

ARTECH HOUSE

COMPUTER SECURITY SERIES

implementing
**Electronic Card
Payment Systems**



C R I S T I A N R A D U

General Library System
University of Wisconsin - Madison
728 Stat.
Madison, WI 53706-1494
U.S.A.

Implementing Electronic Card Payment Systems

Cristian Radu



Artech House
Boston • London
www.artechhouse.com

General Library System
University of Wisconsin - Madison

728 State Street
Madison, WI 53706-1494
U.S.A.

Library of Congress Cataloging-in-Publication Data

Radu, Cristian.

Implementing electronic card payment systems. / Cristian Radu.

p. cm.—(Artech House computer security series)

Includes bibliographical references and index.

ISBN 1-58053-305-1 (alk. paper)

1. Electronic funds transfers. 2. Credit cards. 3. Debit cards. 4. Payment—Data processing.
5. Electromagnetic compatibility. I. Title. II. Series.

HG1710 .R33 2003

658.8'8—dc21

2002033224

British Library Cataloguing in Publication Data

Radu, Cristian

Implementing electronic card payment systems.—(Artech House computer security series)

1. Electronic commerce—Security measures 2. Electronic funds transfers 3. Smart cards

I. Title

005.8'2

ISBN 1-58053-305-1

Cover design by Yekaterina Ratner

© 2003 ARTECH HOUSE, INC.

685 Canton Street

Norwood, MA 02062

SET™ is a trademark owned by SET Secure Electronic Transaction LLC. CEPST™ is a trademark of CEPSTCo, LLC. EMV™ is a trademark owned by EMVCo, LLC. The author acknowledges that the use of the word EMV™ shall never imply authorization or endorsement by EMVCo, LLC, nor will it imply that the author has obtained EMVCo approval for the content of this book. The author agrees and acknowledges that the use of the EMV™ specifications for writing this book does not create any relationship between EMVCo and the author. 3-D Secure™ is a trademark owned by Visa International. 3-D SET™ is a trademark owned by Visa International. SPA™ is a trademark owned by MasterCard International.

All rights reserved. Printed and bound in the United States of America. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Artech House cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

International Standard Book Number: 2002033224

Library of Congress Catalog Card Number: 1-58053-305-1

10 9 8 7 6 5 4 3 2 1

Contents

Acknowledgments	xv
---------------------------	----

1 Introduction 1

Part I: Magnetic stripe debit and credit cards	3
Part II: Chip migration with EMV™	3
Part III: Remote debit and credit with EMV™	5

Part I: Magnetic Stripe Debit and Credit Cards . 7

2 Payment Card Processing 9

2.1 Payment card processing at a glance	10
2.2 Roles involved in payment card processing	13
2.3 Payment card brands	15
2.4 Credit and debit payment cards	16
2.5 Focusing on the magnetic stripe card	17
2.5.1 Embossed financial data	18
2.5.2 Financial data on the magnetic stripe	20
2.6 Threats and security protections	24
2.6.1 Channel protection versus eavesdropping	25
2.6.2 Cardholder verification versus impersonation	27
2.6.3 Static authenticator versus modifying financial data	30
2.6.4 Timeliness versus card counterfeiting	31
2.6.5 Merchant attacks and colluding attacks	33
2.7 Processing at the point of service	34

2.8	Payment network and interchange messages	37
2.8.1	<i>Message structure</i>	38
2.8.2	<i>Message flows</i>	41
2.9	On-line authorization	45
2.10	Clearing and settlement	47
	References	50

Part II: Chip Migration with EMV™ 51

3 Chip Migration 53

3.1	A business case for chip migration	54
3.2	An overview of the chip card technology	56
3.2.1	<i>Hardware and software structure of chip cards</i>	57
3.2.2	<i>Card file system and file referencing</i>	60
3.2.3	<i>Command and response format</i>	65
3.2.4	<i>Card application and terminal application</i>	66
3.3	Proprietary payment application	69
3.3.1	<i>Encoding data elements with a fixed format</i>	71
3.3.2	<i>Fixed file system organization</i>	73
3.3.3	<i>Preestablished command and response formats</i>	73
3.3.4	<i>Symmetric cryptographic technology</i>	76
3.4	Interoperable payment application	80
3.4.1	<i>Self-determined encoding of data elements</i>	82
3.4.2	<i>Customized file system organization</i>	84
3.4.3	<i>Variable formats for commands and responses</i>	87
3.4.4	<i>Asymmetric cryptographic support</i>	87
	References	90

4 EMV™ Compliant Data Organization 91

4.1	Organization of the EMV™ specifications	92
4.2	EMV™ data elements	96
4.3	EMV™ file system	99
4.3.1	<i>ADFs</i>	99
4.3.2	<i>AEFs</i>	106
4.3.3	<i>Directory definition files</i>	108

4.3.4	<i>Payment system environment</i>	112
4.4	EMV™ application selection	115
4.4.1	<i>Building the candidate list from the PSE</i>	118
4.4.2	<i>Building the candidate list directly</i>	119
4.4.3	<i>Final application selection</i>	121
	References	122

5 EMV™ Certificates 125

5.1	Certification mechanism and algorithm	125
5.2	Public key certificate for RSA scheme	126
5.3	Entities and certifiers	127
5.3.1	<i>Issuer requires a public key certificate</i>	127
5.3.2	<i>ICC requires a public key certificate</i>	128
5.4	Entity public key remainder	129
5.5	EMV™ certification chains	129
5.6	Issuing EMV™ public key certificates	132
5.6.1	<i>Data items included in the certificate</i>	132
5.6.2	<i>Generating the public key certificate</i>	135
5.7	Verifying EMV™ public key certificates	136
5.7.1	<i>Verification of the Issuer Public Key Certificate</i>	136
5.7.2	<i>Verification of the ICC Public Key Certificate</i>	138
5.8	Issuing signed static application data	140
5.8.1	<i>AFL</i>	141
5.8.2	<i>Creating the Static Data to Be Authenticated</i>	142
5.8.3	<i>Generate the Signed Static Application Data</i>	143
5.9	Verifying the Signed Static Application Data	144
	References	145

6 Debit and Credit with EMV™ 147

6.1	Overview of the EMV™ debit/credit transaction	148
6.2	Initiate application processing	152
6.2.1	<i>TVR and TSI—two witnesses of terminal processing</i>	152
6.2.2	<i>PDOL and GET PROCESSING OPTIONS</i>	153
6.2.3	<i>AIP and AFL</i>	154
6.3	Read application data	156

6.3.1	<i>AFL processing</i>	156
6.3.2	<i>Consistency rules for the data objects</i>	158
6.4	<i>Off-line data authentication</i>	160
6.4.1	<i>Selection of the off-line authentication mechanism</i>	160
6.4.2	<i>Off-line SDA</i>	162
6.4.3	<i>Off-line DDA</i>	165
6.5	<i>Processing restrictions</i>	174
6.5.1	<i>Application Version Number</i>	174
6.5.2	<i>Application usage control</i>	175
6.5.3	<i>Application effective/expiration dates checking</i>	178
6.6	<i>Cardholder verification</i>	178
6.6.1	<i>Cardholder verification methods in EMV™</i>	179
6.6.2	<i>Data objects involved in cardholder verification</i>	181
6.6.3	<i>Common processing performed by the terminal</i>	184
6.6.4	<i>Off-line PIN processing</i>	186
6.6.5	<i>RSA digital envelope carrying the PIN</i>	191
6.6.6	<i>On-line PIN processing</i>	194
6.7	<i>Terminal risk management</i>	195
6.7.1	<i>Terminal floor limit</i>	195
6.7.2	<i>Random transaction selection</i>	196
6.7.3	<i>Velocity checking</i>	199
6.8	<i>Terminal action analysis</i>	201
6.8.1	<i>Action codes and security policies</i>	201
6.8.2	<i>The terminal proposes and the card disposes</i>	203
6.8.3	<i>Off-line denial of a transaction</i>	204
6.8.4	<i>On-line transmission of a transaction</i>	206
6.8.5	<i>Default action in a transaction</i>	207
6.8.6	<i>Compute Application Cryptogram with GENERATE AC</i>	208
6.9	<i>On-line processing and issuer authentication</i>	217
6.9.1	<i>Authorization request and response with chip data</i>	218
6.9.2	<i>Issuer Authentication</i>	221
6.10	<i>Issuer scripts</i>	222
6.10.1	<i>Processing of issuer script templates</i>	222
6.10.2	<i>Post-Issuance Commands</i>	225
	<i>References</i>	225

