Secure electronic payments for Islamic finance

Mansour Al-Meaither

Technical Report
RHUL–MA–2004–5
15 December 2004

Royal Holloway
University of London

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
`http://www.rhul.ac.uk/mathematics/techreports`

Secure electronic payments for Islamic finance

by

Mansour Al-Meaither

Thesis submitted to the University of London
for the degree of Doctor of Philosophy

Department of Mathematics
Royal Holloway, University of London

2004

Declaration

These doctoral studies were conducted under the supervision of Chris Mitchell and Peter Wild.

The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, whilst enrolled in the Department of Mathematics as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Mansour Al-Meaither
8 July 2004

Acknowledgements

In submitting the thesis, I have completed an important part of my life, one in which I had the fortune to interact with several great people at Royal Holloway, who provided me with knowledge, insight, and perseverance.

I am deeply obliged to my supervisor Professor Chris Mitchell for his continued support, interest, patience and guidance throughout my Doctoral Studies at Royal Holloway. His style of supervision along with his important comments and suggestions for improvements have encouraged me to pursue my ideas. Without his encouragement and support this thesis would be of much lesser quality. Special thanks also to my advisor Peter Wild for his support.

I also wish to thank Sami Al-Suwailem from the Center for Research and Development of Al-Rajhi Banking and Investment Corporation, for discussions that helped shape some of the ideas in this thesis.

To all my friends in Egham, Reading, and London for sharing wonderful moments and for making me feel at home.

I especially want to thank my parents. Over the years they have continued to support and motivate me in my studies.

Finally, I give special recognition to my wife Amal, for her understanding, endless support and encouragement over the last few years.

List of Publications

A number of papers resulting from this work have been presented in refereed conferences.

- M.A. Al-Meaither, and C.J. Mitchell. A person-to-person Internet payment system. In *Proceedings of the 6th Nordic Workshop on Secure IT Systems*, Lyngby, Denmark, November 2001, pages 5–17.

- M.A. Al-Meaither, and C.J. Mitchell. A secure electronic Murabaha transaction. In *Proceedings of the 16th Bled eCommerce Conference*, Bled, Slovenia, June 2003, University of Maribor, pages 662–674.

- M.A. Al-Meaither, and C.J. Mitchell. A secure electronic payment scheme for charity donations. In *Proceedings of EC-Web 2003, 4th International Conference E-Commerce and Web Technologies*, Prague, Czech Republic, September 2003, Springer-Verlag (LNCS 2738), Berlin (2003), pages 50–61.

- M.A. Al-Meaither, and C.J. Mitchell. Extending EMV to support Murabaha transactions. In *Proceedings of the Seventh Nordic Workshop on Secure IT Systems*, Gjovik University College, Norway, October 2003, pages 95–108.

- M.A. Al-Meaither, and C.J. Mitchell. A secure GSM-based Murabaha transaction. In *Proceedings of the 1st International Conference on Information and Communication Technologies from Theory to Applications*, Damascus, Syria, April 2004, IEEE Press, pages 77–78.

Abstract

Secure electronic payment systems are of paramount importance in supporting the further development of electronic commerce. While an electronic payment system must meet the needs of both businesses and consumers, most of the current electronic payment schemes are based on the traditional methods of finance we are familiar with in the western world. The main aim of this thesis is to develop new secure electronic payment schemes that satisfy the requirements posed by Islamic finance principles, which forbid the payment or receipt of interest.

After providing a generic model for an electronic payment system, a description of some of the properties that distinguish the various types of electronic payment systems is given. The thesis then reviews examples of electronic payment schemes that are relevant to this thesis. The main concepts underlying Islamic finance are also introduced.

The main contribution of this thesis is to propose four protocols that can be used to conduct secure electronic commerce transactions in a way that is consistent with Islamic financial principles. In the theme of developing new schemes to enable new participants to benefit from electronic payments, we also propose a simple and secure interpersonal payment system.

EMV compliant IC cards have been developed to secure traditional Point of Sale debit/credit transactions. In this thesis, we propose a way to use EMV-compliant cards to conduct an electronic Murabaha transaction with the goal of exploiting the widespread deployment of EMV cards.

The Internet is the platform on which most electronic commerce transactions are performed. To build upon this base, this thesis presents a method for conducting a secure electronic Murabaha transaction using the Internet.

The increase in ownership of mobile phones suggests that they can be an effective means of authorising payment in electronic commerce transactions, offering security and convenience advantages by comparison with on-line payments conducted using PCs only. Therefore, this thesis proposes a new GSM-based payment system that enhances the security of Internet Murabaha transactions.

Although many charities have a web presence, almost all of them have been designed to accept credit cards as the only means for making donations. The anonymity requirements of many donors, however, make the existing means of donation inappropriate for them. A new scheme supporting anonymous donations and distribution of these donations is therefore proposed.

Abbreviations

3-D	Three Domain
3GPP	Third Generation Partnership Project
AAC	Application Authentication Crypogram
AAV	Accountholder Authentication Value
AC	Application Cryptogram
ACH	Automatic Clearing House
ACS	Access Control Server
AHS	Authentication History Server
APK	Anonymous Public Key
ARPC	Authorisation Response Cryptogram
ARQC	Authorisation Request Cryptogram
AUTN	Authentication Token
B	Buyer
B2B	Business-to-Business
B2C	Business-to-Consumer
C	Charity
C2B	Consumer-to-Business
C2C	Consumer-to-Consumer
CA	Certification Authority
CAVV	Cardholder Authentication Verification Value
CK	Cipher Key
CSBLK	Certified Seller Block

D	Donor
DDA	Dynamic Data Authentication
E-Commerce	Electronic Commerce
EMV	Europay MasterCard Visa
ETSI	European Telecommunications Standards Institute
FSTC	Financial Services Technology Consortium
GSM	Global System for Mobile Communications
HLR	Home Location Register
HTTP	Hyper Text Transfer Protocol
IBAN	International Bank Account Number
IC	Integrated Circuit
IK	Integrity Key
IMSI	International Mobile Subscriber Identity
M	Merchant
MAC	Message Authentication Code
ME	Mobile Equipment
MN	Mobile Number
P	Provider
PAN	Primary Account Number
PC	Personal Computer
PG	Payment Gateway
PI	Purchase Information
PIN	Personal Identification Number
PKI	Public Key Infrastructure
POS	Point of Sale
PS	Pseudonym server
R	Recipient
SBLK	Seller Block
SDA	Static Data Authentication

SET	Secure Electronic Transaction
SEMT	Secure Electronic Murabaha Transaction
SIM	Subscriber Identity Module
SMS	Short Message Service
SMSC	Short Message Service Centre
SSL	Secure Sockets Layer
TC	Transaction Certificate
TLS	Transport Layer Security
UCAF	Universal Cardholder Field
UMTS	Universal Mobile Telecommunication System
URL	Uniform Resource Locator
USIM	User Services Identity Module
VLR	Visitor Location Register

List of Figures

Contents

Chapter 1

INTRODUCTION

Contents

This chapter describes the context of the research in this thesis, its contribution to the field of secure electronic payments, and presents the overall structure of this thesis.

1.1 Motivation

Commerce is a basic economic activity that involves the buying and selling of goods. As commerce became more complicated and inconvenient, humans invented abstract representations of value. Through history, physical means of payment have evolved from coins made of gold and silver to bank notes, payment orders, cheques, and credit cards.

Over the last decade, there has been a massive move from conventional to electronic transactions. A powerful force in this transformation has been the emergence of the Internet as a medium for conducting electronic commerce (e-commerce). Many products are now promoted and sold over the Internet.

Motivated by the convenience that e-commerce provides for consumers, merchants, and financial institutions, transactions have evolved to become a fully electronic process. Advances in computers, networking technology, mobile networks, and smart cards have enabled new carriers for consumer financial data to emerge. These include magnetic stripe cards, chip cards, PCs, and mobile phones. Meanwhile, electronic payment infrastructures have emerged connecting financial institution computers to merchant Point Of Sale (POS) terminals to support conventional face-to-face transactions.

As is the case with physical payments, many forms of electronic payment have been proposed. These include electronic money, electronic cheque, chip card transactions, and remote debit/credit card transactions.

Security has always been very important when dealing with money, and a substantial effort has been invested to make creating false money very difficult and costly. Unlike physical means of payment, electronic payments systems do not require that the parties to the transaction be physically present at the same location. Therefore the opportunities for transaction participants or unauthorised third parties to cheat are increased in an electronic payment scheme. Perhaps the simplest attack on an electronic payment scheme is for a third party to use stolen account details to conduct a fraudulent transaction. Thus, where user authentication is not enforced (as is typically the case for current e-commerce transactions) electronic payment transaction data need to be protected against disclosure to potential adversaries. Also, electronic payment transaction data need to be verifiably authentic and uncorrupted, lest false or modified data be used to perform a fraudulent transaction. It is also important that, once a transaction has been made, neither a merchant nor a customer can deny receiving or making a payment. The issue of security is thus a particular concern for electronic payment schemes.

For a Muslim, many of the current electronic payment schemes can be used as long they do not involve paying interest. However, when a Muslim wishes to buy goods on credit, then a conventional credit card transaction will not be acceptable since it involves paying interest. Muslims are required by their religion not to pay or receive interest on financial transactions. To address this issue, some financial institutions have offered other financial instruments that are consistent with Islamic law; an example of such an instrument is the Murabaha sale.

Despite the development of many electronic payment schemes, no work appears to have been done to enable Islamic financial instruments to be performed electronically. This motivates research on the design of electronic equivalents of Islamic financial instruments, such as means for conducting electronic Murabaha transactions at the point of sale and via the Internet. Apart from meeting Islamic requirements, such protocols clearly also need to be convenient and secure to use by the participants.

1.2 Main contributions

This thesis proposes a variety of new secure electronic payment schemes designed to enable parties currently excluded from e-commerce to enjoy the benefits it brings. The main contributions of this thesis are as follows.

- Three novel electronic payment protocols have been proposed that satisfy Islamic finance principles.
- A scheme is proposed that allows secure electronic charitable donations and distribution using the Internet and smart cards.
- A protocol is described that enables person-to-person Internet payments.

While EMV debit card transactions are consistent with Islamic finance principles, since the payment is cleared immediately without payment of interest, EMV credit card transactions are not, since they involve dealing in interest. To address this issue, a way to use EMV-compliant cards to conduct a face-to-face Murabaha transaction is proposed (chapter 5). The scheme only requires users to possess an EMV-compliant debit/credit card. Necessary changes to the standard EMV payment transaction are described.

The emergence of the Internet has led to the development of a variety of electronic payment protocols. However, many of these schemes are based on the credit card transaction model. Therefore, we propose a scheme for conducting a secure electronic Murabaha transaction using the Internet (chapter 6). The scheme builds upon the SET protocol.

The increase in ownership of mobile phones suggests that they can be an effective means of authorising payment in electronic commerce transactions, offering security and convenience advantages by comparison with on-line payments conducted using PCs only. Therefore, in chapter 7 we propose a new GSM-based payment system that enables Internet Murabaha transactions to be performed securely. By comparison with the scheme proposed in chapter 6, this scheme has the advantage of supporting user mobility.

Debit/credit cards provide a simple way for individuals to pay businesses for products and services, but they do not provide private individuals with the means to make payments to one another. There do exist schemes by which a payer can make a payment to anyone with an e-mail address. However, these schemes require both the payer and the payee to move money from their bank accounts to their account with the scheme provider in order to benefit from the service. Moreover, the payment instruction does not contain any information that links the goods being sold to the payment itself. Therefore, these schemes are not particularly appropriate for the case where a payment is made in exchange for goods. In chapter 8 we propose a new person-to-person Internet payment system that uses the existing relationships between buyers, sellers, and their respective banks to perform remote payments, and that links payment details to the goods being sold.

Finally although many charities have a web presence, almost all of these web sites have been designed to accept credit cards as the only means for making donations. The anonymity requirements of many donors, however, make the existing means of donation inappropriate for them. In chapter 9 we therefore propose a scheme that uses an anonymous electronic cash technique to make donations, and that employs smart cards for donation distribution.

1.3 Structure of the thesis

This thesis is organised as follows.

Chapter 2 is a preliminary chapter that introduces the security services and mechanisms used throughout this thesis.

Chapter 3 provides a generic model for an electronic payment system, followed by a description of some of the properties that distinguish the various types of electronic payment systems. A review of examples of electronic payment schemes that are relevant to this thesis is then given.

Chapter 4 introduces the main concepts underlying Islamic finance. We start by briefly describing the relevant aspects of the religion of Islam. The view of Islam regarding the payment of interest is presented. Islamic banking as an alternative to conventional banking is described, followed by a description of the notion of a Murabaha sale. Finally, we describe the importance of charity to Muslims.

Chapter 5 proposes a way of using EMV cards to conduct secure Murabaha transactions at the point of sale. An overview of the EMV card payment transaction procedure is first given. This is followed by an analysis of security and Murabaha requirements and a description of a possible modification to the EMV specifications that allows an EMV card to be used to conduct a Murabaha sale transaction. An analysis of how the proposed extension to EMV matches the identified requirements is also given.

Chapter 6 proposes a method to conduct secure electronic Murabaha transactions on the Internet. First, a model for a secure electronic Murabaha transaction is described. We then identify the security and Murabaha requirements for an electronic Murabaha transaction, and present the Secure Electronic Murabaha Transaction (SEMT) scheme, designed to address the identified requirements. Finally, we analyse how the proposed protocol matches the identified requirements.

In chapter 7 a way of using the GSM security services to conduct a secure electronic Murabaha e-commerce transaction is proposed. An overview of the wireless infrastructure and services which are relevant to the proposed protocol is first given. A model of a GSM-based electronic payment system is then discussed. This is followed by a description of a method for mobile secure electronic Murabaha transactions using a combination of the Internet, a mobile phone, and a hash-chain scheme related to S/KEY. A security analysis of the proposed protocol is then given. Finally, a description of how the protocol can be extended to make use of the security features of UMTS instead of GSM is presented.

Chapter 8 proposes a new Internet person-to-person payment system. We start by describing a general model for interpersonal Internet payments. Security requirements are then identified for a person-to-person Internet payment system. A payment protocol designed to address the identified security requirements is proposed. Finally, we analyse how the proposed protocol matches the identified security requirements.

Chapter 9 proposes a scheme that supports secure anonymous electronic charity donations. First, a model for a secure electronic charity donation scheme is described. We then identify the security requirements such a scheme should fulfill, and propose a scheme that uses an anonymous electronic cash technique to make donations, and that employs smart cards for donation distribution. Finally, we analyse how the proposed scheme matches the identified security requirements.

Finally, chapter 10 gives the conclusions of this thesis.

Chapter 2

SECURITY SERVICES AND MECHANISMS

Contents

This chapter introduces the security threats, services, and mechanisms used throughout this thesis.

2.1 Introduction

In this chapter we briefly describe the generic security threats and the basic security services (section 2.2 and 2.3) relevant to this thesis, as well as the mechanisms (section 2.4) which can provide these security services. For a more thorough introduction to all the necessary cryptography see, for example, [66].

2.2 Security threats

A security threat is a potential violation of security of a system. The following generic security threats can be identified within a networked information system [58].

- Impersonation, whereby an attacker pretends to be an authorised entity which is entitled to participate in a transaction.
- Eavesdropping, whereby an attacker monitors network activity to obtain sensitive information.
- Data replay, where a previous message, in its entirety or in part, is retransmitted after its recording.
- Manipulation of the content of exchanged data by substitution, insertion, deletion, or reorganisation.
- Denial of service, i.e. preventing or delaying access to resources by authorised users, by the prevention or interruption of communications. Denial of service attacks may be caused deliberately or accidentally.
- Repudiation or denial of participation in part or in all of a communications session by an entity.

2.3 Security services

To counter the threats identified in section 2.2, we identify four main security services which are of importance in this thesis. They are authentication, confidentiality, integrity, and non-repudiation. The following definitions are based on those given in [50].

2.3.1 Authentication

This service can be sub-divided into the following:

- *Entity authentication*, which provides assurance to one party regarding the identity of a second party involved in a protocol, and that the second has actually participated. This service provides confidence at the establishment of (or during) a communications connection that an entity has the identity claimed, and that the current communications session is not a replay of a previous connection.
- *Data origin authentication*, which provides to a party which receives a message, assurance of the identity of the party which originated the message. However, it does not, in itself, provide protection against duplication or modification of data. These latter properties are issues addressed by a data integrity service.

2.3.2 Confidentiality

This service provides protection for data against unauthorised disclosure. It can be sub-divided into the following:

- *Connection confidentiality*, which provides confidentiality for all user data transferred using a connection,

- *Connectionless confidentiality*, which provides confidentiality for all user data transferred in a single connectionless data unit, i.e. a data packet,
- *Selective field confidentiality*, which provides confidentiality for selected fields within user data transferred in either a connection or a single connectionless data unit, and
- *Traffic flow confidentiality*, which provides confidentiality for information which might be derived from observation of traffic flows, e.g. the time at which data is sent, the volumes of data sent to, or received by, particular recipients, or the length of individual messages.

2.3.3 Integrity

This service provides protection for data against active threats to the validity of transferred data. It can be sub-divided into the following:

- *Connection integrity with recovery*, which provides integrity protection for all the user data transferred using a connection, and detects any modification, insertion, deletion or replay of data within an entire data unit sequence. The term 'with recovery' means that, if some form of modification is detected, then the service attempts to recover the correct data, typically by requesting the data to be re-sent.
- *Connection integrity without recovery*, which is the same as previously, but with no recovery attempted if an integrity failure is detected.
- *Selective field connection integrity*, which provides integrity protection for selected fields within the user data, or within a data unit transferred over a connection.
- *Connectionless integrity*, which provides integrity assurance to the recipient of a data unit. More specifically, it enables the recipient of a connectionless data unit to determine whether that data unit has been modified. Additionally, a limited form of replay detection may be provided.

- *Selective field connectionless integrity*, which provides integrity protection for selective fields within a single connectionless data unit.

2.3.4 Non-repudiation

This service can be sub-divided into the following:

- *Non-repudiation with proof of origin*, where the recipient of data is provided with evidence of the origin of data. This evidence will protect against any subsequent attempt by the sender to falsely deny sending the data. This service is usually abbreviated to non-repudiation of origin.
- *Non-repudiation with proof of delivery*, where the sender of data is provided with evidence of delivery of data. This evidence will protect against any subsequent attempt by the recipient to falsely deny receiving the data.

In both cases the evidence provided by the service must be of value in helping to resolve disputes using a trusted third party acting as an arbiter (e.g. a judge in a court of law).

2.4 Security mechanisms

In the context of this thesis, a security mechanism is the means by which a security service is provided. This section outlines the security mechanisms used in the design of the electronic payment schemes proposed in this thesis. Note that we use $X||Y$ throughout the thesis to denote the concatenation of data items X and Y.

2.4.1 Cryptographic hash-functions

A cryptographic hash function is an algorithm that takes a message of any size as an input and outputs a fixed-length 'hash code' (or message digest). Cryptographic hash functions are used as a building block to help construct other cryptographic mechanisms. For example, hash functions are used as a component in almost all practical digital signature schemes.

A cryptographic hash function must satisfy the following properties [66].

- *Pre-image resistance*: Given a random hash code, it must be computationally infeasible to find an input which the hash function maps to that hash code,
- *2nd pre-image resistance*: Given any input, it must be computationally infeasible to find a second input which gives the same hash code, and
- *Collision resistance*: It must be computationally infeasible to find two inputs which give the same hash code.

2.4.2 Symmetric cryptography

Symmetric cryptography includes symmetric encryption and message authentication codes. The use of symmetric cryptography requires the sender and receiver to agree on a shared secret key. Thus a major issue with symmetric cryptography is to find an efficient method to agree upon, or to securely exchange, keys between pairs of communicating parties.

Symmetric encryption

Symmetric encryption uses a secret key to encrypt a message into ciphertext and the same key to decrypt the ciphertext into the original message. For the purposes of this thesis, $E_K(M)$ denotes the symmetric encryption of message M using secret key K. There are two widely used classes of symmetric encryption schemes: block ciphers and stream ciphers — see, for example, [66] for a description of a number of symmetric encryption algorithms.

Message authentication codes

A Message Authentication Code (MAC) function takes as input a message and a secret key and outputs a fixed length MAC. MACs are used to help guarantee the source and integrity of a message. A MAC is sent together with the message it is protecting. For the purposes of this thesis, $\mathrm{MAC}_K(M)$ denotes a MAC computed on message M using the secret key K.

There are a variety of means for computing MACs, typically either based on the use of a block cipher or a cryptographic hash function (see, for example, [66]).

2.4.3 Asymmetric cryptography

The use of asymmetric, or public key, cryptography, as introduced by Diffie and Hellman [25], involves the use of key pairs instead of single secret keys, as is the case for symmetric cryptography. Moreover, each key pair is associated with a single entity, as opposed to secret keys which are typically shared by pairs or groups of entities. That is, each entity using public key cryptography has to obtain its own key pair.

Each key pair is made up of public key and a private key. The public key is typically widely distributed (and does not need to remain secret), whereas the private key should be a secret known only to its owner. The ways in which public and private keys are used depends on the type of cryptographic scheme involved. Asymmetric cryptographic schemes are typically based on complex mathematical functions, and tend to operate rather more slowly than symmetric cryptographic schemes.

Key management for asymmetric cryptography is a rather different problem than is the case for symmetric cryptography. Instead of establishing shared secret keys, the main problem is to reliably distribute copies of public keys for individual entities. This is addressed through key management systems known as Public Key Infrastructures (PKIs). As part of a typical PKI, one or more Certification Authorities (CAs) issue digitally signed public key certificates, whose purpose is to bind a public key to an identifier for the owner of the public key (the certificate subject), together with an expiry date and other information associated with the subject and the public key. X.509 [57] is a widely used standard that specifies the format of public key certificates.

There are a number of different types of asymmetric cryptographic schemes, including encryption schemes and digital signatures.

Asymmetric encryption

In an asymmetric encryption scheme, the public key is used for encryption and the private key for decryption. Asymmetric encryption can be used to provide confidentiality. For the purposes of this thesis, $e_{P_X}(M)$ denotes the asymmetric encryption of message M using the public key (P_X) of entity X.

Digital signatures

Digital signatures help guarantee the origin and integrity of a message. A signature scheme consists of three components, namely a key generation algorithm, a signing algorithm, and a verification algorithm. The key generation algorithm produces pairs of keys, made up of a private signing key which should be kept secret, and a public verification key which should be distributed to everyone who may wish to verify a digital signature. The signing algorithm takes as input the private signing key and the message that needs to be signed, and outputs a digital signature for that message. The verification algorithm takes as input the public verification key, a digital signature and, possibly, the message that was signed. It either outputs valid, which indicates that the given digital signature is a correct signature for that message, or invalid, which indicates that the given digital signature is not a correct signature for that message.

The idea behind a digital signature is that it should be computationally infeasible for any party to compute a correct digital signature without knowing the private signing key.

For the purposes of this thesis, $s_{S_X}(M)$ denotes the signature of entity X on message M using the private signature key (S_X) of entity X. Note that we implicitly assume that the signature scheme in use provides message recovery, i.e. the signature verification process takes as input the signature verification key V_X and the signature, and outputs the message and an indication of whether or not the signature is valid. If a signature scheme is used which does not have this property, i.e. the verification process requires the plaintext message as one of the inputs, then the notation $s_{S_X}(M)$ should be read as being equivalent to $M||s_{S_X}(M)$.

There are many signature schemes available (see, for example, [66]).

Anonymous Public Keys

Anonymous public keys (APKs) are certified public keys where the owner is anonymous to the verifier of the certificate [72]. These public keys can be used in the same way as non-anonymised public keys. Although the CA knows the user's identity, the CA cannot eavesdrop on the user's encrypted communication or forge a digital signature of the user.

We now sketch one method of implementing APKs (as described by Oishi, Mambo and Okamoto [72]), which applies to public keys for a discrete logarithm based signature scheme.

1. A user X registers his identity id_X and public key P_X with the CA, where P_X is generated using a discrete logarithm base g.
2. The CA converts the pair (g,P_X) to another pair (g',P'_X). These two pairs, however, are associated with the same private key S_X.
3. Next, the CA generates an anonymous public key certificate that consists of the converted public key P'_X, additional information (e.g. the identity of the CA, validity period, etc.) and a signature on this data generated by the CA.
4. Finally, the CA sends the anonymous public key certificate $\text{Cert}_{P'_X}$ to X.

2.4.4 S/KEY

One-time passwords are a strong technique for remote user authentication, as such passwords can only be used once and are hence resistant to eavesdropping. That is, an interceptor who learns a password will not be able to use it to impersonate the user because it will only be accepted once.

S/KEY is a one-time password scheme which has been published as an Internet RFC [40], and is based on a scheme originally proposed by Lamport, [63]. It uses one-time passwords to control user access to a remote host.

In the S/KEY scheme, the user and the host which the user wishes to access share a one-way hash function f. The user first selects a password p. The host is assigned an initial seed value d and a count value c which defines the number of user authentications to be allowed using this seed value. In addition, the host is given the verifier $f^c(s)$ for subsequent authentication, where s is the result of the bit-wise exclusive-or of the user secret password p with d, i.e. $s = p \oplus d$. Here, as throughout this thesis, $\oplus$ denotes the bit-wise exclusive-or operation. When the user identity is to be verified, the following procedure is followed.

1. The host decrements its stored counter c for that user and sends the new value of c to the user in conjunction with the seed value d.
2. On receipt of d and c, the user employs his secret password p to compute $f^c(s)$, and sends this value back to the host.
3. On receipt of $f^c(s)$, the host computes $f(f^c(s))$ and compares the result with its stored verifier $f^{c+1}(s)$. If the two values agree then the user is authenticated and the 'old' stored verifier is replaced with $f^c(s)$. Otherwise the user is rejected.

An advantage of the S/KEY scheme is that it does not require the host to store secret information about the remote user, since it only keeps the ‘old’ verifier, from which the new verifier (and the secret password p) cannot easily be derived. However, note that if p is poorly chosen, then knowledge of $f^c(s)$, c and d can be used as the basis of an exhaustive search for p. Also, S/KEY should only be used where the user can be confident of the host identity, otherwise attacks are possible [67].

Chapter 3

ELECTRONIC PAYMENT SYSTEMS

Contents

In this chapter electronic payment systems are introduced.

A generic model of an electronic payment system is first provided (section 3.2). This is followed by a description of some of the properties that distinguish the various types of electronic payment systems (section 3.3). A review of some examples of electronic payment schemes that are relevant to this thesis is given in section 3.4. Conclusions are provided in section 3.5.

3.1 Introduction

Over the last few years there has been a formidable increase in the use of electronic commerce (e-commerce) as a means to perform business transactions. The integration of information and communications technology in business has revolutionised relationships between businesses and individuals. E-commerce brings many benefits to its users, e.g. lower prices, greater choice, and better access to information.

There exists no unique definition for e-commerce, but it can be understood to include any normal business function executed via electronic networks. However, in this thesis we follow the definition given by Kalakota and Whinston [61]:

- From a *communications perspective*, e-commerce is the delivery of information, products/services, or payments via telephone lines, computer networks, or any other means.
- From a *business process perspective*, e-commerce is the application of technology toward the automation of business transactions and workflows.
- From a *service perspective*, e-commerce is a tool that addresses the desire of firms, consumers, and management to cut service costs while improving the quality of goods and increasing the speed of service delivery.
- From an *on-line perspective*, e-commerce provides the capability for buying and selling products and information on the Internet and other on-line services.

This thesis focuses on e-commerce applications primarily from the on-line and communications perspectives.

Meanwhile, in conventional payment systems, various means of payment are used to facilitate the exchange of goods and services. Common to all payment methods is the fact that the actual flow of money takes place from the payer's account to the payee's account. Currently, six major payment methods, with variations, are used to facilitate almost all commerce transactions: cash, cheques, giros, automatic clearing house, wire transfer, and debit/credit cards [73].

- **Cash payments**: Cash is a simple and effective means of payment for face to face commerce. Cash transactions have the advantage of settlement on the spot and do not carry any processing cost or paperwork (although for retailers there are costs associated with secure storage, transport and deposit of cash). Moreover, they can be conducted by anyone with sufficient cash, and both parties may remain anonymous. Cash transactions are relatively inexpensive to process and cost retailers less than other types of payments [87]. The major security threats in dealing with cash are theft, forgery and counterfeiting.

- **Cheque payments**: Paper cheques are an often convenient means of making payments, especially for large value transactions. However, cheques are costly for both users and banks, where the cost includes both cheque fabrication and cheque processing. Moreover, the task of clearing the cheque takes time.

- **Giro payments**: A giro is an instruction to the payer's bank to transfer funds to the payee's bank. The payment is not initiated unless the payer has the funds available, thus reducing the cost associated with returned cheques. Giro payments are easier than cheques to conduct electronically, since the correct processing of payment does not require sending the signed document through the clearing system [73].

- **Automatic Clearing House (ACH)**: The growing number of cheque and giro payments has made paper-based clearing increasingly difficult; this has led to the development of ACH payments [73]. ACH is an automated means of payment which operates in a similar way to paper clearing, except that the payment instructions are in electronic form. Examples of such systems include direct debits, automatic payroll deposits, and utility bill payments.

- **Wire transfer**: While ACH payments are used for low-to-mid value transactions, wire transfer payments are used for high value transactions. In these transactions the risk level is high (because of high transaction value), and thus different procedures involving more security are required [73]. Typically, participants in these systems include corporations, banks, and governments.

- **Debit/credit card payments**: Debit/credit cards appear to be the most expensive form of payment, with a transaction cost five times higher than the cost of a cash transaction [87]. Despite this, debit/credit cards have gained in popularity because of strong demand from consumers, who typically do not pay the additional transaction costs, at least in any direct way. Compared to cash payments, that only require the exchange of banknotes between buyer and seller, debit/credit card transactions are considerably more complex. A typical payment card transaction requires the participation of several entities in addition to the cardholder and the seller. For example, before a credit card payment transaction can be conducted, the cardholder (issuer) bank may be contacted by the merchant bank (acquirer) to authorise the transaction. The authorisation process uses pre-established criteria to make a decision, such as use of spending limit and/or a maximum allowed number of transactions in a specific period.

  After the transaction has been conducted by the merchant and cardholder, the transaction is cleared and settled between the issuer and acquirer using the network established by the card brand. With current payment cards the sellers have no cash handling problem, although they do have to pay additional fees to their acquiring bank.

  A debit card is linked to a bank account and used to pay for goods and services by debiting the cardholder's account at the time the transaction takes place. The transaction value is then transferred from the payer to the payee bank account. In the case of credit cards, the cardholder pays the balance at the end of a predefined period (e.g. monthly). However, the balance owing on a cardholder's credit card account need not necessarily be paid at the end of the defined period. The cardholder can choose to make a lesser payment, and then pay interest on the outstanding balance; in this case the card is used as an ongoing source of credit.

