Chapter 3

SECURITY IN M-COMMERCE

3.1 Introduction

Three billion people are expected to own mobile phones in the globe by 2010. There are currently 225 million mobile phones in India and 100 million are added every year. In a few years more than 500 million people are expected to have mobile phones in India [219]. Mobile commerce is a natural successor to electronic commerce. The capability to pay electronically coupled with a website is the engine behind electronic commerce. Electronic commerce has been facilitated by automatic teller machines (ATMs), shared banking networks, debit and credit card systems, electronic money and stored value applications, electronic bill presentment and payment systems. Mobile payments are a natural evolution e-payment schemes that will facilitate mobile commerce. A mobile payment or m-payment may be defined, for our purposes as any payment where a mobile device is used to initiate, authorize and confirm an exchange of financial value in return for goods and services [220]. Mobile devices may include mobile phones, PDAs, wireless tablets or any other device that connect to mobile telecommunication network and make it possible for payments to be made [221]. The realization of mobile payments will make possible new and unforeseen ways of convenience and commerce. Unsuspected technological innovations are possible. Music, video on demand, location based services identifiable through mobile handheld devices procurement of travel, hospitality, entertainment and other uses are possible when mobile payments become feasible and ubiquitous. Mobile payments can become a complement to cash, cheques, credit cards
and debit cards. It can also be used for payment of bills (especially utilities and insurance premiums) with access to account-based payment instruments such as electronic funds transfer, Internet banking payments, direct debit and electronic bill presentment.

Several mobile payment companies and initiatives in EU have failed and many have been discontinued [222]. In Europe and North America with few exceptions such as Austria, Spain and Scandinavian countries the development of mobile payments have not been successful. However, mobile payment services in Asia have been fairly successful especially in South Korea, Japan and other Asian countries (e.g., Mobile Suica, Edy, Moneta, Octopus, GCash). NTT DoCoMo has 20 million subscribers and 1.5 million of them have activated credit card functionality in Japan. The main difference between successful implementations of mobile payment services in the Asia Pacific region and failure in Europe and North America is primarily attributed to the ‘payment culture’ of the consumers that are country-specific.

### 3.2 Mobile Payment Characteristics

A mobile payment service in order to become acceptable in the market as a mode of payment the following conditions have to be met [221]:

(a) **Simplicity and Usability**: The m-payment application must be user friendly with little or no learning curve to the customer. The customer must also be able to personalize the application to suit his or her convenience.

(b) **Universality**: M-payments service must provide for transactions between one customer to another customer (C2C), or from a business to a customer (B2C) or between businesses (B2B). The coverage should include domestic, regional and global environments. Payments must be possible in terms of both low value micro-payments and high value macro-payments.

(c) **Interoperability**: Development should be based on standards and open technologies that allow one implemented system to interact with other systems.
(d) **Security, Privacy and Trust:** A customer must be able to trust a mobile payment application provider that his or her credit or debit card information may not be misused. Secondly, when these transactions become recorded customer privacy should not be lost in the sense that the credit histories and spending patterns of the customer should not be openly available for public scrutiny. Mobile payments have to be as anonymous as cash transactions. Third, the system should be foolproof, resistant to attacks from hackers and terrorists. This may be provided using public key infrastructure security, biometrics and passwords integrated into the mobile payment solution architectures.

(e) **Cost:** The m-payments should not be costlier than existing payment mechanisms to the extent possible. A m-payment solution should compete with other modes of payment in terms of cost and convenience.

(f) **Speed:** The speed at which m-payments are executed must be acceptable to customers and merchants.

(g) **Cross border payments:** To become widely accepted the m-payment application must be available globally, word-wide.

### 3.3 Mobile Payment Solutions

Mobile payment solutions may be classified according to the type of payment effected, and based on the technology adopted to implement the solution. There are a variety of combinations of these frameworks technology adopted and mode of payment, a survey of which would constitute a study in itself. There are three different models available for m-payment solutions on the basis of payment [224]:

(a) Bank account based

(b) Credit card based

(c) Telecommunication company billing based
3.3.1 Bank Account based M-Payment

Banks have several million customers and telecommunication operators also have several million customers. If they both collaborate to provide an m-payment solution it is a win-win situation for both industries. In this model, the bank account is linked to the mobile phone number of the customer. When the customer makes an m-payment transaction with a merchant, the bank account of the customer is debited and the value is credited to the merchant account.

3.3.2 Credit Card based M-Payment

In the credit card based m-payment model, the credit card number is linked to the mobile phone number of the customer. When the customer makes an m-payment transaction with a merchant, the credit card is charged and the value is credited to the merchant account. Credit card based solutions have the limitation that it is heavily dependent on the level of penetration of credit cards in the country. In India, the number of credit card holders is 15 million [225]. Only this small segment of the population will benefit in the credit card based model. Though limited in scope, there may be high demand within this segment for a payment solution with credit cards and also, may provide high volumes of transactions.

3.3.3 Telecommunication Company Billing of M-Payments

Customers may make payment to merchants using his or her mobile phone and this may be charged to the mobile phone bills of the customer. The customer then settles the bill with the telecommunication company [226]. This may be further classified into prepaid airtime (debit) and postpaid subscription (credit).

3.4 Technologies for Mobile Payments

The mobile technology landscape provides various possibilities for implementing m-payments. Essentially, a GSM mobile phone may send or receive information (mobile data service) through
three possible channels: SMS, USSD or WAP/GPRS. The choice of the channel influences the way m-payment schemes are implemented. Secondly, the m-payment client application may reside on the phone or else it may reside in the subscriber identity module (SIM).

### 3.4.1 Short Message Service (SMS)

This is a text message service that enables short messages (140-160 characters) that can be transmitted from a mobile phone that are stored and forwarded by SMS centers. SMS messages have a channel of access to phone different from the voice channel [227]. SMS can be used to provide information about the status of ones account with the bank (informational) or can be used to transmit payment instructions from the phone (transactional).

### 3.4.2 Unstructured Supplementary Services Delivery (USSD)

Unstructured Supplementary Service Data (USSD) is a technology unique to GSM. It is a capability built into the GSM standard for support of transmitting information over the signaling channels of the GSM network. USSD provides session-based communication, enabling a variety of applications. USSD is session oriented transaction-oriented technology while SMS is a store-and-forward technology. Turnaround response times for interactive applications are shorter for USSD than SMS.