1	EMV™ Chip Migration Issues	227
7.1	EMV™ regulatory framework	228
7.1.1	<i>Business objectives</i>	229
7.1.2	<i>Functional requirements</i>	231
7.1.3	<i>Security politics</i>	233
7.2	Deriving ICC specifications by issuers	236
7.3	Selection criteria of the ICC architecture	239
7.3.1	<i>ICC hardware resources</i>	239
7.3.2	<i>ICC software platform</i>	241
7.4	Multiapplication ICC	242
7.4.1	<i>Choice of a set of card applications</i>	243
7.4.2	<i>Card layout definition</i>	246
7.5	Issuer's business case	253
7.5.1	<i>Availability of the financial service</i>	253
7.5.2	<i>Improved security</i>	254
7.5.3	<i>Reduced operational costs</i>	255
7.6	Adaptive initiate application processing	255
7.7	Design criteria for CAM selection	259
7.7.1	<i>On-line CAM</i>	260
7.7.2	<i>Off-line static CAM</i>	261
7.7.3	<i>Off-line dynamic CAM</i>	262
7.7.4	<i>Security considerations regarding CAM</i>	263
7.8	Design criteria for CVM	267
7.8.1	<i>Enciphered PIN verified on-line</i>	267
7.8.2	<i>Plaintext/enciphered PIN verification by ICC</i>	268
7.8.3	<i>Requirements for the implementation of various CVM</i>	269
7.8.4	<i>Criteria for the definition of the CVM List</i>	270
7.9	Processing restrictions	271
7.9.1	<i>Application usage control</i>	271
7.9.2	<i>Application Version Number</i>	272
7.9.3	<i>Application effective/expiration dates</i>	272
7.10	Card risk management	273
7.10.1	<i>CRM Components</i>	273
7.10.2	<i>The set of CRM functions</i>	274
7.10.3	<i>CRM data</i>	278
7.10.4	<i>CRM function definitions</i>	283
	References	286

Part III: Remote Debit and Credit with EMV™ . 289

8 Remote Card Payments and EMV™ 291

8.1	A model for remote card payments	293
8.2	Security aspects of remote card payments	295
8.2.1	<i>Threats environment</i>	296
8.2.2	<i>Security services for remote transactions</i>	300
8.2.3	<i>Security services realization</i>	304
8.3	Remote payment method based on TLS	306
8.3.1	<i>TLS handshake protocol</i>	307
8.3.2	<i>TLS record protocol</i>	309
8.3.3	<i>Security limitations of the TLS protocol</i>	309
8.4	SET-based solutions	310
8.4.1	<i>SET model</i>	311
8.4.2	<i>Setup of the SET payment scheme</i>	311
8.4.3	<i>Registration of participants</i>	315
8.4.4	<i>Secure SET channel over insecure networks</i>	317
8.4.5	<i>SET dual signatures</i>	321
8.4.6	<i>SET payment method</i>	322
8.5	TLS versus SET or wallet servers and EMV™ cards	332
8.5.1	<i>Security makes the difference</i>	332
8.5.2	<i>Acceptability is a main concern</i>	333
8.5.3	<i>Improved solutions with wallet servers and EMV™ cards</i>	336
8.6	Transaction processing for chip e-commerce	340
8.6.1	<i>EMV™ application context in the cardholder system</i>	342
8.6.2	<i>Purchase initialization (PInitReq/PInitRes)</i>	346
8.6.3	<i>Cardholder verification</i>	347
8.6.4	<i>Terminal action analysis</i>	349
8.6.5	<i>Purchase request and response</i>	350
8.6.6	<i>Authorization request/response</i>	353
8.6.7	<i>Completion of the EMV™ transaction</i>	355
	References	356

Appendix A: Security Framework 359

Reference	361
-----------	-----

Appendix B: Generic Security Threats	363
Appendix C: Security Services	367
C.1 Service description	367
C.2 Realization of security services	370
References	371
Appendix D: Security Mechanisms	373
D.1 Encryption	373
D.1.1 Symmetric encryption	374
D.1.2 Asymmetric encryption	375
D.2 Cryptographic hash functions	376
D.2.1 Hash function	377
D.2.2 MAC	379
D.3 Digital signature schemes	380
D.3.1 Signature scheme with appendix	382
D.3.2 Signature scheme with recovery	383
D.4 Public key certificates	384
D.4.1 Authenticity of public keys	384
D.4.2 Public key certificate generation	385
D.4.3 Public key certificate verification	386
D.5 Cardholder verification mechanisms	387
D.5.1 Manual signature	387
D.5.2 Enciphered PIN verified on-line	387
D.5.3 Plaintext PIN verification performed by the chip card	388
D.5.4 Symmetric enciphered PIN verification	389
D.5.5 Asymmetric enciphered PIN verification	390
D.5.6 Cardholder verification based on biometrics	391
D.6 SDA mechanisms	392
D.6.1 MAC-based SDA mechanism	392
D.6.2 Signature-based SDA mechanism	393
D.7 DDA mechanisms	394
D.7.1 MAC-based DDA	394
D.7.2 Digital signature-based DDA	395
D.7.3 One-time passwords	396
References	397

Appendix E: Block Ciphers	399
E.1 Definition and parameters	399
E.2 Modes of operation	400
E.3 DES, Triple-DES, and AES	402
E.4 MAC using a 64 bit-length block cipher	404
E.5 Key derivation	405
References	406
 Appendix F: RSA Encryption and Signature Scheme	 407
F.1 Key generation	407
F.2 Public and secret RSA operations	409
F.3 Digital signature giving message recovery	410
<i>F.3.1 Signature generation</i>	411
<i>F.3.2 Signature verification</i>	412
F.4 Digital signature and encryption with PKCS#1	414
References	416
 Appendix G: E-Commerce and M-Commerce Related Technologies	 419
G.1 E-commerce and m-commerce	419
G.2 SIM, STK, SMS, and WAP	420
G.3 Access devices for remote card payments	421
G.4 WAP protocol suite compared to Internet	426
References	427
 About the Author	 429
 Index	 431

CHAPTER

2

Contents

- 2.1 Payment card processing at a glance
- 2.2 Roles involved in payment card processing
- 2.3 Payment card brands
- 2.4 Credit and debit payment cards
- 2.5 Focusing on the magnetic stripe card
- 2.6 Threats and security protections
- 2.7 Processing at the point of service
- 2.8 Payment network and interchange messages
- 2.9 On-line authorization
- 2.10 Clearing and settlement

Payment Card Processing

The last three decades witnessed major advancements in payment technologies. On one hand, the payment infrastructure was created. This infrastructure consists of highly reliable computer networks connecting automatic teller machines (ATMs) and POS terminals with powerful mainframe computers of financial institutions. On the other hand, new carriers of the customer's financial data emerged. The most common are the magnetic stripe cards and ICCs, also known as chip cards. These developments provided the framework for the electronic processing of payment transactions.

This chapter considers only the electronic processing of credit and debit payment cards implemented with magnetic stripes. For those impatient to discover the overall picture, Section 2.1 gives a quick, broad view on payment card processing. This landscape reveals roles of the acquirer, the payment system operator, and the issuer, and Section 2.2 highlights their attributions in the payment card processing. Many actors can play the role of a payment system operator. In many countries, national bank associations have created dedicated companies to run a national payment network. The most reputed actors, however, are the multinational card associations like Visa, MasterCard, American Express, Europay, Diners Club, Discover, and JBC, which provide global coverage of their payment card processing and have boosted the payment card to the status that it enjoys today. Section 2.3 explains the concepts of card association and brand, which are at the core of the card business. Section 2.4 explains the main features of credit and debit cards, highlighting the differences between these two

payment card products. Starting with the Section 2.5, we move quickly to more detailed functionality. First, we discuss the content of the financial data that is embossed or printed on a payment card and stored on its magnetic stripe. The possibilities of reading this information at the point of service are also outlined. Section 2.6 describes potential attacks on payment cards, as well as the corresponding security protections available for this type of card. This security analysis highlights the limitations of the magnetic stripe cards, revealing the need for migrating to ICC implementations. Section 2.7 focuses on the processing performed by the terminal. The rest of the chapter deals with what is called the payment system backbone. Its functionality is described in terms of payment messages exchanged between the acquirer, the payment system operator, and the issuer to complete various transactions. The concepts of authorization, clearing, and settlement are explained; these are basic processes carried out by the payment system. It is important to stress that the processing in the payment system backbone does not change drastically for the credit/debit payment transactions carried out with ICC.