Electronic payment systems are clearly a prerequisite for e-commerce, given that on-line consumers must pay for products and services. However, such systems suffer from the same security problems as traditional means of payment, e.g., money can be counterfeited, signatures can be forged, and cheques can bounce. In addition, unlike the case with paper-based documents, digital information can be effortlessly and perfectly copied; moreover, a buyer's name can typically be associated with every payment, unlike the case of cash payments which are inherently anonymous. Thus, without secure electronic payment systems, widespread e-commerce is very risk-prone. Moreover, it seems reasonable to assume that the level of consumer adoption of e-commerce will be restricted if e-commerce gains a reputation for possessing major security flaws. For example, on-line privacy, which is a significant factor in consumer trust, is increasingly being viewed as an imperative for e-commerce success [46].

In general, electronic payment systems are cheaper than their paper-based equivalent. The costs associated with electronic payments usually lie between a third and a half of the costs posed by cheques or other paper based transactions [47].

Numerous schemes for electronically paying for goods or services across a network have been proposed. Many of these systems rely on, or extend, traditional payment methods such as cash, cheques or debit/credit cards. An overview of electronic payment systems can be found in [73].

In this thesis, an electronic payment system is defined as a system where the processing of payment instructions between buyers and sellers is facilitated electronically and is backed by a bank or an intermediary.

3.2 Electronic payment model

This section gives a generic model for an electronic payment system. This model is widely used — see, for example, [13]. The model is shown in figure 3.1. It involves four roles, namely a payer, a payee, an issuer, and an acquirer. The precise interactions between these roles will vary depending on the transaction type. During a transaction, on-line connectivity may be limited to certain subsets of roles.

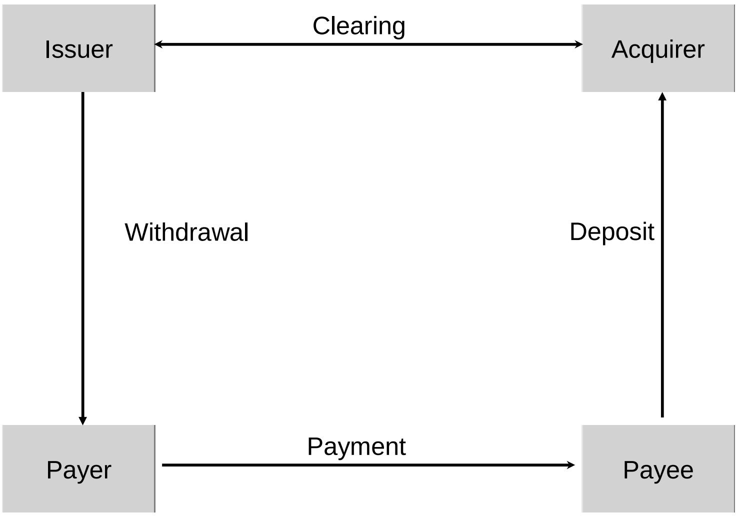

Figure 3.1: Model of an electronic payment system

- **Payer**: This is the entity that makes a payment using an electronic payment instrument obtained from the issuer. The payer is sometimes called the buyer or the consumer.

- **Payee**: This is the entity receiving the funds resulting from the payment. The payee sends the received payment instrument to the acquirer for clearance and settlement. The payee is sometimes called the seller or the merchant.

- **Issuer**: This is a financial institution that provides electronic payment instruments to the payer to use in a payment. These instruments need to provide assurance to the payee that they will be honoured.
- **Acquirer**: This is a financial institution associated with the payee that verifies the validity of the deposited payment instrument by clearing it with the issuer. After the settlement of funds between issuer and acquirer, the acquirer credits the payee's account with the monetary value stated by the deposited instrument.

For the purpose of this thesis we assume that the issuer and the acquirer trust each other and communicate by means of a secure link.

3.3 Characteristics of electronic payment systems

The payment model described in section 3.2 is generic. However, the actual meaning of a transaction, and the order in which the steps in a transaction are processed, will vary depending on the specific payment system.

In this section, we describe some of the properties that distinguish various types of electronic payment systems. These characteristics can be used to analyse an electronic payment system. Moreover, these properties help to determine the interpretation of the overall model, and the security requirements of participants in such a payment system [13, 35, 64, 74].

3.3.1 Information flow

The information flow in a payment scheme describes the way monetary value changes hands during a transaction. As in [3], electronic payment systems can be categorised according to the underlying flows of information into the following.

Direct cash-like systems

In a system of this type, the payer withdraws funds from his account at the issuer in the form of an electronic payment instrument. The payer hands the payment instrument to the payee, who in turn deposits it at his acquirer. The acquirer then asks the issuer for settlement. Direct cash-like payment systems resemble conventional cash (see section 3.1). Participants exchange electronic instruments that represent value, just as banknotes determine the value of paper money. The payer's instruments can be kept locally at the payer PC or in a payer smart card, or they can be managed centrally at the issuer. If no payer information is included in the electronic payment instrument, or in other payment data, then the transaction can be anonymous.

Direct account-based systems

These systems resembles the use of a conventional cheque (see section 3.1). The payer creates and signs an electronic payment instrument and sends it to the payee. The payee then presents the payment instrument to its acquirer for deposit in the payee account. The acquirer in turn redeems it from the issuer. The issuer arranges for the funds to be transferred from the payer's account to the payee account, and notifies the payer after completion of the payment. Such a scheme does not allow the payer's identity to be hidden from the issuer, nor can the payer's information be hidden from the payee.

Indirect push systems

In a system of this type, the payer instructs the issuer to transfer funds to the payee's account at the acquirer. The system requires the issuer to be on-line for every payment transaction. It is assumed that both the payer and the payee have an account with a bank. If they use separate banks it is assumed that the banks concerned are able to communicate securely. These systems resembles the use of giro payments (see section 3.1). The payee's only involvement is to receive notification of the incoming payment. Systems of this type do not allow anonymous transactions. The payer's identity must be known to the issuer in order to directly transfer funds from his account. The payer does not need to communicate any of his details to the payee, although in practice it is necessary for the payee to be provided with an identifier for each payment, so that the payee knows with which transaction the payment is associated.

Indirect pull systems

In this type of system the payee instructs the acquirer to charge the payer's account at the issuer. Usually, there must be a pre-existing agreement between payer and payee. These systems resembles the use of ACH payments (see section 3.1). The payer's only involvement in the transaction is simply to receive a notification of the outgoing payment.

3.3.2 Parties involved

The business relationships between the participants in an electronic payment system determine the type of payment instrument that can be used. Moreover, these relationships also affect how the system is designed. From the perspective of the buyer and seller relationship, e-commerce transactions can be divided into the following categories [19]:

1. Business-to-Consumer (B2C), where the seller is a business organisation and the buyer is a consumer. This is the electronic version of conventional retailing.
2. Business-to-Business (B2B), where both the buyer and the seller are business organisations, and relations are typically characterised by long-term stability.
3. Consumer-to-Consumer (C2C), where both the seller and the buyer are consumers.
4. Consumer-to-Business (C2B), in which a consumer specifies the requirements to a business, which provides a product that meets these requirements.

3.3.3 On-line versus off-line authorisation

An important characteristic of any payment system is whether there is a need to obtain an on-line authorisation for a transaction, or whether it is possible to proceed with a transaction without communicating with the issuer. Many electronic payment systems require the payee to contact the issuer (via the acquirer) at the time of the transaction in order to receive authorisation for the payment. The method used to obtain an authorisation depends on the particular payment system. Authorisation might take the form of verifying that an electronic coin has not been previously spent in the case of an electronic money system, or a credit check in the case of a credit card system.

Clearly, the use of an on-line authorisation system implies a higher cost per transaction, and a greater processing load is placed on the issuer. Moreover, it assumes that the issuer can always be contacted. If the communications link to the issuer becomes unavailable, then potentially no transactions can be accepted. In the case of network congestion, an on-line transaction may also be subject to time delays. On-line authorisations are, however, probably essential when the value of the payment is high.

In an off-line electronic payment system, only the payer and the payee have to be in contact at the moment of the transaction. This kind of payment must include a way of verifying the validity of the transaction. In most off-line electronic payment systems, authorisation is provided using a smartcard. Such systems usually implement limits on how much the user can spend. In the case of electronic cash, verifying that an electronic coin has not been previously spent can typically be established after the fact by cryptographic means, thereby increasing security. In an off-line system payees are required to contact their acquirer on a regular basis for clearing received payments. Off-line schemes typically impose greater transaction complexity on the payer and payee than on-line schemes, since transaction security must be maintained without the on-line assistance of the acquirer or the issuer.

3.3.4 Hardware versus software

In an electronic payment system, payers need some kind of device to play their part in a payment transaction. An important issue is whether the device contains tamper-resistant hardware or not. A software-only system is one in which no part of the device is required to be tamper-resistant.

In a software-only system, additional security features must be provided to prevent users from obtaining any benefit from tampering with software, stored data or exchanged data. One advantage of a software-only system is that it can be distributed easily at low cost. Moreover, it can be run on any computer, and hence there is typically plenty of computing power and disk storage available. A disadvantage of software-only electronic payment schemes is that a software application must be installed and initialised on a specific computer. Software-only schemes are usually specifically designed for transactions over the Internet.

By contrast, some electronic payment systems require specific hardware to process transactions, and require some part of this hardware to be tamper-proof. In the case of electronic cash systems, tamper-resistant hardware can help solve the double spending problem in an off-line environment. A disadvantage of hardware dependent electronic payment schemes is the expense required to establish the scheme. Smart cards must be securely distributed and smart card readers must be securely installed.

3.3.5 Size of the payment

An electronic payment system in which relatively large amounts of money can be exchanged is usually referred as a macropayment system. On the other hand, if a system is designed for small payments, it is called a micropayment system [73].

A payment system is usually only designed for a specific range of transaction values, and will not be well-suited for values outside this range. This is primarily due to a trade-off between the transaction costs and the level of security offered by the scheme. In macropayment schemes, transaction overheads such as processing and communications costs are of lesser importance, because these costs tend to be small relative to the transaction value, and large transactions are also relatively infrequent. Further, anonymity is not always required, as an audit trail is not only usually required by law, but also makes dispute resolution easier. By contrast, in the micropayment case, transaction costs must be minimised. Reducing computational complexity can, in turn, reduce the requirements on the devices performing the transaction, which leads to a potential reduction in the transaction cost. Other cost reductions are possible by performing off-line authorisations (see section 3.3.3) and by grouping the clearance of transactions [84].

3.3.6 Transaction medium

Electronic payment systems are often designed for specific transaction media. Currently, Point Of Sale (POS) terminals, the Internet, mobile networks, and pay TV are all used to support electronic payment transactions. These media each have specific restrictions and requirements that affect the way an electronic payment transaction takes place. For example, while the Internet is generally a fast and cheap medium, mobile networks such as GSM tend to be relatively slow and expensive. Therefore, payment systems designed for this latter type of medium must implement special features to limit the amount of data to be transferred and the duration of the connection.

3.4 Examples of electronic payment systems

In this section examples of electronic payment systems of a variety of different types are discussed.

3.4.1 Face-to-face payments

Currently, debit/credit card payments are widely used in POS transactions, where payees rely on paper signature checking to verify the payer's identity and on the magnetic stripe card's physical properties to authenticate the card. However, magnetic stripe card technology is not sufficiently robust to prevent a variety of frauds against debit/credit card POS payments. In order to help combat debit/credit card payment fraud, the major card brands have therefore developed an industry standard to employ IC (Integrated Circuit) cards instead of the existing magnetic stripe cards. The goal of the standard is to reduce both fraud and the costs associated with on-line transaction authorisation at the POS. This collaboration between MasterCard and Visa has resulted in the card/terminal specifications known as EMV [27, 28, 29, 30]. The EMV specifications standardise interactions between a debit/credit IC card and a POS IC terminal. The EMV transaction is discussed in detail in Chapter 5.

3.4.2 Remote payments with debit/credit cards

Although many electronic payment systems have been proposed for use over the Internet, currently most transactions are conducted using debit/credit cards. The transaction model is essentially identical to that used for any other 'cardholder not present' transaction, e.g. a mail order/telephone order transaction. That is, the merchant will use the card account number and associated details, such as the accountholder name and the card expiry date, provided by the cardholder to seek an on-line authorisation from the card issuer, just as would be the case for any other transaction. The only significant security measure particular to such a transaction is that the SSL/TLS protocol is widely used to secure the communication link between the payer and the payee, hence preventing disclosure of the card details to anyone monitoring the Internet link.

Other electronic payment protocols such as SET, Visa 3-D Secure, and one-time credit cards, have been proposed to eliminate some of the vulnerabilities associated with the naive use of

SSL/TLS for securing electronic payments.

An advantage of the current model for debit/credit card electronic payments is its ease of use and scalability. However, disadvantages of this model, and debit/credit card transactions generally, include the lack of buyer anonymity and the lack of support for small payments. Moreover, the fact that debit/credit card issuers gain access to payer account details and spending behaviour poses potentially serious security and privacy concerns.

SSL/TLS

In a typical Internet debit/credit card transaction, the payer uses a web browser to send his debit/credit card details over a secure channel to the payee's web server. This secure channel is provided by the Secure Sockets Layer (SSL) protocol [38], or its successor the Transport Layer Security (TLS) protocol [24]. SSL/TLS was not designed specifically to secure payments over the Internet; however, because SSL/TLS is integrated into popular web browsers, it is widely used by payee web servers to set up secure sessions with payer PCs during Internet debit/credit card transactions.

SSL/TLS provides data confidentiality, payee authentication, (optional) payer authentication and data integrity. SSL/TLS authentication is based on asymmetric cryptography, with each entity to be authenticated requiring a signature key pair and a matching public key certificate issued by a trusted third party. After successful authentication, all messages exchanged are encrypted and integrity protected using symmetric cryptography. The actual cryptographic algorithms used are negotiated at connection setup. Note that, although payer authentication is optionally available as part of SSL/TLS, this is almost never performed in practice since the payer will not have a suitable signature key pair and corresponding public key certificate.

Once the debit/credit card details have been passed to the merchant, they are processed in the 'normal way'. That is, an on-line transaction authorisation would typically be sought from the issuer and, after the transaction has been completed, the issuer and acquirer will clear the resulting payment.

One problem with this model is that a payer has to disclose his debit/credit card number to each payee (as is the case for conventional debit/credit card transactions). Thus fraud is possible through the misuse of the payer debit/credit card details. Even if the payee is honest, there is still the risk that payer card details will be disclosed to a fraudster that hacks into the payee web server. Moreover, without payer authentication it is possible for a fraudulent third party to pretend to be a card owner by presenting valid credit card details (again as is the case for any 'cardholder not present' transaction).

SET

Visa and MasterCard developed the Secure Electronic Transaction (SET) protocol [80], to provide a complete and secure solution for credit card based electronic payments. The SET security model provides confidentiality and integrity protection for transaction data and authentication of all participants. Confidentiality is provided using symmetric cryptography, while integrity and authentication are supported using digital signatures. Digital certificates are issued to all participants by a SET-specific Certification Authority.

In SET, a payer is required to send an order information message to the payee for processing; at the same time, a payment instructions message is required by the acquirer. SET uses the concept of dual signature to protect a payer's payment instruction from both eavesdroppers and dishonest payees. Moreover, a dual signature provides confidentiality of purchase order information with respect to the acquirer. It also provides a link between a message and an identity without the need to be able to see the message contents. In SET, the dual signature is constructed using the following steps:

- Message digests are generated for both the order information and the payment instruction.

- The two message digests are concatenated to produce a new block of data which is hashed again to provide a final message digest.
- The final message digest is digitally signed by the sender. A recipient of either message (order information or payment instruction) can check its authenticity by generating the message digest on its copy of the message, concatenating it with the message digest of the other message (as provided by the sender) and computing the message digest of the result. If the newly generated digest matches the verified dual signature, the recipient can trust the authenticity of the message.

In a typical SET transaction, as shown in figure 3.2, the following steps are executed.

1. The payer sends an initiate request to the payee.
2. In response, the payee generates an 'initiate response' message containing his public key certificate and that of the acquirer. This message is then digitally signed and sent to the payer.
3. Upon receipt of the initiate response message, the payer verifies the certificates of both the payee and the acquirer. Next, the payer creates the order information, which identifies the goods to be purchased, and the payment instruction which contains the payer account number, price, and a unique transaction identifier. The transaction identifier links the order information and payment instruction together. Finally, the payer generates a purchase request message containing dual-signed order information and a dual-signed payment instruction that is digitally enveloped using the acquirer's public key (i.e. encrypted using a random secret key, and this key, in turn, is encrypted with the public key of the acquirer). The entire purchase request is then sent to the payee along with the payer certificate.
4. When the payee receives the purchase request message, he verifies the payer certificate, as well as the signed order information. The payee generates a purchase response message to the payer which includes the signed transaction identifier.

5. The payee then generates and signs an authorisation request message, which includes the amount to be authorised, the transaction identifier from the order information, and other information about the transaction. This information is digitally enveloped using the acquirer public key. The authorisation request message, the payer payment instruction, and the payee certificate are transmitted to the acquirer.

6. The acquirer verifies the payee signature on the authorisation request message and decrypts the payment instruction. The acquirer transmits the payer data to the issuer to request a confirmation of the payer's identification and card information.

7. If everything is valid, the acquirer generates and digitally signs an authorisation response message which is then digitally enveloped using the payee public key and sent to the payee.

8. Upon receipt of the authorisation response message, the payee verifies the message. If the purchase is authorised, the payee then completes processing of the payer's order by shipping the goods.

9. When the order-processing portion of the protocol is completed with the payer, the payee requests the acquirer to capture the payment. The payee generates the capture request, which includes the final amount of the transaction, the transaction identifier, and other information about the transaction. This message is then digitally enveloped using the acquirer public key and transmitted to the acquirer.

10. After the capture request is received by the acquirer, its contents are verified. Funds are then transferred to the payee's account. The acquirer generates the capture response, which includes information pertaining to the payment for the transaction requested. This response is then digitally enveloped using the payee public key and is transmitted back to the payee. Upon receipt of the capture response from the acquirer, the payee decrypts the message, verifying the signature and message data. The capture response is stored by the payee for reconciliation with payment received from the acquirer.

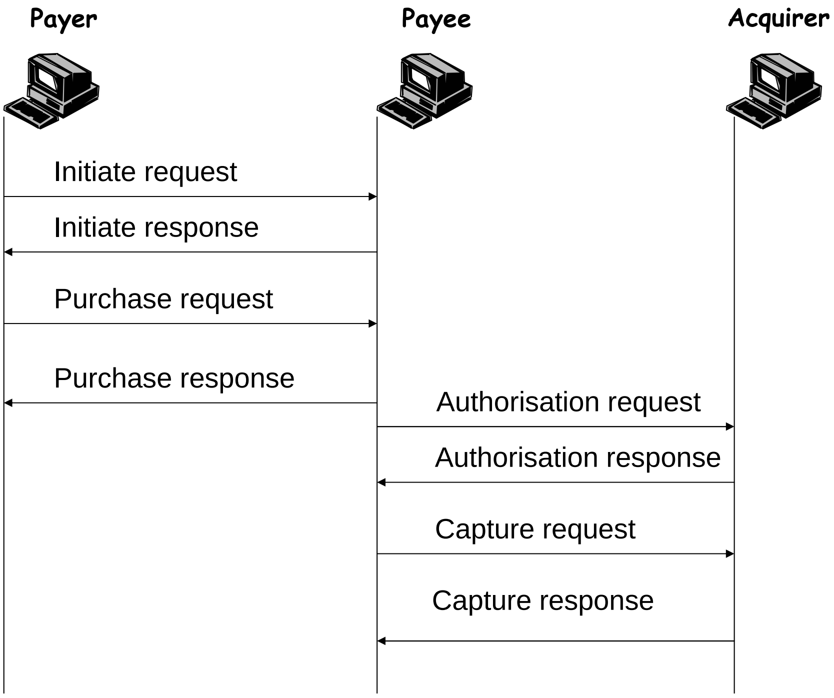

Figure 3.2: SET transaction

In SET, even if unauthorised access to a payee web server occurs, the confidentiality of payer payment details will not be endangered since these details are encrypted using an acquirer public key. Thus SET can prevent credit card fraud arising from transmission and storage of sensitive data. Moreover, order and payment information are encrypted separately for specific recipients. That is, the payee public key is used to encrypt order information, and the acquirer public key is used to encrypt payment information. Payers can thus be assured that their payment details will not be compromised by a fraudulent payee.

Although SET was designed to secure remote electronic payments it has, however, not been widely adopted. The primary reason for this is probably its overall complexity. For example, the complexity of the specifications meant that software development and testing was considerable. Other factors impeding its adoption include: the need for all parties, including cardholders, to generate key pairs and have their public keys certified, the major changes to the payment model for merchants, and the difficulties in simultaneously assembling a critical mass of payers and merchants to use the protocol, [73].

Visa 3-D Secure

The Three Domain (3-D) Secure protocol was developed by Visa to provide payer authentication when paying on-line using a debit or credit card [88]. As the name suggests, the system involves functionality in three domains:

- the Issuer domain: in which the issuer operates an Access Control Server (ACS) and handles communication with payers and a centralised Visa directory.
- the Acquirer domain: in which functionality implemented in a payee (merchant) server plug-in handles communications with the Visa directory and the acquirer.
- the Interoperability domain: in which an intermediary between payees and issuers, called the Visa directory server, is operated.

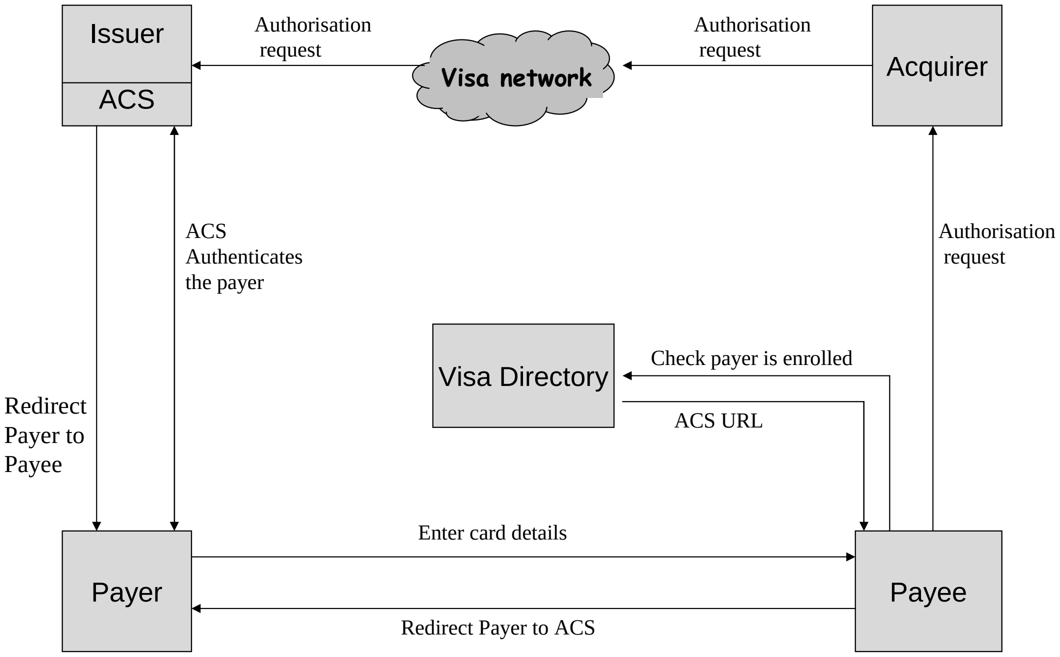

Figure 3.3: 3-D Secure transaction

In 3-D Secure, all the main communications links, including between payer and payee server, and between payer and issuer ACS, are protected using SSL/TLS.

The transaction flow for a 3-D Secure transaction, as shown in figure 3.3, begins when the payer decides to buy goods or services offered via a payee web site. The payer is first requested to enter debit/credit card details, including the Primary Account Number (PAN), into a payee form. A plug-in is activated (on the payee server), which queries the Visa directory server to determine whether the payer is enrolled in the 3-D Secure scheme. If so, then the directory server provides the payee plug-in with the URL of the ACS of the payer's card issuer. Using the HTTP redirect functionality, the payee redirects the payer web browser to the ACS URL. The ACS then authenticates the payer using a card issuer specific method. If payer authentication is successfully completed, the ACS formats an authentication response, which is digitally signed by the ACS. The ACS redirects the payer's web browser back to the payee, and the authentication response is simultaneously transferred to the payee. This response includes a unique cryptographic value based upon the transaction data called the 'Cardholder Authentication Verification Value' (CAVV). A copy of the authentication response message is also (optionally) sent to an Authentication History Server (AHS). When the payee server receives the authentication response, the payee server verifies the digital signature of the issuer, and submits an authorisation request to the acquirer. This authorisation request includes the CAVV and a unique transaction identifier. The acquirer passes this into the Visa payment network, where a server verifies the CAVV received with the copy stored on the AHS, and forwards the authorisation request to the issuer who processes the transaction in the normal way.

In 3-D Secure, the payee has the benefit of authenticating the payer, thus reducing the likelihood of fraud.

One-time credit card numbers

To protect against the fraudulent use of credit card numbers, the notion of one-time credit card account numbers has been introduced. In such a scheme, a temporary credit card number that is valid for one transaction only is assigned to the payer by the issuer. The issuer associates the temporary account number with the permanent account number of the payer for payment purposes. This temporary credit card number can then be used by the payer just like an ordinary card number, and sent over an SSL/TLS link to a payee Internet web server. The payee submits the credit card number to the issuer for authorisation, just as in a standard credit card transaction. The one-time property protects against fraudulent use of stolen payer credit card numbers, since the issuer will only authorise one transaction for any given temporary card number. Since different purchases are paid for with different credit card numbers, which cannot be linked together by the payee, such a scheme also provides a measure of payer privacy, although this is typically very limited since payers are normally required to submit their account name with their card number. Moreover, transactions can still be linked by the issuer. Various schemes have been proposed to generate one-time credit card numbers. An example of such a scheme is Shamir's SecureClick [83], which was commercialised by Cyota[1].

In a typical one-time credit card scheme, the issuer provides a browser plug-in to the payer when he joins the system. This plug-in is invoked whenever the payer clicks the pay button on a participating Internet site. The plug-in initiates the payer and issuer pre-approval protocol. In this protocol both the parties are authenticated and details of the transaction are send to the issuer, who provides a one-time credit card number along with an expiry date back to the payer. The payer copies this number to the payee Internet form and continue the transaction as normal. A disadvantage of this scheme is that if a large number of payers simultaneously connect to an issuer using SSL, the performance of the issuer server will potentially become a bottleneck.

3.4.3 Electronic cheques and account transfer

An electronic cheque resembles an ordinary paper cheque. In a typical electronic cheque transaction, the payer orders goods from the payee, and the payee sends an electronic invoice to the payer. As a payment, the payer sends a digitally signed electronic cheque. The payer is typically required to endorse the cheque by applying a further digital signature before sending it to the acquirer. Using the existing financial networks, the issuer and the acquirer can ensure that the correct amount is withdrawn from the payer's account and credited to the payee's account. To ensure the availability of funds during a purchase, the cheque will typically be cleared on-line. After receiving the cheque from the payer, the payee can ship the goods or deliver the services ordered.

The concept of an electronic cheque was developed by the Financial Services Technology

[1] http://www.cyota.com

Consortium (FSTC)[2]. The objective was to use the Internet to transfer money between parties with accounts at different banks. Many electronic cheque systems have been proposed in the literature including NetCheque [71], Mandate II [22], and NetBill [85]. The FSTC has implemented an off-line electronic cheque scheme [12], where each payer and payee uses a smart card to hold their public/private keys and digital certificates (issued by their bank). Moreover, the smart card maintains a register of which cheques have recently been signed or endorsed. It is assumed that both issuer and acquirer are issued certificates by a certification authority. To provide confidentiality, a secure link is assumed between the participants, e.g., as provided by SSL/TLS, when transporting electronic cheques. Using his smart card, the payee endorses the cheque when it is received, before forwarding it to the acquirer where it is processed in a similar way to a paper cheque. Advantages of e-cheques over debit/credit cards include that they can be used for high-value transactions and business-to-business payments, and they allow for easy integration with existing banking systems.

Account transfers

Another type of payment scheme that has recently gained widespread adoption for Internet payments is the centralised account transfer. In such a scheme a payer can make a payment to anyone with an e-mail address, giving that the payee has an account with the same system. In this model, the payer connects to the payment system, and authorises it to transfer a certain sum from the payer account to the payee account. SSL/TLS is usually used to secure the communications when connecting to the payment system, and authentication is usually password based.

O'Mahony, Peirce and Tewari [73] identify three methods for funding the payer account, namely using a debit/credit card, transferring funds from a regular bank account, or using a prepaid card.

In the first method, the payer uses a credit card to fund his account; an example of such a

[2] http://www.echeck.org/

scheme is PayPal[3]. PayPal links user's credit card numbers to their e-mail addresses. One PayPal user can transfer money to another PayPal user by logging into the central PayPal web site with a password, and submitting the e-mail address of the other user and the amount of money to be transferred. PayPal will then debit the payer's account and credit the payee's account with the relevant amount. Compared to an Internet credit card payment transaction secured using SSL/TLS, PayPal has the advantage that the payer does not need to send sensitive information directly to the payee, and a payment requires payer authentication. On the other hand, sensitive information is kept for a longer period of time at the PayPal central server and email messages can be intercepted by a third party.

The second method is to transfer funds from a bank account. When signing up, the bank account details are given to the payment system. At the request of the payer, funds can then be transferred using an existing financial network from the payer's bank account to the payment system on-line account. Yahoo! PayDirect[4] is an example of this method.

In the third method, the payer purchases a physical prepaid card that is associated with a unique number and contains a specific financial value. The card number and the corresponding stored value are stored by the payment system. The value held on the card is protected with a PIN code which, for example, can be retrieved by the cardholder from a scratch-off card supplied with the prepaid card. On acquiring the card, the payer submits the account identifier on the prepaid card to the payment system's web site. Once the account number has been entered, the funds allocated for that account are available for spending. These systems are only suited for relatively small payments; however, they do provide a means for anonymous payment. They also provide a payment solution for people who do not have a debit or credit card, or do not want to send their card number across the Internet. InternetCash[5] is an example of a pre-paid account based system.

[3] http://www.paypal.com/

[4] http://paydirect.yahoo.com/

[5] http://www.internetcash.com/

3.4.4 Electronic money

Unlike debit/credit card and cheque payments, cash payments inherently provide payer anonymity because there is no information to link the payment to the payer. Developments in cryptography have made it possible to implement an electronic version of cash payments [45].

A unit of electronic money is usually referred to as a 'electronic coin'. In a typical non-anonymous electronic money transaction, the payer *withdraws* electronic coins from his account and pays for these electronic coins by some means (e.g. using cash or by debiting a bank account). Each coin is digitally signed by the issuer to show that it is authentic. Electronic coins can be carried on a smart card or stored on a PC hard disk. The payer can then make purchases from any payee that accept the electronic coins of that issuer. To make a payment, the payer executes a *payment* operation to send the coins to the payee. The payee can verify the issuer's signature on each coin. The payee then *deposits* the coins at the issuer for redemption.