### 3.4.3 WAP/GPRS

General Packet Radio Service (GPRS) is a mobile data service available to GSM users. GPRS provides packet-switched data for GSM networks. GPRS enables services such as Wireless Application Protocol (WAP) access, Multimedia Messaging Service (MMS), and for Internet communication services such as email and World Wide Web access in mobile phones.
3.4.4 Phone-based Application (J2ME/BREW)

The client m-payment application can reside on the mobile phone of the customer. This application can be developed in Java (J2ME) for GSM mobile phones and in Binary Runtime Environment for Wireless (BREW) for CDMA mobile phones. Personalization of the phones can be done over the air (OTA).

3.4.5 SIM-based Application

The subscriber identity module (SIM) used in GSM mobile phones is a smart card i.e., it is a small chip with processing power (intelligence) and memory. The information in the SIM can be protected using cryptographic algorithms and keys. This makes SIM applications relatively more secure than client applications that reside on the mobile phone. Also, whenever the customer acquires a new handset only the SIM card needs to be moved (Card Technology, 2007). Incase the application is placed on the phone, a new handset has to be personalized again.

3.4.6 Near Field Communication (NFC)

NFC is the fusion of contactless smartcard (RFID) and a mobile phone. The mobile phone can be used as a contactless card. NFC enabled phones can act as RFID tags or readers. This creates opportunity to make innovative applications especially in ticketing and couponing (Ondrus and Pigneur, 2007). The Pay-Buy Mobile project launched by the GSM Association (fourteen mobile operators are part of the initiative) targets 900 million mobile users with a common global approach using NFC (Card Technology Today, 2007).

3.4.7 Dual Chip

Usually the m-payment application is integrated into the SIM card. Normally, SIM cards are purchased in bulk by telecom companies and customized for use before sale. If the m-payment application service provider has to write an m-payment application in the SIM card, this has to be done in collaboration with the telecommunications operator (the owner of the SIM). To avoid this,
dual chip phones have two slots one for a SIM card (telephony) and another for a payment chip card. Financial institutions prefer this approach as they can exercise full control over the chip and the mobile payment process[221], but, customers would have to invest in dual chip mobile devices.

3.4.8 Mobile Wallet

A m-payment application software that resides on the mobile phone with details of the customer bank account details or credit card information which allows the customer to make payments using the mobile phone. Customers can multi-home with several debit or credit payment instruments in a single wallet. With several implementations of wallets that are company-specific are in use globally.

3.5 A Generic Architecture for M-Payments

This is a simple, illustrative conceptual model that describes the relationship between the major participants in an m-payment scenario 3.1. There is the customer and the merchant who would like to use an m-payment service. The M-Payment Application Service Provider (MASP) provides the necessary technical infrastructure (hardware and software) to facilitate m-payments and acts as an intermediary between the financial institutions and mobile network operators. The MASP registers users who would like to avail of the m-payment service. The users (customers and merchants) have to be registered with the MASP prior to using the service. At the time of registration the MASP collects the bank account details (or credit card details) of the customer and merchant as well as their valid digital certificates. The mobile phone numbers of the customer and the merchant are mapped to their respective bank accounts and this mapping is maintained by the MASP. The users are provided with a client m-payment application (mobile wallet) that is either resident on their phones or else in the SIM card. This application may be provided over the air to the users. The mobile wallet will normally interact with the MASP server.
A mobile phone user communicates with a merchant and makes an economic transaction (e.g., buying a ticket from an airline over the phone). The merchant obtains the phone number of the customer and initiates the m-payment transaction request stating the amount for which payment is required. The customer confirms the request and authorizes payment. The MASP receives the authorization and verifies the authenticity of the customer. The MASP then debits the customer account and credits the merchant account by interacting with the bank. Once the electronic funds transfer is successful a confirmation message is sent to the customer and the merchant advising them of the debit and credit respectively. The Certifying Authority also shown in Fig. 1 supplies digital certificates for the users in the system to provide security. This model can be extended to handle the interaction between the MASP and the financial system taking into account inter-bank payments and settlement.
3.6 Security Issues

For widespread use and customer acceptance of m-payment services, both perceived and technical levels of security should be high. For customers, privacy should not be compromised and there should be no possibility of financial losses. For businesses, customer authentication is important. As per the general framework of any secure messaging system - confidentiality, integrity, non-repudiation and authentication should be guaranteed by the m-payment services [228]. The transport layer security offered by GSM/CDMA networks sufficiently guarantees confidentiality (that messages cannot be read by anyone else) and message integrity (the assurance that the message has not been altered in transit). Authentication (identifies the author of the transaction) and non-repudiation (makes sure that any of the users in the system cannot later deny the message they sent) can only be guaranteed with the help of wireless public key infrastructure (WPKI) and digital certificates. Secure mobile payment transactions can be implemented using existing national public key infrastructure, which is independent of financial institutions, mobile network operators and mobile payment application service providers but can be used by all of them. Their proposed technological solution to provide secure mobile payment transaction is briefly described below.

3.6.1 Public Key Infrastructure and SIM cards

Every user of the system is listed in a publicly available directory. A would like to send a message to another user B. A first obtains Bs public key from the directory and encrypts the message using it. Since only B has the private key only he can read the message (after decryption) and no one else and A can digitally sign the message. In this scheme anybody can verify that A did indeed send the message and the message was not altered during transmission. A Certification Authority (CA) maintains the publicly available directory, which is responsible for issuing and revoking digital certificates. A digital certificate contains the public key of a user in the system. This framework is known as public key infrastructure (PKI).

A user normally maintains his or her private key confidentially in a personal secure environment. SIM cards have the ability to store and process private keys. In terms of key management, there
must be an administrative system to issue key pairs to genuine citizens in a country.

3.6.2 Protocols

A sample protocol that describes the transaction between a customer and a merchant, each using his or her mobile phone and a m-payment application service provider as an intermediary is outlined in this section. It is assumed that customer and merchant are registered as users with the m-payment application service provider (with their respective bank account details) and both of them have valid digital certificates. The transactions are detailed below.

1. Service Request

Customer → Merchant

Customer makes a service request to the merchant

2. Product Options

Merchant → Customer:

Merchant sends his product options and his certificate

3. Product Selection

Customer → Merchant:

Customer selects a product; the selection is signed by the customers private key

4. Payment Request

Merchant → M-payment Application Solution Provider (MASP) Customer:

The payment request (containing the invoice amount) is signed using merchants private key. Customer can verify that the merchant is genuine by using his certificate (sent earlier in step 2). The MASP also authenticates the merchant before passing the payment request to the customer.
5. Payment Authorization
Customer → MASP: The customer authorizes the payment request by digitally signing the authorization using the customers private key. The MASP transfers the money from the buyers account to the sellers account by communicating to the bank(s).