2.1 Payment card processing at a glance

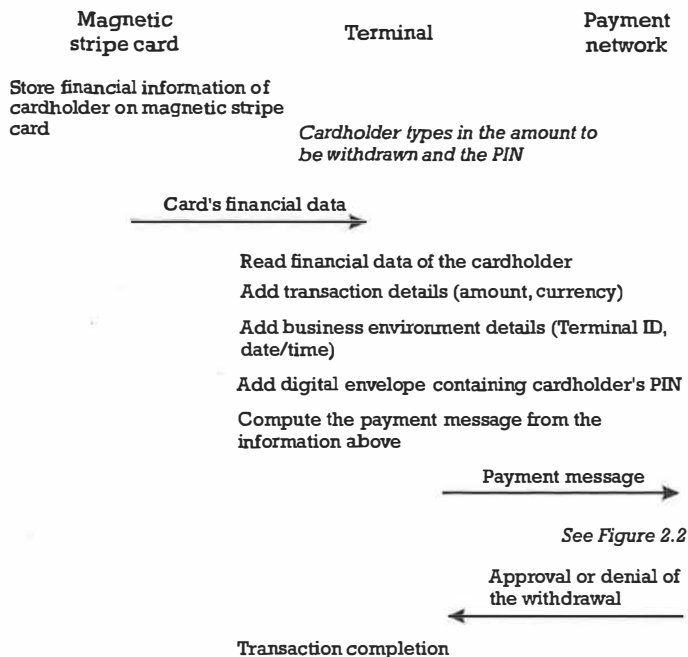
We address this section to those impatient to have at a glance the overall picture of payment card processing. We have chosen for this purpose the cash withdrawal transaction at an ATM terminal. As with any quick, general view, the reader can benefit from a fast orientation to the topic of payment card processing, but not all the details will be visible.

The overall picture is split into two parts: Figure 2.1 shows the things people usually see when using a debit card for withdrawing money at an ATM; Figure 2.2 reveals the unseen part of the payment card processing, which is sometimes referred to as the network and back-office processing. The financial institutions, including banks, payment system operators, and card associations, are involved in the completion of this part of the processing for each transaction carried out with a payment card.

On the backside of the debit card used for performing the money withdrawal, there is a magnetic stripe. It contains the financial data related to the cardholder in connection with an account kept with a bank. While passing this card through the magnetic stripe reader of the ATM, the financial data is read by the terminal and stored in its random access memory (RAM).

The cardholder is prompted to type in his or her personal identification number (PIN), which establishes the link between the user at the point of service and the legitimate cardholder. The cardholder is also prompted to

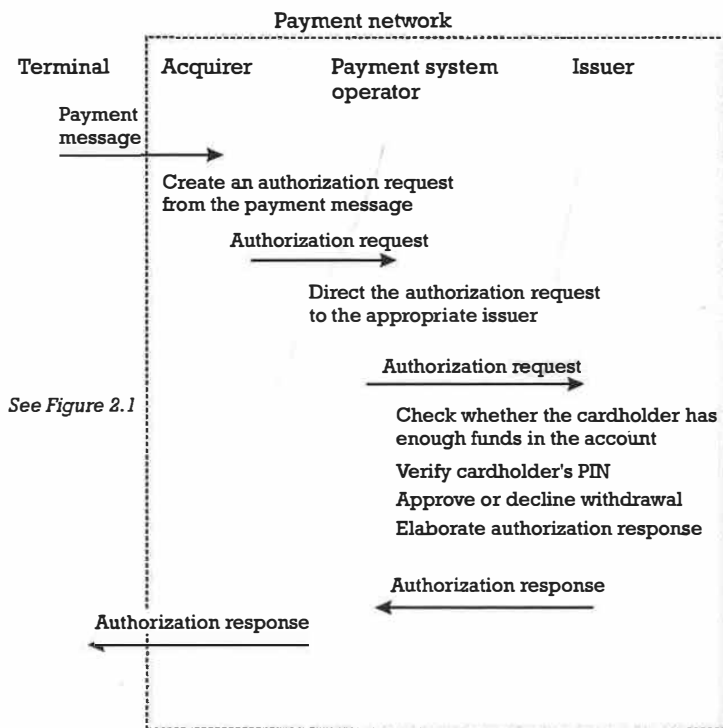
Figure 2.1
Payment card
processing—
things one can
see.



type in the amount of cash to withdraw. After capturing this information from the cardholder, the terminal constructs the payment message. It contains the amount of the cash withdrawal combined with other information about the business environment at the point of service. This information includes but is not limited to the currency in which the amount is expressed, the identifier of the terminal, the date and time when the transaction took place, and the current serial number of the transaction performed by the terminal. The payment message also includes a cryptogram of the cardholder's PIN, which is the result of an encryption algorithm applied to the PIN.

At this point the processing performed by the ATM terminal ends, and the long journey of the payment message through the payment network begins. Figure 2.2 schematizes this journey to an oversimplified circuit composed of the acquirer, the payment system operator, and the issuer. It is important to notice that not all transactions at the point of service are directed on-line to the issuer for authorization. It could be that, after capturing the whole information from the card, the terminal has enough information for concluding the transaction off-line. In the case of an ATM money withdrawal, however, the issuer is usually involved on-line in the

Figure 2.2
Network and
back-office
processing of
payment card
transactions.



transaction for a better risk management. We will assume this situation for the remainder of this section.

The ATM terminal at the point of service forwards the payment message to the acquirer, which further submits the received message to an intricate electronic processing system. Several cooperating parties (the acquirer, the payment system operator, and the issuer of the payment card) participate in completing this processing. Payment messages are exchanged in real time or compiled in batch files following clearly established protocols. Each payment message conveys certain data elements depending on the scope of the processing. While the message is sent from one party to another, each party performs a predetermined set of transformations on the data elements contained in the message according to their business role in the payment system. The following steps roughly describe the processing.

The acquirer creates the authorization request. To this end, the acquirer adds to the payment message received from the terminal some data elements kept in its terminals database. These data elements include the location of the terminal involved in the withdrawal transaction, the type of

terminal at the point of service and its capabilities, and the identification information of the acquirer node that creates the authorization request message. In addition, the acquirer translates the initial PIN cryptogram received from the terminal into a second cryptogram that can be deciphered by the payment system operator's secure module. The acquirer attaches the translated cryptogram to the authorization message.

The acquirer transmits the authorization message to the payment system operator's node in the payment network to which its acquirer host is connected. If the ATM withdrawal occurs in a foreign country, after receiving the authorization message, the payment system operator adds to it the actual exchange rate between the currency of the amount requested at the point of service and the currency used in the home country by the cardholder's issuer. Afterwards, the authorization message is directed to the destination node in the payment system operator's network, which is connected to the issuer host. It could be that the payment system operator performs a third conversion of the cardholder's PIN cryptogram using cryptographic parameters that are accessible to the secure module attached to the issuer host.

The issuer host is the final destination point for the authorization message. Based on the financial data in the authorization message, the account of the cardholder is identified. The cryptogram containing the PIN of the cardholder is deciphered in the secure module of the issuer host. This value is used to compute the PIN image control value, which is compared to a similar witness value stored in the cardholders' database since the personalization of the card. If the two values are equal, the issuer accepts the link between the user of the card at the point of service and the legitimate cardholder. Then, the issuer converts the cash amount in its own currency, using the currency exchange rate indicated by the payment system operator. If after deducting the amount requested in the transaction, the balance of the cardholder account is still higher than a floor limit (which could be either a negative value or zero), the issuer accepts the cash withdrawal transaction. Through an authorization response, the issuer host informs the acquirer about the approval or denial of the transaction. The acquirer instructs the ATM terminal at the point of service either to provide the required cash or to decline the transaction.

2.2 Roles involved in payment card processing

Several parties, also referred to as roles, are involved in the processing of a payment card transaction—namely, the issuer, the cardholder, the

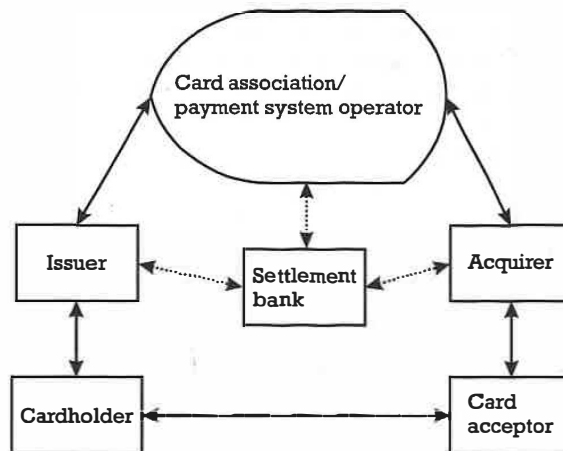
merchant, the acquirer, the card association, and the settlement bank. An overview of the payment system, including all these roles, is presented in Figure 2.3.