In order to prevent a coin from being spent twice (double spending), the issuer maintains a database of the serial numbers of all spent coins. If a coin has already been spent then its serial number will be present in the database. Otherwise, the payment is valid, and the coin serial number is entered into the database. In on-line schemes, the database is consulted for each payment. Some off-line schemes use a tamper-resistant smart card to hold coins, which can be used to prevent double spending.

Schemes with payer anonymity

Different electronic money schemes provides varying levels of payer anonymity. Jakobsson *et al.* [59], classify electronic money payment schemes into those with perfect payer privacy, schemes with revocable privacy, and schemes without privacy or with limited privacy mechanisms.

Schemes with perfect privacy provide unconditional anonymity to the payer. In these schemes, the identity of a payer in a particular transaction remains unknown to all parties, even if those parties collude. Payer anonymity may be established at the time of withdrawal of the electronic coin from the issuer. The issuer signs the electronic coins during withdrawal in such a way that the issuer cannot determine the serial numbers on the coins he is signing. The issuer is now unable to link a specific withdrawal with a specific deposit. Ecash [79] is an example of an on-line payment protocol that offers unconditional anonymity for the payer.

However, full anonymity may cause problems at the legal level and also make it easy to cheat without being caught [89]. To address these issues, some schemes provides conditional anonymity. In such a scheme a trusted third party can revoke payer anonymity under certain circumstances, such as if double spending has been detected. Such anonymity revocation is usually made possible by encoding the payer identifier into the electronic coin in an encrypted form such that only a trusted third party can decrypt it. When anonymity revocation is needed, the encrypted payer identifier is given to the third party, which then decrypts it. Stadler, Piveteau and Camenisch [86] propose a scheme that offers revocable anonymity.

Schemes with limited privacy mechanisms allow the identity of the payer to remain unknown to the payee, but not to the issuer. The electronic coins contain the payer's identity information. For example, in the Mondex scheme[6], each card has a unique identification number that is linked to the person to whom the card was issued at the bank. A user cannot buy a Mondex card without revealing his identity.

3.5 Conclusions

In this chapter we introduced electronic payment systems, and we gave a generic model for such systems. Characteristics that distinguish various types of electronic payment systems were also identified. These characteristics can be used to analyse an electronic payment system. Moreover, these properties help to determine the interpretation of the overall model.

[6] http://www.mondexusa.com/

A selection of electronic payment schemes relevant to this thesis was also discussed.

Chapter 4

ISLAMIC FINANCE

Contents

This chapter introduces the main concepts underlying Islamic finance.

We start by briefly describing the relevant aspects of the religion of Islam (section 4.1). The view of Islam regarding the payment of interest is presented in section 4.2. Islamic banking as an alternative to conventional banking is described (section 4.3), followed by a description of the notion of a Murabaha sale (section 4.4). Finally, we describe the importance of charity to Muslims (section 4.5). Conclusions are provided in section 4.6.

4.1 Introduction

There would appear to be no reliable source of information regarding the worldwide population of Muslims; nevertheless, some sources estimate that Islam is the world's second largest religion with nearly a billion followers [4].

The word 'Islam' is derived from the arabic word *salaam*, which means peace, submission, and obedience [65]. In the religious sense, Islam is the acceptance of, and obedience to, the teachings of God, which He revealed to his last prophet, Muhammad [48]. Under Islam, individuals have five fundamental duties [18], the first of which is a state of faith, and the other four of which are exercises of faith:

- Witnessing that Allah (God) is one and Muhammad is his messenger,
- Establishment of the daily prayers,
- Payment of obligatory charity to the needy (Zakat),
- Fasting during the month of Ramadan, and
- Pilgrimage to Mecca, at least once in a lifetime.

In Islam there is no separation between church and state. Islamic law is a comprehensive discipline regulating all public and private behaviour. It covers practicalities related to worship, as well as dealing with societal issues including political and economic relationships as well as social activities.

The following are the fundamental sources of Islamic law, listed in decreasing order of priority.

- The Quran, which Muslims believe to be God's words presented to the prophet. The Quran is the primary source and must be followed at all times; it is eternal and cannot be changed.

- Sunna, which is the authentic tradition of the prophet. It includes his sayings about specific cases and his conduct regarding them. Like the Quran, Sunna is everlasting and cannot be altered.
- Ijma, which is a consensus of scholastic opinions on a question of law. The concepts and ideas found in the Ijma are not found explicitly in the Quran or the Sunna.
- Qiyas, which is a form of reasoning by legal analogy, i.e. a comparison with similar questions that have already been settled. Qiyas covers issues not explicitly found in the Quran or Sunna, and not given in the Ijma.

4.2 Islam and the payment of interest

Most types of trade (buying and selling) are permitted in Islamic law. In Islamic law, a valid trade is concluded if a seller and a buyer exchange an offer and an acceptance of that offer, both of which specify the object of sale and the price, and they both agree. Therefore, any financing conducted through valid trading by mutual consent is permissible [26]. However, a key concept in Islamic law is the prohibition of payment and receipt of interest on deposits and loans. Instead, the sale of goods and the sharing of profits and losses among parties to any business transaction are encouraged.

Interest (Riba) is strongly condemned in verse (2:275) of the Quran, ‘Those who eat Riba will not stand (on the Day of Resurrection) except like the standing of a person beaten by the Evil leading him to insanity. That is because they say: ‘Trade is only like Riba’, whereas Allah has permitted trading and forbidden Riba’ [5]. Moreover, it is reported in the Sunna that the prophet Muhammad cursed the one who accepted Riba, the one who paid it, the witness to it, and the one who recorded it. Riba is an arabic word that means extra [65]; with respect to Islamic law, Riba has various forms, one of which is the lending of money for a specific period of time, after which the lender receives his money back with an extra amount agreed upon added to it. Riba is often referred to as interest or usury. It is interesting to note that the pejorative term 'usury' is used in Islam with respect to any interest-bearing transaction, whereas in common English usage the term is used only in connection with interest-bearing transactions where the interest charged is deemed to be excessive.

4.3 Islamic banking

The current worldwide banking system is based on the use of interest. Modern banking systems were introduced into the Muslim countries in the late 19th century. Many Muslims have confined their involvement with these banks to transaction activities such as current accounts and money transfers; borrowing from banks was strictly avoided in order to avoid dealing in interest, which is prohibited in Islam.

With the passage of time, however, and the demands for more involvement in financial activities, avoiding interactions with the banks became impossible. As the need to engage in banking activities became unavoidable and urgent, governments, businesses and individuals were obliged to transact business with interest-based banks, whether or not they really wished to. This situation drew the attention and concern of Muslim intellectuals, and resulted in a series of books and articles on interest-free banking both in English and Arabic during the 1950s and early 1960s.

The first modern attempt to establish an Islamic bank on individual initiative was in Malaysia in the mid-1940s [42]. The objective of this institution was to invest pilgrim savings in real estate and plantations in accordance with Islamic law; however, it did not turn out to be the immediate success that was anticipated. Another attempt was made in Pakistan in the late 1950s, and a small experimental Islamic bank was established in the rural area. The bank was closed down after few years of operation because of a shortage of funds [93]. A third attempt was made in Egypt during the 1960s. Established as an interest-free banking alternative, the bank mainly focused on providing savings facilities based on the profit and loss sharing concept. The experiment continued until 1967 [11].

During the 1970s, development of Islamic banking took a different direction. Unlike the above-mentioned three saving banks, the new Islamic banks were oriented toward profitable investments. In 1972, the Nasir Social bank was established in Egypt and declared as an interest-free commercial bank. The initial capital of the bank was provided by the Egyptian government, which remained the sole owner of the bank.

The progress of the Nasir Social bank was a major source of encouragement for Islamic banking in other countries. The first private interest-free bank, the Dubai Islamic bank, was set up in 1975 by a group of Muslim businessmen from several countries. Two more private banks were founded in 1977 under the name of the Faisal Islamic bank in Egypt and Sudan. In the same year the Kuwaiti government set up the Kuwait Finance House [11].

In the ten years following the establishment of the first private commercial bank in Dubai, more than 50 interest-free banks came into being. Though nearly all of them are in Muslim countries, some are in Western Europe, including in Denmark, Luxembourg, Switzerland and the UK [49].

Beyond establishing and running independent Islamic banks at the state and individual levels, an inter-governmental bank was established in 1975. The main objective of the Islamic development bank was to provide financial capital for industrial and agricultural projects in Muslim countries. The activities of the bank are limited to mechanisms that are allowed in Islamic law. The membership of the Islamic Development Bank stands now at 53 countries.

In most countries the establishment of interest-free banking has been by private initiative and the adoption of Islamic banking principles were confined to that bank. In Pakistan, Iran and Sudan, however, the adoption of Islamic banking principles has occurred as the result of a government initiative and covered all banks in the country. The governments in both these countries took steps in 1981 to introduce interest-free banking. They claim to have converted their entire financial system to a wholly interest-free scheme. Other countries, like Malaysia, Saudi Arabia, and some other Arab countries employ a hybrid financial system where interest-based (conventional) banks may coexist with interest-free (Islamic) banks, and both types of financial institutions are regulated by the monetary authorities of those countries.

The main characteristic of Islamic banks is the prohibition of the payment or receipt of interest. Although most Muslim countries allow conventional banks to operate alongside Islamic banks, there is nevertheless a growing desire among many Muslims to abide by Islamic law, which has helped the growth of Islamic banking. It is difficult to obtain exact figures on the size of the Islamic financial sector; it is nevertheless experiencing strong growth. According to [56], assets of worldwide Islamic banks grew from $5 billion in 1985 to over $100 billion in the late nineties.

Currently, financing is the primary function of an Islamic bank. Between 50 and 80 percent of the total assets in Islamic banks are used for financing activities [43]. Different financing instruments are used, including those listed below.

- *Musharaka* – this very much resembles the structure of a joint venture.
- *Mudaraba* – this is identical to an investment fund in which managers handle a pool of funds. The capital providers and the bank share the profits and the losses.
- *Salam* – this is a forward sale.
- *Murabaha* – this a form of a cost-plus financing.
- *Ijara* – this is a type of lease financing.

The main focus of much of the remainder of this thesis is the Murabaha instrument, which we describe in more detail immediately below.

4.4 Murabaha Sale

The Murabaha sale is one of the most commonly used forms of financing provided by Islamic banks. Hasanin [44] notes that Murabaha is the mode of contract most frequently used in Islamic banking, in some cases accounting for 90% of all financing.

Murabaha is an Arabic term that means obtaining profit, and is a type of trust trading. Financially, it means cost plus profit sale, but in Islamic law it is a term that refers to a particular kind of sale [44].

4.4.1 Outline of a Murabaha transaction

A customer wishing to purchase goods requests the Islamic bank to purchase these items on his behalf, and then at a later date sell them to him with a certain amount of profit agreed upon added to the initial cost. In the period up to the resale the bank has title to the goods, and hence a legal responsibility. The basic component of Murabaha is that the seller discloses the actual cost he has incurred in acquiring the goods, and then adds some profit thereon. When a customer approaches an Islamic bank to finance a purchase through Murabaha, the payment of the price is usually deferred, and most commonly paid in instalments.

4.4.2 Rules Governing a Murabaha Sale

The validity of a Murabaha transaction depends on certain conditions, that must be properly observed to make the transaction acceptable in Islamic law. These principles, as stated in [44], are as follows.

- The two sale transactions making up a Murabaha payment, one through which the financial institution acquires the commodity and the other through which it sells it to the customer, should be separate and real transactions.

- The financial institution must own the commodity before it is sold to the customer.
- It is essential to the validity of the Murabaha sale that the customer is aware of the original price, including the costs necessary to obtain the commodity, and the profit. This is because Murabaha is a sale with a mark-up, and if the customer did not know the basic price then a violation of the Murabaha sale conditions has taken place.
- Both parties, i.e. the financial institution and the customer, have to agree on the profit for the financial institution from the sale, where the sum of the cost and profit is equal to the selling price charged by the financial institution.
- Murabaha is valid only where the exact cost of a commodity can be ascertained. If the exact cost cannot be ascertained, the commodity cannot be sold on a Murabaha basis.
- It is necessary for the validity of Murabaha that the commodity is purchased from a third party. The purchase of the commodity from the customer on a “buy back” agreement is not allowed in Islamic law. Murabaha based on a “buy back” agreement would be nothing more than an interest-based transaction.
- Cash is not permitted to be withdrawn on a Murabaha basis.

Unless these conditions are fully observed, a Murabaha transaction becomes invalid under Islamic law.

4.5 Charity

Giving charity is a common activity for many individuals. However, for Muslims it is an important part of their religion (see section 4.1). Generally, every Muslim who owns more than a certain amount of wealth must pay annually a fixed rate of charity (Zakat) to those in need. This payment can be in cash.

Another occasion for making charitable donations is the month of Ramadan where, after completing a month of fasting, Muslims celebrate Eid, the festival of the breaking of the fast. It is also an occasion to make a special donation (Zakat ulfitr) to the poor. This is a type of charity given at the end of the month of Ramadhan to the poor and the needy. All Muslims who have enough money to take care of their own family's needs must make this donation.

It is obligatory for a Muslim, young or old, male or female, if he finds that he has a surplus of property after having catered for his core needs on the day and night of Eid, such as food, shelter, clothes etc. He must give this on his own behalf and on behalf of all his dependents, young or old, provided that the dependent is not able to give the Zakat on his or her own behalf. The amount of the donation is the same regardless of income. It is best that Zakat ulfitr be given one or two days before the Eid prayer. Islamic law insists that Zakat ulfitr should not be given as money but as food (about 2.5 kg) that is similar to the normal diet of the country in which the recipient resides; hence it is possible that it be given as rice in some countries and wheat in others, etc. It is possible that a person gives his own Zakat ulfitr and that of his dependents to one person.

Usually, if a Muslim cannot find anyone who merits being given this charity, he can pay an amount of money that is equivalent to the cost of food to a charity that distributes it as food in other places.

4.6 Conclusions

Any of the current electronic payment schemes (such as those described in chapter 3) can be used by a Muslim as long it does not involve paying interest on payments. However, when a Muslim wishes to buy goods on credit, then a conventional credit card transaction will clearly not be acceptable since it involves paying interest (unless full payment is made shortly after receipt of the bill from the card issuer). In such a case, one of the financial instruments described in section 4.3 could be used instead of a conventional credit card transaction.

Despite the recent growth in the use of e-commerce as a means to perform business transactions, no work appears to have been done to enable Muslims to benefit from this growth by performing Islamic financial transactions electronically. There is therefore a need to fill this gap by designing electronic equivalents of these financial instruments. For example, means for conducting electronic Murabaha transactions at the point of sale and via the Internet would be potentially hugely beneficial both to individual Muslims and to businesses wishing to sell their goods to Muslims.

In parallel with this, Islamic charities still use manual methods to receive and distribute Zakat donations. This method has a high overhead in terms of storage of donated goods and the time consumed in distributing them. On the other hand, recent developments in smart card technology and the popularity of the Internet could be used to make charitable donations easier and more cost effective for donors, charities, and recipients.

These issues are addressed in the remainder of this thesis.

Chapter 5

EXTENDING EMV TO SUPPORT MURABAHA TRANSACTIONS

Contents

This chapter proposes a way of using EMV cards to conduct secure Murabaha transactions at the point of sale. An overview of the EMV card payment transaction procedure is first given in section 5.2. This is followed by an analysis of security and Murabaha requirements, and a description of a possible modification to the EMV specifications that allows an EMV card to conduct a Murabaha sale transaction (section 5.3). Finally, we analyse how the proposed extension to EMV matches the identified requirements (section 5.4). Conclusions are provided in section 5.5.

Note that much of the material in this chapter has previously been described in [7].

5.1 Introduction

Debit/credit card transactions at the POS typically take place between two parties who do not need to have any prior knowledge of each other. This is made possible by the contractual relationship that exists between the card issuer and the acquirer. This relationship between the two banks is established by membership of a payment system, which also provides the network for authorising and clearing payment transactions [92].

In recent years, a variety of security measures have been added to debit/credit cards by the card associations to protect against frauds in POS transactions, including embossed printing, magnetic stripes, holograms, and card verification values. However, a variety of frauds are still possible against POS debit/credit card transactions.

Meanwhile, the advent of smart cards and the growing volume of fraud in credit/debit card transactions at the POS has led the MasterCard and Visa card associations to develop means to replace the magnetic stripe payment cards by chip-based cards. The EMV specifications [27, 28, 29, 30] have been developed, which define the physical and electrical characteristics of the IC card, the IC terminal specifications, and how the IC terminal communicates with the IC card. The latest version of these specifications is known as EMV 2000 [27, 28, 29, 30]. These specifications are now managed by a jointly owned organisation known as EMVCo.

The main objectives for the introduction of EMV by the card associations were [16]:

- To establish standards for global interoperability that allow a single merchant terminal to be used with cards issued by different card issuing banks and different brands.
- To reduce the number of on-line authorisations, leading to reduced acquiring costs for merchants.
- To reduce the fraud costs associated with counterfeit cards.

- To provide a secure platform to enable off-line cardholder verification.
- To allow card issuers to manage card parameters remotely.

While EMV debit card transactions are consistent with Islamic finance principles, since the payment is cleared immediately without payment of interest, EMV credit card transactions are not, since, as discussed in chapter 4, they involve dealing in interest. Therefore, if Islamic finance principles are to be applied to EMV credit card payments, a new and secure card payment process that is consistent with Islamic principles is required. Meeting this need by extending the EMV specifications to enable EMV cards to conduct Murabaha transactions is potentially attractive for a number of reasons.

1. There will potentially be a significant reduction in Murabaha sale transaction expenses.
2. There will undoubtedly be a significant increase in Murabaha transactions, which will result in additional revenue stream for both merchants and issuers.
3. EMV is a good basis for inter-operability and global coverage.
4. Many EMV cardholders already have a trust relationship with their issuer through the use of EMV cards for debit and ATM transactions.
5. Exploiting the existing EMV infrastructure provides a cost-effective solution.

As a result, in this chapter an extension to the EMV specifications is proposed to enable EMV cards to be used to conduct Murabaha transactions.

5.2 The EMV transaction process

In this section we give an overview of the EMV transaction flow, with a focus on the security mechanisms. A more detailed description of these mechanisms can be found in many places — see, for example, [28, 84].

The security services supported by the EMV specifications include the following:

- Card authentication, by means of which a terminal can be certain that a card is genuine,
- Risk management, where the card and the terminal independently decide which transactions need to be referred back to the card issuer at the time of the transaction,
- PIN verification on the card itself, and
- Transaction authorisation by which a card issuer can be certain that a transaction has come from a specific and authentic card, as well as the card ensuring that the approval/decline response has been sent by the authentic issuer.

In line with the model introduced in section 3.2, an EMV card payment transaction involves interactions between four parties: the cardholder, the merchant, the acquirer, and the issuer, with roles as follows.

- **Issuer**: A financial institution that issues a payment card to the cardholder.
- **Cardholder**: An authorised holder of a card supplied by the issuer. The card stores the cardholder's payment data and is capable of generating authentication data and verifying a cardholder's PIN. The cardholder is associated with a Primary Account Number (PAN), stored on the card, that identifies the cardholder account and the issuer. During a transaction, the card has a connection only to the merchant, which passes authorisation messages to the issuer via the acquirer.
- **Merchant**: This is the business that accepts the card payment for purchased goods. It uses a terminal to interact with the card. The terminal also interacts with the issuer (via the acquirer) to receive authorisation for transactions.
- **Acquirer**: This is a financial institution that processes card payment authorisations and payments for the merchant. The acquirer and the issuer communicate via a secure financial network.

5.2.1 EMV transaction security

EMV employs digital signatures and MACs to provide authentication and integrity for card transactions. EMV transaction security is accomplished in two phases:

1. **Authentication.** Card authentication to the terminal is achieved using digital signatures. A chain of trust is established from the card brand, which acts as the top-level certification authority (CA). Each terminal has a trusted copy of the brand CA's public key; moreover the brand CA signs a certificate for each issuer public key, and the card issuer installs such a certificate on each card. The CA's public key and the issuer public key certificate (sent by the card to the terminal) are used by the terminal to obtain a trusted copy of the issuer public key. This public key is then used to verify the authenticity of data stored in the card and/or messages sent by the card to the terminal during a transaction.

   Cardholder authentication is performed by PIN entry at the terminal. The PIN can be verified off-line by the card, or on-line by the issuer. If supported, PIN encryption for off-line PIN verification is performed by the terminal using a card public key with an asymmetric encipherment mechanism [28]. The card may have a separate key pair for PIN encryption, or the signature key pair may be re-used. The card's public key is used by the PIN pad to encrypt the PIN, and the corresponding private key is used by the card to decrypt the encrypted PIN, which is then verified against a copy stored in the card.

2. **Transaction authorisation.** Transactions can be approved either off-line by the card or on-line by the issuer. In both cases, symmetric cryptographic mechanisms are used to generate and verify an Application Cryptogram (AC). The ACs exchanged by the issuer and the card are cryptographically secured using MACs. These are computed using a session key derived from a long term secret key shared between the card and the issuer.

The EMV Specifications allow both phases to be completed off-line, without any communications with the issuer. However, the card or the terminal may force the transaction on-line, in which case an authorisation request message is sent to the issuer.

5.2.2 EMV transaction flow

The EMV transaction flow, as shown in figure 5.1, begins when the buyer card is inserted into the merchant terminal. The terminal reads data from the card for use in its risk management procedures and to establish the card authenticity. There are two types of card authentication, Static and Dynamic Data Authentication (SDA and DDA), where not all cards support DDA. For the card to support DDA it must have its own signature key pair and the means to generate

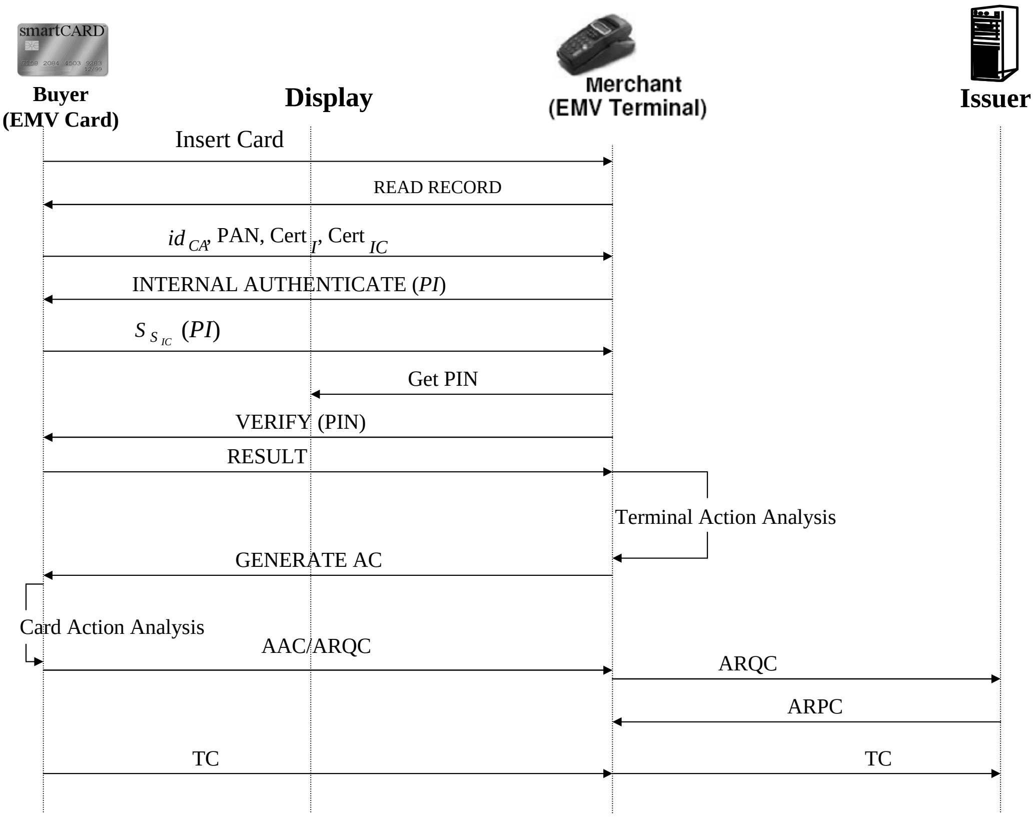

Figure 5.1: EMV transaction

signatures. In both cases the terminal uses a stored copy of the card brand public key to verify the issuer public key certificate; in DDA, the terminal also verifies an issuer-signed certificate for the card public key. In SDA, the terminal verifies the issuer's signature on critical card resident data so that unauthorised alteration of issuer data after personalisation is detected. In DDA, the terminal uses a signature based challenge-response protocol to authenticate the card and verify the integrity of card resident data [28].

Next the Cardholder verification method is invoked to ensure that the person presenting the card is the one to whom the card was issued. For this purpose EMV uses a secret PIN. As described in section 5.2.1, this PIN can be verified either off-line by the card or on-line by the issuer. Upon successful cardholder verification, the terminal then decides whether the transaction should be approved off-line, declined off-line, or an on-line authorisation is necessary. Providing it does not reject the payment at this stage, the terminal passes the payment request to the card in the form of a GENERATE AC command. In response, the card performs 'action analysis'. Depending on the card risk management policy the card's action analysis can return one of three results [29].

1. A Transaction Certificate (TC), when the payment is approved off-line.
2. An Authorisation ReQuest Cryptogram (ARQC), when either the card or the terminal want to go on-line so that the issuer can authorise or reject the transaction. This ARQC is sent by the terminal, via the acquirer, to the issuer. The issuer then responds to the terminal with an Authorisation ResPonse Cryptogram (ARPC) intended for the card to verify and act upon. The terminal then issues a second GENERATE AC command that includes the issuer ARPC and, possibly, a command script, if one was sent by the issuer. If the transaction is approved by the issuer, the card computes and returns a Transaction Certificate (TC).
3. An Application Authentication Cryptogram (AAC), when the request is declined.

Finally, by returning either a TC or an AAC to either the first or second GENERATE AC command issued by the terminal, the card indicates that it is willing to complete transaction processing. If the terminal decides to go on-line, completion shall be achieved when the second GENERATE AC command is issued.

5.3 Using EMV cards for a Murabaha transaction

The method proposed here for using EMV to support a Murabaha transaction involves the same participants and roles as the standard EMV payment process. However, using EMV for a Murabaha transaction requires extensions to the current security model and message flows. As described in chapter 4, a key feature of a Murabaha transaction is that it is composed of two transactions, one between the merchant and the Murabaha finance provider, and the other between the cardholder and the Murabaha finance provider. We suppose here that the card issuer is also acting as the Murabaha finance provider, i.e. it will buy the goods from the merchant and then resell them to the cardholder at a later date. Therefore, using an EMV card for a Murabaha transaction in this way requires on-line communication with the issuer during every transaction. Moreover, the following assumptions are made.

1. The issuer has an agreement with the cardholder to sell him goods on a Murabaha basis. The issuer undertakes to purchase commodities as specified by a cardholder, and then resell them on a Murabaha basis to the cardholder for the cost price plus a margin of profit agreed upon previously by the two parties. The issuer does not make a purchase unless the cardholder requests it and makes a prior promise to purchase.

2. The EMV card is DDA capable.

3. Every acquirer participating in the scheme has their own signature key pair.

4. Every acquirer participating in the scheme has obtained a certificate for their public key from the brand CA, and this certificate is loaded into every merchant terminal supported by this acquirer.

5. Every issuer is equipped with a copy of the public key of the brand CA, as necessary to verify acquirer certificates. In fact this would normally be the case for existing EMV card issuers, since possession of the brand CA public key enables an issuer to verify issuer public key certificates supplied to the issuer by the brand CA.

6. Every terminal is equipped with its own signature key pair. Moreover, every terminal is provided with a certificate for its public signature verification key, signed by the acquirer. [Alternatively the acquirer could append this certificate to every signed message sent from a merchant terminal to an issuer, as it passes through the acquirer network]. Additionally, every terminal has the means to compute signatures, as well as a secure, i.e. confidentiality and integrity protected, location in which to store its private key.

7. In a standard EMV transaction, terminal risk management is performed to protect the system against fraud. It provides positive issuer authorisation for high-value transactions and ensures that transactions initiated from the card go on-line periodically to protect against threats that might be undetectable in an off-line environment [29]. However, since this function is related to off-line transactions, and the proposed extension requires the terminal to go on-line for every transaction, terminal risk management is not performed in the proposed extension to EMV.

5.3.1 Security requirements

No participant in any transaction will want to suffer any loss. Therefore, we need to define precisely the security requirements to meet the needs of the transaction participants. We thus next identify which security services are required for a secure card-based Murabaha transaction. As discussed in section 2.3, security services can be divided into four categories: authentication, confidentiality, integrity, and non-repudiation.

Authentication

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Entity authentication of the merchant terminal to the issuer.* This will enable the issuer to verify that the merchant is as claimed.

2. *Entity authentication of the payment card to the merchant terminal.* The merchant needs to be sure that the payment card is genuine.

3. *Entity authentication of the cardholder to the merchant terminal.* The merchant needs to be sure that the cardholder is the legitimate owner of the payment card.

4. *Origin authentication to the issuer for the payment instruction.* The issuer needs to be sure that the source of the payment instruction is a legitimate card.

5. *Entity authentication of the cardholder to the issuer.* No third party, even if they have stolen a card, should be able authorise a Murabaha transaction.

Confidentiality

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Confidentiality of the PIN.* The cardholder PIN must be kept secret from non-authorised parties.

2. *Confidentiality of the order information.* The cardholder may require that his order information is not available to parties other than the merchant, e.g. for privacy reasons.

Integrity

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Integrity protection for the payment authorisation sent by the cardholder to the issuer.* The cardholder payment authorisation must be protected against alteration, or at least any alteration must be detectable.

Non-repudiation

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Non-repudiation of origin for issuer messages sent to the merchant.* The merchant must have evidence that the issuer has bought the goods and authorises him to supply the goods to the cardholder.
2. *Non-repudiation of origin for cardholder payment authorisation messages sent to the issuer.* The issuer must possess evidence that the cardholder has authorised payment for the goods on a Murabaha basis.

5.3.2 Murabaha requirements

In this section we identify the requirements that enable EMV cards to be used to conduct valid Murabaha transactions, in line with the principles enumerated in section 4.4.2.

1. *Cardholder knowledge of the original price and profit charged.* The cardholder must be aware of the original price of the goods being purchased and the amount of profit the issuer is charging him before buying the goods. This is important for the transaction to be compatible with Murabaha sale conditions.
2. *Issuer ownership of the goods.* The cardholder requires assurance that the issuer owns the goods being offered.

5.3.3 Interaction

We now describe the processes necessary to complete an EMV-based Murabaha transaction.

Figure 5.2 illustrates a transaction in which a cardholder uses his EMV card to purchase goods on a Murabaha basis from a merchant. It begins when the card is inserted into the merchant terminal (step 1). To authenticate the card, DDA is performed.

The terminal first issues the READ RECORD command to the card (step 2) which returns (step 3) the Primary Account Number (PAN), the CA identifier id_{CA}, the issuer public key certificate Cert_I, and the card public key certificate Cert_{IC}.

