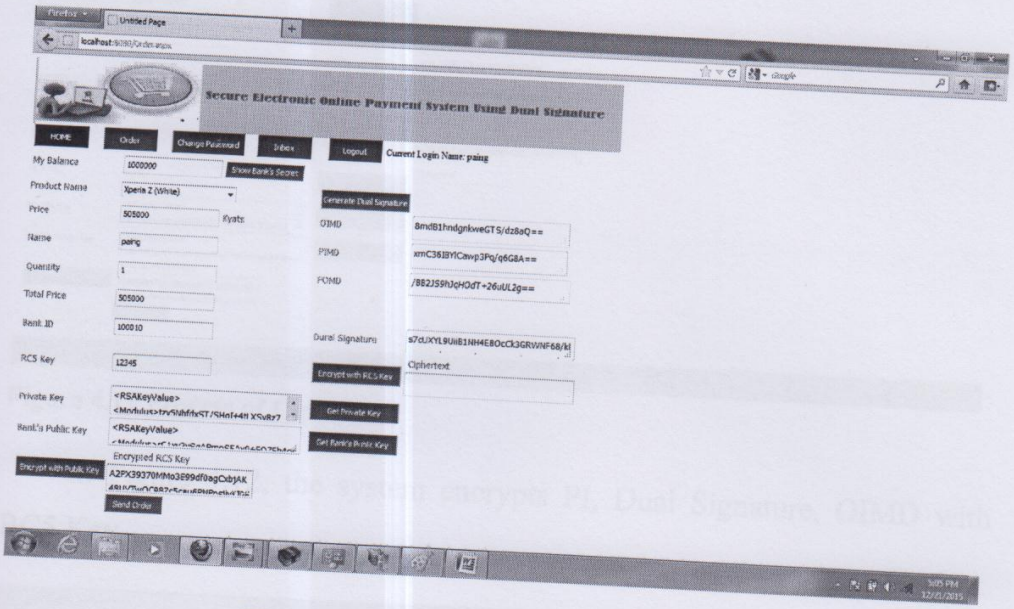

Figure 4.11 Generation of Dual Signature

In Figure 4.11, user generates the dual signature keys that include OIMD, PIMD, POMD before transmitting other processes such as bank or merchant.

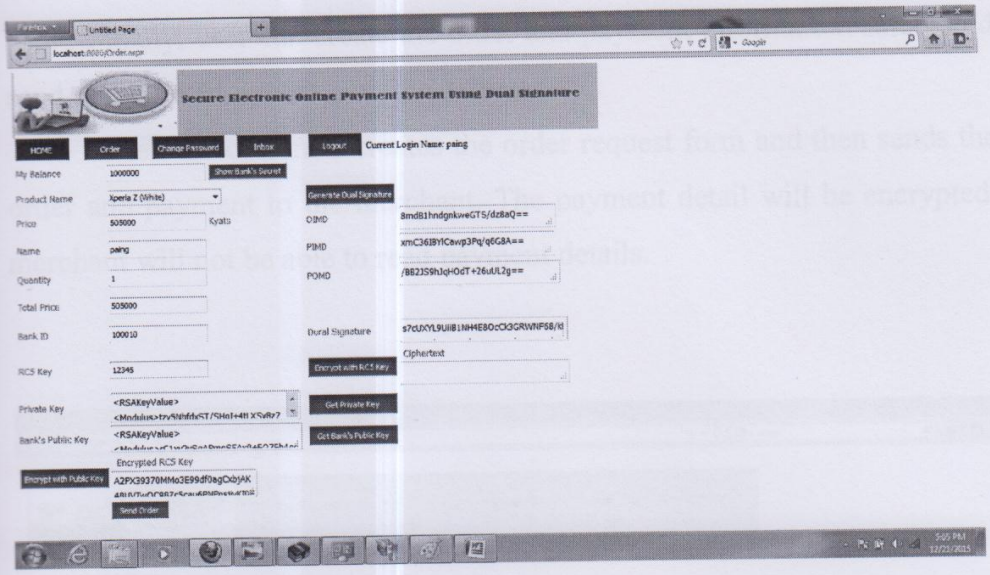

Figure 4.12 Process of Ciphertext

In Figure 4.12, the system encrypts PI, Dual Signature, OIMD with RC5 Key.

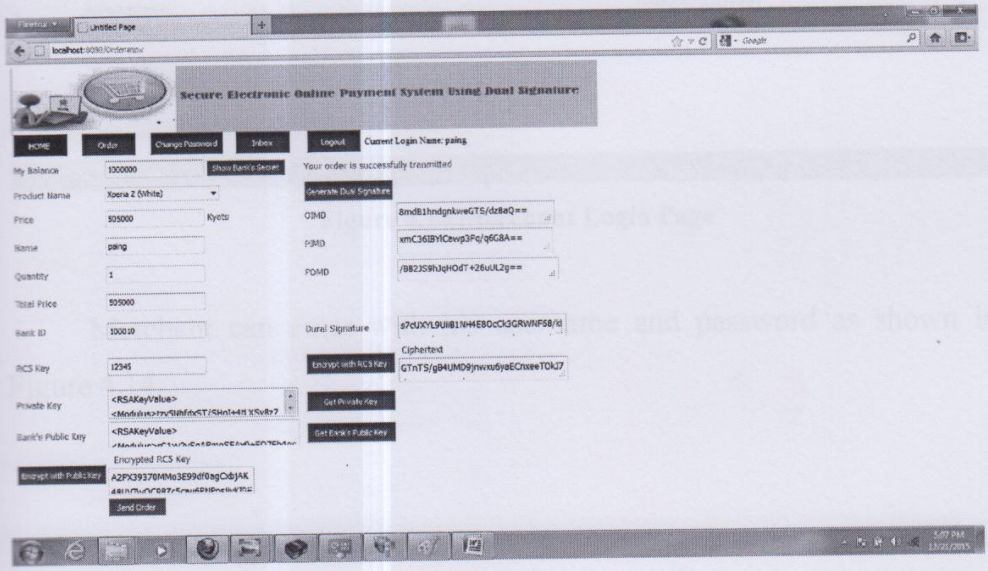

Figure 4.13 Successfully Order

Finally, user can create the order and payment information safety and send to merchant by using online deliver.