The issuer is a licensed financial institution or its agent that issues the payment card to the cardholder and is responsible for the provision of responses to authorization requests. The issuer remains unchanged during a transaction. The financial institution can be a bank, also referred to as the issuing bank, that is member of a card association and adopts a payment card product promoted by the card association. The issuer keeps the accounts of cardholders to which they charge bills or directly debit the amounts involved in the financial transactions performed by the cardholders. The issuer guarantees payment for authorized transactions, processing the payment card in accordance with the payment card product regulations and local legislation. The issuer supports the clearing and settlement functions between the cardholder and the acquirer. The issuer host is the computing system that accesses the cardholder accounts database and represents the issuer during the authorization, clearing, and settlement.

The cardholder is a customer of the issuer that uses a payment card in a B2C payment transaction. The cardholder is associated with the primary account number (PAN). The PAN gives enough information for identifying the account of the cardholder to which bills are charged or from which amounts are directly debited and the financial institution that keeps this account.

The card acceptor is the party that accepts a payment card at the point of service, formats the data of the transaction in a payment message, and forwards the payment message to the acquirer. The card acceptor can be a

Figure 2.3
Roles involved in
payment card
processing.



merchant selling goods or services or a branch bank disbursing cash for its customers. A supermarket selling groceries, a vending machine for beverages, a travel agent, or a small café can play the role of the merchant.

The acquirer is a licensed financial institution or its agent that acquires the payment message relating to a transaction from the card acceptor and feeds that data into an interchange system. The acquirer remains unchanged during a transaction. The financial institution can be a bank, also referred to as the acquiring bank, that is a member of a card association and adopts the payment card product designed and operated by that card association. From this perspective, the acquirer intermediates between the card acceptor and the card association, providing the card acceptor with operational support and needed infrastructure for card authorization, clearing, and settlement through the interchange system. The acquirer keeps the accounts of merchants, to whom they credit the amounts involved in the financial transactions performed at their point of service. The acquirer host is the computing system that accesses the merchant accounts database and represents the acquirer during the authorization, clearing, and settlement processes.

The card association is the owner of the payment card product. The card association's members are issuers and acquirers, providing the direct retail financial service for cardholders and merchants. The card association is responsible for running the interchange system that exchanges transaction data between acquirers and issuers during authorization, clearing, and settlement. The interchange system is implemented as a payment network with processor nodes connected to issuer hosts and acquirer hosts.

The settlement institution is a financial institution that holds the accounts of issuers and acquirers that are members of the card association. The settlement institution can be a bank that is subcontracted by the card association. This settlement bank acts on the transaction information provided by the issuers, acquirers, and the card association by transferring the appropriate funds between their accounts.

2.3 Payment card brands

A payment card is a payment instrument that allows bills in payment transactions to be charged to a designated account, which is linked to the card.

A card association or a payment system operator plays the central role in the card business. They have founded payment card brands to promote their payment card products, to establish and enforce rules for their use and acceptance, and to provide networks to connect end-to-end issuing and acquiring financial institutions. The brand signifies acceptance of a payment

the CCD (if using track 3) can be verified locally for assessing the card authenticity if the terminal at the point of service has the necessary cryptographic material to perform the MAC verification algorithm.

If the card encodes the amount remaining this cycle field on track 3, the risk management at the point of service is improved with a rudimentary card risk management component intended for protecting the issuer against cardholders that overspend. The transaction is authorized off-line if the transaction amount is smaller than both the upper floor limit (acquirer's requirement) and the amount remaining this cycle parameter, which specifies the spending limitation as imposed by the issuer.

The payment messages describing the transactions authorized off-line are gathered in the permanent memory of the terminal at the point of service. Periodically, or when the capacity of the permanent memory is exhausted, the payment messages are compiled in batch files, which are forwarded to the acquirer. Periodically, the acquirer transmits these batch files in the interchange system for clearing.

2.8 Payment network and interchange messages

Terminals located at various points of service are connected to the acquirer host via the acquirer's network. The formats of (1) the payment messages generated by the terminal at the point of service and forwarded to the acquirer host and (2) the confirmation messages returned by the acquirer host to the terminal are proprietary to the acquirer's network.

In the simplest scenario, the acquirer managing the terminal at the point of service and the issuer of the card involved in the payment transaction are subscribers to the services offered by the same payment system network. This network is the property of a card association or a payment system operator that is responsible for managing the network and for settling funds between the issuer and the acquirer following a payment transaction. Each acquirer host (AH) and issuer host (IH) is connected to separate nodes of the payment network, referred to as the acquirer node (AN) and issuer node (IN), respectively. In order to increase the availability of the issuer's service, an issuer can duplicate the functionality of an IH through a second computer acting as an active reserve. The payment system operator can provide a stand-in processing facility to an issuer, through which the payment system operator can answer to an authorization request on behalf of the issuer in case none of its hosts is available in the authorization process.

In a more complex scenario, the acquirer managing the terminal at the point of service and the issuer of the card involved in the payment

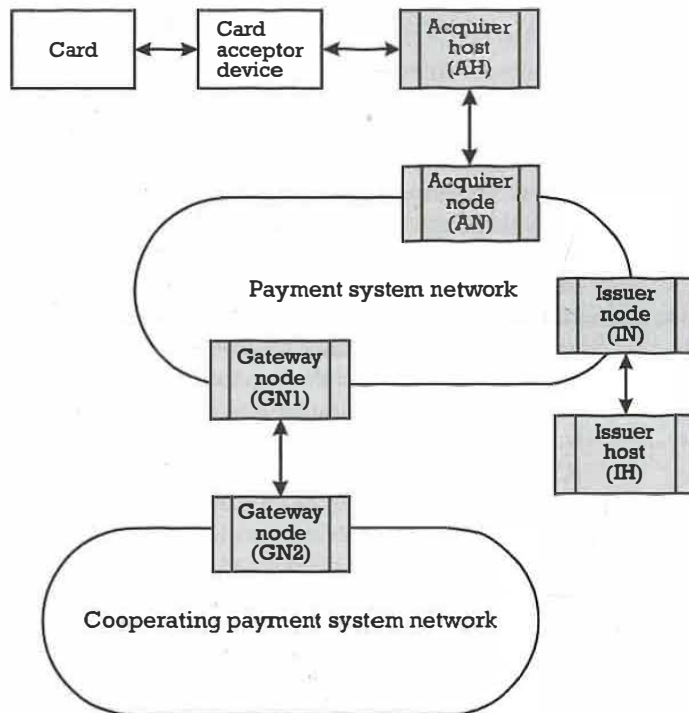
transaction are subscribers to the services offered by different payment system networks, which have established mutual supporting agreements. In order to guarantee the compatibility between these two different networks, two gateway nodes, GN1 and GN2 (one on each payment network), must provide the message translation between the two heterogeneous environments. Different rules concerning the definition of the transaction messages and their flow could govern the two interconnected payment networks. A possible topology of a payment network is schematized in Figure 2.4.

In this topology a national payment system operator transacting in one single currency could manage the payment system network, while an international card association transacting in multiple currencies could manage the cooperating payment system network.

2.8.1 Message structure

The acquirers and issuers must exchange messages towards completing authorization, clearing, and settlement processing. Both the acquirer and

Figure 2.4
Payment
network
topology.



the issuer can play the role of a sender or receiver of a message. In order to facilitate the interconnection between payment system networks that cooperate, the ISO/IEC 8583:1993 standard [17] defines the format of these messages, which have the following structure:

- Message type identifier;
- Bitmap representation, which consists of one or two bitmaps of 64 bits each, giving the index of the data elements that are incorporated in the message;
- A series of data elements, in the order specified in the bitmap representation, which represents the message's body.

The standard defines the dictionary of data elements that can be used in interchange. When a bit is set in the message's bitmap representation, the corresponding data element is included in the message's body. Some data elements are of fixed length. Some other data elements are of variable length, which is specified in a fixed length prefix. The standard, however, does not preclude the use of additional data elements not specified in the dictionary, which could be required by the specific needs of payments system operators for private use.

The message type identifier is a numeric field consisting of four digits. The first digit identifies the message version number as follows:

- 0: messages defined in the previous issue of the standard referred to as the ISO 8583:1987;
- 1: messages defined in the current issue of the standard referred to as the ISO 8583:1993;
- 2–7: reserved for ISO use;
- 8: reserved for national use;
- 9: reserved for private use.

The second digit encodes the message class as follows:

- 0—ISO reserved: Messages in this class are reserved for ISO use.
- 1—authorization: Messages in this class are used for an authorization transaction, which is an approval or guarantee of funds given by the card issuer to the acquirer. Following the authorization, the transaction

amount that is approved by the issuer is not immediately billed to the cardholder's account. This is postponed until a financial message, which is sent by the acquirer to the issuer, confirms the completion of that transaction at the point of service, following the authorization.