6. Payment Confirmation
MASP → Customer:
MASP confirms payment made to merchant
MASP → Merchant:
MASP informs merchant of successful payment

The customer and the merchant can verify their respective bank accounts as to whether payment has been made. The Institute for Development and Research in Banking Technology (IDRBT) has an experimental, proofof-concept project where PKI enabled m-payment applications have been demonstrated to be feasible.

3.6.3 Stakeholders

There are many different stakeholders in the process of implementing mobile payments[221]. They are:

(a) Consumers
(b) Merchants
(c) Mobile Network operators
(d) Mobile device manufacturers
(e) Financial institutions and banks
(f) Software and technology providers
(g) Government

Each player has different incentives and strategies. Sometimes these interests and strategies between different players may be in conflict e.g., the telecommunications network provider would like to maximize revenues through each m-payment transaction whereas customers and merchants would like to minimize costs for each m-payment transaction. The expectations of each of the stakeholders are outlined below.

**Consumer Expectations**

- Personalized service
- Minimal learning curve
- Trust, privacy and security
- Ubiquitous anywhere, anytime and any currency
- Low or zero cost of usage
- Interoperability between different network operators, banks and devices
- Anonymity of payments like cash
- Person to person transfers

**Merchant**

- Faster transaction time
- Low or zero cost in using the system
- Integration with existing payment systems
- High security
- Being able to customize the service
• Real time status of the mobile payment service

Banks

• Network operator independent solutions
• Payment applications designed by the bank
• Exceptional branding opportunities for banks
• Better volumes in banking more card payments and less cash transactions
• Customer loyalty

Telecom Network Providers

• Generating new income by increase in traffic
• Increased Average Revenue Per User (ARPU) and reduced churn (increased loyalty)
• Become an attractive partner to content providers

Mobile Device Manufacturer

• Large market adoption with embedded mobile payment application
• Low time to market
• Increase in Average Revenue Per User (ARPU)

Government

• Revenue through taxation of m-payments
• Standards
3.6.4 Challenges for M-Payments

Standards
M-payments lack cohesive technology standards that can provide a universal mode of payment. Consolidation of standards in the mobile commerce arena is critical and it will enable producers and consumers to make investments that produce value. The lack of standards will give rise to a lot of local and fragmented versions of m-payments offered by different stakeholders (network operator centric models and bank centric models). Standards need to address security and privacy concerns of consumers as well as interoperability between various implementations. Standards formation is a process of negotiation between various stakeholders; it is rather more political negotiations in nature rather technical discussions. First movers benefit from this situation by creating de facto standards and major market share. There is no consensus among the players in terms of m-payments standards setting. Certain start up companies have proposed standards and they hope to make these de facto by being first movers with strategic advantage and early market selection. The battle over standards occurs at the firm level and at the inter-consortia level [224].

Business Models
Since there are several stakeholders in the system, a viable and sound business model needs to be developed that will provide a framework for revenue sharing.

Regulatory Issues
Although m-payments may allow parties to make economic exchanges, it is not legal tender in the sense it lacks the status of other payment instruments such as cash, which is a medium of exchange that is authorized, adopted and guaranteed by the government. At best m-payments will have to be backed by the issuers promise to pay. To overcome this problem legislation has to be put in place that will make m-payments legal tender. The regulations for players in the financial industry are different from those governing the telecommunications industry, which means that each industry has its own particular standards body to comply with.
3.6.5 Micro-Payment Systems Overview

Micro-payments refer to low value electronic transactions. They provide an alternative revenue source for content providers beyond advertising and subscriptions (W3C ’99). They may also provide revenue streams for service providers. Based on Thomas [230], the m-commerce, part of which are the micro-payment systems, will be worth £2.8bn by 2005. According to Pierce [231], Micro-payments involve:

- A buyer / client
- A vendor / data editor
- One or more brokers / intermediates / billing servers.

According to the World Wide Web Consortium (W3C) (W3C ’99) the requirements of micro-payment systems are:

- Embedding must be simple (click & pay interface)
- Embedding must minimize TCP/IP packet round trips
- Embedding must use standard browser features

The basic architecture of a micro-payment system consist of:

- On the client side:
  - A browser,
  - A module which is communicating with the micro-payment server
  - One or more electronic wallets
- On the vendor side:
  - An HTTP server
The following are the micro-payment implementations that were examined/analyzed: PayDirect, FirstVirtual, PayItMobile, CyberCoin, QPass, Millicent, Banxafe, Bibit, Earthport, Digicash, Internet Dollar, FirstGate, Pay2See, MicroMint, Genion, CyberCent, Mobipay, Paybox. The ePSO (electronic Payment Systems Observatory), managed by the European Commission/Joint Research Centre/IPTS, has been used for the identification of the above MPS. These systems represent the state of the art from an international point of view (20 different countries). Details for each system can be seen in appendix A. The biggest risk of such systems is that of user acceptance. The “modern e-consumer” [232] has a lot more requirements than a traditional consumer of the 90s. The system must be simple and easy to use and able to earn the users trust. Although some of the above systems are still operational, they are far from being successful, mostly due to the failure of understanding user preferences. In agreement with Shirky, [233], micro-payment systems, typically, treat cheap resources as precious commodities, while treating the users time as if it were so abundant as to be free.

3.6.6 Micro-Payment Systems Analysis

Five digital money systems were chosen for further study. These are: CyberCashs CyberCoin, DigiCashes eCash, Carnegie Mellons NetBill, Digitals Millicent, and METEORE 2000. These systems have provided leadership, creativity and understanding of the markets micro-payment needs. As it will be seen they provide different solutions to the micropayment transaction problems, hence they were chosen for this analysis. The subsequent sections characterize these schemes using the following criteria:

1. General Concept Description

2. Ease of Use

3. Anonymity & Privacy

4. Security

The general concepts involved in this process are as follows:
• **CyberCashs CyberCoin**: CyberCoin services are distributed through banks, which offers online merchants the CyberCoin service and offers consumers co-branded “Wallets”. The merchants pay the bank a per-transaction fee to use the system. This fee is a tier-based pricing model based on the transaction size.