When the customer creates the order request form and then sends the order and payment to the merchant. The payment detail will be encrypted; merchant will not be able to read payment details.

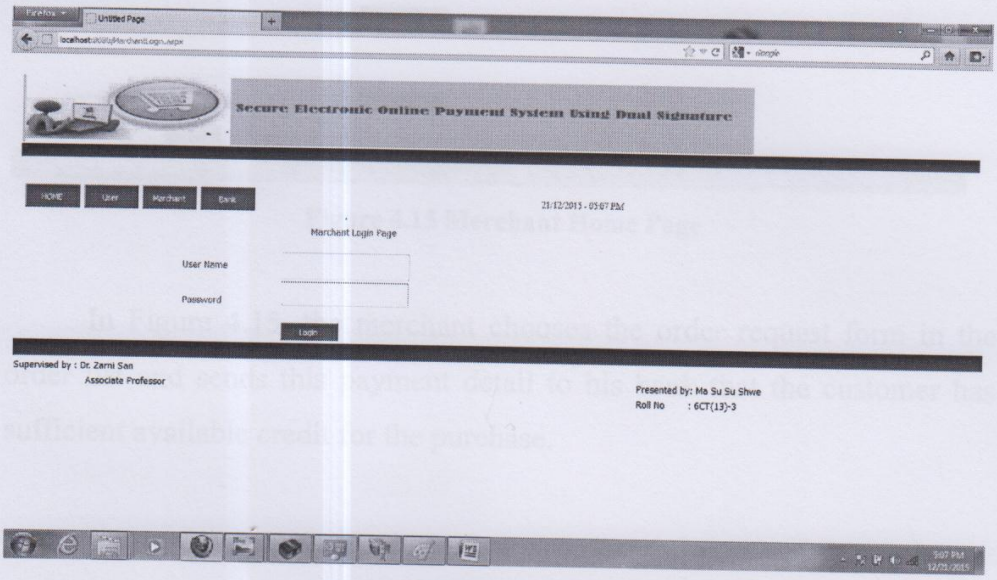

Figure 4.14 Merchant Login Page

Merchant can enter with his username and password as shown in Figure 4.14.

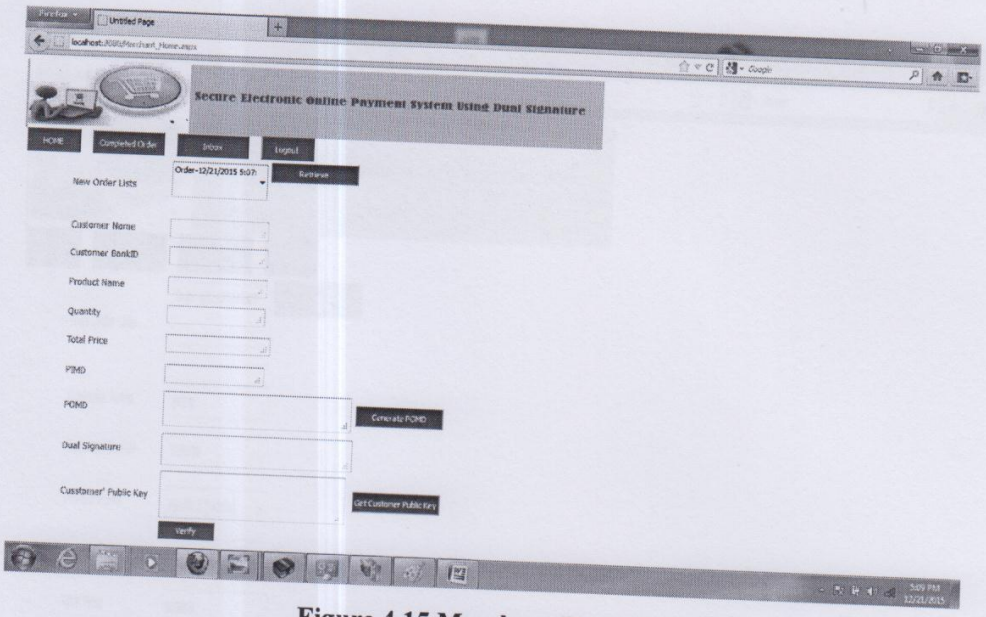

Figure 4.15 Merchant Home Page

In Figure 4.15, the merchant chooses the order request form in the order list and sends this payment detail to his bank that the customer has sufficient available credit for the purchase.