2. Terminal → Card :   READ RECORD

3. Card → Terminal : id_{CA}, PAN, Cert_I, Cert_{IC}

In order to authenticate the card's public key in Cert_{IC}, the terminal first verifies the issuer public key certificate Cert_I using its copy of the brand CA public key as identified by id_{CA}. The issuer signature on Cert_{IC} is then verified using the Issuer public key obtained from Cert_I.

After successful verification of the card certificate Cert_{IC}, the terminal constructs the Purchase Information (PI), which contains a description of the goods, the price of the goods, and a time-stamp. Moreover, the terminal generates and stores authentication data (*Data*) which contains a random number generated by the terminal, the current date and time, and the card PAN. The terminal then sends a challenge to the card using an INTERNAL AUTHENTICATE command containing $Data||PI$ (step 4). The EMV specification allows the INTERNAL AUTHENTICATE command to carry data strings of size up to 252 bytes [29], which should be sufficient for the authentication data and the additional PI.

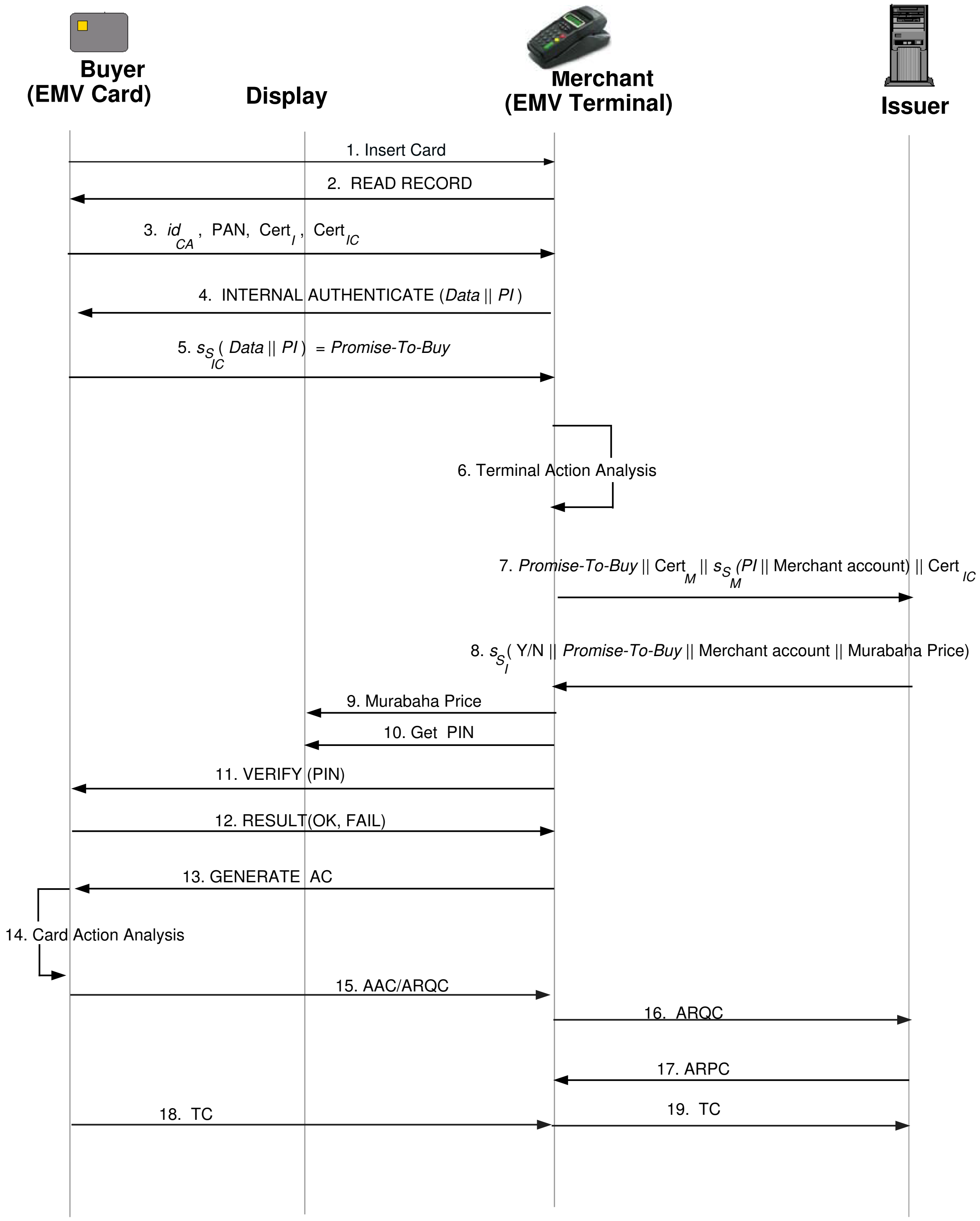
Buyer (EMV Card)
Display
Merchant (EMV Terminal)
Issuer
1. Insert Card
2. READ RECORD
3. id_{CA}, PAN, $Cert_I$, $Cert_{IC}$
4. INTERNAL AUTHENTICATE (*Data* || *PI*)
5. $s_{S_{IC}}$(*Data* || *PI*) = *Promise-To-Buy*
6. Terminal Action Analysis
7. *Promise-To-Buy* || $Cert_M$ || s_{S_M} (*PI* || Merchant account) || $Cert_{IC}$
8. s_{S_I}(Y/N || *Promise-To-Buy* || Merchant account || Murabaha Price)
9. Murabaha Price
10. Get PIN
11. VERIFY (PIN)
12. RESULT(OK, FAIL)
13. GENERATE AC
14. Card Action Analysis
15. AAC/ARQC
16. ARQC
17. ARPC
18. TC
19. TC

Figure 5.2: EMV card-based Murabaha transaction

4. Terminal → Card : INTERNAL AUTHENTICATE ($Data \parallel PI$)

Upon receipt of the message in step 4, the card computes the signature $s_{S_{IC}}(Data \parallel PI)$ and sends it to the terminal (step 5). The card response acts as a promise from the cardholder to buy the goods from the issuer on a Murabaha basis, i.e. *Promise-To-Buy* $= s_{S_{IC}}(Data \parallel PI)$.

5. Card → Terminal : $s_{S_{IC}}(Data \parallel PI)$

The terminal verifies the card certificate Cert_{IC} retrieved in step 3, and uses the card public key obtained from Cert_{IC} to verify the signature $s_{S_{IC}}(Data \parallel PI)$. The terminal also checks that the random number sent in step 4 is present in the signed data.

After successful verification of the signature received in step 5, the terminal action analysis is performed (step 6), where the decision is made as to whether the transaction should be declined off-line, or continued on-line. (Note that all transactions in our extended EMV scheme require on-line processing, and hence the option to continue off-line is not available here).

6. Terminal : action analysis

If the terminal's decision is to reject the transaction, the terminal will issue a GENERATE AC (step 13) asking for an Application Authentication Cryptogram (AAC) from the card.

13. Terminal → Card : GENERATE AC

If, on the other hand, the terminal's decision is to go on-line, the terminal constructs a message that contains the cardholder *Promise-To-Buy*, sent in step 5 and the merchant signature over its bank details ('Merchant account') and PI. The terminal then sends this message (step 7) to the issuer along with its public key certificate Cert_M and the card public key certificate Cert_{IC}.

7. Terminal → Issuer : $\textit{Promise-To-Buy} \parallel \mathrm{Cert}_M \parallel s_{S_M}(PI \parallel \text{Merchant account}) \parallel \mathrm{Cert}_{IC}$

The information sent in step 7 notifies the issuer that a cardholder wishes to buy the goods, with the description given in PI, on a Murabaha basis. The issuer checks the cardholder *Promise-To-Buy*.

If the issuer decides to proceed with the transaction, he first buys the goods from the merchant by crediting the goods price to the 'Merchant account' received in step 7. Then, the issuer constructs and sends a signed authorisation message to the terminal (step 8). This message contains the issuer decision as to whether to proceed with or decline the transaction, and the price at which the goods will be sold to the cardholder ('Murabaha price'). In addition, this message authorises the merchant to deliver the goods to the cardholder. The signature in the message sent in step 8 can be verified by the merchant terminal using the issuer's public key obtained during step 2.

8. Issuer → Terminal : $s_{S_I}(\text{Y/N} \parallel \textit{Promise-To-Buy} \parallel \text{Merchant account} \parallel \text{Murabaha price})$

After successful verification of the signature received in step 8, the terminal displays the 'Murabaha price' to the cardholder (step 9) and requests the buyer to enter his PIN (step 10).

9. Terminal → Display : Murabaha price

10. Terminal → Display : Get PIN

EMV allows the PIN to be verified off-line by the card or on-line by the issuer. In the proposed scheme, both options remain valid. If the decision is to perform off-line PIN verification, the terminal sends a VERIFY command to the card (step 11).

11. Terminal → Card : VERIFY (PIN)

On receipt of the VERIFY command, the card returns a VERIFY response message (step 12), which indicates success or failure. In addition to ensuring that the person presenting the card is the person to whom it was issued, correct PIN entry by the cardholder is regarded as agreement by the cardholder to purchase the goods from the issuer on a Murabaha basis at the specified price (the ‘Murabaha price’).

12. Card → Terminal : RESULT(OK, FAIL)

If, on the other hand, the decision is to perform on-line PIN verification, the PIN shall be protected upon entry by encipherment according to ISO 9564-1 [51], and the terminal shall transmit the PIN to the issuer according to the payment system rules.

After successful cardholder verification, the terminal sends a GENERATE AC command to the card (step 13).

13. Terminal → Card : GENERATE AC

Next, the card action analysis process (step 14) begins, where a card performs its own risk management to protect against fraud or excessive credit risk [29]. Details of card risk management algorithms within the card are specific to the issuer.

14. Card : Action Analysis

A card may decide to complete the transaction on-line or reject the transaction off-line. If the outcome of the decision is to reject the transaction off-line, an AAC is returned by the card to the terminal (step 15) and the transaction ends. If the outcome of the decision is to complete the transaction on-line, an ARQC is generated and sent by the card to the terminal (step 15) and then forwarded to the issuer (step 16).

15. Card → Terminal : AAC/ARQC

16. Terminal → Issuer : ARQC

The issuer responds to the ARQC with an ARPC (step 17).

17. Issuer → Terminal : ARPC

If the transaction is accepted, the card generates a TC (step 18) and sends it to the terminal which forwards it to the issuer (step 19) and the transaction ends.

18. Card → Terminal : TC

19. Terminal → Issuer : TC

An advantage of the proposed extension to EMV is that the modified scheme does not involve any changes to the card functionality, i.e. an existing EMV card could be used to perform a Murabaha transaction. The proposed changes affect only the merchant, the acquirer, and the issuer.

5.3.4 Changes to standard EMV

In the proposed scheme, most of the steps in the transaction procedure are similar to those in the standard EMV payment process. However, some messages have been modified and new messages have been added to satisfy the Murabaha sale rules.

1. The INTERNAL_AUTHENTICATE command (step 4) includes the Purchase Information (PI) in addition to the authentication data required by the EMV specifications. Therefore, the response signature (step 5) computed by the card must be computed on the PI in addition to the authentication data.

2. Completely new messages (steps 7 and 8) have been added between the merchant and the issuer. These messages are necessary to complete the first transaction as specified

in section 4.4.2, which is not part of the EMV standard.

3. Extending the EMV standard payment process requires that the terminal firmware be upgraded to allow the storage of a terminal-specific signature key pair. Additionally, the acquirer and the issuer transaction processing software must be modified to carry the new EMV messages.

5.4 Analysis

In this section, we examine to what extent the identified requirements outlined in section 5.3.1 and section 5.3.2 are met by the extended EMV transaction.

5.4.1 Authentication

1. *Entity authentication of the merchant terminal to the issuer.* The "standard" EMV transaction does not provide mechanisms to authenticate the merchant terminal to the cardholder or to the issuer. However, in our extension to EMV, the merchant terminal has its own public key certificate Cert_M, which is sent along with the terminal signature in step 7 to the issuer. The issuer verifies the merchant certificate and that the merchant signature is valid; if this verification process fails then the transaction is not completed. The merchant signature is computed on a data string that includes a current date/time-stamp, and hence entity authentication of the merchant is provided to the issuer. (Nevertheless, it is still possible for the cardholder to interact with a different merchant than intended, since a dishonest terminal could pretend to the cardholder that the issuer verification process has been successfully completed. However, this is also the situation for EMV as it operates at present.)

2. *Entity authentication of the payment card to the merchant terminal.* This is performed using DDA. The merchant terminal verifies that the card certificate Cert_{IC} retrieved in

step 3 is valid. In addition, the terminal verifies the validity of the card response (step 5).

3. *Entity authentication of the cardholder to the merchant terminal.* This is accomplished using PIN entry, which can be verified either off-line by the card or on-line by the issuer. The EMV Specifications limit the number of unsuccessful PIN entries [29].

4. *Origin authentication to the issuer for the payment instruction.* The issuer can use the *Promise-To-Buy* and Cert$_{IC}$ received in step 7 to verify the origin of the payment instruction. Moreover, the ARQC (step 16) and TC (step 19) are generated using a key shared between the card and the issuer.

5. *Entity authentication of the cardholder to the issuer.* The cardholder PIN is assumed to be a secret known only to the cardholder. Therefore, nobody but the cardholder can authorise an EMV card-based Murabaha transaction. This is based on the assumption that the merchant terminal displays the correct transaction data to the cardholder. This is a standard assumption for merchant terminals, where the cardholder is required to trust the reputation of the merchant when using a card in the merchant premises. In addition a fraudulent merchant is relatively easy to track and prosecute.

5.4.2 Confidentiality

1. *Confidentiality of the PIN.* If PIN verification is performed using the card (off-line), then the terminal can encrypt the cardholder PIN when sent from the PIN pad to the card. The card could use either a separate key pair or its signature key pair for PIN encryption. The card public key is used by the terminal (PIN pad) to encrypt the PIN, and the private key is used by the card to decrypt the encrypted PIN [28] in order to verify it.

2. *Confidentiality of the order information.* Order information is not encrypted and can be read by the terminal, the acquirer, and the issuer. Therefore, this requirement is not satisfied, although this is also the case for ‘standard’ EMV transactions.

5.4.3 Integrity

1. *Integrity protection for the payment authorisation sent by the cardholder to the issuer.* This requirement is met, because a MAC is used to protect the integrity of the AC generated by the card. The card and the issuer can verify MACs generated by each other using a shared secret key.

5.4.4 Non-repudiation

1. *Non-repudiation of origin for issuer messages sent to the merchant.* The merchant can verify the message received in step 8 from the issuer. If it verifies successfully, then it provides a payment guarantee from the issuer, because it contains the signed issuer agreement to buy the goods, the goods description, and the merchant bank details ('Merchant account').

2. *Non-repudiation of origin for cardholder payment authorisation messages sent to the issuer.* Entry of the correct PIN by the cardholder upon the display of the 'Murabaha price' will trigger the generation of an ARQC and a TC by the card. The TC sent to the issuer in step 19 can be regarded as evidence of cardholder authorisation; however, since the TC is generated using a secret key shared by the issuer and the card, it is of limited value in providing non-repudiation unless it is combined with evidence from audit trails, e.g. held by the acquirer.

5.4.5 Murabaha

1. *Cardholder knowledge of the original price and profit charged.* Since the cardholder chose the goods that are to be bought on a Murabaha basis from the merchant, we assume that he/she is aware of the original price of the goods. Moreover, in step 8 of the interaction, the terminal receives confirmation of the issuer willingness to sell the goods on a Murabaha basis with the 'Murabaha price' being the price at which

the goods should be sold to the cardholder. This is followed by the terminal asking the cardholder for PIN entry. Entry of the correct PIN is taken as confirmation of the willingness of the cardholder to continue the transaction.

2. *Issuer ownership of the goods.* By sending the signed message in step 8 to the terminal, the issuer provides an undeniable assurance to the terminal that it has purchased the goods. The buyer has to trust the merchant terminal to display the 'Murabaha price' only if the message in step 8 is verified correctly and indicates that the issuer has credited the 'Merchant account'.

5.4.6 Mobility and user privacy

The proposed scheme does not restrict the buyer to a fixed location; in fact, the buyer can use the EMV card in any terminal that is capable of executing a Murabaha transaction. On the other hand, user privacy is not protected since the buyer order information is not encrypted and can be read by the terminal, the acquirer, and the issuer. Moreover, the PAN of the buyer is sent to the issuer for every transaction.

5.4.7 Denial of service attack

The proposed scheme has the following weakness which may result in the issuer being the target of a denial of service attack. An attacker who manages to get hold of an EMV card can start many Murabaha transactions without completing them. The attack works as follows.

An attacker starts a Murabaha transaction with a merchant, and steps 1-9 of the transaction are executed normally. However, at step 10, the cardholder verification process will fail because the attacker does not know the real PIN. By that time the issuer has bought the goods from the merchant (step 8), on the basis of the Promise-To-Buy message received from the card (step 7) which has been sent before authenticating the cardholder. Such an attack will therefore result in a significant loss to the issuer, who will be persuaded to buy potentially large quantities of goods for which no buyer exists.

We therefore now propose a simple modification to the EMV Murabaha scheme which improves its robustness against such an attack. The revised scheme is similar to that proposed in section 5.3.3, except for the following modification.

Additional cardholder verification steps are added after step 6 to ensure that an authentic cardholder is responsible for the card sending a Promise-To-Buy message.

6. Terminal : action analysis

Terminal → Display : Get PIN

Terminal → Card : VERIFY (PIN)

On receipt of the VERIFY command, the card returns a VERIFY response message, which indicates success or failure. In addition to ensuring that the person presenting the card is the person to whom it was issued, correct PIN entry by the cardholder is regarded as agreement by the cardholder to send a Promise-To-Buy message to the issuer (step 7).

Card → Terminal : RESULT(OK, FAIL)

After successful cardholder verification, the terminal then sends the following message (step 7).

7. Terminal → Issuer : $\textit{Promise-To-Buy} \parallel \mathrm{Cert}_M \parallel s_{S_M}(PI \parallel \text{Merchant account}) \parallel \mathrm{Cert}_{IC}$

and the rest of the transaction is the same as in section 5.3.3.

5.5 Conclusions

In this chapter we have proposed an extension to the EMV specifications to enable cards to conduct Murabaha transactions at POS terminals. We described the new payment method in detail, and explained how it meets the identified security and Murabaha requirements. In the proposed extension to EMV, most of the transaction procedures are similar to those in the standard EMV payment process. However, additional messages have been included to satisfy the Murabaha sale rules. The proposed extension can be seen as a step towards adapting existing deployed electronic payment schemes to the Islamic economic system.

The proposed modifications to the EMV process do not affect the card functionality. Therefore issuers would not need to issue new EMV cards in order to support Murabaha transactions. However, extending the EMV standard payment process requires that the terminal firmware be upgraded to allow the storage of a terminal-specific signature key pair. Additionally, the acquirer and the issuer transaction processing software must be modified to carry the additional EMV messages required to complete the transaction. The cost of these changes could be covered by the increase in Murabaha transactions, which will result in an additional revenue stream for both merchants and issuers.

Chapter 6

A SECURE ELECTRONIC MURABAHA TRANSACTION

Contents

This chapter proposes a method to conduct secure electronic Murabaha transactions on the Internet. First, a model for a secure electronic Murabaha transaction is provided (section 6.2). We then identify the security and Murabaha requirements (sections 6.3 and 6.4) for an electronic Murabaha transaction, and present the Secure Electronic Murabaha Transaction (SEMT) scheme (section 6.5), designed to address the identified requirements. Finally, we analyse how the proposed protocol matches the identified requirements (section 6.6). Conclusions are provided in section 6.7.

Note that much of the material in this chapter has previously been described in [8].

6.1 Introduction

As we discussed in chapter 3, the emergence of the Internet has led to the development of a variety of electronic payment protocols designed with the objective of reducing both the cost of buying goods as well as the merchant's cost of selling. However, as explained in section 4.4, none of these protocols have been designed to conduct electronic transactions based on Murabaha principles.

Many electronic payment schemes are based on the credit card transaction model, where buyers make a purchase from a merchant, the merchant receives payment from their acquiring bank, the acquiring bank is recompensed by the buyer's issuing bank, and the buyer eventually pays the issuing bank, often with interest charges included. However, in a Murabaha transaction, the sale is concluded using two transactions, one between the bank and the merchant and the other between the buyer and the bank.

In this chapter a new scheme is proposed that allows a buyer to conduct a Murabaha transaction electronically. Specifically, the scheme allows a Murabaha transaction to be conducted in an e-commerce setting. Cryptographic techniques are used to provide security features for the transaction. The protocol is designed to satisfy the principles for a Murabaha transaction, as specified in section 4.4.

6.2 Electronic Murabaha Transaction Payment Model

In this section we describe our model for an electronic Murabaha transaction. The model identifies the entities involved and includes a brief description of their interactions.

6.2.1 Entities involved

An electronic Murabaha transaction involves interactions between three parties: the buyer, the merchant and the provider. Their roles are straightforward.

- **Buyer (B)**: This is the entity that wishes to buy goods from a merchant, but does not have the cash immediately available to complete the transaction.
- **Merchant (M)**: This is the entity that offers the goods which the buyer wishes to purchase. The merchant has a formally established agreement with the provider that defines the trust and liability relationship.
- **Provider (P)**: This is a financial institution that acts as an intermediary between the buyer and the merchant. It undertakes the purchase of commodities as specified by a buyer, and then resells them on a Murabaha basis to the buyer for the cost price plus a margin of profit agreed upon previously by the two parties. It does not make a purchase unless the buyer requests it and makes a prior promise to purchase. The provider has a formally established agreement with the buyer that defines the trust and liability relationship.

6.2.2 Interactions

In this model for an electronic Murabaha transaction, we first suppose that a buyer, shopping at an Internet merchant site, chooses to pay using Murabaha through a specified provider. The merchant redirects the buyer to the provider so that the provider can complete the purchase on the buyer's behalf. If the provider chooses to proceed with the sale, the provider first calculates the profit for the transaction and then sends a promise to sell the goods to the buyer once they have been bought. In return, the buyer promises to buy the goods from the provider on a Murabaha basis for the cost of the goods plus the agreed upon profit. This promise is not binding on either the buyer or the provider, and is not an actual sale. It is just a promise to effect a sale in future on the basis of Murabaha. At this stage the relationship between the provider and the buyer is that of a promisor and a promisee.

Based on the goods description supplied to the provider by the merchant, the provider communicates with the merchant Internet server and complete the purchase of the goods. One possible additional benefit of this procedure is that the provider is in a better position to obtain discounts from the merchant, who in most cases will prefer dealing with a provider, as the merchant will receive payment more quickly and with less risk.

Once the purchase of the goods is settled between the provider and the merchant, the provider notifies the buyer of completion of the purchase. The buyer then sends a payment authorisation to the provider, in order to purchase the goods from the provider on a Murabaha basis. The buyer can then obtain the goods from the merchant, although the goods remain the property of the provider until the transaction is completed. At some later stage (the time of which is specified in the payment authorisation provided by the buyer to the provider), the payment is actually made by the buyer to the provider.

6.3 Security requirements

As with any payment system, the scheme we propose provides the potential for considerable financial gains for those who attack it successfully. There is thus a need to protect participants in the proposed scheme from any loss. We therefore identify what security services are required for a secure electronic Murabaha transaction.

6.3.1 Authentication

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Entity authentication of the provider to the buyer.* This will enable the buyer to verify that the provider is as claimed.
2. *Entity authentication of the buyer to the provider.* The provider needs to be sure that the buyer is legitimate and has an account with the provider.
3. *Origin authentication to the provider for the payment authorisation.* The provider needs to be sure that the source of the payment authorisation is a genuine buyer.
4. *Entity authentication of the merchant to the provider.* This will enable the provider to verify that the merchant is as claimed.

6.3.2 Confidentiality

Confidentiality for information exchanged between the transaction participants is needed. This is especially important in the Internet environment where information may travel through network segments that are not necessarily trusted. This security service can be sub-divided into the following.

1. *Confidentiality of the buyer payment authorisation.* The buyer authorisation must be kept secret from non-authorised parties.
2. *Confidentiality of the order information.* The buyer may require that his order information is not available to parties other than the provider, e.g. for privacy reasons.

6.3.3 Integrity

This security service can be sub-divided into the following.

1. *Integrity protection for the payment authorisation sent by the buyer to the provider.* The buyer payment authorisation must be protected against alteration, or at least any

alteration must be detectable.

2. *Integrity protection for the transaction data sent between the participants.* Such transaction data should be protected against modification and replay.

6.3.4 Non-repudiation

In the context of our transaction model, this security service can be more specifically defined as follows.

1. *Non-repudiation of origin for the buyer payment authorisation sent to the provider.* The provider must have evidence that the buyer has authorised payment for the goods on a Murabaha basis.

6.4 Murabaha requirements

In this section we identify the requirements that must be fulfilled for a buyer to conduct a valid Murabaha transaction over the Internet. These requirements have been derived by applying the general Murabaha requirements given in section 4.4 to the specific transaction model given in section 6.2.

1. *Buyer knowledge of the original price and profit charged.* The buyer must be aware of the original price of the goods being purchased and the amount of profit the provider is charging him before buying the goods. This is important for the transaction to be compatible with the Murabaha sale conditions.

2. *Provider ownership of the goods.* The buyer requires assurance that the provider owns the goods being offered.

6.5 The SEMT Protocol

We now describe the proposed Secure Electronic Murabaha Transaction (SEMT) protocol in detail. SEMT (as shown in figure 6.1) consists of five phases: the *Registration phase*, in which the provider registers the buyer and issues him digital certificates that allow the buyer to participate in the SEMT protocol; the *Transaction request phase*, in which the buyer finds goods he wishes to buy at an Internet merchant site, and decides to use Murabaha to pay for the goods; the *Promising phase*, invoked by the provider, wherein the provider promises to sell the buyer the goods he is interested in, while the buyer promises to buy the goods from the provider, once the provider has ownership; the *Purchase phase*, invoked by the provider, wherein he buys the goods requested by the buyer from the merchant, and the *Murabaha phase*, invoked by the provider, wherein the buyer verifies the provider's ownership of the goods offered and sends authorisation to the provider to buy the goods at the agreed price.

The SEMT protocol makes use of the concept of a 'digital envelope' for data confidentiality, which combines symmetric and asymmetric encryption. The sender of a message requiring confidentiality protection first generates a random ('one time') secret key for use with a symmetric encryption algorithm. This key is then used with the symmetric algorithm to encrypt the message. The secret key is then encrypted with the public asymmetric encryption key of the intended recipient, and the digital envelope then consists of the concatenation of the asymmetrically encrypted random secret key with the symmetrically encrypted message. The digital envelope on message M to be sent to entity X is denoted by $\mathrm{Env}_X(M)$, where $\mathrm{Env}_X(M) = E_K(M) \parallel e_{P_X}(K)$ and K is a randomly chosen secret session key.

6.5.1 Specific Requirements

In order to execute the protocol, the following requirements must be satisfied by the SEMT participants.

1. Each participant must have two asymmetric key pairs: one pair used for encryption and decryption and the other used for the creation and verification of digital signatures. This requirement applies not only to buyers and merchants but also to the provider.

2. The buyer, the merchant and the provider must be using the same public key encryption scheme and the same digital signature scheme.

3. Since this is the first attempt to design an electronic equivalent to manual Murabaha transactions, it was important to win the confidence of financial institutions by offering the highest possible compliance with manual Murabaha transactions. Therefore, for the purposes of this specification of SEMT we suppose that SET is used by the provider to secure payments to the merchant. It is therefore assumed that providers and merchants have obtained SET certificates, generated by one of the SET brand CAs, for their public keys. Both provider and buyer are assumed to possess two such certificates, one for the signature verification public key, and one for the public encryption key. The buyer and provider are also assumed to possess the public verification key for the SET brand CA.

4. The buyer must have a pre-established account with the provider, which can be used to pay for the goods — the term ‘account number’ is used below to refer to an unambiguous number for this account, e.g. as would be fulfilled by an IBAN (International Bank Account Number) [31].

6.5.2 Registration phase

This phase contains five steps. Typically this phase will involve a face to face meeting, or some other form of trusted exchange (e.g. modelled on the SET registration procedure). If the buyer B already has an established relationship with the provider P for other purposes then certain aspects of this process might be simplified. First, in step 1, the provider verifies the buyer’s identity (by some provider-specific means).

1. Buyer $\leftrightarrow$ Provider : Authenticate B

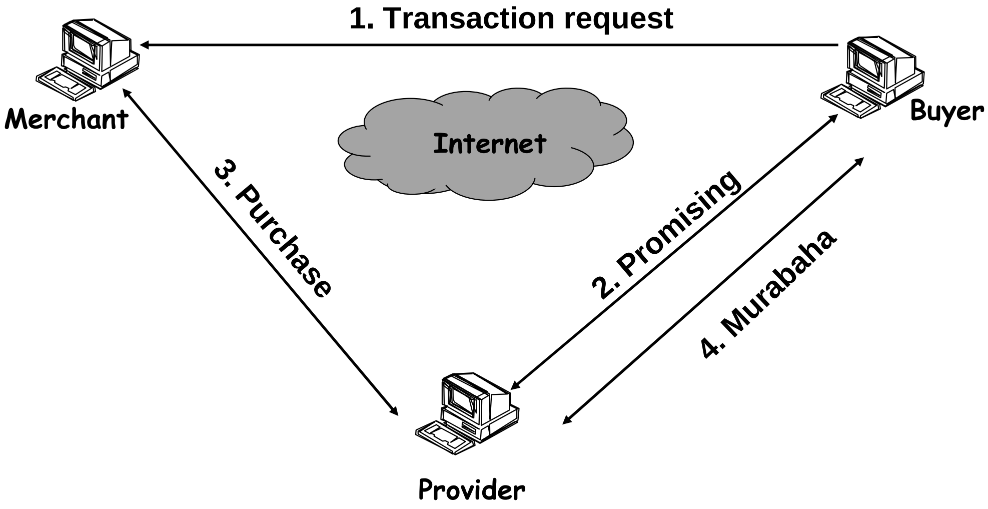

Figure 6.1: SEMT transaction

Then, in step 2, the buyer provides his ‘account number’ (see requirement 4 in section 6.5.1) to the provider, including any other information necessary for the provider to make deductions from this account.

2. Buyer → Provider : account number

After verifying the validity of the account number (and checking the association between this account and the buyer), the provider generates a unique identifier id_B for B, for use in this protocol, and creates a personalised copy of a ‘buyer SEMT application’ for installation by the buyer. This PC application will be used by the buyer to make SEMT purchases, and is assumed to contain copies of the buyer identifier id_B, the provider public encryption key P_P, the SET brand CA signed certificate for the provider public encryption key Cert_{P_P}, and the provider public signature verification key V_P. In step 3 the provider sends this application to the buyer.

3. Provider → Buyer : buyer SEMT application

As step 4, the buyer then installs this application on his/her PC, and the application automatically generates and stores two key pairs for the buyer (one for digital signatures and one for encryption/decryption).

4. Buyer: install application and generate key pairs

In step 5, the buyer transfers the public keys to the provider, which creates certificates for them both, and returns these certificates to the buyer.

5. Buyer ↔ Provider : Submit public keys and obtain Cert_{P_B} and Cert_{V_B}

The buyer then stores these certificates in the SEMT application.