• **DigiCashs eCash**: This system is based on what is called a single use token system. The user generates blinded electronic bank notes and sends them to his bank to be signed with his bank’s public key (PK). The bank signs the notes, deducts the amount from the user’s account, and sends the signed notes back to the user. The user removes the blinding factor and uses them to purchase at the shop. The shop verifies the authenticity of the bank notes using the bank’s corresponding public key and sends them to the bank where they are checked against a list of notes already spent. The amount is deposited into the shop’s account, the deposit confirmed, and the shop in turn sends out the goods. All communication over the network is protected by encryption. The system involves software for both the consumer and the merchant to conduct the transactions. The customer runs a ”wallet“ program. The user can spend the digital money at any shop accepting eCash, without the trouble of having to open an account there first, or having to transmit credit card numbers. Because the received eCash is the value involved with the transaction, shops can instantly provide the goods or services requested.

• **Carnegie Mellons NetBill**: Once a Client approves a purchase of a good that can be transferred through the Internet, a digitally signed request is sent to the Vendor. The Vendor computes a checksum of the goods, and sends the encrypted goods with a time stamp to the Client. The Clients software computes a checksum of the goods and then sends this, along with the accepted price, the product identifier, and the timestamp, back to the Vendor. After the goods have been received and verified, the Vendors software appends the decryption key and endorses it with the Clients digital signature. The merchant then sends this to the NetBill server, which acts as the Broker. The Broker verifies that the product identifiers, prices and checksums are all in agreement. If the customer has the necessary funds or credit in his account, the Broker debits the customer’s account and credits the merchant’s account, logs the transaction, and saves a copy of the decryption key. The Broker then returns to the Vendor a
digitally signed message containing an approval, or an error code indicating why the transaction failed. The Vendor forwards the Brokers reply and (if appropriate) the decryption key to the Client.

- **DECs Millicent**: Millicent is a decentralized micro-payment scheme, which is designed to allow payments as low as 1/10 of a cent. It uses a form of electronic currency, which is called scrip. It is designed to make the cost of committing a fraud, more than the value of the actual transaction. It uses symmetric encryption for all data transactions. The principal actors of the scheme are the Broker, the Customer and the Vendor. Fig 3.2.

![Diagram](image)

**Figure 3.2: Millicent’s Scheme**

The illustrations of the constituents involved are as follows:

**The Broker**: The Broker mediates between Vendors and Customers in order to simplify the tasks they perform. He acts like a bank and provides the electronic currency ("scrip") for the micro-payments. A Broker, after coming to a deal with the Vendor, can either generate his own valid "Vendor-specific“ scrip, or buy a large amount of scrip, from the Vendor, using a macro-payment system. The Broker is then selling the scrip to the customers via macropayment transactions. As it is seen, Brokers are just credit intermediates that buy huge amounts of scrip from the Vendors and sell large amounts of scrip to the Customers. During Customer purchases (either from Broker or
Vendor), no transactions between the Broker and the Vendors are taking place.

**The Customer:** The Customers buy scrip from the Brokers, using real money, via a macro-payment system. The amount should be sufficient to cover the transaction cost plus to produce financial gain for both the Broker and the Vendor (scrip is Vendor specific). The Customer can then use the scrip to perform micro-payment purchases. No transactions with real money are taking place in any given time between Customers and Vendors.

**The Vendor:** The Vendor is the "data bank". He supplies customers with data, services or both. He accepts his specific scrip as the only method of payment. The scrip was either generated by him (the Vendor) or by a licensed broker. Of course some validation and authentication is necessary to ensure that no double spending will take place. After that the Vendor can transmit the requested data back to the Customer, using a given encryption algorithm for avoiding fraudulent use.

According to DEC, initial tests of the Millicent implementation in Japan produced the following results:

- 14000 pieces of scrip can be produced per second
- 8000 payments can be validated per second.
- 1000 Millicent requests per second can be received from the network and validated.

**METEORE 2000:** METEORE is a unified Internet/mobile payment solution for contents and services, to be used in the so-called "Mobility Portals”. A mobility portal is defined as Web/WAP information based system, which provides information or services related to mobility:

- Information's related to a geographical position (which can be the position of the consumer or the one specified by him) or movement (how to go from a point to another one)
- Services like ticketing (entertainment, reservation, parking, etc.)
– Emergency services: reception of SMS signalling events (strikes or delays for travels, stock exchange conditions, etc.)

– Advertisement and advantages related to position or interest profile of the end-user.

A mobility portal has the major characteristics to address multiple terminals: fixed terminals like PC’s or mobile terminals like mobile phones or PDA’s. It also addresses multiple payment modes: aggregated and single payment. The following figure1 illustrates the high level architecture of the METEORE system.

![Diagram of METEORE system](image)

**Figure 3.3: METEORE scheme**

The client can access the sites of on-line sellers to buy coupons, which are stored in the CPS. The client can then buy e/m contents using these coupons, which the CPS authenticate with an intermediary bank. Alternatively the client can pre-pay the bank and create an account with MicroCOMM. The client can then use the CPS to buy e/m contents from online sellers, without dealing
with the bank at all. The core system architecture combines an authentication layer at the core payment system that connects to an aggregation engine and a single payment gateway that interfaces to an external payment system in charge of authorisation and money transfers. Other important functional blocks are:

- Web back-offices: merchant back-office, consumer front-office, system/application back-office, that are all implemented as https portals,

- The system interconnection block.

The Core Payment System offers both aggregated and single payment mode, the authentication depending from the terminal capability. The METEORE model does not specify how the back-offices and front-offices work but only state their existence. Each implementation will use its specific interfaces. Proxywallet is a payment access solution that interfaces to multiple banking systems and especially SSL bank intermediaries.

It can be used as a unique access point either for direct connections to central authorization / payment systems or to secondary access system like SSL intermediaries. ProxyWallet is used for the single payments. Micro-COMM is a typical third party aggregation system built for contents. It uses strong authentication of users accessing via PC to contents sites, through a security agent that wraps communication on http. Micro-COMM is used for the aggregated payments.