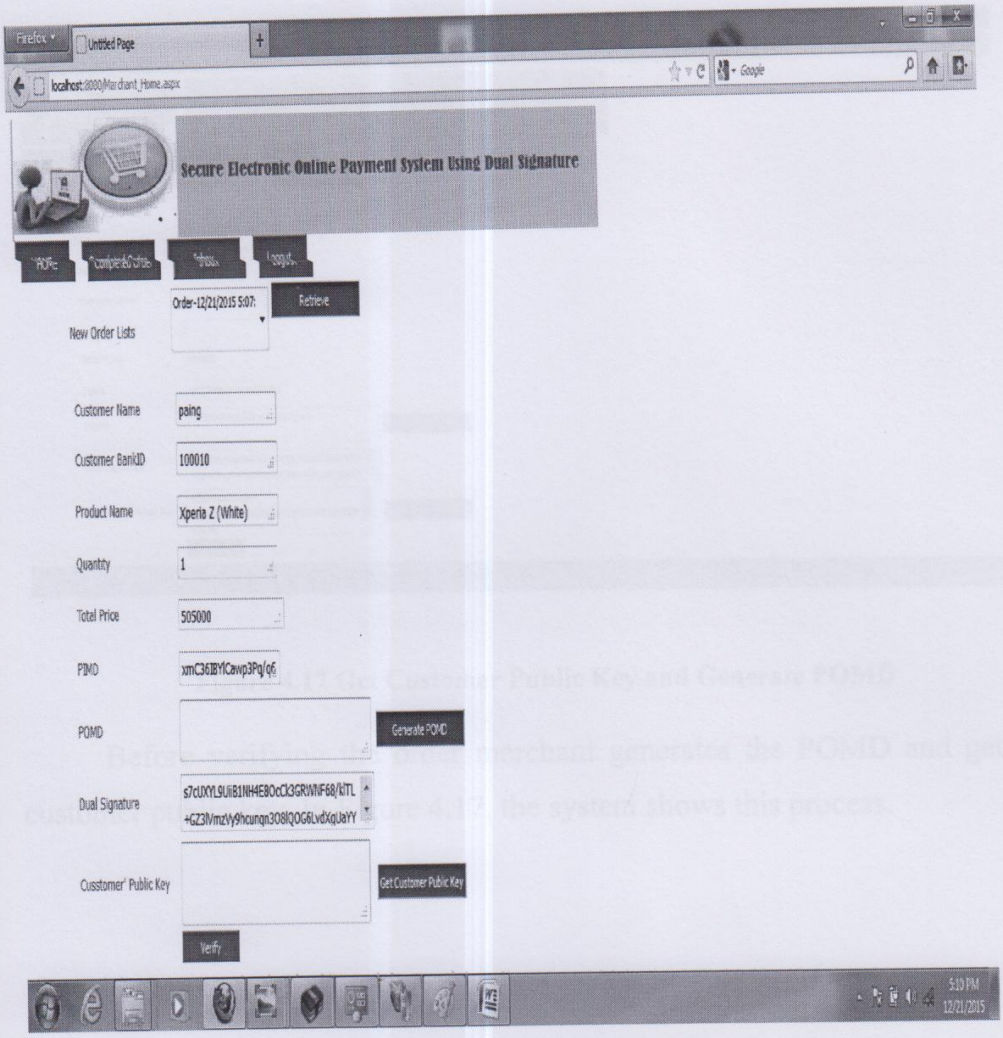

Figure 4.16 Merchant Receive Order

Figure 4.16 illustrate, when merchant receive order, will retrieve customer name, bank ID, productname, price, PIMD, Dual Signature.

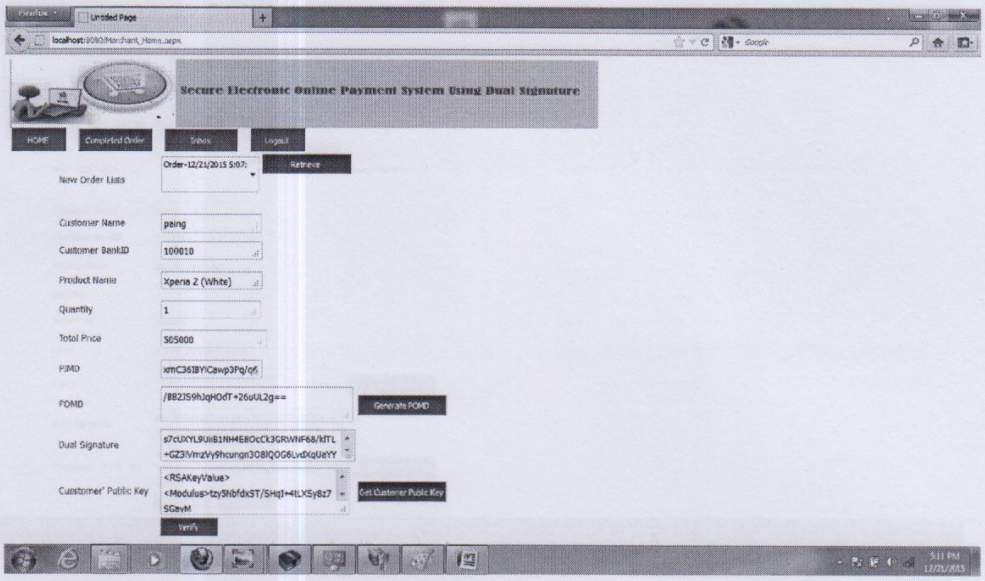

Figure 4.17 Get Customer Public Key and Generate POMD

Before verifying the order merchant generates the POMD and gets customer public key. In Figure 4.17, the system shows this process.