- 2—financial: Messages in this class perform a financial transaction that directly bills the transaction amount to the cardholder's account.
- 3—file action messages: Messages in this class allow the initiation of a remote control transaction over the file system hosted in a device, like adding, changing, deleting, and replacing a record in a file or adding/deleting a file in the file system. These messages are used to perform the management of blacklists, which contain those cards that were compromised (stolen, abused through overspending, cards with compromised cryptographic material, etc.).
- 4—reversal/chargeback: Reversal transactions are used to undo the effect of a previous authorization or financial transaction. The acquirer triggers a reversal transaction any time the result of an authorization or financial transaction is not received from the issuer in due time, or when the cardholder voluntarily cancels the authorization or financial transaction at the point of service. Chargeback transactions also undo the effect of a previous authorization or financial transaction, but at the initiative of the issuer. The reasons that the issuer can invoke for performing a chargeback transaction include a customer dispute, an infringement of rules concerning the use of a certain type of card product on a specific terminal, the use of an expired card, or an invalid transaction.
- The messages in the classes 5—reconciliation, 6—administrative, 7—fee collection, and 8—network management are not explained in this book, but the interested reader can find their definition in the ISO 8583:1993 standard.

The third digit of the message identifier specifies the context in which the message is used. Three different situations are identified:

- The sender addresses a request message (third digit = 0) to inform the receiver that a transaction is in progress and its response is required to complete it. The receiver evaluates the request and approves or denies it, transmitting back to the sender its decision in a request response message (third digit = 1).

- The sender informs the receiver with an advice message (third digit = 2) about a certain activity that has been completed at the point of service. The receiver is not required to approve or deny the advice, but it has to elaborate an advice response message (third digit = 3), which is sent back to the sender.
- The sender informs the receiver with a notification message (third digit = 4) about a certain activity that has been performed. The notification message requires no response back from the receiver to the sender.

The fourth digit identifies the originator of a transaction and whether the current transaction is a repeat of a previous transaction.

- 0: The acquirer is the transaction originator.
- 1: The acquirer is the originator of the repeated transaction.
- 2: The issuer is the transaction originator.
- 3: The issuer is the originator of the repeated transaction.
- 4: The other role is the transaction originator.
- 5: The other role is the originator of the repeated transaction.

It is important to mention that there are some interdependencies between the last three digits of the message type identifier; for example, a reversal transaction shall use only advice messages (1420/1431 and 1430/1431) or notification messages (1440/1441).

2.8.2 Message flows

The standard also specifies the possible message flows that describe the circumstances when a message shall (or may) be sent, and the relationship between messages. In the remainder of this section we focus on several message flow examples that correspond to typical situations that appear during transaction processing. The message flows depicted in the figures below do not represent the AN, the IN, and the payment system network, but just the acquirer and the issuer.

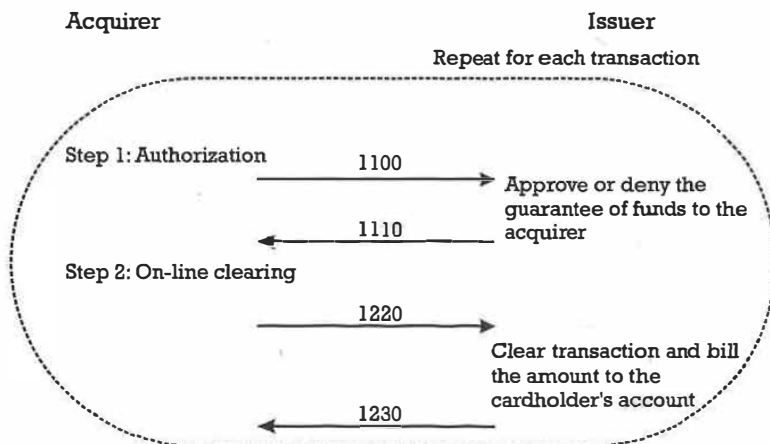
When a terminal is connected on-line to the acquirer and the amount involved in the transaction is greater than a risk threshold limit, the terminal triggers an authorization transaction. After receiving the appropriate

transaction data from the terminal (see Section 2.7), the acquirer performs an authorization phase. If the authorization does not impact the cardholder's account, a subsequent clearing stage follows.

In a dual message network, the authorization phase is performed with an authorization request message (1100). Following the evaluation of this request by IH, the guarantee of funds is approved or denied by the issuer according to the financial situation of the cardholder. The acquirer is informed about the appropriate action through an authorization request response message (1110). Following the authorization, the transaction amount that is approved by the issuer is not immediately billed to the cardholder's account. This is postponed until a separate financial message, which is sent by the acquirer to the issuer, confirms the completion of that transaction at the point of service, following the authorization. This financial message performs the clearing stage. One can distinguish between two possible approaches:

- *On-line clearing:* Following the reception of each authorization request response message (1110), the acquirer forms a financial advice message (1220), which bills the transaction amount to the cardholder's account. The issuer informs the acquirer about the result of this operation with a financial advice response message (1230). An overview of the on-line clearing in dual message networks is presented in Figure 2.5.
- *Off-line clearing:* Following the reception of each authorization request response message (1110), the acquirer forms a financial notification

Figure 2.5
On-line
transaction in
a dual message
network with
on-line
clearing.



message (1240), which is stored in a clearing batch file. Periodically, this file is forwarded to the issuer, which bills each and every transaction amount of each financial notification message to the appropriate cardholder account. An overview of the off-line clearing in dual message networks for transactions approved on-line is presented in Figure 2.6.

In a single message network, the authorization phase and the clearing phase are simultaneously performed with a financial request message (1200). Following the evaluation of this request by the IH, the guarantee of funds is approved or denied by the issuer according to the financial situation of the cardholder. The acquirer is informed about the appropriate action through a financial request response message (1210). Following the financial authorization, the transaction amount that is approved by the issuer is immediately billed to the cardholder's account. An overview of the financial authorization is given in Figure 2.7.

When a terminal has no on-line connection or the amount involved in the transaction is below a risk threshold limit—which is accepted by both the issuer and the acquirer—the authorization phase can be completed locally between the card and the terminal. In this case the terminal initiates no authorization transaction, and consequently, the acquirer generates

Figure 2.6
On-line
transaction in
a dual message
network with
off-line
clearing.

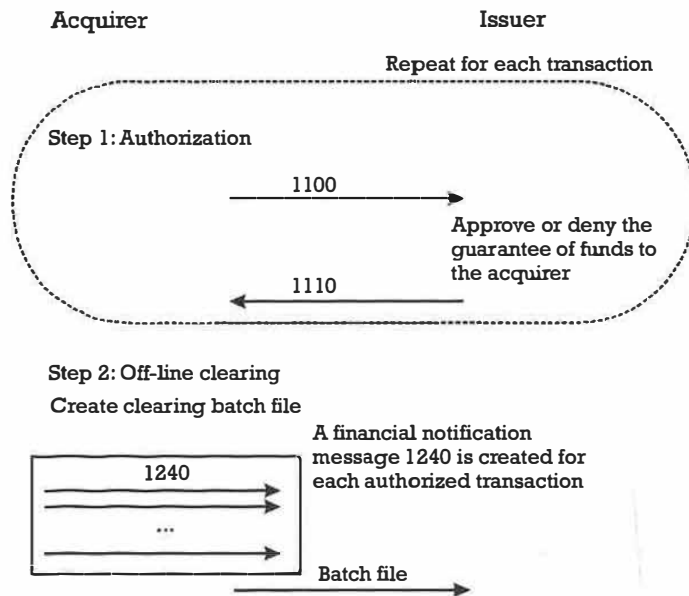
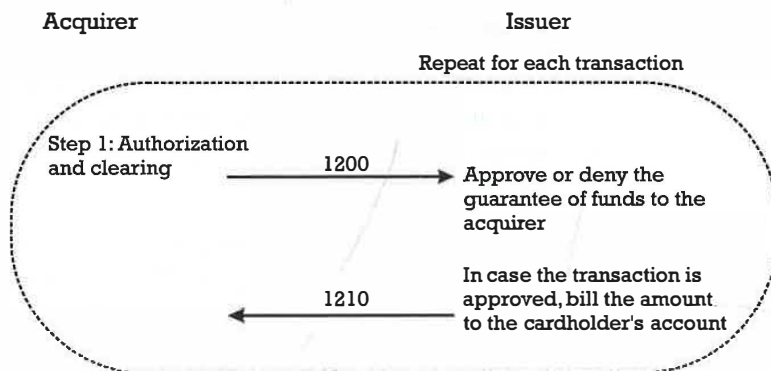