**Ease of Use**

This section examines the actions/steps a customer has to undertake in order to use the system, and how complex these steps are. The section was summarized in table 3.1.
Table 3.1: MPS Ease of Use

<table>
<thead>
<tr>
<th>Type</th>
<th>Relationships</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyber Coin</td>
<td>- Customer needs to establish a relationship with a bank that supports the CyberCash Wallet</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Merchant needs to establish relationship with the CyberCash system</td>
<td>- Debit card transactions</td>
</tr>
<tr>
<td>Ecash</td>
<td>- Customer needs to establish a relationship with a bank that supports the Digicash system.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Vendor needs to establish relationship with the Digicash system</td>
<td>- By transferring the digital money into the HDD of the consumer, communication with the bank is avoided at the time of a transaction. The vendor authenticates the digital cash during the transaction and then later transmits it to the issuer to check for double spending.</td>
</tr>
<tr>
<td>NetBill</td>
<td>- A NetBill server maintains accounts for both Clients and Vendors</td>
<td>- Client requests product</td>
</tr>
<tr>
<td></td>
<td>- Clients replenish funds in their NetBill account as necessary using a credit card or bank account</td>
<td>- Vendor sends an encrypted version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- After verification of transfer takes place, Vendor solicits Broker a funds transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Broker makes transfer and responds to Vendor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Vendor forwards decryption key to Client</td>
</tr>
</tbody>
</table>
| Millicent | - Client enter into long-term relationships with brokers  
|           | - Brokers buy and sell vendor scrip as a service to Clients and Vendors  
|           | - Once a Clients scrip is set at the vendor, transactions take place and the broker only acts as accountant keeping record and managing scrip  
|           | - Consumer purchase scrip from brokers using a bank account or a credit card.  
| METEORE 2000 | - Client has to register with a bank that is using the system  
|           | - Client has to register with the system.  
|           | - Client has to buy vendor specific coupons.  
|           | - Vendors have to register with system and bank.  
|           | - The client makes a secure connection to the broker to get some broker scrip  
|           | - If the client doesn’t already have scrip for a particular vendor, he contacts the broker to buy some using his broker scrip  
|           | - If the broker doesn’t already have scrip for that vendor, he buys some from the vendor  
|           | - The broker returns vendor scrip and change (in broker scrip) to the client  
|           | - The customer uses the vendor scrip to make a purchase from the vendor.  
|           | - Option for aggregated and single payment.  
|           | - Low cost contents can be bought using aggregated payments which minimise overheads, and normal cost contents can be bought using single payment.  
|           | - Setting up the system is easy and only requires an internet connection. Registering for using the system though is a complex and long-winded procedure.  
|           | - Once set-up, the system is easy to use and all complex procedures are hidden from the client.  

According to EU directives, micro-payment systems have to provide amongst other things anonymity and privacy for every single transaction. Both are not optional and are considered to be two of the
most important features of any system making transactions with, [234], Table 2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Anonymity</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CyberCoin</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Ecash</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>NetBill</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>METEORE 2000</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Millicent</td>
<td>Depends on Millicent protocol (see below)</td>
<td></td>
</tr>
</tbody>
</table>

- **CyberCoin**: The Client always has a record of his or her transactions and can prove them, but the Vendor will not know the Client's identity unless the Client reveals it.

- **ECash**: unlike paper cash, is unconditionally untraceable. The computations carried out by the Client's PC makes it impossible for anyone to link the payment to payer. Clients can prove that they did or did not make a payment, without revealing anything more. This level of privacy limits exposure to future data-privacy legislation and reduces record-keeping costs.

- **NetBill**: Client and account details are not shared with anyone except as required by law. Vendors are not provided with access to Clients proprietary information included in an electronic payment order. By design, the Vendor will never have access to the Clients bankcard number.

- **METEORE 2000**: The client does not reveal any personal details other that his/her digital signature, or phone number (that is registered with the system), and a password. The unique identifier of the client is his/her phone number.

- **Millicent**: three distinct protocols enable different levels of privacy (and security) according to customer's needs.
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Efficiency Ranking</th>
<th>Secure</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrip in the clear</td>
<td>1</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Encrypted connection</td>
<td>3</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Request signatures</td>
<td>2</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

- **Scrip in the clear**: No network security is provided. The scrip is not getting encrypted. The purchased data are not getting encrypted either.

- **Encrypted connection**: The scrip is getting encrypted using a symmetric algorithm. The encryption key uniquely identifies the Customer and the transaction. Data are getting encrypted as well.

- **Request signatures**: Digital signatures are being used for uniquely identifying the transaction and the customer. Faster than the previous protocol because no encryption is taking place.

The security analyses are as follows:

- **CyberCoin**: the financial information is encrypted and digitally signed, but the message itself is not.

- **eCash**: provides the highest security possible by applying public key digital signature techniques. Additional security features include the protection of eCash withdrawals from the Clients account with a password that is only known to him; not even to his bank.

- **NetBill**: uses a combination of public-key cryptography and a variant of symmetric-key cryptography to make sure that all its communications are secure, and all transactions are authorized. Their approach is based on the well-tested Kerberos protocol, [235].

- **Millicent**: Each transaction requires that the customer know the secret associated with the scrip. The protocol never sends the secret in the clear, so there is no risk due to eavesdropping. No piece of scrip can be reused, so a replay attack will fail. Each request is signed with the secret, so there is no way to intercept scrip and use the scrip to make a different request.
• *METEORE 2000*: All transactions are done in XML, are digitally signed using a PKI, and are using an encrypted (128bit key) SSL carrier. The network itself is secured with firewalls and NIDSs, following a no-man architecture. The administration is done remotely using SSL and X509 certificates, and the integrity is achieved using mirroring and backup techniques.

### 3.6.7 Micro-Payment Transactions

The broker is the party that provides and/or holds the electronic currency. The vendor is the party that provides the e/m contents. The following general transaction categories are taking place in the examined MPS:

1. *Broker & Vendor*: Must trust each other, that the Vendor should provide what the Broker has promised to the Client. The Broker pay the Vendor for the used scrip. Both parties must have the same security procedures and patterns, as a chain is as secure as its weakest link.

2. *Client & Broker*: The Broker must comply with standards holding personal information. Data security is of the essence as the client trust is what keeps Brokers in business. The transactions between these two players are macro-payment transactions. Broker has to apply relevant security solutions: Privacy infringement, Authentication, Repudiation, Integrity and Denial of service, providing the Client with valid scrip.

3. *Client & Vendor*: The Vendor must be able to authenticate and validate the scrip. The Vendor must always be able to provide data (denial of service, interception problem).

As with all systems we strive for confidentiality, integrity availability, and non-repudiation. These concepts constitute three of the four goals of information security, and are examined in this chapter.