6.5.3 Transaction request phase

This phase begins when a buyer, shopping at an Internet merchant site using a web browser, indicates that he wishes to make a specific purchase using SEMT through a specified provider (step 1). The buyer also provides the provider public key certificate Cert_{P_P} to the merchant.

1. Buyer → Merchant : Request to pay using SEMT through $P \parallel \text{Cert}_{P_P}$

In return, the merchant prepares a quotation to be presented to the provider to enable the purchase of the specified goods by the provider on behalf of the buyer. The quotation prepared by the merchant (step 2) contains data related to the goods being offered, namely the specified goods information ('items'), the price of the goods ('price'), the time of expiry of the quotation (‘expiry’), and the address of the merchant web site (‘Merchant_URL’). Additionally, the merchant includes in the quotation his identifier id_M and the provider identifier id_P.

2. Merchant : Generate ‘quotation’ = ($id_M \parallel id_P \parallel$ items $\parallel$ price $\parallel$ expiry $\parallel$ Merchant_URL)

In order to protect the quotation contents against disclosure to eavesdroppers, the quotation is encrypted in a digital envelope constructed using the public encryption key of the provider P_P. The merchant can obtain this key by verifying Cert$_{P_P}$, as provided in step 1 of this phase. After preparing the quotation, the merchant sends the above quotation to the provider using the buyer’s web browser (step 3). This is achieved using the HTTP redirect facility; that is, the merchant redirects the buyer’s web browser to the provider web server, and at the same time sends the quotation to the provider server.

3. Merchant → Provider : Env$_P$(quotation)

6.5.4 Promising phase

This phase commences when the provider receives the encrypted quotation and attached certificate from the merchant (as a result of step 3 of the Transaction request phase). As step 1 of this phase, the provider decrypts the quotation using its private decryption key. The data in the quotation is then checked, including that the identifiers are as expected, and that the ‘expiry’ time has not passed. Step 1 can thus be summarised as:

1. Provider : Decrypt quotation (from step 3 of the Transaction request phase).

If the quotation contains a valid set of data, the expiry date is still valid, and the provider chooses to proceed with the sale, then the provider will first calculate his profit (‘profit’), and select and store a unique transaction identifier ‘Trans_ID’. Then, in step 2, the provider generates a signed *Promise-To-Sell* message to be sent to the buyer. That is:
Promise-To-Sell = $s_{S_P}(id_P \,||\, id_B \,||$ Trans_ID $||\, T \,||$ items $||$ price $||$ profit $||$ Due_Date).

In addition to the identifiers id_P and id_B, this message includes 'Trans_ID', used to uniquely identify the transaction; the time T that the message is created; 'items', a description of the goods being purchased (as given in 'quotation'); 'price', the price charged by the merchant for the goods; 'profit', the value of the profit requested by the provider; and 'Due_Date', i.e. the date on which the provider expects the buyer to pay for the goods. The inclusion of the profit requested by the provider ('profit') in this message is to satisfy the conditions set out in section 4.4.2. Finally, the message is encrypted by the provider and sent to the buyer. This message promises the buyer that the provider will sell the requested goods to the buyer, once bought from the merchant. That is, step 2 can be summarised as:

2. Provider → Buyer : Env_B (*Promise-To-Sell*)

After receiving the message in step 2, decrypting it, and then verifying the provider signature, the buyer checks that the goods promised by the provider ('items') are the requested goods. Also, the buyer will check that both the profit ('profit') and the due date ('Due_Date') offered by the provider are acceptable to him. If the buyer chooses to proceed with the sale, then he has to promise the provider that he will buy the goods once the provider has ownership of them. This is achieved by generating a *Promise-To-Buy* message, which is a buyer signature over the same information that was signed by the provider to create the *Promise-To-Sell*. I.e. $\textit{Promise-To-Buy} = s_{S_B}(id_B \,||\, id_P \,||\, \text{Trans\_ID} \,||\, T \,||\, \text{items} \,||\, \text{price} \,||\, \text{profit} \,||\, \text{Due\_Date})$. Note that T here represent the time that the buyer created the *Promise-To-Buy* message. As step 3, the buyer then encrypts the *Promise-To-Buy* message and sends it, along with the buyer's signature verification key certificate, to the provider.

3. Buyer → Provider : $\text{Env}_P(\textit{Promise-To-Buy}) \,||\, \text{Cert}_{V_B}$

6.5.5 Purchase phase

This phase involves the transfer of five messages. When the provider receives the encrypted and signed *Promise-To-Buy* message from the buyer, the provider decrypts it and then verifies the buyer signature to check the origin and integrity of the received message. Moreover, the provider checks that the value of 'Trans_ID' is the same as the value that was sent in step 2 of the Promising phase.

Assuming that the provider is using the SET protocol to submit payment to the merchant, the provider generates a SET protocol *Purchase-Request* message based on the goods description ('items') contained in the *Promise-To-Buy* received from the buyer. As step 1 of this phase, the provider sends this message to the merchant web server at the address ('Merchant_URL') specified previously in step 3 of the Transaction request phase, i.e.:

1. Provider → Merchant : *Purchase-Request*

In step 2, the merchant uses the *Purchase-Request* message to produce a SET protocol *Auth-Request* message asking for a payment authorisation from the acquirer. Note that a SET protocol option must be set to make the *Auth–Request* / *Auth–Response* message exchange result in an actual transfer of money, i.e. simultaneous authorisation and capture.

2. Merchant → Acquirer : *Auth-Request*

Step 3 involves the acquirer sending a message via the financial network to obtain a payment authorisation. If a payment authorisation is successfully received, the acquirer generates and digitally signs a SET protocol authorisation response message *Auth-Response*, indicating success or failure and the actual captured amount. This is then sent to the merchant.

3. Acquirer → Merchant : *Auth-Response*

When the merchant receives the authorisation response message *Auth-Response*, the acquirer signature is first verified. If the signature verifies successfully, in step 4 the merchant then generates and digitally signs a SET protocol *Purchase-Response* message and transmits it to the provider:

4. Merchant → Provider : *Purchase-Response*

In order to give the buyer evidence that the provider has bought the goods, the provider must, in step 5, forward the response message *Purchase-Response* received from the merchant to the buyer along with the merchant signature verification key certificate Cert_{V_M}:

5. Provider → Buyer : *Purchase-Response* $\|$ Cert_{V_M}

This message is used to convince the buyer to complete the next phase. The buyer can be sure that the acquirer has authorised the payment since this is indicated in the data fields *AuthStatus* and *CapStatus* within the SET *Purchase-Response* message.

SET has been used here primarily for the purposes of illustration to bankers in the Islamic banking community who were skeptical the ability of the technology to satisfy the rules of a Murabaha transaction. Whilst the Purchase phase described above is based on SET, other methods of Internet payment could easily be used to complete the transaction.

For example, 3-D Secure (see section 3.4.2) can be used by the provider to submit payment to the merchant. When the provider receives the encrypted and signed *Promise-To-Buy* message from the buyer, the provider decrypts it and then verifies the buyer signature to check the origin and integrity of the received message. Moreover, the provider checks that the value of 'Trans_ID' is the same as the value that was sent in step 2 of the Promising phase. The provider submits his payment information to the merchant site which queries the Visa directory server to determine whether the provider is enrolled in the 3-D Secure scheme. If so, then the directory server provides the merchant site with the URL of the ACS of the provider's card issuer. The merchant redirects the provider to the ACS URL to authenticate the provider.

If the provider authentication is successfully completed, the ACS formats an authentication response, which is digitally signed by the ACS. The ACS redirects the provider back to the merchant, and the authentication response is simultaneously transferred to the merchant. When the merchant server receives the authentication response, the merchant server verifies the digital signature of the issuer, and submits an authorisation request to the acquirer. This authorisation request includes the CAVV and a unique transaction identifier. The acquirer passes this into the Visa payment network, where a server verifies the CAVV received with the copy stored on the AHS, and forwards the authorisation request to the issuer, who processes the transaction in the normal way.

6.5.6 Murabaha phase

Once the buyer receives the message sent in step 5 of the Purchase phase, the buyer decrypts it and then verifies the merchant signature. The buyer retrieves *AuthStatus* and *CapStatus* from the *Pay-Response* and checks that the acquirer has authorised the payment, i.e. that the provider has bought the goods.

If convinced, the buyer sends his payment authorisation to the provider using the following *Murabaha-Payment* message:

1. Buyer → Provider :
$\mathrm{Env}_P(s_{S_B}(id_B \,||\, id_P \,||\, \text{account number} \,||\, \text{Trans\_ID} \,||\, T \,||\, \text{items} \,||\, \text{price} \,||\, \text{profit} \,||\, \text{Due\_Date}))$

Upon receipt of the buyer payment authorisation, the provider first decrypts it and then verifies the buyer signature. After verifying that the value of 'Trans_ID' is the same as the value that was sent in step 2 of the Promising phase, and that 'account number' is valid and associated with id_B, the provider instructs the merchant to dispatch the goods to the buyer address.

6.6 Analysis

In this section, we examine to what extent the identified requirements outlined in section 6.3 and section 6.4 are met by the SEMT protocol.

SEMT is similar to the SET protocol in that it provides confidentiality and integrity for payment information using public key cryptography. Moreover, it uses digital signatures to authenticate all parties involved in the payment process. However, there are two important differences. The first is that, unlike SET, the buyer does not submit his payment information through the merchant. The second difference is that SEMT involves two separate transactions, one between the provider and the merchant, and the other between the buyer and the provider.

SEMT also has some similarities to 3-D Secure, as described in section 3.4, in that the issuer must be involved in every transaction. However, 3-D Secure does not require the buyer to have a digital certificate. Instead another security mechanism, namely SSL, is used to secure communication between the cardholder and the merchant.

6.6.1 Authentication

Authentication in SEMT is accomplished using digital signatures and public key certificates. An attacker cannot impersonate another participant except by stealing that participant's private signature key.

1. *Entity authentication of the provider to the buyer.* This requirement is met, because the buyer can verify the provider digital signature received in the *Promise-To-Sell* message of the Promising phase, using the provider signature verification key. The freshness

of the message can be determined by checking that the time stamp T is within its acceptance window.

2. *Entity authentication of the buyer to the provider.* This requirement is met, because the provider can verify the buyer digital signature received in the *Promise-To-Buy* message of the Promising phase, using the buyer verification signature verification key. The freshness of the message can be determined by checking the Trans_ID, previously generated by the provider (which acts here as a nonce).

3. *Origin authentication to the provider for the payment authorisation.* Buyer authorisation is performed by including the account number in the signed *Murabaha-Payment* message of the Payment phase, and no-one but the buyer has the private key necessary to create the required signature.

4. *Entity authentication of the merchant to the provider.* It is assumed (see section 6.5.1) that the provider is using the SET protocol to secure payment of the purchased goods from the merchant. Therefore, authentication of the merchant to the provider is provided within the SET protocol.

6.6.2 Confidentiality

A significant advantage of SEMT is that the buyer does not need to send any private information via the merchant, unlike in conventional e-commerce schemes where a credit card number is sent to a merchant protected using SSL/TLS. This avoids any concerns regarding the ability of the merchant to store buyer private information in a secure manner. Moreover, this keeps the identity of the buyer anonymous to the merchant, since the buyer does not need to reveal his identity to anyone but the provider.

1. *Confidentiality of the buyer payment authorisation.* The buyer payment authorisation sent in step 1 of the Murabaha phase is protected against disclosure to eavesdroppers

since it is encrypted in a digital envelope constructed using the public encryption key of the provider P_P, an authentic copy of which is possessed by the buyer.

2. *Confidentiality of the order information.* When transmitted, all transaction information (e.g. pricing and payment details) in SEMT is encrypted. However, order information can be read by merchant, acquirer, and the issuer, but the merchant will not be able to associate it with a specific buyer.

6.6.3 Integrity

Integrity in the secure electronic Murabaha transaction is provided using digital signatures.

1. *Integrity protection for the payment authorisation sent by the buyer to the provider.* The payment authorisation sent to the provider in message 1 of the Murabaha phase is digitally signed by the buyer. The provider can verify the signature using the buyer signature verification key obtained from Cert_{V_B}, itself sent in message 3 of the Promising phase. Therefore, the provider will discover any unauthorised modification to the buyer payment authorisation.

2. *Integrity protection for the transaction data sent between the participants.* Messages exchanged between the participants that contain transaction data are digitally signed by their originator, thus providing integrity protection. Moreover, messages exchanged between the provider and the buyer include a unique Trans_ID, and therefore any attempt to re-use messages from previously completed transactions will be detected by the provider.

6.6.4 Non-repudiation

1. *Non-repudiation of origin for the buyer payment authorisation sent to the provider.* This requirement is met because the buyer signs the message sent to the provider in step 1

of the Murabaha phase, which associates the buyer unique identifier id_B and the buyer account number with the purchased goods. The provider can verify the buyer signature using the verification key V_B found in the buyer certificate.

6.6.5 Murabaha

1. *Buyer knowledge of the original price and profit charged.* This requirement is met because the *Promise-To-Buy* sent in step 3 of the Promising phase to the provider contains the original price and the amount of profit the provider is adding.

2. *Provider ownership of the goods.* This requirement is met because the buyer can verify the merchant signature on the *Purchase-Response* message, where a copy is forwarded to the buyer in step 5 of the Purchase phase, and that *AuthStatus* and *CapStatus* are set.

6.6.6 Mobility and user privacy

The proposed scheme assumes that the buyer is using a PC which contains his public key pair. Therefore, the buyer is restricted to use of this particular machine whenever a Murabaha transaction is to be made. On the other hand, an advantage of SEMT is that the buyer does not need to send any private information via the merchant, unlike in conventional e-commerce schemes where a credit card number is sent to a merchant protected using SSL/TLS. This avoids any concerns regarding the ability of the merchant to store buyer private information in a secure manner. Moreover, this keeps the identity of the buyer anonymous to the merchant, since the buyer does not need to reveal his identity to anyone but the provider.

6.7 Conclusion

In this chapter, we have proposed the Secure Electronic Murabaha Transaction (SEMT) protocol, which provides a secure Murabaha sale service on the Internet. We described the protocol in detail, and explained how it meets the identified Murabaha and security requirements.

The protocol uses the Internet to exchange messages between the participants; therefore it is assumed that all participants have access to the Internet. Public key cryptography is used to provide the required security services, which adds to the complexity of the scheme; however, costs could potentially be reduced by replacing SET with some other secure payment technique. Moreover, the buyer is expected to have a public key pair stored in the buyer PC, and hence the buyer has to use this particular machine every time a transaction is to be made. This is not practical for users who might use different PCs. Although a smart card could be employed to store the key and enhance mobility, not many user PCs are equipped with smart card readers.

Chapter 7

GSM AND ELECTRONIC MURABAHA TRANSACTIONS

Contents

This chapter proposes a way of using the GSM security services to conduct a secure electronic Murabaha e-commerce transaction.

An overview of the wireless infrastructure and services which are relevant to the proposed protocol is first given (section 7.2). A model of a GSM-based electronic payment system is then discussed (section 7.3) and requirements for such a system are developed (sections 7.4 and 7.5). This is followed by a description of a method for mobile secure electronic Murabaha transactions using a combination of the Internet, a mobile phone, and a hash-chain scheme related to S/KEY (section 7.6). A security analysis of the proposed protocol is then given (section 7.7). Finally, a description of how the protocol can be extended to make use of the security features of UMTS instead of GSM is presented (section 7.8). Conclusions are provided in section 7.9.

Some of the work described in this chapter has previously been published in [10].

7.1 Introduction

Two very important technological trends in recent years have been the wide acceptance of mobile phones around the world and the growth of e-commerce. For example, mobile phone usage is forecast to reach almost 80% of the population in Europe by 2005 [14].

Key characteristics of the mobile phone include the fact that it is ubiquitous, personal, and that the average user is reasonably competent in using it. This fact suggests that it can be used for authentication and authorisation in electronic payment transactions, since it already contains a physically secure cryptographic device (i.e., the Subscriber Identity Module (SIM)). However, despite all the advantages that a mobile phone has as a means of electronic payment, there are also several drawbacks, including the following.

- **Usability**

  Mobile phones are small, have limited processing power, use low-bandwidth communication technologies, use batteries with limited life spans, and often have a relatively limited user interface.

- **Theft**

  A number of current mobile communication systems store all the subscriber-specific information needed to use a mobile phone inside a smart card (e.g., a GSM SIM). To protect the phone, the user can be required to enter a PIN, although it would appear that this security measure is not widely used. However, even if a PIN is used, the smart card typically remains unlocked until the phone is switched off or the SIM loses power. From a security point of view the most important risk that arises is loss or theft of the phone and the embedded smart card.

  An attacker with a stolen SIM is potentially able to make fraudulent transactions at the owner's expense, at least until the SIM is reported stolen and blocked, assuming that the SIM is not PIN protected. Furthermore, if the SIM was stolen in an unlocked state, an attacker could gain access to any personal information in the SIM and Mobile Equipment (ME). Thus a payment scheme based on use of a GSM SIM to help secure the transaction may need to be combined with another authentication method, e.g. username and password, if the fraud threat arising from use of a stolen SIM is to be addressed. In such a case an attacker would need both the user SIM and password to impersonate the user to a seller.

- **Radio interface threats**

  Mobile communications between a mobile phone and a serving network appear to be more susceptible to eavesdropping and interference than current Internet traffic through fixed networks. This is because of the intrinsic greater vulnerability of radio networks to interception. Moreover, the quality of the communications channel can be highly variable when a mobile device is used in a built environment. Thus transmitted data should be protected in terms of authentication, confidentiality, and integrity.

Despite these disadvantages, the use of GSM enables us to devise a Murabaha e-commerce payment system with certain advantages over the scheme presented in chapter 6. The protocol described in chapter 6 requires the buyer to have a public key pair. This key pair would typically be stored in the buyer PC, and hence the buyer has to use this particular machine every time a transaction is to be made. Although a smart card could be employed to store the key and enhance mobility, not many user PCs are equipped with smart card readers. However, the GSM-based scheme described in this chapter does not require such a key pair to be generated, since its security relies on the secret keys stored in a GSM SIM. Moreover, basing security on a mobile phone also inherently supports a level of user mobility.

7.2 Mobile infrastructure and services

The existence of a suitable transport infrastructure is important for enabling the use of mobile phones in electronic payments. In this section we describe the security services provided by the GSM and UMTS mobile communications systems. A brief description of the SMS and SIM Application Toolkit services is also given.

7.2.1 GSM security

A GSM network can be divided into three functional entities [91]. These are the mobile station carried by the subscriber, consisting of a Mobile Equipment (ME) with its Subscriber Identity Module (SIM), the network subsystem which performs the switching of calls between the users and between mobile and fixed network users, and the Base Station subsystem, which controls the air interface between the mobile station and the network subsystem.

The main security services provided by the GSM air interface are, [34, 91]:

- Subscriber identity confidentiality,
- Subscriber identity authentication, and
- Data confidentiality.

Each mobile network operator maintains two databases: the Home Location Register (HLR), and the Visitor Location Register (VLR). The HLR is used to store information regarding the subscribers of this operator. The VLR holds information on subscribers which have roamed into its network. GSM air interface security is based on a secret key shared by the subscriber's home network and the SIM. The secret keys of the subscribers of a network are stored in an Authentication Center (AC) maintained by that network, which generates security parameters on request by the HLR. The AC is usually implemented as part of the HLR [75].

Each SIM has a unique international mobile subscriber identity (IMSI) and a secret key K_i shared only with the subscriber's network operator AC. During authentication, two keyed functions ($A3, A8$), and a stream cipher encryption/decryption algorithm $A5$ are used. To authenticate a subscriber (holder of a SIM) to the network, the subscriber sends its IMSI to the VLR, which, in turn, sends a request to the subscriber's HLR. The HLR requests the AC to generate a triplet (R, SRES, K_c), where R is a random challenge, SRES (the expected response to the challenge) = $A3_{K_i}(R)$, and K_c (the session encryption key) = $A8_{K_i}(R)$. This triplet is then provided to the VLR, which sends R to the mobile device, and hence to the SIM, which recomputes SRES and K_c using its stored copy of K_i, and returns SRES. If the returned value agrees with the value in the triple, the mobile is deemed authentic, and data exchanged between the mobile and the network is subsequently encrypted using K_c. This encrypted channel is also used to transfer a temporary identity (TMSI) for the mobile to provide a measure of mobile anonymity (avoiding the need for the IMSI to be routinely sent across the wireless channel).

7.2.2 UMTS/3GPP security

UMTS (Universal Mobile Telecommunication System) is a third generation (3G) mobile telecommunication system whose security system is somewhat similar to, although more sophisticated than, that used in GSM. UMTS offers the following security services in addition to those provided by GSM, [17]:

- Mutual authentication between the user and network,
- Assurance that authentication information and keys are not being re-used,
- Integrity protection of signalling messages against replay or modification,
- Encryption is mandatory, and the encryption algorithm used is stronger, and
- Termination of the encryption further into the core network to encompass microwave links.

As described in the 3GPP specifications [1], UMTS security is based on a secret key K shared between a User Services Identity Module (USIM) and the user's home network AC. In addition, UMTS uses a set of predefined cryptographic functions '$f1 - f5$' to generate the security parameters needed during the authentication and key agreement procedures. Specifically, $f1$ is a MAC function, $f2$ is a function used to generate authentication responses, $f3$ is a cipher key generation function, $f4$ is an integrity key generation function, and $f5$ is an anonymity key generation function.

To authenticate a subscriber to the network, the subscriber's device sends the IMSI to the VLR, which, in turn, uses it to request authentication data from the subscriber's HLR. Upon receipt of the request, the HLR/AC responds with an array of quintets (R, XRES, CK, IK, AUTN), where R is a random challenge, XRES (expected response) = $f2_K(R)$, CK (cipher key) = $f3_K(R)$, IK (integrity key) = $f4_K(R)$, and AUTN is an authentication token. Each quintet can be used for just one authentication and key agreement exchange between the VLR and the USIM.

The VLR selects a quintet from the array held in the VLR and sends the parameters R and AUTN to the USIM. On receipt of these values, the USIM uses R to recompute part of AUTN and checks whether AUTN can be accepted; if so, the USIM computes the response RES = $f2_K(R)$ and sends it back to the VLR. Finally the USIM computes the cipher key $CK = f3_K(R)$ and the integrity key $IK = f4_K(R)$. Upon receiving RES, the VLR compares it with XRES (the expected response to the challenge). If they match the VLR considers the authentication and key agreement exchange to be successfully completed and selects the corresponding CK and IK from the quintet. Data subsequently exchanged between the mobile and the network is encrypted using CK and integrity protected using IK.

7.2.3 Short message service

The GSM short message service (SMS) allows users with mobile phones to send or receive text messages. All messages sent via SMS are sent in clear text format. A single SMS message can consist of up to 140 bytes. However, several SMS messages can be concatenated to create a longer message. SMS messages are not sent directly from sender to recipient, but always via an SMS Centre (SMSC) in the operator network. The sender and the receiver are identified by their respective IMSIs. When an SMS message is received by the SMSC, it sends an SMS request to the HLR of the recipient. When the HLR receives this request, it responds to the SMSC with the recipient's status, which is either inactive or active. If the recipient is inactive, the SMSC will hold on to the message for a period of time. When the recipient becomes active, the HLR sends an SMS Notification to the SMSC, and the SMSC will attempt delivery. If delivery is successful, the SMSC receives verification that the message was received by the end user. SMS messages are protected by the data confidentiality service provided by GSM/UMTS. However, this protection is only for the wireless part of the network and does not cover the SMSC [32].

A disadvantage of SMS messaging is that messages cannot be blocked. Mobile phones can become the subject of denial of service attacks, viruses and spamming through receipt of multiple unsolicited SMS messages.

7.2.4 SIM/USIM Application Toolkit

The SIM [33] and USIM [2] Application Toolkits have been proposed as means of expanding the ME functionality by allowing the addition of applications to a SIM/USIM card. They define how an application program running on the SIM/USIM can register menu elements and listen to events such as the receipt of an SMS message or user initiated selections. When an event occurs, a procedure on the SIM/USIM is executed. The procedure can invoke other functions of the ME; for example, it can display a message, ask for input, or send an SMS. SIM and USIM Toolkit applications use SMS as the bearer channel, and therefore each transaction incurs a cost based on the number of SMS messages involved.

7.3 The GSM-based electronic Murabaha transaction model

The main goal of this section is to present a protocol that combines use of the GSM infrastructure, a one time password scheme (see section 2.4.4), and the Internet, to conduct a secure electronic Murabaha transaction supporting buyer mobility.

7.3.1 Entities involved

The GSM-based electronic Murabaha transaction defined here involves interactions between four parties (see Figure 7.1): the buyer, the merchant, a GSM Authentication Centre, and the provider.

- **Buyer (B)**: This is the entity that wishes to buy goods from a merchant via the Internet, but does not have the cash immediately available to complete the transaction. The buyer must have access to the Internet and a SIM Application Toolkit compliant mobile phone. It is assumed that the SIM in the buyer mobile phone contains a SIM Toolkit compliant payment application.
- **Merchant (M)**: This is the entity that offers the goods for sale (via the Internet) which the buyer wishes to purchase. We assume that the merchant and provider share a communications link that offers confidentiality, integrity, and origin authentication, for example as provided by the Transport Layer Security (TLS) protocol [24] with both client and server authentication.
- **Provider (P)**: This is a financial institution that acts as an intermediary between the buyer and the merchant. It undertakes the purchase of commodities as specified by

a buyer, and then resells them on a Murabaha basis to him for the cost price plus a margin of profit agreed upon previously by the two parties. It does not make a purchase unless the buyer both requests it and makes a prior promise to purchase the goods from the provider. It is assumed that the provider has a contractual agreement with the mobile network operator to regulate the relationship. In order to communicate with the buyer, the provider has a GSM-enabled device (with a SIM) via which it can send and receive SMS messages; it is further assumed that the SIM is issued by the same network operator that issued the buyer's SIM, and contains an appropriate SIM toolkit compliant payment application. In order to expand his customer base, the provider may have several SIMs issued by different mobile network operators. Moreover, we assume that the buyer trusts the provider. This trust is explicit as the buyer is assumed to have a formally established agreement with the provider that defines the trust and liability relationship. The provider is also assumed to have a secure communications link with the AC of the mobile network operator which issued it the SIM.

- **Authentication Centre (AC)**: This is the AC belonging to the GSM mobile network operator shared by the buyer and the provider. In our protocol it is used as a key distribution centre, where it receives requests from the provider to generate a session key used to secure communications between the buyer and the provider. We assume that the communications link between the provider and the AC offers confidentiality, integrity, and origin authentication, for example as provided by TLS with both client and server authentication.

7.3.2 Interaction

In the proposed payment scheme, we first suppose that a buyer, shopping at an Internet merchant site, chooses to pay using Murabaha through a specified provider. The merchant contacts the selected provider to complete the transaction. If the provider chooses to proceed with the transaction, the provider first calculates the profit for the transaction and then sends

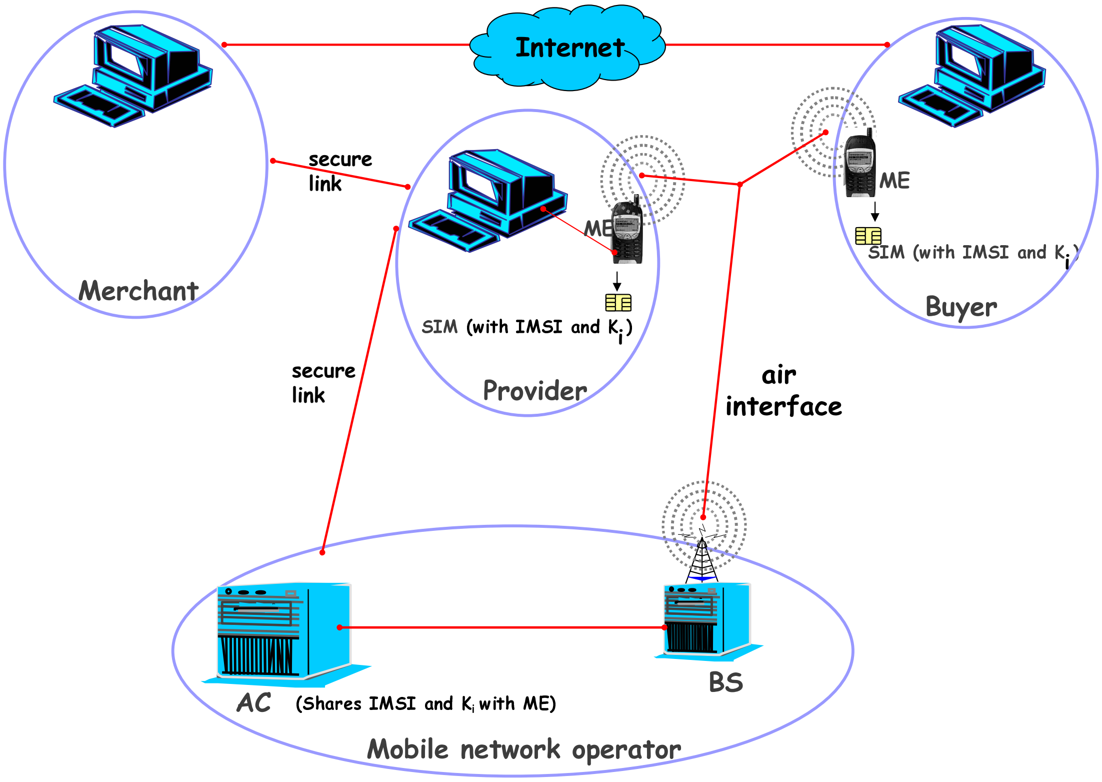

Figure 7.1: GSM-based Murabaha transaction infrastructure

the buyer an SMS message promising to sell the goods to the buyer. In return, the buyer replies with an SMS message in which the buyer promises to buy the goods on a Murabaha basis for the cost of the goods plus the agreed upon profit. This promise is not binding on either the buyer or the provider, and is not an actual sale. At this stage the relationship between the buyer and the provider is that of promisor and promisee. Based on the response received from the buyer, the provider communicates with the merchant Internet site and completes the purchase of the goods. Once the purchase of the goods is settled between the provider and the merchant, the provider sends an SMS message to notify the buyer of completion of the purchase and to offer him the goods. If the buyer agrees, a payment authorisation is sent back to the provider (an SMS message) committing the buyer to purchase the goods from the provider on a Murabaha basis. Finally, the provider asks the merchant to deliver the goods to the buyer.

7.4 Security requirements

No participant in any transaction wants to suffer any loss. Therefore, we need to define precisely the security requirements to meet the needs of the transaction participants. We therefore next identify what security services are required for a secure GSM-based electronic Murabaha transaction. The security services can be divided into four categories: authentication, confidentiality, integrity, and non-repudiation.

7.4.1 Authentication

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Entity authentication of the merchant to the provider.* This will enable the provider to verify that the merchant is as claimed.

2. *Entity authentication of the AC to the provider.* This will enable the provider to verify that the AC is as claimed.

3. *Entity authentication of the provider to the buyer.* This will enable the buyer to verify that the provider is as claimed.

4. *Entity authentication of the buyer to the provider.* The provider needs to be sure that the buyer is the legitimate owner of the SIM.