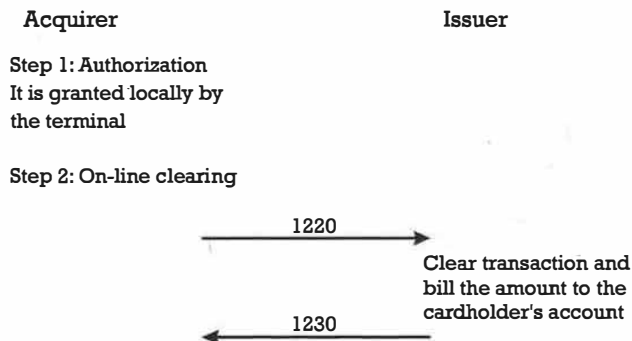
Figure 2.7
On-line
transaction in a
single message
network.



neither an authorization request message (1100) nor a financial request message (1200) for the issuer. After the terminal reports all the transactions it performed off-line during a certain period, the acquirer informs the issuer about the local completion of these transactions at the point of service, during a clearing stage. Two possibilities can be envisaged, depending on the features supported by the payment network:

- A financial advice message (1220) is issued for each transaction that is completed off-line, which allows an on-line clearing of the transactions. Each financial advice response message (1230) informs the acquirer whether the issuer accepted the transfer of liability corresponding to a transaction completed off-line. The corresponding amount involved in the locally authorized transaction is billed to the appropriate cardholder account. This is characteristic of a single message network. The corresponding message flow is schematized in Figure 2.8.

Figure 2.8
Off-line
transaction in a
single message
network with
on-line clearing.

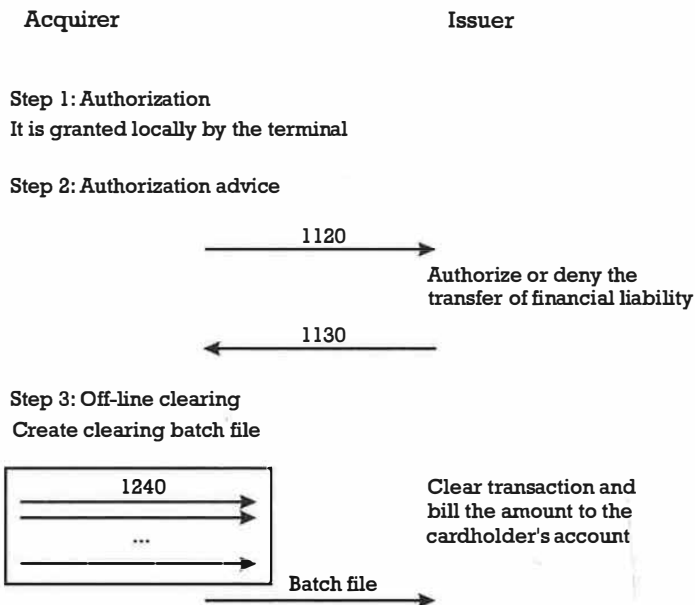


- In a dual message network, the acquirer firstly informs the issuer with an authorization advice message (1120) about the authorization transaction that was completed off-line at the point of service. In the authorization advice response message (1130), the issuer informs the acquirer whether it accepts or rejects the proposed transfer of financial liability. The acquirer forms a batch file, where it generates a financial notification message (1240) for each transaction accepted by the issuer. During an off-line clearing stage the batch file is forwarded to the issuer, which bills the cardholders involved. The corresponding message flow is schematized in Figure 2.9.

2.9 On-line authorization

On-line authorization is the process through which the IH performs the risk management associated with a transaction carried out at the point of service. On-line authorization provides a better control of the risk for the issuer. The cardholder is better assured against fraudulent use of his or her card. Indeed, in an authorization request message the PIN of the cardholder is included in an encrypted form, which allows its verification by the issuer.

Figure 2.9
Off-line
transaction in a
dual message
network with
off-line clearing.



The benefit to the merchant is that the payment system guarantees its payment for each on-line authorized transaction.

On-line authorization can be triggered for the types of transactions listed below:

- Cash withdrawals at ATM terminals or branch banks, which are performed with either debit or credit cards;
- Payments performed at POS terminals with debit cards;
- Payments performed at POS terminals with credit cards, in case the transaction amount is greater than a floor limit established by the payment system.

In the explanation below we refer to a payment network topology like that described in Figure 2.4. After receiving the demand of on-line authorization from the terminal, the AH checks the completeness and correctness of the transaction data received from the terminal. The AH verifies whether the brand of the card product involved in the transaction is accepted, and whether the service required is appropriate for that brand. If all the verifications are successfully passed, the AH forms an authorization request message (1100) according to the specifications of the payment system network to which is directly connected. This message also includes the identifier of the acquirer. The AH forwards the 1100 message to the AN, which represents the entry point of the AH in the payment system network. The AN verifies whether the 1100 message comes from a member acquirer and whether the data it contains is complete and accurate. If these verifications are passed, the AN uses the issuer identification number (IIN, which is a component of the PAN) for further routing the 1100 message to the adequate destination IN in the payment system network. If the AN knows the IIN, the message is forwarded to the appropriate IN to which the destination IH is connected. Otherwise, the AN forwards the message to the gateway node GN1, which tries to identify the IIN in the cooperating payment system network to which is connected. After performing the appropriate adaptations, the message is forwarded from one payment network to the other, until the destination IH is reached.

The correctness, completeness, and integrity of the 1100 message are verified once it arrives at the IH. If all the controls are passed, the authorization request message is stored in the accounts database. The validity of the PAN and the expiration date of the transacting card are checked. If an encrypted PIN was sent for the cardholder's verification, then the secure module of the IH computes the PIN image control value, which is compared

with the PIN image stored value kept in the accounts database for the corresponding PAN. If a debit card is used, the balance of the account is checked for enough funds. If a credit card is used, the issuer checks that the cardholder did not reach the spending limit associated with the card. The guarantee of funds is finally approved or rejected and a response code is included in the authorization request response (1110), giving further details in case the guarantee of funds has been denied.

If the payment network fails to reach the appropriate IN or if the destination IH is not available, then the payment network can stand in for the issuer in elaborating an authorization request response (1110), which is returned to the originating AH. This service can be provided if there is a business agreement between the issuer and the payment system operator and if the issuer has delegated enough approval data to the payment network.

The authorization request response message (1110) is identified with the same reference number as the corresponding authorization request message (1100). The 1110 message is sent back to the terminal over the payment network(s) and the AH. When the card acceptor's terminal receives the approval, cash is disbursed in case of an ATM terminal or the purchase is handed to the cardholder. In case of an attended point of service and if a credit card product was used, a paper slip is printed and the cardholder is required to sign it. The signature on the paper slip is compared against the witness signature on the back of the card for the cardholder's verification. The terminal records the completion of the transaction in order to send it later to the acquirer for clearing. The acquirer records each approved transaction for submitting to the clearing process.

2.10 Clearing and settlement

Clearing is the process initiated by an acquirer through which issuers are informed about the payment transactions performed by their cardholders. These transactions were completed at points of service managed by card acceptors that are the clients of the acquirer. Following the clearing process, the issuer can bill the transaction amounts to the accounts of the appropriate cardholders and can settle its debt with each of the acquirers that accepted its payment cards. Settlement is the process through which funds are exchanged between the members of the payment system(s) for the value of the transaction amounts cleared during a certain period [17].

The payment network topology in Figure 2.4 shows the clearing and settlement process. We assume that a national operator, handling transactions

only in the national currency, manages the upper payment system network, while an international card association, able to handle multiple currencies, manages the cooperating payment system network. We also assume for simplicity that both networks support the off-line clearing with dual messages (see Section 2.8.2).

The transaction records stored at the points of service by either POS or ATM terminals are transmitted electronically to the AH. After checking and validating the received transaction records from one terminal, the transaction records database of the card acceptor kept with the AH is appropriately updated. The AH forms a financial notification message (1240) from each valid transaction record. Then it creates a separate clearing batch file with all the 1240 messages having the same IIN.

- If a clearing batch file gathers transactions, the IIN of which corresponds to an issuer located in the same country as the acquirer and subscribing to the same national payment system network, then the acquirer directly transmits the clearing batch file to the appropriate issuer.
- If a clearing batch file gathers transactions performed with cards issued by members from other countries than the acquirer but subscribing to the same international payment system network with which the national payment system network cooperates, this file is transmitted over the gateway nodes (GN1, GN2) to the card association. The currencies in which these transactions will be settled are different from the currency in which the transaction amount is expressed at the point of service.

The card association continues the clearing process, collecting the clearing batch files containing transactions in various currencies from the acquirer members. Clearing batch files received from different acquirers in different currencies are sorted according to the IIN of the card, and a separate IIN batch file is compiled for each issuer. Each IIN batch file contains transactions performed with one type of card product, while the transaction amounts are expressed in various currencies. These currencies are different from the billing currency, which is the currency in which the billing of the cardholder account is performed.