1. *Confidentiality*: Assets must be accessible by authorized parties, hence:

   (a) Client’s’ private data must remain private in Brokers database (database security, transaction security).

   (b) Client’s’ scrip should not be "stolen" by third parties (transaction security, encryption issues).
(c) Vendor’s’ data are accessible by valid Clients only (authentication, repudiation).

2. **Integrity**: Assets can be modified by authorized parties and in authorized ways, hence:

   (a) Brokers must issue valid scrip (trust, forgery issues).

   (b) Scrip can only be generated by valid Brokers (authentication, repudiation).

   (c) Vendors data cannot be modified by third parties (database security, transaction security, encryption issues).

3. **Availability**: Assets must always be accessible to authorized parties, hence:

   (a) Broker must always be able to generate and supply scrip to clients (denial of service issues, software & hardware protection issues)

   (b) Vendor must always be able to supply data to valid clients (denial of service issues, software & hardware protection issues)

The above security requirements are in agreement with the logical security layer of on-line transactions that was presented by Hoogenboom and Steemers [236]. According to them:

- Both parties must be able to authenticate each other (mutual authentication).

- The integrity of the information exchanges must be verifiable (data integrity).

- The ordering party must have access to the desired service (service access).

- The confidentiality of the transaction must be guaranteed (information encryption).

- The transaction must be verified through at least two channel

### 3.6.7.1 Micro-Payment Systems Vulnerabilities

As with all computing systems, micro-payment systems have vulnerabilities that can be categorized by the following broad categories of their resources:

- Hardware
3.6.7.2 Hardware

All hardware components that are not necessary for the operation of the MPS should be removed. By hardware we mean floppy drives, CD-ROMs, parallel and serial ports as well as USB and infrared ports. Furthermore the unused data ports on the motherboard itself should be disabled or removed. The option of the wireless network is not an option as the security risk is so high that the cost involved for securing such a system does not justify it. All computers that participate in the MPS should be protected by a BIOS password. A member of staff though with physical access to the circuits of those computers can bypass this security feature by removing the battery of the motherboard for an approximate of 1 minute (as tested in our labs). Anything less and the voltage on the circuit itself will be able to maintain the CMOS data. The test in our lab showed that a hostile internal attacker (dissatisfied member of staff, member of staff employed by a competitor...) with a screwdriver as his only weapon only needs an approximate of 3 minutes to bypass the BIOS password.

After gaining access to the BIOS an attacker can choose to boot the computer from an alternative device, hence it is essential to have the absolute minimum hardware installed. He will even be able to install an alternative HDD and use it to retrieve data that otherwise it would be impossible for him to do. After an experiment in our lab, we where able to retrieve the private data of a Debian user, as well as all the sensitive configuration files of the operating system. The suggested countermeasure to the above vulnerability is to physically secure the room that the computers are in and physically lock their cases to the ground. Furthermore, a sophisticated lock should be installed between the cases of the computers and their covers to prevent access to the computer circuits by the use of a simple screwdriver. There have been a lot of discussions on network security and how
to better secure a corporate network. One of the common outcomes is the single point of entry. Should there is only one path for an attacker to come in the CPS then the ISec officers will have an easier job to do. That was not the case though to some of the MPS that was examined. Every single point of entry should have the same level of security. The weakest link destroys the game altogether. The two concepts that the ISec officers should be concerned with is the robustness of each machine participating in the MPS and the robustness of the connection between the servers and between the servers and the clients, [237], [238], [239], [240], [241], [242].

3.6.7.3 Software

Banners
All services and applications in general, identify themselves by giving away into the world "banners". The "banners" contain all the necessary information for an individual or a software program to recognise exactly the type of service, its version and some times set-up parameters as well. As it was discussed in other sections of this report, threat agents can use these "banners" to collect information for the system and then interrogate vulnerability databases in order to find ways to break them. Because of the nature of the systems it is best practice to put fraudulent information in those "banners" in order to mislead any potential attackers.

Services
Necessary services should be offered from each computer that is part of any computing system[238] [245] [239] [243]. Even if a computer is not directly connected to the Internet it has to be certain that no unnecessary networking services are offered and never have too many security layers installed in a system. Only secure and authorised computers should be part of the system. Firewalls, [240] [246] must be installed between the intranets and the Internet and also between the Intranets themselves. The firewalls must not only check the sessions but the packets as well offering the same level of security and protection. A portscanner should be used in a daily basis to monitor for unauthorised services.

Network Traffic
Cryptographic techniques should be used for all network traffic, both inside and outside the system [247] [248] [249]. The financial and personal data of the users, stored in the MPS, should also be encrypted. Such a countermeasure will prevent the modification of the data in the case of an intruder gaining access to the servers. The MPS should be a no-man zone. No user should be allowed to run any sort of programs on the network other than the ones dedicated to maintenance.

DNS
Some MPS are using the DNS service to simplify their operations. The question is what computers are contained in the DNS database and which computers are running the service. The DNS should not be public in order to minimize the possibility of an attacker gaining information about the private machines running the MPS. Furthermore, there should be a backup DNS for robustness purposes. Will that server be allowed to perform zone transfers? If yes, this poses a major vulnerability that needs to be countered at any costs. Zone transfers must be restricted to authorized servers only. The "allow-transfer" directive in the "named.conf" file must be used to enforce the above restriction. Furthermore, the firewall must be configured to deny all inbound connections to TCP port 53. One common implementation of the DNS is BIND, which was used by most MPS implementations. The first step in setting up BIND is the creation of the DNS data. The configuration details are spread over multiple files. For reference purposes the file that maps host names to addresses will be called db.wallet and the file that maps addresses to host names will be called db.addr. Two more configuration files are the db.cache and the db.127.0.0. To tie all the files together a startup file is required. The file is normally called named.boot and it is stored in the /etc directory. Needless to say that the permissions for this file should be the same as for the /etc directory: -rw-r- -r- -. For maintenance purposes at least one user should have to write permissions, but actions should be audited, as an unauthorized entry to this file, even as unintentional, will cause a denial of service.