5. *Origin authentication to the provider for the payment authorisation.* The provider needs to be sure that the source of the payment authorisation is a genuine SIM.

7.4.2 Confidentiality

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Confidentiality of the buyer authorisation.* The buyer authorisation must be kept secret from non-authorised parties.

2. *Confidentiality of the order information.* The buyer may require that his order information is not available to parties other than the merchant, e.g. for privacy reasons.

7.4.3 Integrity

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Integrity protection for the transaction data sent between the participants.* Such transaction data should be protected against modification and replay.

2. *Integrity protection for the payment authorisation.* The buyer payment authorisation must be protected against alteration, or at least any alteration must be detectable.

7.4.4 Non-repudiation

In the context of the proposed protocol, this security service can be sub-divided into the following:

1. *Non-repudiation of origin for the buyer payment authorisation sent to the provider.* The provider must have evidence that the buyer has authorised payment for the goods on a Murabaha basis.

7.5 Murabaha requirements

In this section we identify the requirements that must be fulfilled for a buyer to conduct a valid Murabaha transaction over the Internet. These requirements have been derived by applying the general Murabaha requirements given in section 4.4 to the specific transaction model given in section 7.3.

1. *Buyer knowledge of the original price and profit charged.* The buyer must be aware of the original price of the goods being purchased and the amount of profit the provider is charging him before buying the goods. This is important for the transaction to be compatible with the Murabaha sale conditions as stated in section 4.4.2.

2. *Provider ownership of the goods.* The buyer requires assurance that the provider owns the goods being offered.

7.6 The protocol

We now describe the proposed GSM-based electronic Murabaha transaction (see figure 7.2) in detail. The protocol consists of five phases: the *Registration phase*, in which the buyer sets up an S/KEY system with the provider; the *Transaction request phase*, in which the buyer finds goods he wishes to buy at an Internet merchant site, and decides to use Murabaha to pay for the goods; the *Promising phase*, invoked by the provider, wherein the provider promises to sell the buyer the goods he is interested in, while the buyer promises to buy the goods from the provider, once the provider has ownership; the *Purchase phase*, invoked by the provider, wherein he buys the goods requested by the buyer from the merchant, and the *Murabaha phase*, invoked by the provider, wherein the buyer verifies the provider's ownership of the goods offered and sends a commitment to the provider to buy the goods at the agreed price.

The transaction is based on a combination of secret key encryption, computation and verification of Message Authentication Codes (MACs), and a hash-chain scheme related to S/KEY. The provider needs to store one time passwords (hash values) released by the buyer as evidence that the buyer authorised the Murabaha transaction.

In the protocol descriptions we make use of the following notation.

- c denotes the number of the buyer authentications remaining before the S/KEY system needs to be re-initialised,
- d denotes the initial seed value used in the S/KEY system,
- f denotes a one-way hash function,
- $\mathrm{MAC}_K(M)$ denotes a MAC computed on message M using a variant of key K (note that it is important that the key used to compute the MAC is not precisely the same as the key used for encryption, particularly if the MAC is a CBC-MAC [66]),

- MN is the buyer mobile number,
- p denotes the buyer secret password, and
- R_i denotes a random nonce generated by entity i.

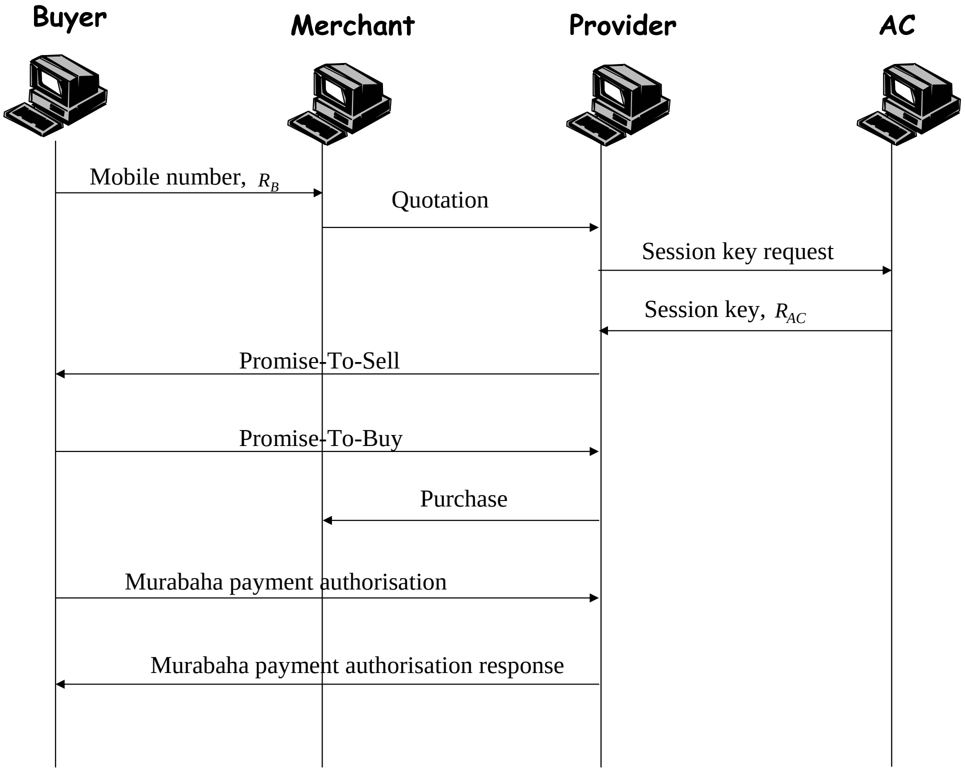

Figure 7.2: GSM-based Murabaha transaction

7.6.1 Registration

Initially, the provider and the buyer mobile (in fact the SIM) set up an S/KEY system by means of a secure channel, e.g. as provided using Bluetooth. The one-way hash-function f must also be agreed.

- The provider selects a value of c and securely passes it either to the buyer or directly to the buyer's SIM Toolkit application.

- The buyer invokes his SIM Toolkit payment application, inputs his own secret password p, and inputs the value of c selected by the provider; the application selects a random value for d.
- The SIM Toolkit payment application computes $s = p \oplus d$.
- The SIM Toolkit payment application computes $f^c(s)$, where f^c is the recursive application of f a total of c times, and securely transfers $f^c(s), c$ and d to the provider for subsequent Murabaha transaction authorisations.

After the allowed number of buyer authentications c has expired, the buyer is expected to re-initialise the S/KEY system using the steps described above. The provider selection of the value of c is a trade off between user convenience and security. A large value for c means greater user convenience because of less frequent re-initialisations, while a small value for c reduces the risks associated with compromised S/KEY passwords.

7.6.2 Transaction request

This phase begins when a buyer, shopping at an Internet merchant site, indicates that he wishes to make a specific purchase using Murabaha through a specified provider, and fills in an Internet form that contains a field for a GSM phone number and a random number. The buyer invokes the SIM Toolkit payment application in his mobile phone, which, in turn, generates, stores and displays a random number R_B. The buyer enters R_B and his mobile number MN in the merchant form. On receipt of the form, the merchant prepares a quotation that contains data related to the goods being offered, such as the goods description, price, validity of the quotation, MN and R_B. After preparing the quotation, the merchant sends it to the specified provider in order to finalise the transaction. This quotation is optionally signed by the merchant. As in section 7.3.1, we assume that the merchant and the provider are using appropriate security measures that provide entity authentication and integrity protection to protect messages exchanged during the course of the transaction.

7.6.3 Promising

In this phase, consisting of a total of five steps as listed below, both the provider and the buyer use the GSM security infrastructure to establish a shared secret session key, which they can use to provide security for the rest of the transaction. Moreover, the provider starts to negotiate with the buyer to assert his willingness to buy the goods specified in the previous phase.

1. $P \rightarrow AC : \, MN$
2. $AC : K_s = A8_{K_i}(R_{AC})$
3. $AC \rightarrow P : R_{AC} \,||\, MN \,||\, K_s$
4. $P \rightarrow B : \, R_{AC} \,||\, E_{K_s}(\, PI \,||\, R_B || \, \text{Trans\_ID}) \,||\, \text{MAC}_{K_s}(PI \,||\, R_B || \, \text{Trans\_ID})$
5. $B \rightarrow P : \, E_{K_s}(\, PI \,||\, \text{Trans\_ID} \,||\, Y/N) \,||\, \text{MAC}_{K_s}(PI \,||\, \text{Trans\_ID} \,||\, Y/N)$

Upon receipt of the quotation prepared in the Transaction request phase, and after successfully verifying the merchant's signature, the provider sends a session key request to the AC, as shown in step 1. The request contains the buyer mobile number MN. The AC retrieves the buyer shared secret key K_i corresponding to MN, generates a new random number R_{AC}, and uses them to derive a random session key K_s for this transaction (step 2).

The AC then constructs and sends a response message to the provider (step 3). In addition to K_s, this message contains the random number R_{AC} used to generate the session key K_s and the buyer mobile number MN to enable the provider to link the messages sent in steps 1 and 3.

After receiving the message in step 3, the provider constructs the Purchase Information (PI), which contains an abbreviated goods description, the cost of the goods to the provider, the profit requested by the provider, and the date the provider expects the buyer payment. The inclusion of the profit requested by the provider is to satisfy the conditions set out in section 4.4.2. The provider next generates and stores a transaction identifier, labelled 'Trans_ID', that uniquely identifies the transaction. This must be generated in such a way that the same identifier is never used twice. The provider then creates and sends an SMS message called the *Promise-To-Sell* that contains R_{AC} and a commitment to sell the requested goods to the buyer once they have been bought from the merchant (step 4). The promise includes the PI, the random number R_B received in the transaction request phase, and Trans_ID. The provider encrypts the promise using the session key K_s. A MAC is also computed on the *Promise-To-Sell* message using the key K_s. The enciphered information and the MAC are then sent to the buyer. It should be noted that the key used to compute the MAC is not precisely the same as the key used for encryption; we assume that the encryption and MAC functions use different keys derived from K_s.

Upon receipt of the SMS message (the *the Promise-To-Sell*) in step 4, the SIM Toolkit payment application in the buyer SIM calculates the key K_s using the received R_{AC} and the SIM stored key K_i as inputs to the key derivation algorithm $A8$ shared with the AC, i.e. $K_s = A8_{K_i}(R_{AC})$. The buyer SIM then decrypts the encrypted part of the *Promise-To-Sell* message and also verifies the MAC to check message integrity and origin using K_s. Moreover, it checks that the value of R_B included in the message is the same as the value sent during the Transaction request phase.

The buyer then checks that the goods promised by the provider (as described in the *PI*) are the requested goods. Also, the buyer checks that both the profit and the due date offered by the provider are acceptable to him. If the buyer chooses to proceed with sale, the buyer first stores the various transaction details, including Trans_ID, for later use in the Murabaha phase. The buyer next responds to the provider with a *Promise-To-Buy* SMS message (step 5) to indicate his willingness to buy the goods once the provider has ownership of them. The message includes the PI, the Trans_ID, and B's agreement to continue the transaction (Y/N), encrypted and MACed using K_s (as previously, this involves deriving two separate keys from K_s for the two cryptographic functions involved).

7.6.4 Purchase

When the provider receives the *Promise-To-Buy* SMS message from the buyer (step 5 of the Promising phase), he decrypts the encrypted part using the shared session key K_s and checks that the contents are as expected and that the Trans_ID matches the value sent in step 4 of the Promising phase. The provider also verifies the MAC on the message using K_s. If all the checks succeed then the provider completes the purchase of the goods from the merchant. We assume that the provider signs and sends a message to the merchant that guarantees the merchant payment. Once the merchant has verified the promise of payment for the goods from the provider, the merchant displays a message to the buyer confirming that the transfer of ownership of the goods to the provider has taken place. This gives the buyer assurance that the provider has purchased the goods, and also serves as a prompt to the buyer to purchase the goods from the provider.

7.6.5 Murabaha

In this final two-step phase the buyer purchases the goods from the provider. The two messages below are both sent as SMS messages.

1. $B \rightarrow P : E_{K_s}(\,PI \,||\, \text{Trans\_ID} \,||\, f^{c-1}(s)) \,||\, \text{MAC}_{K_s}(PI \,||\, \text{Trans\_ID} \,||\, f^{c-1}(s))$
2. $P \rightarrow B : E_{K_s}(\text{Trans\_ID} \,||\, f^{c-1}(s)) \,||\, \text{MAC}_{K_s}(\text{Trans\_ID} \,||\, f^{c-1}(s))$

If the buyer chooses to complete the transaction, then he authorises the SIM to construct and send an encrypted and MAC-protected payment authorisation to the provider, as shown in step 1. The payment authorisation contains PI and Trans_ID. Moreover, it contains the current value in the password chain ($f^{c-1}(s)$) which confirms the buyer agreement to buy the goods on a Murabaha basis. To ensure that only the rightful owner of the phone can generate the authorisation, the SIM Toolkit payment application can additionally ask the buyer to authenticate himself before sending this message, e.g. using a PIN; the application then sends the payment authorisation to the provider.

Upon receipt of the message in step 1, the provider first decrypts it and checks the MAC. The provider then uses the one way function f to check whether $f(f^{c-1}(s)) = f^{c}(s)$ using its stored value of $f^{c}(s)$. If the check succeeds, then the provider saves the current $f^{c-1}(s)$ as evidence of the transaction authorisation, decrements c, and updates $f^{c}(s)$ with $f^{c-1}(s)$. If the provider does not receive $f^{c-1}(s)$, or receives an incorrect value for $f^{c-1}(s)$, then the provider terminates the transaction.

After successful verification of the buyer authorisation, the provider instructs the merchant to dispatch the goods to the buyer address (which is already known to the provider at registration). Moreover, the provider sends a completion message to the buyer (step 2). Upon receiving the message in step 2, the buyer SIM Toolkit payment procedure decrypts $E_{K_s}(\text{Trans\_ID} \,||\, f^{c-1}(s))$. If 'Trans_ID' is the same as the value sent in step 1 and the MAC verifies correctly, then the buyer SIM Toolkit payment application decrements c and ends the application.

7.7 Analysis

In this section, we examine to what extent the generic security requirements outlined in section 7.4 and 7.5 are met by the proposed transaction.

7.7.1 Authentication

1. *Entity authentication of the merchant to the provider.* It is assumed (see section 7.3.1) that the provider is using a security protocol such as SSL/TLS to authenticate the merchant. Therefore, this requirement is met.

2. *Entity authentication of the AC to the provider.* It is assumed (see section 7.3.1) that the provider is using a security protocol such as SSL/TLS to authenticate the AC. Therefore, this requirement is met.

3. *Entity authentication of the provider to the buyer.* When the buyer verifies the MAC received in message 4 of the Promising phase, this implies the freshness of the transaction because the message contains the value R_B, previously chosen by the buyer. The buyer thus authenticates the provider. However, because R_{AC} is not authenticated, the risk remains that the MAC has been generated using an old R_{AC} with a compromised K_s.

4. *Entity authentication of the buyer to the provider.* A legitimate owner of the SIM will need to enter the correct PIN to use it. Even if an attacker has stolen the SIM and impersonates the buyer to purchase goods, he will not be able to gain financially because the goods will be delivered to the buyer address contained in the provider database. The proposed protocol does not prevent this attack until the buyer SIM is reported stolen and blocked by the mobile network operator.

5. *Origin authentication to the provider for the payment authorisation.* The payment authorisation sent in step 1 of the Murabaha phase is MAC-protected using a key K_s shared between the buyer and the provider. The K_s can only be generated by a SIM that has a legitimate K_i. Therefore this provides an evidence to the provider that the source of the payment authorisation is the legitimate SIM.

7.7.2 Confidentiality

1. *Confidentiality of the buyer authorisation.* The value $f^{c-1}(s)$ generated by the buyer is sent to the provider in an encrypted SMS message. The encryption key used is known only to the buyer, the provider and the AC. The AC is trusted not to reveal the key and hence $f^{c-1}(s)$ will not be available to unauthorised parties. Furthermore, the value $f^{c-1}(s)$ is used only once.

2. *Confidentiality of the order information.* All messages exchanged between the buyer and the provider are encrypted using a shared session key. An advantage of our scheme is that the buyer does not need to send any private information via the merchant, unlike conventional e-commerce schemes where a credit card number is sent to a merchant protected using TLS. This avoids any concerns regarding the ability of the merchant to store buyer private information in a secure manner. However, the buyer will need to provide the MN and R_B to the merchant via the purchase form. This information is sent to the provider using a secure channel, e.g. as provided using TLS.

7.7.3 Integrity

1. *Integrity protection for the transaction data sent between the participants.* While the communication link between the AC and the provider is assumed to be secure, the AC uses SMS messages to communicate with the buyer SIM. If an attacker modified the R_{AC} sent by the AC to the buyer SIM, then the SIM will generate a different session key from the one AC sends to the provider. Therefore, in such a case, the buyer SIM and the provider would not be able to establish a secure channel.

   On the other hand, an attacker can force re-use of an old R_{AC} for which the corresponding key K_s is known. It is therefore important for the AC to generate R_{AC} values from a source with good randomness properties, such that the probability of sending the same R_{AC} twice to the buyer SIM is negligible.

2. *Integrity protection for the payment authorisation.* The payment authorisation sent in step 1 of the Murabaha phase is protected against unauthorised modification through the use of a MAC. Without the key K_s, it is assumed to be infeasible to generate a valid MAC for a modified authorisation message.

7.7.4 Non-repudiation

1. *Non-repudiation of origin for the buyer payment authorisation sent to the provider.*

   The one way function f is used to achieve non-repudiation. In the ith session, the buyer provides $f^{c-i}(s)$ to authorise the Murabaha transaction. The provider can verify the correctness of $f^{c-i}(s)$ but cannot derive $f^{c-i}(s)$ from $f^{c-i+1}(s)$. Therefore, $f^{c-i}(s)$ can be used as evidence of the ith authorisation.

   Moreover, the buyer authorisation $f^{c-i}(s)$ is different for each transaction, and therefore $f^{c-i}(s)$ is not replayable, or usable as proof for some other transaction.

7.7.5 Murabaha

1. *Buyer knowledge of the original price and profit charged.* This requirement is met because, in order for the provider to complete the transaction, the buyer must respond to the message sent in step 4 of the Promising phase. If the buyer replies with the message in step 5 then he must know the original price and the amount of profit the provider is adding, since it is included in the PI.

2. *Provider ownership of the goods.* We assume that the merchant, once it has sold the goods to the provider, will display a message to buyer indicating the transfer of goods ownership to the provider. However, this is not verifiable by the buyer.

7.7.6 Mobility and user privacy

The proposed scheme gives the buyer the freedom to use any PC with an Internet connection to make a Murabaha transaction. Regarding user privacy, the buyer does not need to send his identity to the merchant, However, the merchant knows at least the buyer mobile phone number. This number does not necessarily reveal a buyer's real identity (as would be the case for an ordinary credit card payment.) Besides the issue of potential identity disclosure, other privacy issues exist with passing a mobile phone number to a third party; for example, this number might be used for advertisement purposes. Finally, the provider knows exactly which buyers are buying goods from which merchants and for what amounts of money.

7.7.7 Some potential problems

The following possible problems with the protocol can be identified.

- The protocol possesses a single point of attack, namely the AC. The AC is trusted by the participants to make trustworthy decisions about the authenticity of a participant. Compromise of the AC would be disastrous. Obviously, the usefulness of a system like this also increases in proportion to the number of users who subscribe, although the effects of an outage (deliberate or accidental) also increase. It is assumed that the GSM network operator will not abuse this trust and that the AC will be well-protected against service denial and illegal access. This is likely to be true in any event, as AC failure would also bring down the entire GSM network.
- The 160 character limit on SMS messages should be considered when designing the SIM Toolkit payment application.
- There is an additional inconvenience if the user loses his mobile phone, because the mobile network operator and the provider must be contacted again to re-initialise the system.

7.8 Extending the protocol to UMTS/3GPP

Since the subscriber authentication system employed by UMTS is similar to that of GSM, our protocol can be readily adapted to take advantage of the UMTS security features (as described in section 7.2.2). Specifically, $f1$ can be used to generate the necessary MACs and $f3$ can be used to generate the session encryption key K_s. Moreover, the session integrity key IK_s can be used to compute the MAC instead of using a variant of the session key K_s, thereby avoiding any key separation issues.

Accordingly, messages in the *Promising phase* would follow the same sequence described in section 7.6.3. However, in step 2, the AC will use the function $f4$ shared with the buyer SIM to generate a session integrity key IK_s. This key is sent to the provider along with K_s (step 3) for use in MAC calculations when communicating with the buyer.

1. $P \rightarrow AC:\ MN$
2. $AC: K_s = f3_{K_i}(R_{AC}), IK_s = f4_{K_i}(R_{AC})$
3. $AC \rightarrow P: R_{AC} \,||\, MN \,||\, K_s \,||\, IK_s$
4. $P \rightarrow B:\ R_{AC} \,||\, E_{K_s}(\,PI \,||\, R_B || \,\text{Trans\_ID}) \,||\, \text{MAC}_{IK_s}(PI \,||\, R_B || \,\text{Trans\_ID})$
5. $B \rightarrow P:\ E_{K_s}(\,PI \,||\, \text{Trans\_ID} \,||\, Y/N) \,||\, \text{MAC}_{IK_s}(PI \,||\, \text{Trans\_ID} \,||\, Y/N)$

Moreover, messages in the *Murabaha phase* would be modified to use IK_s in the MAC calculations.

1. $B \rightarrow P: E_{K_s}(\,PI \,||\, \text{Trans\_ID} \,||\, f^{c-1}(s)) \,||\, \text{MAC}_{IK_s}(PI \,||\, \text{Trans\_ID} \,||\, f^{c-1}(s))$
2. $P \rightarrow B: E_{K_s}(\text{Trans\_ID} \,||\, f^{c-1}(s)) \,||\, \text{MAC}_{IK_s}(\text{Trans\_ID} \,||\, f^{c-1}(s))$

7.8.1 Related work

There exist other payment systems which uses GSM phones to support e-commerce transactions. Claessens et al. [21] proposed an electronic payment system in which the world wide web and the GSM environment are combined to improve overall security, mobility, and functionality. GiSMo [73] is a scheme that was developed by Millicom International Cellular in 1999. In this scheme, every Internet transaction is validated with a random code sent in an SMS message transmitted via a central server. This random code is then entered via the computer in order to pay. Finally, Khu-smith and Mitchell proposed an e-commerce protocol [62] that relies on the fact that GSM subscribers and network operators share a secret key K_i.

7.9 Conclusion

In this chapter we have a proposed a secure GSM-based electronic Murabaha payment protocol, where the GSM infrastructure is used to support authentication and payment authorisation. In addition to the Internet, the scheme uses SIMs equipped with a special SIM Toolkit payment application. We have described the scheme in detail, explained how it meets the identified security requirements, and we have also shown how it can be extended to use the UMTS/3GPP security features.

The proposed protocol makes use of the mobile network operator to generate some of the required security parameters. The protocol requires the provider and the AC of the mobile network operator to be involved in every Murabaha transaction. Although this is not a service currently offered by mobile network operators, it may nevertheless be attractive for operators to offer this service on the basis of a charge paid by the participating providers. To provide the required security services and payment authorisation, the protocol involves seven SMS messages and two active Internet connections. The buyer is expected to have a mobile phone and access to the Internet, services available to large number of potential consumers.

Chapter 8

A PERSON-TO-PERSON INTERNET PAYMENT SYSTEM

Contents

This chapter proposes a new Internet person-to-person payment system.

We start by describing a general model for interpersonal Internet payments (section 8.2). Security requirements are then identified for a person-to-person Internet payment system (section 8.3). A payment protocol designed to address the identified security requirements is proposed in section 8.4. Finally, we analyse how the proposed protocol matches the identified security requirements (section 8.5). Conclusions are provided in section 8.6.

It is important to note that much of the material in this chapter has previously been published in [6].

8.1 Introduction

Debit and credit cards provide a simple way for individuals to pay businesses for products and services, but they do not provide private individuals with the means to make payments to one another (here, private individuals means consumers and non-professional merchants). For example, a private seller who wants to obtain permission to accept credit-card payments has to apply for a merchant processing account, for which transaction processing costs may be up to 5%. Moreover, such a merchant account will only be granted under certain conditions, which many individuals may not meet. Thus 'standard' debit/credit card transactions are simply not appropriate for most interpersonal payments.

On the other hand, as explained in chapter 3, there are schemes by which a payer can make a payment to anyone with an e-mail address. In these schemes, the payer connects to the financial institution, and authorises it to transfer a certain amount from his account to a specified payee account. SSL/TLS is usually used to secure the communications when connecting to the financial institution, and payer authentication is usually password based. However, these schemes require both the payer and the payee to move money from their bank accounts to their account with the scheme provider in order to benefit from the service. Moreover, the payment instruction does not contain any information that links the goods being sold to the payment itself. Therefore, these schemes are not particularly appropriate to the case where a payment is made in exchange for goods.

In general, it is difficult for people unknown to each other to perform e-commerce transactions unless they possess credentials from a trusted common source, and this is often difficult to arrange. However, most individuals have a well-established relationship with a bank, and there are existing worldwide trading relationships between banks. This observation motivates the design of a person-to-person Internet payment system that exploits these existing trust and contractual relationships. In such a system the buyer authorises payment from his bank, the seller verifies payment has been received by his bank, and the seller then supplies the goods to the buyer. The seller will be able to see if the buyer has enough funds by debiting the buyer account in real time.

To summarise, in this chapter we propose a new person-to-person Internet payment system that uses the existing relationships between buyers, sellers, and their respective banks to perform remote payments, and that links payment details to the goods being sold.

8.2 A model for person-to-person Internet payments

In this section we describe our model for an interpersonal payment system. The model identifies the entities involved and includes a brief description of their interactions.

8.2.1 Entities involved

A person-to-person Internet payment system involves interactions between: the buyer, the seller, and a trusted third party, which we call the "Payment Gateway", together with the buyer and the seller banks. Their roles are straightforward.

- **Buyer (B)**: This is the entity that makes the payment, using money in a bank account, in exchange for goods or services. The buyer has a digital certificate, issued by the trusted payment gateway, that links the buyer account number to the buyer public key. The buyer has the capability to set up a secure channel to the payment gateway, which we assume is provided by the use of SSL/TLS with client authentication.
- **Seller (S)**: This is the entity that receives money from the buyer during a payment transaction in return for providing the buyer with goods or services. The money is deposited into the seller bank account. Although the seller trusts the payment gateway, the buyer and the seller need not trust each other. The seller has a digital certificate

issued by the payment gateway that links the seller account number to the seller public key.

- **Payment Gateway (PG)**: This is a trusted third party which links electronic payments to the transfer of “real money”. It certifies the trustworthiness of the buyer and the seller by issuing, maintaining and checking the status of digital certificates that link the seller and buyer bank accounts to their respective public keys. The payment gateway and the buyer’s bank are assumed to enjoy a degree of mutual trust and share an infrastructure for secure communication, e.g. a financial network. In the event of a dispute, the payment gateway shall be capable of producing evidence of buyer authorisations to the buyer bank. The payment gateway shall store those authorisations securely.

- **Buyer bank**: This is a financial institution that establishes an account for a buyer. It trusts the payment gateway to pass it buyer-approved payment instructions. It is assumed that the buyer has authorised the buyer bank to accept payment gateway messages. When requested, the buyer bank must be able to show the destination of the transferred money.

- **Seller bank**: This is a financial institution that establishes an account for a seller. It accepts money transfers made by buyers and credits them to the seller account. When requested, the seller bank must be able to demonstrate the source of the transferred money.

The payment gateway issues two X.509 certificates [55] for every participant, one that binds the signature key pair to the user bank account number, and the other suitable for use in setting up an authenticated SSL session. The payment gateway will reissue these certificates upon expiry, provided that the bank account is in good standing and that no theft or misuse of the corresponding private key has been reported.

8.2.2 Interactions

In our payment model, the seller sends his bank account number and the description of the goods in a seller block (SBLK) to the payment gateway, which certifies the correctness of the account number by creating a certified seller block (CSBLK). The concept of seller block is introduced here to give the buyer confidence that the seller he is dealing with is genuine. Every time a seller agrees with a buyer on the sale of specific goods, the seller should generate an SBLK that is specific to this transaction and that incorporates the description of the goods and the agreed price. The seller must then have it certified by the payment gateway to produce the CSBLK. The seller then sends (or makes available) this CSBLK to the buyer, who uses it to instruct the payment gateway to pay the seller, using the buyer's bank account. A payment to a seller will only be successful in the presence of a valid CSBLK. Once the payment gateway has received a valid payment instruction from the buyer, it communicates with the buyer bank to instruct it to transfer the agreed payment for the goods from the buyer account to the seller account.

8.3 Security Requirements

We next consider what security services are needed to protect the scheme participants against possible threats.

8.3.1 Authentication

This security service can be sub-divided into the following.

1. *Entity authentication of the payment gateway to the seller.* The seller needs to authenticate the payment gateway to prevent transmission of the SBLK to unauthorised third parties.

2. *Entity authentication of the payment gateway to the buyer.* The buyer needs assurance that he is sending the payment instruction to the trusted payment gateway.

3. *Origin authentication to the payment gateway for the payment instruction.* The payment gateway needs to have assurance that the buyer is the source of the received payment instruction.

8.3.2 Confidentiality

1. *Confidentiality of the seller and buyer payment details.* The payment details of the seller and the buyer, as sent in SBLK and CSBLK, must be kept secret from unauthorised third parties.

8.3.3 Integrity

This security service can be sub-divided into the following.

1. *Integrity protection for the payment authorisation sent by the buyer to the payment gateway.* The buyer payment authorisation must be protected against alteration, or at least any alteration must be detectable.

2. *Integrity protection for SBLK and CSBLK.* The contents of the SBLK and CSBLK must be protected against alteration, or at least any alteration must be detectable.

8.3.4 Non-repudiation

This security service can be sub-divided into the following.

1. *Non-repudiation of the seller payment.* The buyer requires evidence that the seller

specified in the CSBLK received the payment, to protect against repudiation by the seller.

2. *Non-repudiation of origin for the buyer payment instruction sent to the payment gateway.* When the payment gateway requests the buyer bank to debit a buyer account by a certain amount, and credit the same amount to the seller account, the payment gateway must be in possession of evidence that the buyer has authorised this payment in order to protect against repudiation by the buyer.

8.4 The Protocol

The proposed scheme (see figure 8.1) is composed of three phases: *the Registration phase*, in which the payment gateway registers the participant and issues him a digital certificate that binds his bank account to his public key; *the Seller Block Certification phase*, invoked by the seller, wherein the payment gateway certifies that the seller bank account number is valid, and *The Payment phase*, invoked by the buyer, wherein the buyer verifies the certified seller bank and sends authorisation to his bank to transfer the agreed amount of money from the buyer bank account to the seller bank account.

8.4.1 Specific Requirements

In order to execute the protocol, the following requirements must be satisfied.

1. Every payment gateway must possess a signature key pair, used both for signing certificates for participant public keys and for creating certified seller blocks during the Seller block certification phase of the protocol. Every payment gateway must also possess a key pair suitable for use in setting up authenticated SSL/TLS sessions.