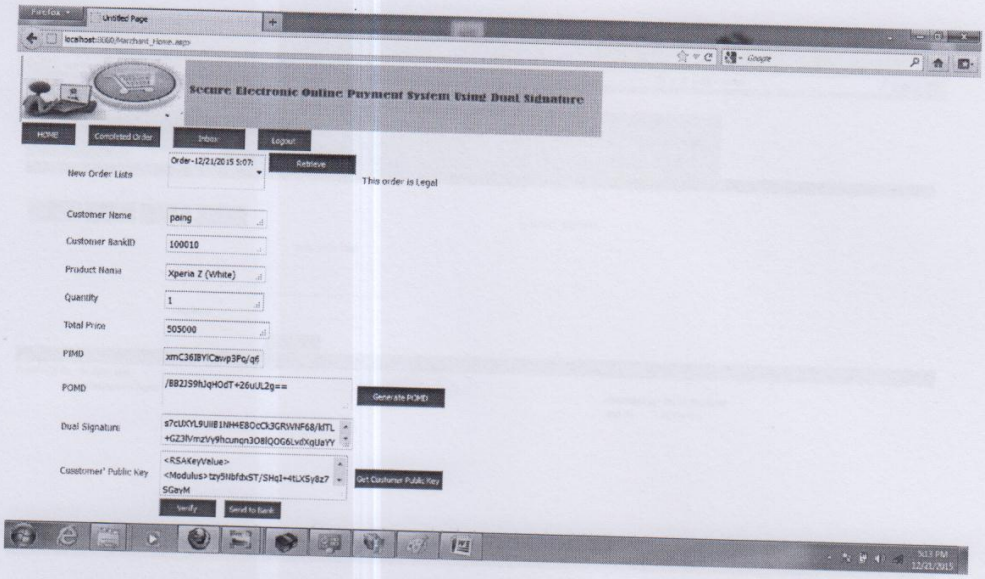

Figure 4.18 Order is legal and then send to Bank

Merchant checks if the order is legal or not. If the order is legal, merchant sends the encrypted message to the bank. If not, merchant checks the information and requirement.

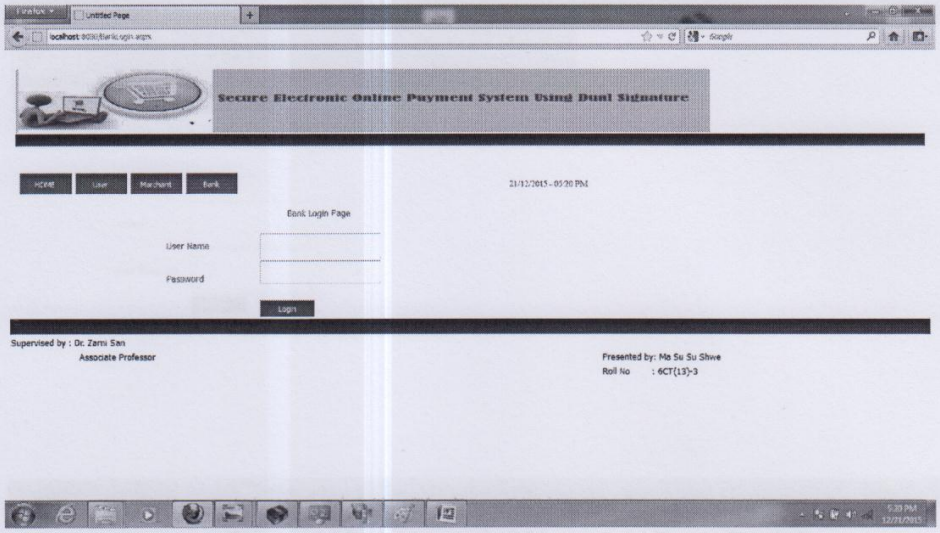

Figure 4.19 Bank Admin Login

In Figure 4.19, bank admin logins with his login name and password. Bank admin can add money in the customer account when he receives customer request.

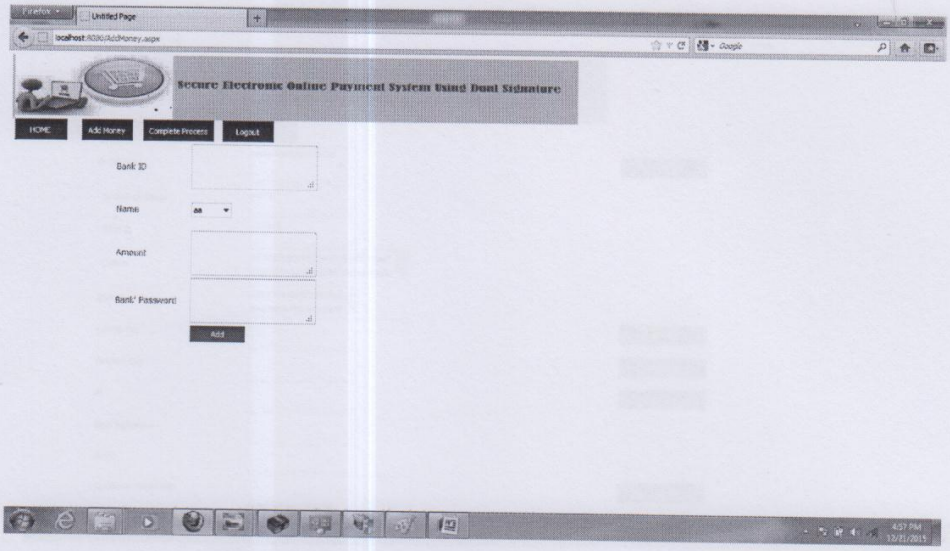

Figure 4.20 Customer's Money Add Form Bank Process

In Figure 4.20, the system shows the bank admin site. This page includes home, add money and complete process.