The issuer can receive IIN batch files compiled by both card associations and local acquirers. Using the unique transaction reference number retrieved from each financial notification message (1240), the issuer matches each 1240 message received during clearing against the

authorization database, containing all the previously accepted or rejected authorization request messages (1100). All the matching records are separately compiled. For each record received from the card association a currency exchange operation is performed from the currency in which the amount at the point of service is expressed to the billing currency agreed by the issuer. The records are sorted according to the PAN corresponding to each cardholder, and a statement file is separately compiled for each cardholder. The issuer knows all the information it needs to settle payment with its customers. The issuer recovers the disbursed amount during the clearing period by billing its cardholders. The total in each statement file, corresponding to a cardholder, is computed. For debit cards, the resulting total amount is debited directly from the cardholder's account kept with the issuer. For credit cards, the total of the statement file is posted to a temporary account linked to the cardholder's account, updating its balance, which is held until the statement billing time. This completes the reimbursement of the issuer.

After the clearing of the transactions carried out in a well-defined time period is performed, *settlement* can be accomplished. The three organizations involved in the processing of a point of service transaction (namely, the card acceptor, the acquirer, and the card association) are reimbursed for their services. The card association computes the financial compensations between issuers and acquirers. The transfer of funds is performed through an international settlement bank, which is designated by the card association and which can handle various currencies. Both acquirers and issuers have an account with this bank.

The settlement process can be schematized as follows. The clearing batch file compiled by the card association containing the transaction data that was submitted for clearing to the issuer with transaction amounts in various currencies is submitted to a currency exchange procedure with currency exchange rates provided by the settlement bank. The amount in each transaction record of the batch file is converted from the card acceptor's currency, which is the local currency at the point of service, to a settlement currency agreed by each issuer. After all the transaction amounts are converted in the issuer's settlement currency, a total is computed that reflects the amount to be debited from the issuer's account kept with the settlement bank. The issuer is notified about the settlement of its debt. The card association repeats this process for each issuer that participated in the clearing.

The card association processes the transaction data file received from each acquirer and computes the total of the transaction amounts of all the records in the file. The resulting value is credited in the acquirer's account kept with the settlement bank. The acquirer is notified about the crediting of

its account. The card association repeats this process for each acquirer that participated in the clearing. Each acquirer credits the amount in each card acceptor master file to the corresponding account, which finishes the settlement process.

References

- [1] Visa, "Products & Services," <http://www.visa.com>.
- [2] MasterCard, "Our Cards," <http://www.vis.com>.
- [3] American Express, "Cards," <http://www.americanexpress.com>.
- [4] Europay International, "Brands," <http://www.europay.com>.
- [5] Diners Club, "Credit Cards," <http://www.dinersclub.com>.
- [6] Discover Financial Services, <http://www.discovercard.com>.
- [7] Japanese Business Cards, <http://www.japanese-business-cards.com>.
- [8] Federal Trade Commission, "Secured Credit Card Marketing Scams," <http://www.ftc.gov>, October 1996.
- [9] Brobeck, S., "Recent Trends in Bank Credit Card Marketing and Indebtedness," Consumer Federation of America, <http://www.abiworld.org>, July 1998.
- [10] ISO/IEC 7811, "Identification Cards—Recording technique—Part 1: Embossing," "Part 2: Magnetic Stripe," "Part 3: Location of Embossed Characters on ID-1 Card," "Part 4: Location of Read-Only Magnetic Tracks—Track 1 and 2," "Part 5: Location of Read-Write Magnetic Track—Track 3," 1985.
- [11] ISO/IEC 7812, "Identification Cards—Numbering System and Registration Procedure for Issuer Identifiers," 1987.
- [12] ISO/IEC 4909, "Bank Cards—Magnetic Stripe Data Content for Track 3," 1987.
- [13] ISO/IEC 7813, "Identification Cards—Financial Transaction Cards," 1987.
- [14] Stern, C., "Micro-Thief That 'Steals' Credit Cards," *Sunday Mirror Magazine*, January 28, 2001.
- [15] Freier, A. O., P. Karlton, and P. C. Kocher, *The SSL 3.0 Protocol*, Internet-Draft, November 1996.
- [16] Dierks, T., and C. Allen, *The TLS Protocol—Version 1.0*, Internet-Draft, November 1997.
- [17] ISO/IEC 8583:1993, "Financial Transaction Card Originated Messages—Interchange Message Specifications," 1995.

CHAPTER**3****Contents**

- 3.1 A business case for chip migration
- 3.2 An overview of the chip card technology
- 3.3 Proprietary payment application
- 3.4 Interoperable payment application

Chip Migration

Payment cards for debit and credit have proven to be a huge business success for the retail financial industry. The magnetic stripe technology is cheap enough to make the cost of cards small. Moreover, the payment network and the terminals at the point of service have been in place for years now. Therefore, there is no further need to invest in infrastructure. Meanwhile, the operation of debit and credit cards increases year after year, both in the number of issued cards and in the geographical coverage. Consequently, profit has increased, so there is no apparent cause for concern.

The first section of this chapter lists several causes of concern that should encourage payment system operators, issuers, and acquirers to consider the migration from magnetic stripe to chip. We believe that this motivation could help a chip solution vendor make his business case when talking to skeptics about switching from the magnetic stripe technology to chip.

The second section reminds the reader of the essentials of chip card technology. In this book the terms ICC and chip card are used interchangeably to refer to one and the same device, which not only stores data in its permanent memory but is also able to process data. Therefore, it would be more accurate to refer to these cards as microprocessor chip cards, to clearly distinguish them from the memory chip cards, which can store data but cannot process it. For the fluency of presentation, however, we will refer to the microprocessor chip cards as simply chip cards or ICCs. In Section 3.2.1 we give an overview of the hardware and software structure of a chip card, as well as the life cycle of the chip card. We then make a diagonal

minimum amount of knowledge someone would need to be able to understand the rest of this book. For an extensive introduction to chip card technology, the reader should refer to [1]. At the end of the section, we present the concepts of terminal application and card application, and their interactions in a client server model when performing a transaction (Section 3.2.4).

After these foundations of the ICC technology are revisited, two possible chip migration paths are outlined: closed proprietary payment applications and open interoperable payment applications. We analyze some of the features of a payment application that allows interoperability (Section 3.4). These features contrast with the homologue features of a proprietary application (identified in Section 3.3), and thus emphasize the price one pays for open design and interoperability.

3.1 A business case for chip migration

A cause of concern against keeping in place the magnetic stripe technology is the increase of abuses in magnetic stripe payment cards reported worldwide. Attackers have great insight about the design details of these cards, which helps them to identify security weaknesses that could lead to fraud.

In face-to-face payment transactions, counterfeiting the magnetic stripe has become a dangerous threat [2, 3]. This threat combined with sophisticated methods of monitoring the cardholder's PIN cause significant damages to financial institutions issuing such card products (see Section 2.6).

Card associations and payment system operators are concerned with decreasing the amount of fraud. In this context, the migration of actual payment card products from implementations using the magnetic stripe as a storage medium to a chip is seen as a necessary security improvement. The term "chip" designates the integrated circuit embedded in the plastic card. For the purpose of this book, we consider only those chips that offer protection against probing their resources. A chip providing this feature is referred to as tamper-resistant. The reduction of fraud becomes possible because of several factors:

- It is very hard to clone chip cards, particularly the secret cryptographic parameters they contain, unless the tamper resistance of the chip is overtaken. Even though more and more papers report methods of subverting the tamper resistance of chip cards, the attacks are far too complicated for common attackers to mount.
- Through its processing power, the chip card is actively involved in the risk management at the point of service. The chip card becomes a remote agent of the issuer that is able to correctly intervene in a local authorization process performed at a terminal that is not connected on-line to the payment network. The chip can enforce the proper policies for an optimal trade-off between the availability of the retail financial service provided to the cardholder and the security of the issuer against fraudulent transactions.
- The chip improves the process of determining counterfeit cards, through implementing the card authentication method with dynamic authentication mechanisms. It also provides greater protection of the cardholder against fraudulent transactions through the off-line verification of the PIN in the card, for transactions authorized off-line.

The cost of the chip migration is impressive. Integrated circuit cards are much more expensive to issue than magnetic stripe cards. This entails significant costs for the issuers. New terminals are needed at the point of service, which are equipped with integrated circuit card readers. This entails high costs for the acquirer. The host computers of issuers and acquirers as well as the payment network must be adapted for chip migration.