2. In order to engage in the interpersonal payment scheme, every buyer and seller will need to install a special application on their PC to perform their part of the payment

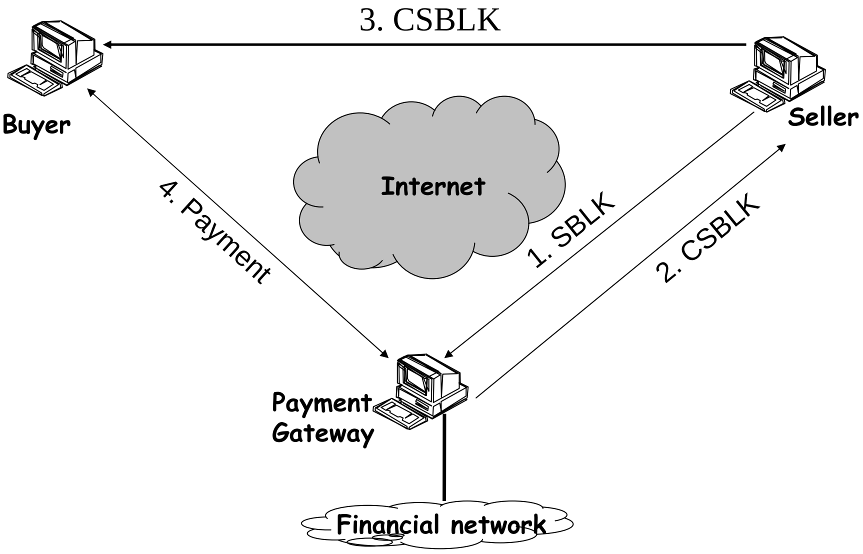

Figure 8.1: Person-to-Person Internet payment system

protocol.

3. Each participant (buyer or seller) must have two asymmetric key pairs: one pair used for setting up an authenticated SSL/TLS session and the other used for the creation and verification of digital signatures. The payment gateway issues two X.509 certificates [55] for every participant, one that binds the signature key pair to the user bank account number, and the other suitable for use in setting up an authenticated SSL session. The payment gateway will reissue these certificates upon expiry, provided that the bank account is in good standing and that no theft or misuse of the corresponding private key has been reported.

8.4.2 Registration phase

This phase has five steps, as described below. We use X to denote the user to be registered.

First, in step 1, the payment gateway PG securely authenticates the user by some implementation-specific means.

1. $X \leftrightarrow$ Payment Gateway : Authenticate X

Then, in step 2, user X provides his bank account number to the payment gateway.

2. $X \rightarrow$ Payment Gateway : account number

After verifying the association between the entity and the supplied bank account details (by implementation-specific means), the payment gateway generates a unique identifier id_X for X, for use in this protocol, and creates a personalised copy of a 'payment application' for installation by the user. This PC application will be used by the user acting as buyer or seller to make person to person purchases, and is assumed to contain copies of the user identifier id_X, the payment gateway public encryption key P_{PG}, and the payment gateway public signature verification key V_{PG}.

In step 3 the payment gateway sends this application to the user.

3. Payment Gateway $\rightarrow X$: payment application

As step 4, the user then installs this application on his/her PC, and the application automatically generates and stores two key pairs for the user (one for creating and verifying digital signatures as part of this application and one for setting up authenticated SSL/TLS sessions).

4. X: Install application and generate key pairs

In step 5, the user transfers the public keys to the payment gateway, which creates certificates for them both, and returns these certificates to the user. Note that one certificate should be tailored to the requirements of this payment application, and the other should be suitable for use in SSL/TLS client authentication.

5. $X \leftrightarrow$ Payment Gateway : Submit public keys and obtain Cert_{P_X} and Cert_{V_X}

The user then stores these certificates in the payment application.

8.4.3 Seller Block Certification phase

Once the seller and the buyer have agreed on the goods to be sold and the price for these goods, the seller begins the process of creating the SBLK for that sale and having it certified by the payment gateway to produce a certified seller block CSBLK. Typically, an SBLK will contain the description of the goods, the price, and the seller bank account number. The SBLK therefore allows the buyer payment to be made to the seller's bank account. The CSBLK is introduced to give the buyer confidence that the seller he is dealing with is genuine, and to allow the payment gateway to check that the seller certificate has not expired or been revoked.

First, the seller S initiates a communication session with PG. We assume that this communications session offers confidentiality, integrity, and origin authentication, for example as provided by the SSL/TLS protocol with both client and server authentication. S signs the concatenation of his identifier and the SBLK, and then sends it, along with his signature verification digital certificate Cert_{V_S}, to PG as step 1:

1. Seller $\rightarrow$ Payment Gateway : $s_{S_S}(id_S \parallel \text{SBLK}) \parallel \text{Cert}_{V_S}$

After receiving the above message and successfully verifying the seller's signature, PG uses the financial network to ensure that the bank account number found in SBLK belongs to the identified seller. Once these checks have been successfully completed, PG creates a CSBLK, which is a signature over the CSBLK expiry time T, the seller identifier id_S, and the received SBLK. I.e., CSBLK $= s_{S_{PG}}(T \parallel id_S \parallel \text{SBLK})$. PG then sends the CSBLK to the seller, as step 2:

2. Payment Gateway → Seller : $s_{S_{PG}}(T \parallel id_S \parallel \text{SBLK})$

8.4.4 The payment phase

After receiving the CSBLK back from the PG, the seller makes the CSBLK available to the buyer by some means, e.g. by email or via the seller website. Thus step 1 is:

1. Seller → Buyer : CSBLK

The buyer then verifies the PG signature within CSBLK, using the buyer's stored (trusted) copy of the PG signature verification public key V_{PG}. The buyer also checks that the expiry time T has not passed, that the seller identifier id_S is as expected, and that the contents of SBLK are as expected.

The buyer next opens a communication session with the payment gateway PG. As in the previous phase, we assume that this communications session offers confidentiality, integrity, and origin authentication, for example as provided by SSL/TLS with both client and server authentication.

In step 2, B instructs the payment gateway to communicate with the buyer bank to transfer the amount specified in the CSBLK from his account to the seller account. B achieves this by creating a signed message that contains the buyer identifier id_B, the current time T, the buyer account number, and the CSBLK. B sends this, along with the certificate for his signature verification key Cert_{V_B}, to PG.

2. Buyer → Payment Gateway : $s_{S_B}(id_B \parallel T \parallel \text{Buyer account number} \parallel \text{CSBLK})\parallel \text{Cert}_{V_B}$

After verifying the signature, processing the payment instruction, and receiving a positive response from the buyer bank that the money has been transferred, the payment gateway PG sends a confirmation message (step 3) to both the buyer B and the seller S signifying the outcome of the payment. It uses the field "*result*" to indicate whether the payment has completed or failed. The message also contains a copy of the CSBLK created for that payment.

3. Payment Gateway → Buyer, Seller :
$s_{S_{PG}}(T \parallel result \parallel id_B \parallel id_S \parallel \text{Buyer account number} \parallel \text{CSBLK})$

After receiving the payment confirmation (step 3) from the PG and successfully verifying the PG signature, the seller checks the field "*result*". If "*result*" indicates the payment has completed, the seller releases the goods or services to the buyer.

8.5 Security analysis

In this section, we will examine to what extent the generic security requirements outlined in section 8.3 are met by the proposed scheme.

8.5.1 Authentication

1. *Entity authentication of the payment gateway to the seller.* The seller uses the SSL/TLS protocol with client and server authentication when initiating the communication session in step 1 of the Seller Block Certification phase. SSL/TLS authentication is based on public key certificates.

2. *Entity authentication of the payment gateway to the buyer.* The buyer uses the SSL/TLS protocol with client and server authentication when initiating the communication ses-

sion in step 2 of the Payment phase. SSL/TLS authentication is based on public key certificates.

3. *Origin authentication to the payment gateway for the payment instruction.* The buyer authorisation is sent in step 2 of the Payment phase, and it includes the buyer bank account number and the CSBLK. This message is digitally signed by the buyer, thus providing origin authentication, since no-one but the buyer has the private key necessary to create the required signature.

8.5.2 Confidentiality

The scheme uses the SSL/TLS protocol to secure communication between the seller and the payment gateway, and the buyer and the payment gateway. SSL/TLS provides confidentiality for transferred data using symmetric encryption. Messages 1 and 2 of the Seller Block Certification phase, that contain SBLK, are sent through an SSL-encrypted channel. Moreover, messages 2 and 3 of the Payment phase, which contain CSBLK, are also encrypted using SSL/TLS. However, the CSBLK is not protected when it is sent from the seller to the buyer in the Payment phase.

8.5.3 Integrity

1. *Integrity protection for the payment authorisation sent by the buyer to the payment gateway.* The buyer payment authorisation in step 2 of the Payment phase is digitally signed by the buyer and sent to the payment gateway along with the buyer signature verification certificate Cert_{V_B}. The payment gateway can verify the integrity of the buyer payment authorisation using the key V_B recovered from the buyer certificate.

2. *Integrity protection for SBLK and CSBLK.* The buyer signature over message 1 of the Seller Block Certification phase provides integrity protection for the SBLK sent to the payment gateway. Moreover, the payment gateway constructs the CSBLK by signing

the SBLK (step 2 of the *Seller Block Certification phase*). Therefore, the buyer and the seller can verify the integrity of the CSBLK contents.

8.5.4 Non-repudiation

1. *Non-repudiation of the seller payment.* The buyer can use the message received in step 3 of the Payment phase as evidence that the payment has taken place, since the payment gateway signs this message, and it includes a data item "*result*" which indicates whether or not the payment has taken place. There is a possibility that the payment gateway might not send the above message even though it has already received the payment message from the buyer. In this case, the buyer does not know whether the transaction was aborted or finalised; however, the buyer can ask his bank for an account statement which can act as a replacement for this receipt.

2. *Non-repudiation of origin for the buyer payment instruction sent to the payment gateway.* This requirement is met, because the buyer signs the message in step 2 of the Payment phase, and the *PG* can verify the buyer signature using the buyer certificate.

8.5.5 Mobility and user privacy

The proposed scheme assumes that the buyer is using a PC which contains his public key pair. Therefore, the buyer is restricted to use of this particular machine whenever a payment transaction is to be made. Regarding user privacy, the scheme allows the buyer and the seller identities to be known to each other and to the payment gateway. Moreover, the payment gateway knows the details of the goods being bought.

8.5.6 Advantages of the scheme

One advantage of this system is its auditability. Once a payment transaction has finished, the buyer can determine who authorised the payment, and that the payment transaction was credited to the seller bank account. In addition, the model is resistant to fraud because the payment gateway will only accept payment instructions from the buyer himself, so it only has to authenticate that it is talking to the person who owns the account being charged, and it does not have to rely on any other party. However, a disadvantage of this model is that the seller must receive a payment confirmation before it releases the goods or services to the buyer.

Guarantee of payment to the seller is clear in this system since the buyer will ask the payment gateway to instruct his bank to transfer the specified amount of money to the seller bank account, which could be an immediate transfer or might take some time. In the latter case, the payment gateway has already received a positive response from the buyer bank, which is a payment guarantee for the seller and allows him to ship the goods. The payment gateway provides liability protection for the seller because it provides him with proof that the buyer bank has authorised the payment to the seller bank account.

8.5.7 Some potential problems

We next briefly mention two possible problems with the proposed scheme.

- A potential problem with this scheme is that it possesses a central point of attack. The payment gateway is trusted by all participants to make trustworthy decisions about the authenticity of a participant. Compromise of the payment gateway would be disastrous. The effects of a denial of service attack on the payment gateway are also severe. Obviously, the usefulness of a system like this also increases in proportion to the number of users who subscribe, although the effects of an outage (deliberate or accidental) also increase. One solution to such problems of service availability is for the payment gateway to employ replication to increase availability.

- Sellers should be able to refund payments to buyers if necessary, although this is not supported by the scheme as described above. However, the seller can be asked to send a digitally signed document to the payment gateway promising to repay the price of the goods, if they are not delivered to the buyer. The payment gateway could hold this signed statement, giving the buyer a guarantee that he will be repaid if the goods are not delivered.
- The proposed scheme will be easier to implement in a country where all banks are assumed to have relationships with each other and are governed by a single authority. However, implementing the system in a worldwide scenario will depend on the number of banks who have relationships with each other.

8.6 Conclusion

In this chapter we have proposed a scheme designed to enable Internet person-to-person payments. We have described the scheme in detail, and explained how it meets the identified security requirements. The scheme uses existing relationships between buyers, sellers, and their respective banks to perform remote payments and to link payment details to the goods being sold.

The protocol uses the Internet to exchange messages between the participants; therefore it is assumed that all participants have access to the Internet. The protocol has the advantage of using the existing relationships between users and their banks. One possible disadvantage of the protocol is that it requires the users to have public key pairs. This key pair would typically be stored in the users PC, and hence the user has to use this particular machine whenever a transaction is to be made. By contrast, the schemes described in chapter 3 are easier to use since they require only an e-mail address; however, they have the disadvantage that they require users to transfer money from their bank account to their account with the scheme provider.

Chapter 9

A SECURE ELECTRONIC PAYMENT SCHEME FOR CHARITY DONATIONS

Contents

This chapter proposes a scheme that supports secure anonymous electronic charity donations.

First, a model for a secure electronic charity donation scheme is described (section 9.2). We then identify the security requirements such a scheme should fulfill (section 9.3), and propose a scheme that uses an anonymous electronic cash technique to make donations, and that employs smart cards for donation distribution (section 9.4). Finally, we analyse how the proposed scheme matches the identified security requirements (section 9.5). Conclusions are provided in section 9.6.

Note that much of the material in this chapter has previously been described in [9].

9.1 Introduction

Individuals are increasingly prepared to donate on-line. According to [23], of those Internet users who are likely to make a donation on-line, 52% have purchased a product or service over the Internet, making on-line buyers more likely to give on-line than other Internet users. Although the necessary technology is already in place, many charities do not take advantage of the ubiquity of the Internet and recent developments in smart card technology. Nevertheless, some charities have established their own web sites with debit/credit cards as the means for making donations.

As discussed in chapter 3, a major advantage of debit/credit card electronic payments is their ease of use and scalability. However, a disadvantage of such payments is that they do not offer anonymity, and hence privacy concerns are a potential barrier to on-line donations just as is the case for e-commerce transactions. According to [23], concerns about privacy and credit card security remain high. 71% of donors said they were concerned about the security of their personal information on-line. Nearly 90% said they would never give their credit card information out to a charity. Therefore, donating on-line using a debit/credit card payment does not satisfy the anonymity requirements of many donors. The current situation could be changed if a new electronic payment scheme for charity donations can be devised.

In this chapter, we introduce the concept of e-donation, the electronic counterpart of a charity donation. We propose a scheme that allows donations to be made anonymously and distributed to recipients using smart cards, allowing recipients to redeem their e-donations directly from a shop. In particular, the scheme involves the donor contributing a specific amount of money to the charity, which then arranges for the recipient to receive goods of precisely the value contributed by the donor. Moreover the donor can specify the nature of the goods to be made available to the recipient. Whilst this is not a model of charitable donation of general applicability, it matches the particular requirements of certain scenarios, notably the obligation on Muslims to make donations during Eid (see section 4.5).

9.2 A model for electronic charity donations

In this section, we describe our model for Internet charity transactions. The model identifies the entities involved and includes a brief description of their interactions.

9.2.1 Participants

We first define the participants in our model.

- **Donor (D)**: A person who wishes to donate to a charity anonymously using the Internet.
- **Charity (C)**: This is an intermediary between the donor and the recipient. It generates and issues e-donations to the recipients. It trusts the Pseudonym Server to issue valid electronic coins to the donors. It issues smart cards to its registered participants, and determines which recipients receive charitable donations. It also maintains relationships with stores from whom its registered recipients may redeem donations.
- **Recipient (R)**: A recipient is the entity that receives goods when redeeming an e-donation from a participating store. Each recipient has a smart card supplied by his charity. This card holds e-donations (specifying particular goods) issued by the charity which the holder can redeem from a participating store at a convenient time.
- **Store (S)**: This is an entity willing to sell its goods through participation in the scheme. It has an agreement with a charity to exchange e-donations generated by that charity for goods described in the e-donations. It uses a smart card terminal to receive e-donations from the recipient smart card.
- **Certification authority (CA)**: A trusted entity that generates public key certificates for charities, stores, and the pseudonym server.

- **Pseudonym server (PS)**: A trusted entity that will bind cryptographic data to participants. It provides an infrastructure for issuing anonymous identities and electronic coins to donors. It is trusted not to revoke the anonymity of a donor at any time, except under certain conditions agreed upon by all participants. It is trusted by all other parties and should be managed in such a way that fraud is very unlikely. To cover the operational costs of providing this kind of service, the PS might make a charge to the donor and/or the charity.

Trust is a critical issue in payment systems. In our model, we assume that the store and the charity trust each other. This trust is likely to be explicit as the store and the charity would be expected to have a formally established agreement that defines the trust and liability relationship. The donor trusts the charity to deliver his donation to a deserving recipient. The donor might need a receipt for his donation from the charity to prove it has been issued to a recipient. All participants trust the CA and the PS to be honest.

9.2.2 Interactions

E-donation will provide a recipient of charity with the digital representation of a right to claim goods of a specified type from a participating store. A participating store will first need to decide which types of goods it will make available for distribution via charitable means — for each such item it will generate an e-donation token. Each such token is a simple data structure containing a description of the goods to be purchased. Associated with each token will also be the cost for a donor to purchase a right for a recipient to receive the goods specified in the token. The charity publishes these e-donation tokens via its web site.

When a donor wishes to donate (see figure 9.1), he first contacts the PS to get an electronic cash coin which can only be used to donate to a charity, and an anonymous identity (pseudonym) used when communicating with a charity. After selecting the kind of donation he wishes to make at a charity web site, the donor makes the donation using the electronic cash coin received earlier from the PS. In response the charity generates an e-donation that satisfies the donor requirements, and keeps it in a database.

When the charity decides to issue an e-donation to a recipient, it retrieves this e-donation from the database and loads it into the recipient's smart card. The recipient collects the goods from a participating store in exchange for the e-donation contained in the smart card. At a later stage, the store sends all the redeemed e-donations to their respective charities for clearing.

An advantage of our scheme is transparency, i.e. the donor knows that a recipient will receive goods exactly as specified by the donor. Moreover, the charity does not need to be contacted during each redemption.

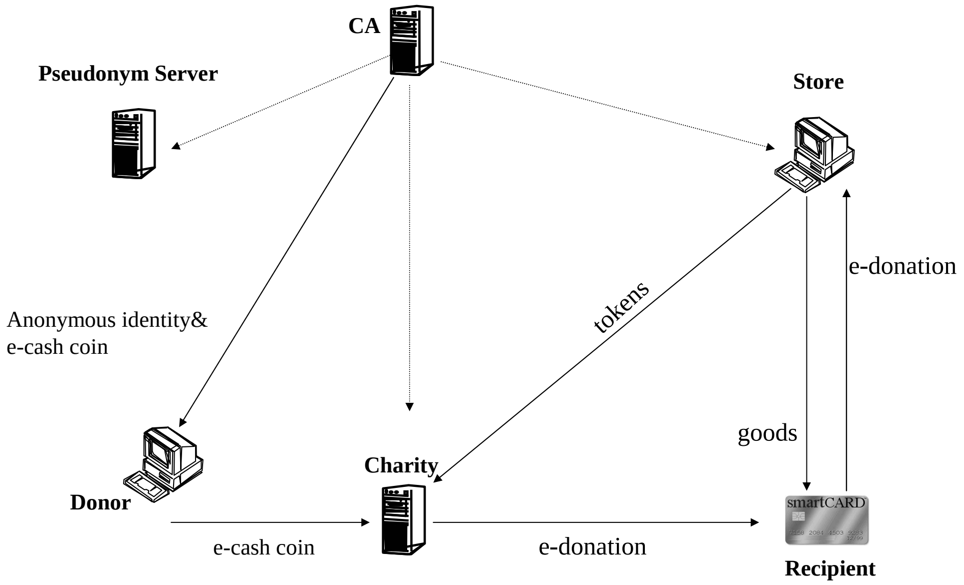

Figure 9.1: Secure electronic payment scheme for charity donations

9.3 Security requirements

The purpose of the scheme is to facilitate the transfer of donations from donors to recipients. However, the scheme provides the potential for considerable financial gains for those who attack it successfully. Therefore security measures must be provided to protect e-donation transactions. We discuss the security requirements that our scheme should satisfy.

9.3.1 Donor Anonymity

As previously discussed, donor privacy is important. This is especially important in an Internet environment where information may travel through network segments that are not necessarily trusted. The donor wants anonymity for his donation; neither the charity nor the recipient should be able to learn the donor's real identity. There are many reasons why anonymity might be required in a payment system [20]; in this case, the donor might not wish charities to be able to link different donations together and build a profile of his/her behaviour.

However, there are situations where anonymous payments can be misused for criminal activities [89]. Furthermore, there may be regulatory and legal constraints limiting anonymous donations. In order to make an anonymous electronic charity payment system acceptable to both donors and governments, a mechanism for limiting donor anonymity may also be needed.

9.3.2 Double-spending

Double-spending refers to the possibility of fraudulently spending the same e-donation more than once. Since e-donations are in digital form, they can readily be duplicated by the store or by the recipient, who may also blame each other for any fraudulent behaviour. If double-spending does take place, the charity will not know this until the store(s) involved send(s) the redeemed e-donations for clearing.

The scheme should protect against recipients attempting to redeem the same e-donation more than once and against stores attempting to deposit an e-donation multiple times. Ideally such double-spending or double-depositing should be prevented, although detection must be possible where prevention is not. Moreover, only the cardholder should be able to initiate a redemption transaction. Stores must be able to detect attempted double-spending without requiring any on-line verification from charities.

9.3.3 Integrity

E-donation data may be manipulated to attack the system. For example, a dishonest recipient may try to change an e-donation to extend its validity period or increase its value. Alternatively, an operator of a false or manipulated store terminal may interrogate the recipient's card and extract information which can later be used to obtain goods from a genuine store (at the expense of the genuine recipient). To combat these threats, it must not be possible to successfully fake or modify an e-donation.

9.4 The scheme

We now present a secure electronic payment scheme for charity donations. The proposed scheme is composed of five phases: the *Initialisation phase*, in which the store provides the charity with e-donation tokens that can be redeemed from the store during a specified interval of time; the *Anonymity phase*, in which the donor obtains an anonymous identity and an electronic coin from the PS; the *E-donation definition phase* wherein the donor selects an e-donation token to donate and pays for it; the *Donation phase* during which the charity loads the e-donation into a recipient smart card for redemption from a participating store; and the *Redemption phase* wherein the recipient pays an e-donation to a participating store in exchange for the described goods.

9.4.1 System set up

When initially establishing the system, the PS must decide on a number of fundamental system parameters, which must be reliably communicated to all parties within the system. These include selecting:

- A signature algorithm,
- A Message Authentication Code algorithm,
- A scheme to generate Anonymous Public Keys (APKs) and APK certificates (see section 2.4.3), and
- An anonymous electronic cash system with revocable anonymity that the PS must operate; an example of such a system is given in [37]. The proposed scheme requires the e-cash system to possess three main functions (typically involving special purpose exchanges of messages):

    1. Withdraw(val): A donor withdraws a coin c of value val from the PS.
    2. Payment(c): A donor pays a charity a coin c to make an e-donation.
    3. Deposit(c'): A charity deposits a spent coin c' with the PS which credits the charity account with the amount of val.

The CA must also generate its own signature key pair used for generating public key certificates, where, as throughout this thesis, the CA-generated certificate for public key P_X is written as Cert_{P_X}.

Prior to use of the system, every participating charity and store must generate and securely store their own secret MAC keys, denoted K_C and K_S respectively. Additionally, the participating organisations (charities, stores, and the PS) must register with the CA operating the system. Registration will involve the organisation (X say):

- Generating a signature key pair, with private key S_X and public key V_X,
- Obtaining a certificate Cert_{V_X} for V_X from the CA, and
- Obtaining a reliable copy of the public certificate verification key of the CA.

Each recipient R must be issued with a smart card by a charity, where smart card personalisation and issue involve the following steps.

1. The recipient card must be equipped with a signature key pair.
2. During smart card personalisation, the charity stores in the card a copy of the CA public key, the card expiry date, the charity public key certificate Cert_{V_C}, the card public key certificate Cert_{V_R} signed by the charity, and the recipient unique identifier id_R.
3. To prevent misuse of stolen or borrowed cards, we assume that the entry of a PIN by the authorised cardholder is required to use a recipient card.

9.4.2 Initialisation phase

In this phase the store provides the charity with a token that can be used as the basis of e-donations to be redeemed from the store during a specified interval of time. I.e.

$$token = s\_data \,||\, \mathrm{MAC}_{K_S}(s\_data)$$

where

$$s\_data = Item \,||\, Value \,||\, Expiry \,||\, id_S \,||\, id_C$$

and where $Item$ specifies the goods, $Value$ denotes the cost of the goods, $Expiry$ indicates the expiry date of the token, and id_S and id_C are identifiers for the store and the charity respectively.

The MAC protects the integrity of the token. The charity publishes the received e-donation tokens on its web site. This gives the donors choices for their donations.

9.4.3 Anonymity phase

This phase involves the donor and the PS. A donor must first obtain an APK certificate from the PS (see section 2.4.3). Donors then withdraw electronic coins from the PS, where these electronic coins can only be used to purchase e-donations from a participating charity. The electronic cash system is operated by the PS, which acts as a bank to donors and charities.

We now describe the Anonymity phase:

1. If the donor D does not already have an APK certificate, then the donor generates a signature key pair (with private key S_D). This key pair must be suitable for use with the APK scheme in use.
2. The donor visits the PS web site and submits: the donor identity id_D, the amount val to be donated, payment information (e.g. an account number), and the public key to be anonymously certified.
3. After collection of the payment from the donor using the specified payment information, the PS uses the provided donor public key to create the anonymised donor public key P_D, and generates an APK certificate Cert_{P_D} for P_D.
4. The donor uses the PS withdraw function to obtain a coin c of value val. The donor can use this coin to make an anonymous e-donation to a participating charity.

9.4.4 E-donation definition phase

This phase starts when a donor visits a charity web site and decides to donate through this charity. After browsing through the available e-donation tokens provided by participating stores, he selects the token that satisfy his requirements for value, donation type (e.g. food or clothing), validity period, and location where the e-donation will be spent. The donor then sends an e-donation request to the charity. To construct the request, the donor signs a message (step 1) that contains the selected charity token and a time stamp T to ensure message freshness. The donor sends the message, along with his APK certificate Cert_{P_D}, to the charity.

1. Donor $\longrightarrow$ Charity : $s_{S_D}(\,token \,||\, T) \,||\,$ Cert_{P_D}

After successful verification of the donor certificate, the signature on the message, and the expiry date within *token*, the charity creates an entry *c_data* in a donation requests database. We assume that the charity keeps a database of all donation requests received and awaiting use for generation of an e-donation. That is, the charity generates

$$c\_data = token \,||\, Serial\_number \,||\, Creation\_time$$

where *Serial_number* is a number that uniquely identifies this entry, and *Creation_time* indicates the date/time that the entry was created by the charity.

On generating the entry, the charity signs and sends a response message to the donor (step 2). The message contains a signed copy of the generated entry, along with the charity signature verification key certificate Cert_{V_C}.

2. Charity $\longrightarrow$ Donor : $s_{S_C}(\,c\_data) \,||\, \mathrm{Cert}_{V_C}$

On receiving the above message, the donor verifies the signature and that the entry was generated according to the donor requirements. If successful the donor and the charity engage in an electronic cash payment protocol that allows the donor to pay the coin c to the charity for the generated entry.

3. Donor $\longleftrightarrow$ Charity : Payment (c)

Upon receiving the payment from the donor, the charity interacts with the PS in an electronic cash deposit protocol to deposit the received coin c'.

4. Charity $\longleftrightarrow$ PS : Deposit (c')

If successful the charity adds the generated entry c_data to its database of donation requests. The donor must trust the charity to spend the donated coin in the way requested.

9.4.5 Donation phase

In this phase, the recipient smart card and the charity terminal engage in an authentication protocol, during which the recipient smart card receives e-donations. This protocol conforms to the mutual entity authentication mechanism specified in clause 5.2.2 of ISO/IEC 9798-3 [52].

This phase begins when the recipient presents his card to the charity to receive e-donations. First, the charity terminal reads the recipient's identity id_R and the recipient card public key certificate Cert_{V_R} from the card. Then, the charity generates a random number R_1, stores it, and sends it to the recipient card along with its unique identifier id_C (step 1).

1. Charity $\longrightarrow$ Recipient card : $R_1 \,||\, id_C$

After receiving the message in step 1, the recipient card generates and stores a random number R_2 as a challenge to the charity. It then creates a signed message that contains R_2, the charity identity id_C and the received random number R_1. The recipient card sends the generated signature to the charity terminal along with R_2.

2. Recipient card $\longrightarrow$ Charity : $s_{S_R}(R_1 \,||\, R_2 \,||\, id_C)$

After receiving the message in step 2, the charity terminal retrieves the recipient card signature verification key from Cert_{V_R} and uses it to verify that the received signature is valid. Moreover, it checks that the value of R_1 included in the message is the same as the value the charity terminal sent in message 1. If the verification fails, the process is terminated and the card is rejected. Otherwise, the charity terminal generates a response message that contains an $e-donation$ and sends it to the recipient card.

When a charity chooses to issue an $e-donation$ to a recipient, it retrieves an entry c_data from the donation requests database and adds $Issue\_time$, the current date/time, and the recipient unique identity id_R to that entry. Then, using its secret key K_C, the charity terminal computes and adds $\text{MAC}_{K_C}(c\_data\,||\,Issue\_time\,||\,id_R)$ to the retrieved entry. Thus,

$$e-donation = c\_data\,||\,Issue\_time\,||\,id_R\,||\,\text{MAC}_{K_C}(c\_data\,||\,Issue\_time\,||\,id_R)$$

The charity terminal now creates a signed message that contains the random numbers received in step 2, the recipient identifier id_R, and the $e-donation$. The charity sends the $e-donation$ and the generated signature to the recipient card.

3. Charity $\longrightarrow$ Recipient card : $e-donation\,||\,s_{S_C}(\,R_2\,||\,R_1\,||\,id_R\,||\,e-donation)$

Upon receiving the message in step 3, the recipient card verifies the stored charity public key certificate to obtain the charity public key and uses this to verify the signature. It then checks that the values of R_1 and R_2 are the same as the values that were sent in message 2. If the check fails, the process is terminated and the card does not accept any information from the terminal. Otherwise the card updates its stored list of e-donations. We assume that the charity keeps a database of all e-donations issued during a specific period. The charity also deletes the c_data used in the $e-donation$ generation from the donation requests database, and adds an entry to the e-donations database.

9.4.6 Redemption phase

In this phase, the recipient card and the store terminal engage in an authentication protocol, during which the store terminal retrieves an e-donation stored in the recipient card. Goods as specified in the e-donation are then provided to the recipient by the store. The recipient must trust the store terminal not to remove e-donations from the card that are not authorised by the recipient.