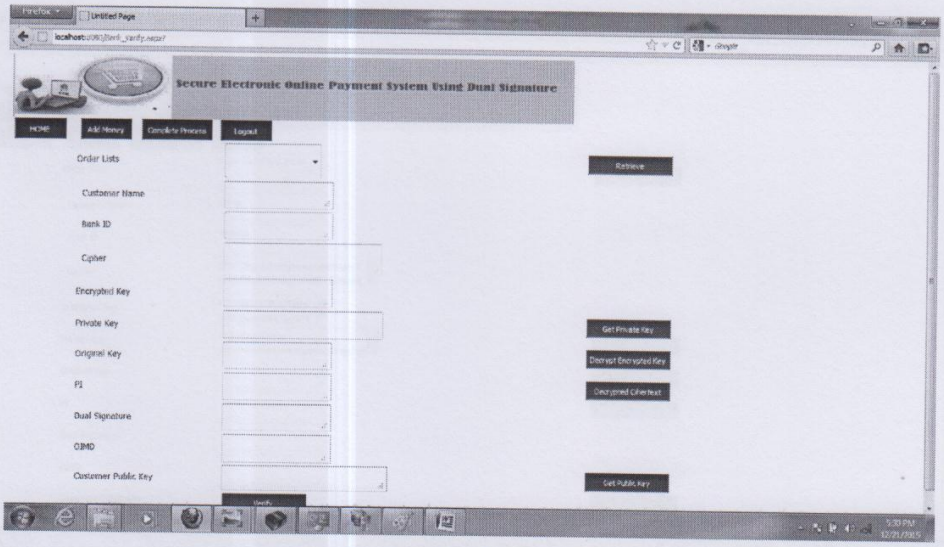

Figure 4.21 Bank Home Page

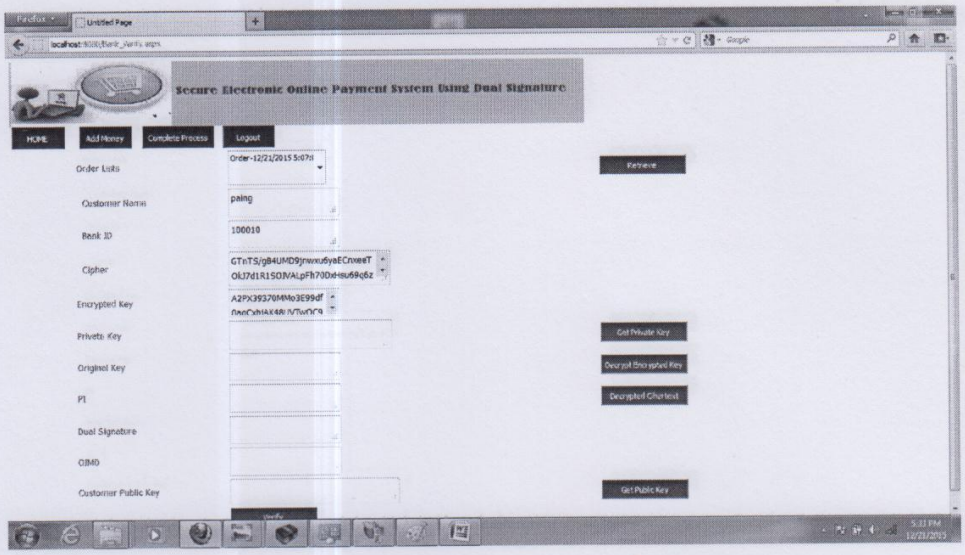

Figure 4.22 Retrieve Customer Information

By selecting the order from the order lists, Bank admin sees the customer information such as bank ID, customer name, cipher and encrypted key.

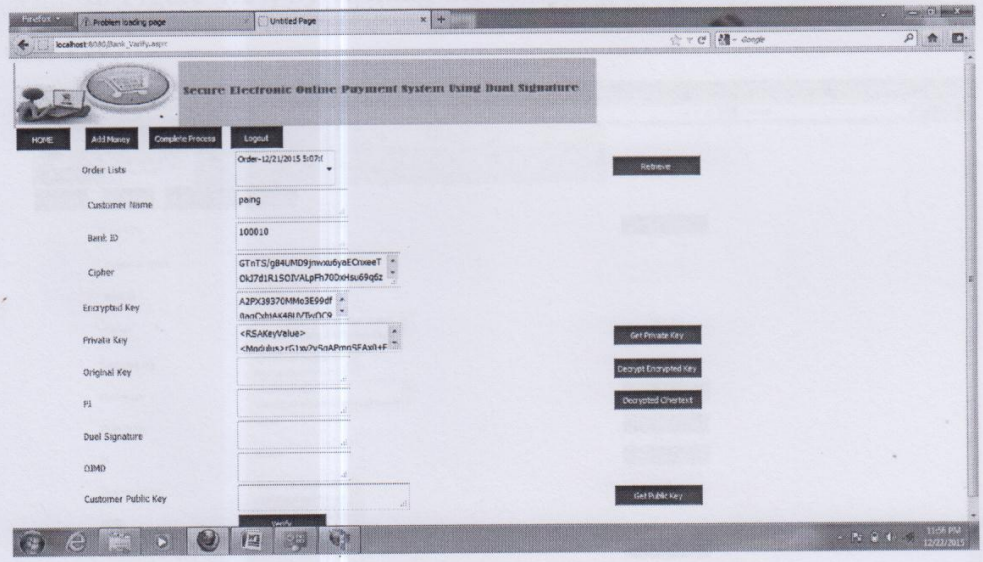

Figure 4.23 Loading Customer Private Key

After clicking the get private key button, the system shows the private key of the customer. The process is illustrated in figure 4.22.