It is a norm for the entries in the db files to have the following structure: SOA record, NS record, other records. The SOA stands for start of authority and indicates the best source of information for a zone. There can be only one such entry, as there can be only one such machine in each zone.
The NS stands for name server and contains the records for the name servers of the zone. HINFO records must NOT be used in the DNS database as they contain very sensitive information that does not offer any sort of functionality other than making DNS administration easier. One of the most popular attacks against domain name servers is the zone transfer. A legitimate zone transfer allows a secondary master server to update its configuration files from the primary master DNS. Only a secondary master DNS should be able to issue a zone transfer request. In our system then, only the secondary firewall must be authorized to interrogate the primary one and ask for a zone transfer.

IDS
There is a variety of IDSs used by the examined MPSs. SNORT is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can produce documentation "like there is no tomorrow". Because the MPS is handling financial transactions, it is believed that legal advice is required to ensure that the output of SNORT will be able to stand in a court of law. SNORT should log all traffic to and from the open ports. The best way of stopping a future attack is by identifying it when it is on the "information gathering" stage. By closely monitoring ICMP and nslookup requests, and by detecting ports scans with the relevant SNORT preprocessor, attackers can be identified and legal actions can be taken. Another preprocessor that should be used is the "Frag2". "Frag2" allows SNORT to perform a full-blown IP de-fragmentation, making it more difficult for hackers to circumvent the detection capabilities of the system. For TCP stream monitoring the "Stream4" preprocessor should be used. TCP streams on the configured ports with small segments will be reassembled and properly evaluated for malicious activity. "Stream4" is able to handle 64,000 simultaneous TCP connections. To close this section, SNORT rules must be nested. Because there is a need for speed and efficiency, complicated and slow contents rules should only by called when something suspicious has been identified by using other faster connection rules. A new administrator should only require a couple of minutes to understand and follow the rules "out of the blue.

Firewalls
Netfilter is a packet-filtering gateway, which uses ipchains/iptables rules for effectively checking
every single packet that comes and goes from the configured ports. First of all the firewalls should not reply to echo requests nor to any type of portscans. It has been observed that once an attacker will identify a firewall it will try to work around it and not through it. Hence, by making it difficult for an attacker to identify firewall machines, we earn more time for allowing the IDS system to successfully identify the attack. Another countermeasure that could be deployed is against route tracing. We do that by restricting firewalls from responding to TTL expired packets. ICMP and UDP tunneling should also be monitored and trapped. ICMP tunneling is the capability of wrapping real data in an ICMP header. ICMP traffic should not be blindly allowed through the firewall. By limiting ICMP traffic through the firewall we also secure our system against a lot of denial of service attacks. By not allowing broadcast ICMP echo requests we effectively protect the system from smurf attacks. Fragments must also be checked on the firewalls. Sanity checking on the fragmentation length must be performed (both for being too small or too large). Carefully constructed packets send to the firewall machines of the MPS would result on a system reboot or a system halt.

**CGI Security**

Some of the MPS were using CGIs for communication purposes. There are many ways to prevent CGI attacks and the easiest and most sensible is to install the web server correctly from the start. Vulnerabilities in the web program should be checked with Mitre.org. The operating system itself must of course be brought up to date with the latest patches. All web CGI programs must be checked for insecurities via scanners to see if they are tempting targets for hackers. According to [253] [254] the developers must take under consideration the following:

- Always include error-handling code to warn if the file isnt actually a file, cannot be created or opened, already exists, doesnt exist, requires different permissions, and so on.

- Watch which directories are used to create or open files. Never write a file to a world-writeable or world-readable directory.

- Always explicitly set the files UMASK.

- Set the file permissions as restrictively as possible. If the file is a dump of user input, such as a visitor list, the file should be readable only by the processes that will engage that file.
• Ensure that the files name does not have metacharacters in it, and if the file is generated on
  the fly, include a screening process to weed out such characters.

• Never run the web servers with root privileges.

• Delete scripts that are not in use to prevent vulnerability from being exploited.

Utilize CGIWrap written by Nathan Neulinger to allow general users to use CGI scripts and HTML
forms without compromising the security of the http server. Scripts should not be using the user ID
of the httpd process. Also, several security checks are performed by CGIWrap on the script, which
will not be executed if any checks fail. Network Reconnaissance Countermeasures SNORT [255]
[256] [257] is one of the best available sources (both open and close) for detecting reconnaissance
activity. The above tool can do nothing though for preventing a hostile party from gathering sensi-
tive information from a system. According to [242] the tool that should be used for that is called
“RotoRouter”3 and counters the above vulnerability by generating fake “traceroute” responses.

**Spoofing Countermeasures**

No two users must be allowed in the same computer in the same time. We must keep in mind
that new vulnerabilities are surfacing every day. By not allowing the above make sure that even as
the computer suffers from uncovered vulnerabilities, no one will be able to exploit them using a
spoofing attack.

**Ping Sweeps Countermeasures**

Ping sweeps are the first step of an active attack. Being able to detect them and find out their
source will greatly help in efficiently protecting the network. The primary method for detecting
ping sweep attacks is network based IDS programs such as SNORT.

**SYN Flooding Countermeasures**

Since 1996 the SYN flood has made news with attacks against the FBI, the White House, NYSE
web sites, along with many others. The use of random spoofed source IP numbers on the IP head-
ers made tracing the attacks back to the source nearly impossible. Hackers developed Zombie”
hosts to attack targets in unison from a master machine. The technique, now famous, is referred to as the distributed denial of service attack. It uses many machines to coordinate attacks and in many cases without the owners knowledge or consent. The result is the same. The target computer fills a very small queue of half open port connections. Once the queue (also called a Backlog) is full, no other connections can be made until a connection times out and the memory space cleared, or the connection is complemented, which moves the connection out of the queue into an application level memory buffer. The computer stops answering requests on the attacked port once the Backlog threshold is reached. In the case of web servers, the attacks are aimed at ports 80 or 443. Once overwhelmed, the Web server stops working as designed and the attack is successful. The defense against a SYN flood DoS attack must start at an organizations boarder router and continue through the network all the way to the target host. Boarder routers of organizations usually represent the boundary between an organizations line of control and responsibility and the wild world. Boarder routers represent the first organizational network device a packet travels through on the way to the target host. A limit of half open connections that can be established through that router must be set. Once the limit is crossed, the router will begin to drop connections at a specified rate. The aggressive interception value is usually half of the normal network threshold of half connections. The next level of defense comes form the firewall machines. Proxy firewalls are the best choice as they can detect the states of the connections and wait for them to be full before forwarding them to the actual hosts. The iptables firewall though is a packet firewall and cannot detect the states of the connections. A change is not recommended for the trial or for a small-scale implementation in a single geographical area. Should the consortium decides to go live with a large scale implementation of the system over a number of geographical areas, then an investigation for justifying the expense of a proxy firewall must be conducted.