The store terminal first retrieves from the card the recipient unique identity id_R, the card expiry date, the charity public key certificate, the recipient card signature verification key certificate, and the list of e-donations. If the card has not expired, the store terminal first verifies the charity public key certificate using its stored copy of the CA public key, and then uses the recovered charity public key to verify the recipient card public key certificate. If successful, the store terminal then displays to the recipient a list of unredeemed e-donations, from which the recipient selects the one that is to be redeemed. Moreover, the store terminal asks the recipient card for a challenge.

1. Store $\longrightarrow$ Recipient card : e-donation to be redeemed

The recipient card responds by generating and storing a random number R_3, which it sends to the store terminal (step 2).

2. Recipient card $\longrightarrow$ Store : R_3

After receiving the message in step 2, the store terminal generates and stores a random number R_4 as a challenge to the recipient card. It then creates a signed message that contains R_4, the store identity id_S and the received random number R_3. The store terminal sends the generated signature to the recipient card along with the store signature verification key certificate Cert_{V_S}.

3. Store $\longrightarrow$ Recipient card : $s_{S_S}(\, R_4||\, id_S||R_3)||\ \mathrm{Cert}_{V_S}$

When the recipient card receives the message in step 3, it uses the stored CA public key to verify the store's signature verification key certificate. If the verification is successful then the card uses the store's public key to verify the received signature. If the signature verifies successfully, the recipient card first checks that that R_3 and id_S are correct, and then responds with a message that contains the selected unspent e-donation and a signature computed over the concatenation of that $e - donation$, R_4, R_3, and the identities of both the recipient and the store.

4. Recipient card $\longrightarrow$ Store : $s_{S_R}(e - donation \,||\, R_4 || \, R_3 \,||\, id_R \,||\, id_S)$

Upon receipt of the message in step 4, the store verifies the recipient card signature. Moreover, the store checks that the values of R_3 and R_4 included in the message are the same as were sent in message 3. If the signature verifies successfully and the check succeeds, then the store uses its secret key K_S to recompute $\text{MAC}_{K_S}(s\_data)$ and then checks the result against the MAC in the *token*, contained in the received $e - donation$. If the check succeeds then it accepts the $e - donation$ as valid and proceeds with providing the goods specified in the $e - donation$ to the recipient. Moreover, the recipient card marks the $e - donation$ used in message 4 as spent.

To help protect the card against fraud by the store, the recipient card logs message 4 for later settlement by the charity.

Similarly, protection of the store against the recipient is provided by exchanging the goods stated in the $e - donation$ for the message in step 4. The store later uses message 4 to collect the corresponding monetary amount from the charity. Typically, the transactions would be sent in a batch, signed by the store so that the charity can verify the integrity and authenticity of the transaction batch.

When the charity receives a copy of message 4 from the store, the charity uses its secret key K_c to verify that the MAC value in $e - donation$ is valid. Moreover, it uses the value of

id_R recovered from the received $e - donation$ to retrieve the signature verification key of the recipient smart card from its database. The charity verifies that the signature received in step 3 is valid, giving the charity assurance that the intended recipient has received the donated goods.

9.5 Security analysis

In this section, we examine to what extent the generic security requirements outlined in section 9.3 are met by our scheme.

9.5.1 Donor Anonymity

The anonymity of the donor is protected from the charity using the APK certificate, which allows a donation to be made to a charity without revealing the donor's real identity. Although the donor is not anonymous to the PS, since the donor makes a payment in exchange for an APK certificate and an electronic cash coin, it is not possible for the PS to know what donation a donor makes because the coin used to make the donation is anonymous. The PS would need to deanonymise the coin deposited by a charity to reveal the identity of the withdrawer.

9.5.2 Double-spending

Protection against e-donation double-spending is provided by means of smart cards. Our e-donation scheme is an off-line system, i.e. the store does not need to contact the charity during every redemption performed by a recipient. Instead, the scheme relies on a tamper-resistant recipient card that uses cryptographic means to recognise when it is communicating with a member of the scheme (i.e. a charity or a store). The charity and the recipient trust the recipient card to update its list of e-donations every time it is involved in a donation or redemption transaction with a member of the scheme. Moreover, since the recipient card is tamper-resistant, an attacker cannot modify the card contents without permanently damaging the card. Therefore, the recipient cannot benefit more than once from the same e-donation. Of course, no card is completely tamper-resistant, and the cost associated with setting up the scheme may be significant.

The charity also maintains a database of all the e-donations which have been issued. The charity uses the redeemed e-donations received from stores and recipient cards logs to detect and deal with double-spending after the event.

Typically, an e-donation would have a limited validity, e.g. the recipient will have a limited number of days to redeem the e-donation, in order to minimise the risks of forgery and to limit the size of the e-donation database. This database will be large, but not infeasibly so. There will need to be one database record per generated e-donation.

9.5.3 Integrity

Integrity protection for the e-donation data is accomplished using MACs and digital signatures. Calculating a MAC over parts of the messages exchanged using a key known only to the authorised parties provides evidence to the verifier that the message content has not been altered or destroyed, accidentally or with malicious intent, since it was created.

For example, if an attacker decided to change the $Item$ field found in the s_data part of $e-donation$ to his benefit, the attacker would need to modify $\text{MAC}_{K_S}(s\_data)$ to reflect the new value of the $Item$ field. However, our scheme assumes that no one other than the store who computed the original $\text{MAC}_{K_S}(s\_data)$ knows the key K_S. Moreover, we assume that the MAC function used is secure. The use of a MAC thus prevents such an attack.

On the other hand, theft of e-donations paid to a store is prevented by making such e-donations depositable only by that store. This is done by including the identity of the store in the signature $s_{S_R}(e-donation||R_4||id_R||id_S)$, which is created by the recipient card in step 4 of the *Redemption phase.*

9.5.4 Denial of service

An attacker could advertise 'fake tokens', containing a dummy MAC. Everything will work — the donor can 'buy' the token, the charity will debit the donor's bank account, and the recipient will get the value. However the recipient will be left with an unredeemable gift, since it is based on a false token which the genuine store will reject (the MAC will be wrong).

9.6 Conclusion

In this chapter we have proposed a scheme to make and distribute charitable donations electronically using the Internet and smart cards. We have described the scheme in detail, and explained how it meets the identified security requirements.

The infrastructure for the proposed scheme is complex. It requires the presence of a certification authority, and the pseudonym server to issue anonymous identities and electronic coins to the donors. The charity must maintain a web site to publish e-donation tokens and receive payments from donors. Moreover, the charity is expected to have the capability to issue smart cards to the recipients, and to provide card readers to load e-donations onto the recipient cards. Every merchant is required to install card readers at their stores to read e-donations from the recipient cards. Also, the merchant and the charity are expected to set up a secure link when sending e-donation tokens and redeeming e-donations. The donor is expected to have a PC and an Internet connection.

While the certification authority and the pseudonym server may cover their running cost by charging participated charities, these two entities could be removed to simplify the scheme (at the cost of losing anonymity for the donor) and to reduce its cost. The initial investment by the charity to set up the scheme could reduce the cost of distributing donations in the long run. Moreover, the charity is not expected to make a profit but only to cover its operational costs. The store's initial investment to set up the scheme will be paid back by the increase in sales of its goods.

Chapter 10

DISCUSSION AND CONCLUSIONS

Contents

This chapter summarises the primary contributions of this thesis and concludes with suggestions for future work.

10.1 Summary and conclusions

This thesis deals with secure electronic payment systems. The primary goal of the research described above was to develop secure payment schemes that satisfy the requirements posed by Islamic finance principles, which forbid the payment or receipt of interest. The main contribution of this thesis is to propose three protocols that can be used to conduct secure electronic commerce transactions based on Murabaha sale, and a scheme to allow Islamic charities to receive and distribute Zakat donations.

The properties that distinguish the various types of electronic payment systems have been identified. Our review of selected electronic payment systems (chapter 3) indicates that such a system can be used by a Muslim as long it does not involve paying interest on payments. However, the review also highlighted the inadequacies of these schemes for Muslims when they wish to buy goods on credit, either at point of sale terminals or using the Internet.

While face-to-face EMV debit card transactions are consistent with Islamic finance principles, since the payment is cleared immediately without payment of interest, EMV credit card transactions are not, since they involve dealing in interest. In chapter 5, a way to use an EMV card to conduct secure Murabaha transactions at the point of sale has been described. The scheme provides cardholder authentication, confidentiality, integrity, and non-repudiation. An advantage of the proposed extension to EMV is that the modified scheme does not involve any changes to EMV card functionality, i.e. an existing EMV card could be used to perform a Murabaha transaction. However, the scheme requires merchant terminal firmware to be upgraded, and both the acquirer and the issuer transaction processing software to be modified.

In chapter 6, we propose a method to conduct secure electronic Murabaha transactions on the Internet. An online Murabaha transaction involves more participants and different transaction interactions than those in conventional online purchases. The scheme uses public key cryptography to provide security services and to satisfy Murabaha requirements. A significant advantage of the scheme is that the buyer does not need to send any private information via the merchant, unlike in conventional e-commerce schemes where a credit card number is sent to a merchant protected using SSL/TLS. The scheme also keeps the identity of the buyer anonymous to the merchant, since the buyer does not need to reveal his identity to anyone other than the provider. Whilst the Murabaha transaction scenario described in this thesis is based on SET, other methods of Internet payment could easily be used to complete the transaction. SET has been used here primarily for the purposes of illustration.

The protocol described in chapter 6 requires the buyer to have a public key pair. This key pair would typically be stored in the buyer PC, and hence the buyer has to use this particular machine every time a transaction is to be made. Although a smart card could be employed to store the key and enhance mobility, not many user PCs are equipped with smart card readers. Therefore, with the goal of supporting user mobility, in chapter 7 we propose a GSM/UMTS-based scheme that does not require such a key pair to be generated; instead its security relies on the secret keys stored in a GSM SIM. Basing security on a mobile phone also inherently supports a level of user mobility.

Debit/credit cards provide a simple way for individuals to pay businesses for products and services, but they do not provide private individuals with the means to make payments to one another. There do exist schemes by which a payer can make a payment to anyone with an e-mail address. However, these schemes require both the payer and the payee to move money from their bank accounts to their account with the scheme provider in order to benefit from the service. Moreover, the payment instruction does not contain any information that links the goods being sold to the payment itself. Therefore, these schemes are not particularly appropriate for the case where a payment is made in exchange for goods. In chapter 8 we propose a new person-to-person Internet payment system that uses the existing relationships between buyers, sellers, and their respective banks to perform remote payments, and to link payment details to the goods being sold. The scheme uses SSL/TLS and additional signature key pairs to provides authentication, confidentiality, integrity, and non-repudiation.

Islamic charities still use manual methods to receive and distribute Zakat donations. This method has a high overhead in terms of storage of donated goods and the time consumed in distributing them. Recent developments in smart card technology and the popularity of the Internet could be used to make charitable donations easier and more cost effective for donors, charities, and recipients. In chapter 9 we therefore introduce the concept of e-donation, the electronic counterpart of a charity donation. We propose a scheme that allows donations to be made anonymously and distributed to recipients using smart cards, allowing recipients to redeem their e-donations directly from a shop. In particular, the scheme involves the donor contributing a specific amount of money to the charity, which then arranges for the recipient to receive goods of precisely the value contributed by the donor. Moreover the donor can specify the nature of the goods to be made available to the recipient. Whilst this is not a model of charitable donation of general applicability, it matches the particular requirements of certain scenarios, notably the obligation on Muslims to make donations during Eid.

Differing characteristics of local environments have created a significant level of variation in the acceptance and growth of e-commerce in different regions of the world. Consumer participation in e-commerce is impacted by access to technology (computers, connectivity, and gateway to Internet), and payment systems for enabling transfer of funds. Access to technology is in a constant state of change and improvement, and we suggest that, in a relatively short time, they will cease to be a very significant obstacle to e-commerce development in many Muslim countries. A review of the involvement of Islamic banks with e-commerce technologies has to date been quite limited.

We think that Islamic banks have an important reason to pursue the conduct of e-commerce. If they fail to respond to the opportunities posed by the Internet and smart card technologies, they could be consigned to a largely secondary role as commerce is increasingly conducted electronically. This thesis has demonstrated that it is possible to design traditional Islamic banking products more efficiently and to develop and sell new products sought by e-commerce participants. Moreover, this thesis has proposed a new area for researchers in electronic payment systems, namely to propose new ways to design the electronic equivalents of Islamic banking products, or to evaluate existing schemes.

10.2 Suggestions for future work

This thesis represents a first attempt to design secure electronic payment schemes aimed at Islamic financial instruments. It is likely that there are many other ways of constructing electronic payment schemes consistent with Islamic principles, perhaps based on some of the other electronic payment schemes examined in chapter 3. Moreover, the thesis suggests that there are many possible directions in which to continue the research, especially with regard to the other financial instruments presented in chapter 4.

The design of the new secure electronic Murabaha transaction indicates that there is still scope for innovative transaction models on the Internet. The proposed scheme in chapter 6 was built upon SET to create a secure e-commerce Murabaha transaction. One obvious area for future work would be to modify recently developed and emerging secure e-commerce payment schemes to support Murabaha payments, notably Visa 3-D Secure. It would also be interesting to investigate how an EMV card can be used to help conduct a Murabaha transaction on the Internet.

The scheme proposed in chapter 9 uses smart cards for the distribution and redemption of charitable donations. The increase in ownership of mobile phones suggests that they could be used as a substitute for smart cards in this scheme. It would therefore be interesting to investigate how the proposed scheme could be modified to allow the recipient to receive and redeem e-donations using a mobile phone instead of a smart card. This would clearly avoid the costs of personalising and issuing a separate smart card to each recipient.

The security analyses performed on the proposed schemes was informal, and we have built the proposed schemes out of existing/standardised schemes as much as possible, in the belief that re-using protocols which have existing proofs of security should help yield secure mechanisms. However, it would be of interest to evaluate the security properties and efficiency of the proposed schemes using mathematical techniques. This is a particularly challenging and time-consuming task given the complexity of electronic payment protocols.

The schemes presented in the thesis were not implemented, and it would also be useful to validate them by means of practical tests. It is likely that new issues will emerge from such prototyping, and the proposed protocols are likely to need to be modified accordingly. However, prototyping would certainly give no guarantees about the security of the schemes.

One of the most significant advantages of smart cards and electronic communications is the potential for reducing the physical storage needs for transaction records, and for speeding up the conclusion of transactions. In Saudi Arabia today, a real estate sale transaction is usually concluded in a land registry office with the buyer and seller present. When the buyer confirms receipt of payment from the seller (e.g. in the form of a cashier cheque) to the land registry, the title is transferred to the buyer. In addition, the land registry records are updated based on the outcome of the sale. However, performing this manual process is time consuming for all parties involved. It would be interesting to investigate the possibility of conducting an electronic real estate transaction that binds the title transfer to an electronic payment and to use smart cards as a secure device for storing the real estate title details.

Finally, the scale of online trading in stocks is rapidly growing. In current online systems, including stock exchanges, brokers act as trusted intermediaries between customers. Brokers guarantee the financial status of customers, and provide anonymity in the market. However, there is currently no electronic means for customers to buy shares on a Murabaha basis. It would be interesting to investigate how the role of brokers might be extended to include buying shares for customers on a Murabaha basis.

Bibliography

[1] 3rd Generation Partnership Project (3GPP). *3GPP TS 33.102: Security architecture V6.0.0*, 2003.

[2] 3rd Generation Partnership Project (3GPP). *3GPP TS 31.111: USIM Application Toolkit (USAT) V6.1.0*, March 2004.

[3] J.L. Abad-Peiro, N. Asokan, M. Steiner, and M. Waidner. Designing a generic payment service. *IBM Systems Journal*, 37(1):72–88, January 1998.

[4] Central Intelligence Agency. World Factbook, 2003. Available at http://www.cia.gov.

[5] M. T. Al-Hilali and M. M. Khan. *Interpretation of the meanings of the noble QURAN.* Darussalam Publishers, Houston, USA, 1996.

[6] M. A. Al-Meaither and C. J. Mitchell. A person-to-person Internet payment system. In Hanne Riis Nielson, editor, *Proceedings of 6th Nordic Workshop on Secure IT Systems*, pages 5–17. Technical University of Denmark, November 2001.

[7] M. A. Al-Meaither and C. J. Mitchell. Extending EMV to support Murabaha transactions. In S. Knapskog, editor, *Proceedings of Seventh Nordic Workshop on Secure IT Systems*, pages 95–108. Department of Telematics, Norwegian University of Science and Technology, Norway, October 2003.

[8] M. A. Al-Meaither and C. J. Mitchell. A secure electronic Murabaha transaction. In R. T. Wigand, Y.-H. Tan, J. Gricar, A. Pucihar, and T. Lunar, editors, *Proceedings*

of eTransformation, 16th Bled eCommerce Conference, pages 662–674, Bled, Slovenia, University of Maribor, June 2003.

[9] M. A. Al-Meaither and C. J. Mitchell. A secure electronic payment scheme for charity donations. In Kurt Bauknecht, A. Min Tjoa, and Gerald Quirchmayr, editors, *Proceedings of EC-Web 2003, 4th International Conference E-Commerce and Web Technologies*, volume 2738 of *Lecture Notes in Computer Science*, pages 50–61, Springer-Verlag, Berlin, September 2003.

[10] M. A. Al-Meaither and C. J. Mitchell. A secure GSM-based Murabaha transaction. In *Proceedings of the 1st International Conference on Information and Communication Technologies from Theory to Applications*, pages 77–78. IEEE Press, April 2004.

[11] A. Al-tyar. *Islamic Banks between theory and application.* Dar alwtan, Riyadh, Saudi Arabia, 1994.

[12] M. Anderson. Architectural overview of the FSTC eCheck system. Available at http://www.echeck.org/.

[13] N. Asokan, P. Janson, M. Steiner, and M. Waidner. The state of the art in electronic payment systems. *IEEE Computer*, 30(9):28–35, September 1997.

[14] N. Barnett, S. Hodges, and M. J. Wilshire. M-commerce: An operators manual. *McKinsey Quarterly*, 3:162–173, 2000.

[15] A. Bhati and S. Sahai. Dial M for money. In *Proceedings of the Second ACM International Workshop on Mobile Commerce (WMC-02)*, pages 95–99. ACM Press, New York, NY, USA, 2002.

[16] D. Birch. The ABC of EMV, 1999. Available at http://www.hyperion.co.uk.

[17] C. W. Blanchard. Security for the third generation 3G mobile system. *Information Security Technical Report*, 5(3):55–65, 2000.

[18] M. I. Bukhari. *Sahih Bukhari*, volume 1. Dar al-Kotob al-ilmiyah Publishers, Beirut, Lebanon, 2003.

[19] H. Chan, R. Lee, T. Dillon, and E. Chang. *E-Commerce Fundamentals and Applications*. John Wiley & Sons, Chichester, West Sussex, U.K., 2001.

[20] D. Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology — CRYPTO '82*, pages 199–203. Plenum Press, 1983.

[21] J. Claessens, B. Preneel, and J. Vandewalle. Combining world wide web and wireless security. In B. De Decker, F. Piessens, J. Smits, and E. Van Herreweghen, editors, *Advances in Network and Distributed Systems Security, Proceedings of IFIP TC11 WG11.4 First Annual Working Conference on Network Security*, pages 153–171. Kluwer Academic Publishers, 2001.

[22] Mandate II Consortium. Mandate final report. European Communities DGXIII Electronic Trusted Services Programme, February 1998.

[23] Craver, Mathews, Smith, and Company. Socially engaged internet users: Prospects for online philanthropy and activism, September 1999. Available at http://www.craveronline.com/.

[24] T. Dierks and E. Rescorla. *The TLS Protocol Version 1.0.* Certicom, January 1999. Internet RFC 2246.

[25] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[26] M. A. El-Gamal. *A Basic Guide to Contemporary Islamic Banking and Finance.* Islamic Society of North America, Plainfield, Indiana, USA, 2000.

[27] EMVCo. *EMV2000: Integrated Circuit Card Specification for Payment Systems: Book 1 — Application Independent IC Card to Terminal Interface Requirements*, 2000.

[28] EMVCo. *EMV2000: Integrated Circuit Card Specification for Payment Systems: Book 2 — Security and Key Management*, 2000.

[29] EMVCo. *EMV2000: Integrated Circuit Card Specification for Payment Systems: Book 3 — Application Specification*, 2000.

[30] EMVCo. *EMV2000: Integrated Circuit Card Specification for Payment Systems: Book 4 — Cardholder, Attendant, and Acquirer Interface Requirements*, 2000.

[31] European Committee for Banking Standards (ECBS), Brussel. *EBS204, IBAN: International Bank Account Number Version 3.2*, 2003.

[32] European Telecommunications Standards Institution (ETSI). *Digital cellular telecommunications system (Phase 2+); Technical Realization of the Short Message Service (SMS, version 7.4.0)*, 1999.

[33] European Telecommunications Standards Institution (ETSI). *Digital cellular telecommunications system (Phase 2+); Specification of the SIM Application Toolkit for the Subscriber Identity Module — Mobile Equipment (SIM-ME) Interface*, August 2000.

[34] European Telecommunications Standards Institution (ETSI). *Digital cellular telecommunications system (Phase 2+); GSM Security Aspects (GSM 02.09 version 8.0.1)*, June 2001.

[35] L. Ferreira and R. Dahab. A scheme for analyzing electronic payment systems. In *Proceedings of 14th Annual Computer Security Applications Conference*, pages 137–146. IEEE Computer Society Press, December 1998.

[36] Mobile Payment Forum. Enabling secure, interoperable, and user-friendly mobile payments. Mobile Payment Forum White Paper, December 2002. Available at http://www.mobilepaymentforum.org/.

[37] Y. Frankel, Y. Tsiounnis, and M. Yung. Fair off-line e-cash made easy. In K. Ohta and D. Pei, editors, *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, volume 1514 of *Lecture Notes in Computer Science*, pages 257 – 270, Springer-Verlag, Berlin, October 1998.

[38] A. O. Freier, P. Karlton, and P. C. Kocher. The SSL protocol, version 3.0, November 1996. Internet Draft.

[39] K. Fu, E. Sit, K. Smith, and N. Feamster. The Dos and Don'ts of Client Authentication on the Web. In Dan S. Wallach, editor, *Proceedings of the 10th USENIX Security Symposium*, pages 251–270, Washington, D.C., August 2001.

[40] N. Haller. *The S/KEY one-time password system.* Bellcore, February 1995. Internet RFC 1760.

[41] B. Hamwi and A. Aylward. Islamic finance: a growing international market. *Thunderbird International Business Review*, 41(4–5):407–420, 1999.

[42] S. Haron. *Islamic Banking: Rules and Regulations.* Pelanduk Publications, Selangor Darul Ehsan, Malaysia, 1997.

[43] S. Haron and B. Shanmugam. *Islamic Banking System Concepts & Applications.* Pelanduk Publications, Selangor Darul Ehsan, Malaysia, 2001.

[44] F. Hasanin. *Murabaha Sale in Islamic Banks.* The International Institute of Islamic Thought, Herndon, VA, USA, 1996.

[45] V. Hassler. *Security Fundamentals for E-commerce.* Artech House, Norwood, MA, USA, 2001.

[46] D. L. Hoffman, T. P. Novak, and M. Peralta. Building Consumer Trust Online. *Communications of the ACM*, 42(4):80–85, April 1999.

[47] D. Humphrey, M. Kim, and B. Vale. Realizing the gains form electronic payments: Costs, pricing and payment choice. *Journal of Money, Credit, and Banking*, 33(2):216–234, 2001.

[48] I.A. Ibrahim. *A brief illustrated guide to understanding Islam.* Darussalam Publishers, Houston, USA, 2nd edition, 1997.

[49] The International Association of Islamic Banks, Jeddah, Saudi Arabia. *Directory of Islamic Banks and financial Institutions*, 1997.

[50] International Organization for Standardization (ISO), Geneva. *ISO 7498-2, Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture*, 1989.

[51] International Organization for Standardization (ISO), Geneva. *ISO 9564-1, Banking – PIN management and security – Part 1: PIN protection principles and techniques*, 1991.

[52] International Organization for Standardization (ISO), Geneva. *ISO/IEC 9798-3, Information technology — Security techniques — Entity authentication mechanisms — Part 3: Mechanisms using digital signature techniques*, 2nd edition, 1998.

[53] International Organization for Standardization (ISO), Geneva. *ISO/IEC 9797-1, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*, 1999.

[54] International Organization for Standardization (ISO), Geneva. *ISO/IEC 9797-2, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a hash function*, 2000.

[55] International Organization for Standardization (ISO), Geneva. *ISO/IEC 9594-8, Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*, 2001.

[56] Z. Iqbal and A. Mirakhor. Progress and challenges of Islamic banking. *Thunderbird International Business Review*, 41(4–5):381–405, 1999.

[57] ITU-T Recommendation X.509. *Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks*, 4th edition, 2000. Geneva.

[58] ITU-T Recommendation X.800. *Security Architecture for Open Systems Interconnection for CCITT Applications*, 1991. Geneva.

[59] M. Jakobsson, D. MRaihi, Y. Tsiounis, and M. Yung. Electronic payments: Where do we go from here? In Rainer Baumgart, editor, *Secure Networking — CQRE [Secure] '99*, volume 1740 of *Lecture Notes in Computer Science*, pages 34–63, Springer-Verlag, Berlin, 1999.

[60] M. Kahf and T. Khan. *Principles of Islamic Financing (A Survey).* Islamic Development Bank — Islamic Research and Training Institute, Jeddah, Saudi Arabia, 1993. http://www.irti.org/.

[61] R. Kalakota and A.B. Whinston. *Electronic Commerce: A Manager's Guide.* Addison-Wesley, Reading, MA, 1997.

[62] V. Khu-smith and C.J. Mitchell. Using GSM to enhance e-commerce security. In *In Proceedings of the Second ACM International Workshop on Mobile Commerce (WMC 02)*, pages 75–81. ACM Press, 2002.

[63] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.

[64] J. K. MacKie-Mason and K. White. Evaluating and selecting digital payment mechanisms. In G. Rosston and D. Waterman, editors, *Interconnection and the Internet*, Lawrence Erlbaum, pages 113–134, 1997.

[65] Ibn Mandoor. *Lesan Al Arab Dictionary.* Dar Al Fikr Publishing, Damascus, Syria, 1997.

[66] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography.* CRC Press, Boca Raton, FL, USA, 1997.

[67] C. J. Mitchell and L. Chen. Comments on the S/KEY user authentication scheme. *ACM Operating Systems Review*, 30(4):12–16, 1996.

[68] Mobile Electronic Transactions. *MeT Account-Based Payment*, 2001. Available at http://www.mobiletransaction.org.

[69] National Institute of Standards and Technology (NIST). *Federal Information Processing Standards Publication 197 (FIPS PUB 197): Specification for the Advanced Encryption Standard (AES)*, November 2001.

[70] National Institute of Standards and Technology (NIST). *Federal Information Processing Standards Publication 180-2 (FIPS PUB 180-2): Secure Hash Standard*, August 2002.

[71] B.C. Neuman and G. Medvinsky. Requirements for Network Payment: The NetCheque Perspective. In *Proceedings of IEEE Compcon '95*, pages 32–36. IEEE-CS Press, March 1995.

[72] K. Oishi, M. Mambo, and E. Okamoto. Anonymous public key certificates and their applications. *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences*, E81-A(1):56–64, January 1998.

[73] D. O'Mahony, M. Peirce, and H. Tewari. *Electronic Payment Systems for E-Commerce.* Artech House, Norwood, MA, USA, 2nd edition, 2001.

[74] B. Pfitzmann and M. Waidner. Properties of payment systems — General definition sketch and classification. Technical Report RZ 2823, IBM Zurich Research Laboratory, May 1996.

[75] S. M. Redl, M. K. Weber, and M. W. Oliphant. *An Introduction to GSM.* Artech House, Norwood, MA, USA, 1995.

[76] R. L. Rivest and A. Shamir. Payword and MicroMint: Two simple micropayment schemes. *CryptoBytes*, 2(1):7–11, 1996.

[77] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[78] A. D. Rubin and R. N. Wright. Off-line generation of limited use credit card numbers. In Yair Frankel, editor, *Proceedings of the 4th International Conference on Financial Cryptography*, volume 1962 of *Lecture Notes in Computer Science*, pages 196–209, Springer-Verlag, Berlin, 2001.

[79] B. Schoenmakers. Basic security of the ecash payment system. In Bart Preneel and Vincent Rijmen, editors, *Computer Security and Industrial Cryptography: State of the Art and Evolution*, volume 1528 of *Lecture Notes in Computer Science*, pages 342–356, Springer-Verlag, Berlin, 1998.

[80] SETCo. *Secure Electronic Transaction Standard: Book 1 — Business Description*, 1997. Available at http://www.setco.org.

[81] SETCo. *Secure Electronic Transaction Standard: Book 2 — Programmers guide*, 1997. Available at http://www.setco.org.

[82] SETCo. *Secure Electronic Transaction Standard: Book 3 — Formal Protocol Definition*, 1997. Available at http://www.setco.org.

[83] A. Shamir. Secureclick: A web payment system with disposable credit card numbers. In Paul F. Syverson, editor, *Proceedings of the 5th International Conference on Financial Cryptography*, volume 2339 of *Lecture Notes in Computer Science*, pages 232–242, Springer-Verlag, Berlin, 2001.

[84] M. H. Sherif. *Protocols for Secure Electronic Commerce.* CRC Press, Boca Raton, FL, USA, 2nd edition, 2003.

[85] M. Sirbu and J. D. Tygar. Netbill: An Internet commerce system optimized for networked delivered services. In G. Rosston and D. Waterman, editors, *Proceedings of IEEE Compcon '95*, pages 20–25. IEEE-CS Press, 1995.

[86] M. Stadler, J. Piveteau, and J. Camenisch. Fair blind signatures. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology – EUROCRYPT '95 Proceedings*, volume 921 of *Lecture Notes in Computer Science*, pages 209–19, Springer-Verlag, Berlin, 1995.

[87] J. Stavins. Effect of Consumer Characteristics on the Use of Payment Instruments. *New England Economic Review*, 3(4–5):19–31, Summer 2001.

[88] VISA. *3-D Secure Protocol Specification: Core functions Version 1.0.2*, January 2003. Available at http://www.international.visa.com/.

[89] S. von Solms and D. Naccache. On blind signatures and perfect crimes. *Computers and Security*, 11(6):581–583, 1992.

[90] D. Wagner and B. Schneier. Analysis of the SSL 3.0 protocol. In Doug Tygar, editor, *the Second USENIX Workshop on Electronic Commerce*, pages 29–40. USENIX Press, 1996.

[91] M. Walker and T. Wright. Security. In F. Hillebrand, editor, *GSM and UMTS: The creation of global mobile communication*, chapter 14, pages 385–406. John Wiley & Sons, 2002.

[92] M. Ward. EMV — The ICC specifications for Payment Systems. *Information Security Technical Report*, 4(2):51–57, 1999.

[93] R. Wilson. *Banking and Finance in the Arab Middle-East.* Palgrave Macmillan, New York, USA, 1983.