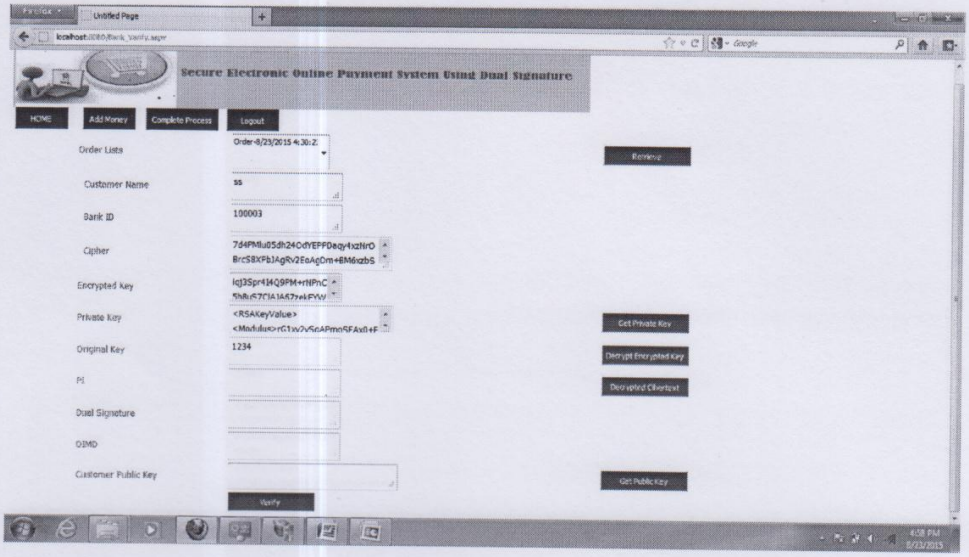

Figure 4.24 Decrypt Encrypted Key with Customer Private Key

In Figure 4.24, by clicking the decrypt private key, admin can see the user key.

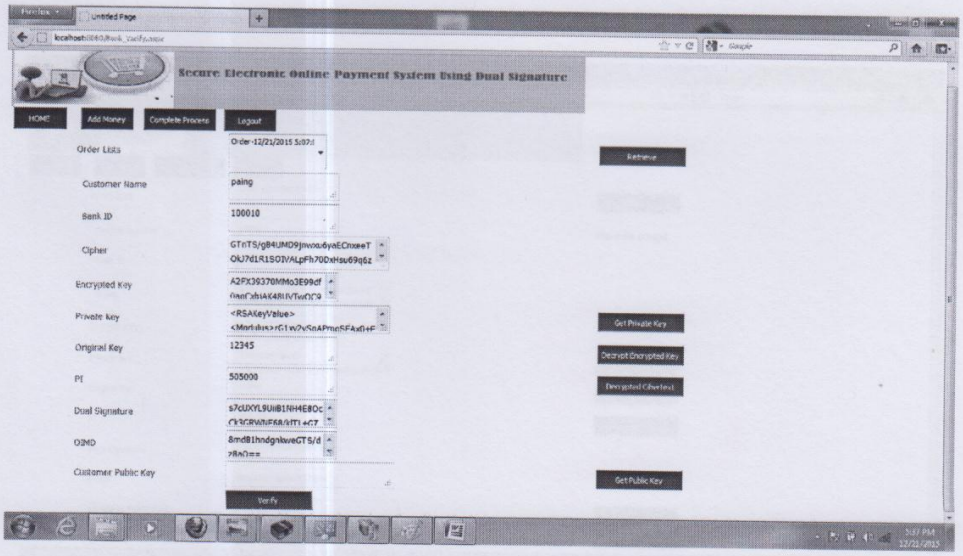

Figure 4.25 Decrypted Ciphertext with RC5 Key

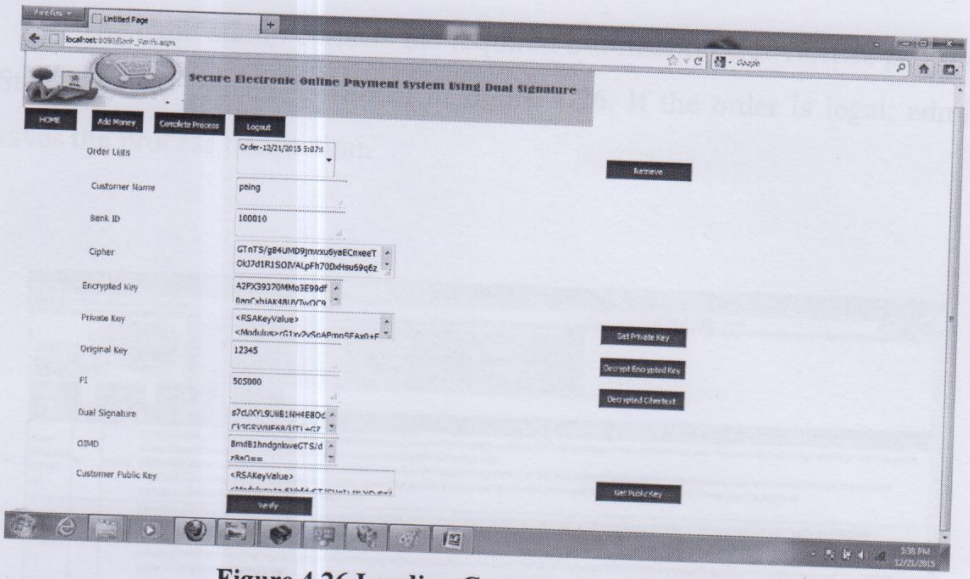

Figure 4.26 Loading Customer's Public Key

In Figure 4.24 and 4.25, admin collects the requirement information such as user private key, original text and customer public key by clicking the associate button.