**SQL Injection Countermeasures**

Most MPS are using some version of SQL to hold data. The following list present defenses against the SQL injection attack:

- Full XML source code audit
  - Review all SQL queries and all variables in them
(ii) Trace SQL queries and look for user data

- Take advantage of stored procedures
- Minimize privileges of database connections
- Disable verbose error messages.

**Data Vulnerabilities**

**Key Management**

- The system itself should manage all the sensitive cryptographic elements such as keys and algorithms.
- Customers and users should not be able to read or modify sensitive cryptographic elements.
- The system should not serve user and customer requests that are not secure and encrypted by the default ciphers with authorised keys.

"Bruteforce" attacks are common and the system should be able to effectively counter them. Calculating the complexity of a bruteforce attack is relatively easy. Incase the key is 56bits long there are 256 possible keys, assuming that a supercomputer can try a million keys per second, it will take 2285 years to find the correct key. There is always a 50% chance, after trying half of them, to accidentally find a match. Incase the key is 128 bits long, the same supercomputer would require 1025 years. A threat agent will not spend millions of Euros and can spend a couple of thousand to bribe a user to get his keys. With an initial question that arises is if that customers have knowledge of what their key is provided the customers will be prone to social engineering attacks. The lifetime of the key should be very short.

Another concern is that only active customers should have access to the system and all the dormant accounts should be getting deleted in a regular basis.
Verifying Keys
The third issue is the key verification. The problem in the computing world has the name of: man-in-the-middle. This method of attack is discussed in another section of this report. Public key cryptography used with digital signatures and key distribution centres is considered to be the best solution. The capabilities required for substituting a key are over those of the category of hackers, crackers, and script-kiddies. Of course organised crime organisations, and rival companies have both the resources and the money to do so. To counter this vulnerability the system could be using customer certificates. In order for the certificates to work though, we must ensure that they are not jeopardized. The only 100% safe way to achieve the above is to send the certificates to the customers using floppy disks via recorded delivery. Crime organizations though can influence almost everything. An even safer scenario would be the use of the branches of the bank that the customer is registered. Most MPS actively involve one or more banking organizations. Because the system is dealing with electronic money the fact that the customer should have to go and collect something from his bank should not be a drawback. More of the point, customers are used in collecting their credit cards and their pin numbers form their banks so they will view it as another safety trigger against possible misuse.

Password Management
According to the NSA guidelines, [244], the first password for the user of the system and the first PIN for the customer should be temporary, in order to give them sufficient access to the system for changing them. The system must automatically ensure that the change must happen in a fixed time period after the receipt of the first password or PIN. In case that period expires before the connection of the customer, the customer should only have the option of a new temporary PIN (for a fixed period of time). The system must keep a history of passwords and PINs and prevent the reuse of the same ones [259]. On top of that, the system must fix the minimum life of the password and PIN as well as the maximum to prevent users changing their passwords until they can reuse their favourite one. The minimum length of the password and PIN for the users and the customers should be eight characters. The administrators should not use their login name in any form, as a password. Furthermore they should not use their first or second name or even any nicknames that they might have. Passwords should not be words contained in English or other dictionaries,
spelling lists or other lists of words. Passwords should not be based on information that are easily obtained from electronic or other sources. Passwords should be a combination of alphanumeric and punctuation characters. To sum up this section, the following three conditions must be fixed:

**Log Files**

All actions should be monitored, especially on the servers of the MPS. It is important to keep all log data in such a way that they will be able to stand in a court of law. As it was discussed, user trust is essential for the survivability of an MPS. That means that in the possibility of a breach, the system will be able to trace it back, find the responsible entity and legally prosecute it. Only authenticated users should be able to view the log files and perform authenticated actions [244] [239] [242]. Furthermore, no one should have permissions to amend the log files in any way. The log files should be backed up daily and the back-ups should be treated in the same way as the log files. Because different stakeholders host different parts of the system there is a need for the security team to be able to cross reference all of the log files. A tool should be developed in order to effectively solve the issue of data fusion and data mining. This issue must be addressed in the information security policy document.

**Administrative Vulnerabilities**

Information Security Policy Document

A policy document should be approved by the stakeholders of the MPS, published and communicated to all the partners accordingly. According to ISO17799 [260] the following guidance should be included as a minimum:

- A definition of information security
- A statement of management intent
- Explanation of the security policies that are important to the organization, for example:
  - Compliance with legislative and contractual requirements
  - Security education requirements
  - Prevention and detection of malicious software
– Consequences of security policy violations
– References to supporting documentation

Having a number of alarms and countermeasures deployed in the system for ensuring its safe operation is considered to be best practice. The alarms though can only inform about malicious actions against the system, they cannot prevent them. It is important for the administrators of the system to know exactly the procedures that they will have to follow for each case of an alarm switching on. As Schneier [261] is saying: “Real security is about people.” All the alarms and countermeasures should be specified and analysed in the security policy document. Because different stakeholders are involved in an MPS, the security policy document should comply with the security policy document of each one of them.

The security policy document should take under consideration at least the following two security standards: ISO17799 [260] & ISO15408 [262]. According to the number of European countries the system will be deployed different laws and legislations will probably apply. Legal advice for each country is required. The security policy document should not be made publicly available, as it will contain sensitive information about the security procedures and countermeasures of the METEORE system. Instead a second document containing the management views and support as well as a small overview of the whole document should be made ready for any customer requests. According to our research the major asset and major vulnerability in the same time, of such systems is the users trust, [263]. The users must know that there is a policy document that covers all the possible security concerns the system will have to face during its operation. The employees that will develop the security policy document must be internal full time employees bound by non-disclosure agreements and not external contractors. As per NSA guidelines, information security is a business responsibility shared by all members of the management team [244]. There is a need to identify the security responsibilities of each one of them and include them in the information security policy document. The above is compliant with the two security standards we are concerned with: the ISO17799 [260] and the ISO15408 [262].
3.7 Conclusion

This chapter discusses about the security considerations in mobile commerce system that is a major application domain for mobile devices enabling users to perform commercial transactions wherever they go. The newness of this area and the rapidness with which it is Emerging makes it difficult to analyze the technological problems that m-commerce introduces and in particular, the security and privacy issues.