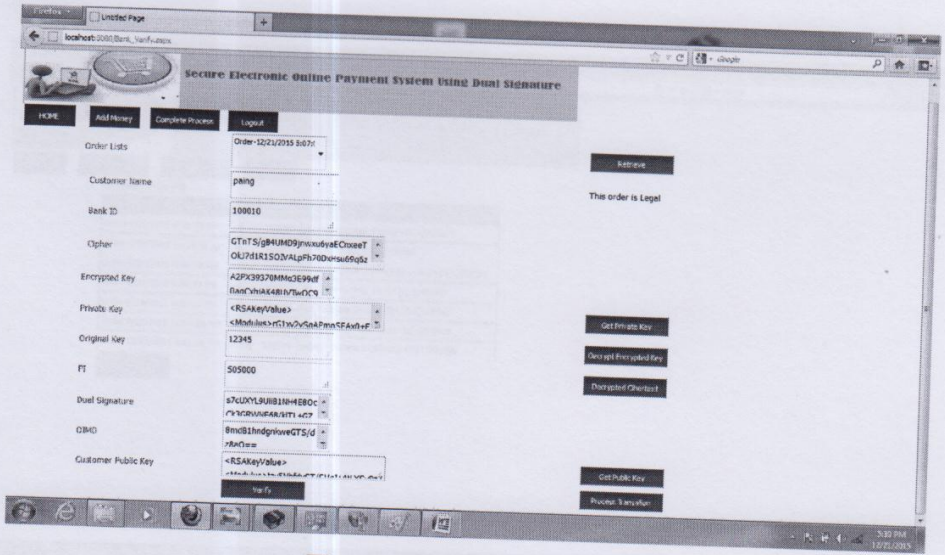

Figure 4.27 Order is Legal

The bank admin collects the required information and verifies process. Step –by-step process is shown in figure 4.26. If the order is legal, admin saves the process transaction.

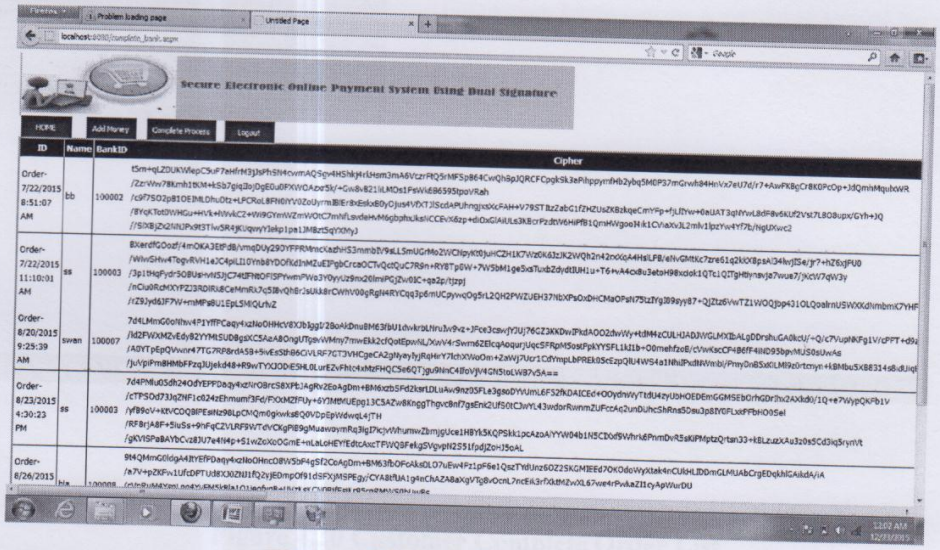

Figure 4.28 Bank Transaction list

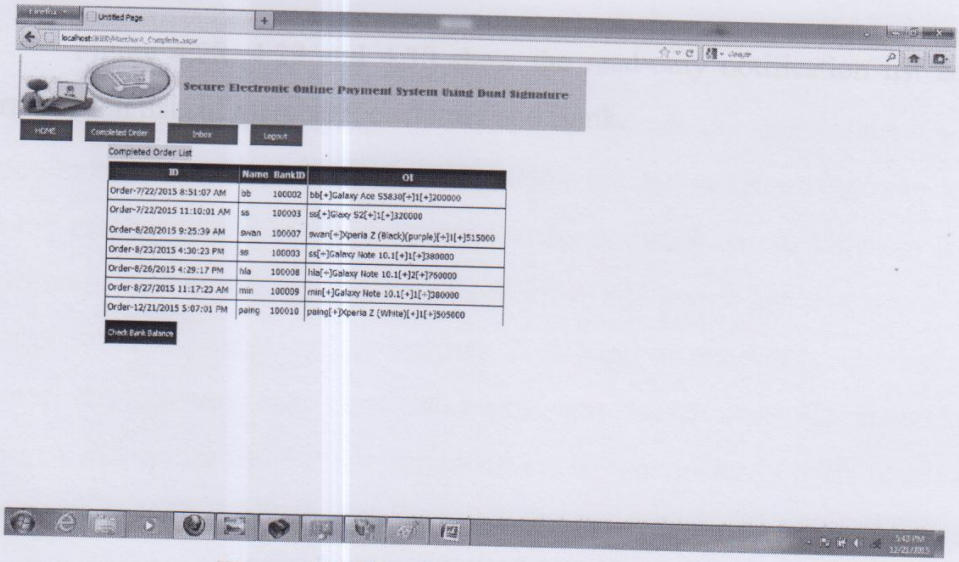

Figure 4.29 Merchant Complete Order List

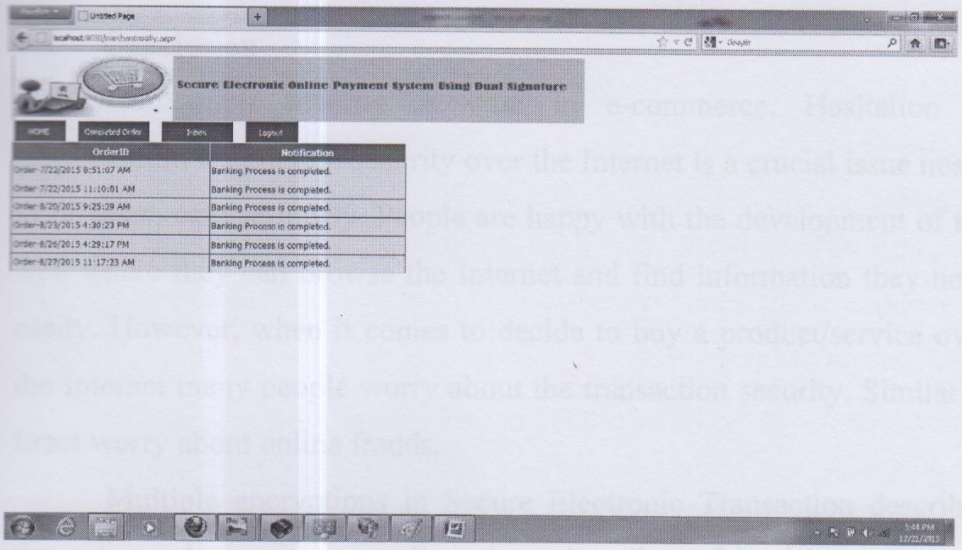

Figure 4.30 Customer Complete Order List

Figure 4.28, 4.29 and 4.30 show the read-only notification list and transaction lists of merchant, customer and Bank.

CHAPTER 5

CONCLUSION, LIMITION AND FURTHER EXTENSION

5.1 Conclusion

Transaction security is vital in e-commerce. Hesitation or scepticism in transaction security over the Internet is a crucial issue needs to be taken care seriously. People are happy with the development of the web where they can browse the Internet and find information they need easily. However, when it comes to decide to buy a product/service over the Internet many people worry about the transaction security. Similarly, firms worry about online frauds.

Multiple encryptions in Secure Electronic Transaction describes the enhanced security as well as integrity of confidential data due to multiple encryption operations.

The ideal of the secure electronic transactions protocol (SET) with multiple encryptions is important for the success of electronic commerce.

5.2 Limitation

SET is specifically a payment protocol. It defines the communication between cardholder, merchant and payment gateway for card purchases and refunds. It defines the communication between the different parties and certification authorities for public key signature. It does not define anything beyond that. Such level of security is required to earn the interest and trust of customers, merchants and financial organizations for online transaction over wireless network. Such level of security is required to earn the interest and trust of customers, merchants and financial organizations for online transaction over wireless network.

The merchant still has access to the payment information, and all information is encrypted using the same key strength. The main advantage over SSL/TLS is that 3-D Secure provides credit card authorization and non-repudiation. On the other hand, prior customer registration is required.

5.3 Further Extension

Secure Electronic Payment schemes have through SSL, SET, and secure communication tunnel. The system can ensure the security of transaction, so it is an excellent solution to the E-business model. Main advantages of Payment System for Internet Transaction are: it uses strong cryptography and authenticity checking models; the merchant is prevented from seeing payment information; the customer can easy to use the system, since he is not required to install additional software for secure payments or to have a digital certificate.

The electronic payment system is to be secure for Internet transaction participants such as Payment gateway server, Bank sever and Merchant server. The security architecture of the system is designed by using Many Security Protocols and techniques, which eliminates the fraud that occurs today with stolen credit card/debit card payment information and customer information. Electronic commerce involves the exchange of some form of money for goods and services over the Internet but today, Internet is an insecure and unreliable media. The asymmetric key cryptosystem Methodology with help of Security Protocol, secure communication tunnel techniques can protect conventional transaction data such as account numbers, amount and other information.

REFERENCE

[1]. A SENGUPTA[1], C MAZUMDAR[1] and MS BARIK[2], "E-Commerce Security – A Life Cycle Approach", [1]Centre for Distributed Computing, Department of Computer Science and Engineering, Jadavpur University, Kolkata 700 032, Inida, [2]Department of Information Technology, Bengal Engineering and Science University, Shibpur 711 103, India. e-mail:sg-anirban@yahoo.co.in; chandanm@vsnl. com; mridul@it.becs. ac.in.

[2]. A.Koponen, "E-commerce Electronic Payments", Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, Email:atkopone@cc.hut.fi.

[3]. Berry Schoenmakers, "Basic Security of The Ecash Payment System", Digi Cash, Kruislaan 419, NL-1098 VA Amsterdam, The Netherlands, berry@digicash.com.

[4]. Electronic Commerce, Seventh Annual edition, Chapter 11 "Payments Systems For Electronic Commerce", www.juntak.com.

[5]. Man Young Rhee "Internet Security-Cryptographic Princles, Algorithms and Protocol", School of Electrical and Engineering, Seoul National University, Republic of Korea. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England. Reprinted July 2003, March 2005, www.wiley.com.

[6]. Niranjanamurthy M[1], DR. Dharmendra Chahar[2],"The Study of E-commerce Security Issues and Solutions", Assistant Professor Dept. of MCA, MSRIT, Bangalore, INDIA[1], HOD. Dept. of CS & IT, Seth G. B. Podar College, Nawalgarh (Jhunjhunu) -333042, INDIA[2], International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 7, July 2013.

[7]. OECD(2006), "Online Payment Systemsfor E-commerce", OECD Digital Economy Papers, No.117, OECDPublishing, http://dx.doi.org/ 10.1787/231454241135

[8]. Singh Sumanjeet, "Emergence of Payment Systems in the Age of Electronic Commerce: the state of ART", University Asia Pacific Journal of Finance and Banking Research Vol. 3. No. 3. 2009. University of Delhi, India, E-mail: sumanjeetsingh@gmail.com.

[9]. William Stallings, "Cryptography and Network Security", Principle and Practice, Third Edition, publication date August 27, 2002.

[10]. Yang Jing, "On-line Payment and Security of E-commerce", School of KeXin, Hebei University of Engineering, Handan, China hdjianghua @126.com.