These economic factors have caused many financial institutions to question whether it is cheaper to continue to support the loss due to fraud or to change the whole infrastructure. This is mainly the case for financial institutions located in developed countries, where the existing payment infrastructure is huge. Moreover, their losses are kept reasonably low, considering that the majority of the transactions, if not all, are authorized on-line, which decreases the risk of fraudulent transactions. In developing countries with large territories, however, where the payment infrastructure is poor, the payment transaction is assessed off-line in the majority of situations. In these cases it makes sense to invest in a chip solution from the beginning, since the security protection is clearly better.

Card associations and/or payment system operators have adopted new operating rules for their chip card products, which has motivated issuers and acquirers to perform the chip migration. Thus, the policy of decreasing the interchange fees for acquirers that do not adapt their terminals to accept

acquirer/issuer that has performed the chip migration.

There is still another strong reason for chip migration. Instead of thinking in terms of reducing fraud, maybe it is better to think in terms of increasing revenue streams as a consequence of chip migration:

1. Because of better decision-making by the chip at the point of service, it is possible to improve authorization controls at a lower cost. This means that communication costs related to the on-line authorization of a transaction can be reduced in situations where the card risk management together with the terminal risk management decides that authorization can be granted locally. This improves the efficiency of debit/credit cards in a segment of payments, which were previously judged too small.
2. Since the chip has computation power, the payment card becomes "smart." Card applications can provide far more flexible financial services and better answer the rapid changes in the retail financial market. The same chip can accommodate several card applications, which provides the multiapplication dimension of the chip cards. This allows issuers to reduce the investment cost per card application and better combine several payment instruments that satisfy different payment behaviors. For example, the same chip card can accommodate a national debit scheme used for domestic payments, an international credit scheme suitable for relatively important payments made while travelling abroad, and a cross-border electronic purse for paying per byte for information on demand bought from Internet providers. Thus, the flexibility of customizing the financial service provided to each cardholder on an individual basis further strengthens the relationship between the cardholder and his or her bank.

3.2 An overview of the chip card technology

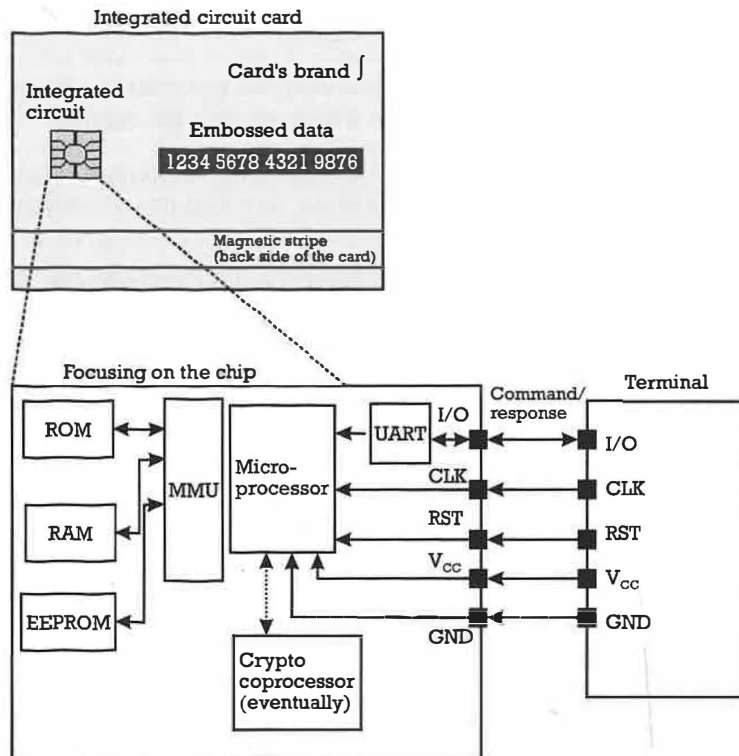
This section presents a quick overview of the chip card technology. The functionality of an ICC is based on the standard ISO/IEC 7816, "Identification

cards—Integrated circuit(s) card with contacts” [4–7]. First, we look at the hardware and software structure of a chip card. Second, we review the basics about a card file system and the methods of referencing files. Third, the format of the command/response pairs sent to and returned from the card as well as the most common commands are briefly presented. Finally, the concepts of terminal application, card application, and their interaction in a client server model are presented.

3.2.1 Hardware and software structure of chip cards

The chip card is a plastic card that incorporates an integrated circuit, which is a single-chip computer. This computer contains a microprocessor that can access read-only memory (ROM), electrically erasable programmable memory (EEPROM), and RAM. The memory management unit (MMU) controls the access to these memories. The hardware structure of the single-chip computer is shown in Figure 3.1.

Figure 3.1
Hardware structure of the single-chip computer in the card.



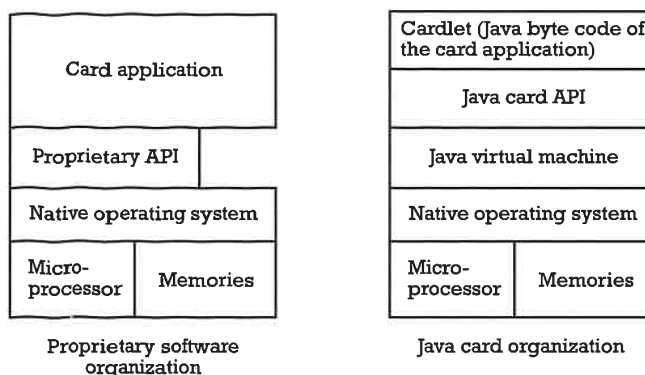
The chip is connected to the outside world through five contacts, which are assigned as follows:

- *I/O*: The chip has only one input/output serial line for communicating with the outside world. The universal asynchronous receiver transmitter (UART) serializes both commands coming from and responses going to the terminal. Among the protocols that can be implemented at the transmission level between the chip card and the terminal, we mention only two protocols known as $T = 0$ and $T = 1$ [4], since they are the only transmission protocols accepted by the EMV™ cards.
- V_{cc} and *GND*: The electrical power for the chip is provided by the terminal on these two contacts.
- *CLK*: The execution of all the processing in the chip is synchronized with a clock that is received from the terminal.
- *RST*: This contact receives the electrical reset signal from the terminal, which brings the chip to an initial status (see Section 3.2.4).

The single-chip computer is a slave depending on the terminal, which can be regarded as a master. The chip does not take initiative, but is simply driven by the terminal. Figure 3.2 shows two possible software architectures of a chip card, which can be regarded as a pile of software packages.

The left side of Figure 3.2 shows a proprietary software organization that does not allow for the portability of card applications. It can be seen that the card application makes direct calls to either a proprietary application programming interface (API) or to the card's operating system. Since each ICC producer has its own operating system and its own proprietary API, the card application is not portable from one chip card to another. Every time an issuer changes the chip card producer, the card application has to be rewritten. Regarding the mapping of hardware resources to the software architecture, it must be noted that a large diversity of possibilities exists, depending on the memory capacity allocated for ROM and EEPROM. The operating system and proprietary API are masked in the ROM, whereas the card

Figure 3.2
Two software
architectures
for chip cards.



application can reside either in the ROM or in the EEPROM. In the majority of the proprietary card implementations, however, the card application physically resides in the ROM and is logically integrated in the operating system instead of being on top of the operating system. The card file system that contains the data structures needed during the processing performed by the card application is always kept in the EEPROM, since both read and write operations must be available on the permanently stored data.

In the right side of Figure 3.2 the software organization of a Java card [8] is presented. In this software architecture the code of the card application is isolated from specific hardware and operating system libraries through the Java virtual machine (JVM). The JVM interprets the byte code corresponding to the Java source of the card application and translates it into instructions that are executable by the hardware and native operating system. Each chip hardware platform has its own JVM, which allows the card application to be independent of the hardware and the native operating system of the card. One of the big benefits of this platform, which can justify its higher price, is the reduced time to get new applications to the market. They also support the downloading of "cardlets," which is the term sometimes used for the applets downloaded to a chip card, even when the card is already in its utilization stage. Last but not least, the applications written for one chip card can be ported to other chip cards, provided they have the same Java card API, which is actually standardized as Java Card 2.1.1 [9]. Thus, this software organization guarantees the interoperability of card applications written for different chip card platforms.

The actual competitor of the Java card is the MULTOS operating system for chip cards, whose specifications are created by the MAOSCO consortium [10]. Card applications are coded in the MULTOS executable language (MEL), which is an interpreted language that is hardware-independent.

Therefore, similar to the Java card, the MULTOS architecture bases its functionality on a MEL interpreter, which can be regarded as a virtual machine, and an application loader. Generally there is a distinction between off-card and on-card virtual machines. In contrast to the Java card, the MULTOS virtual machine is completely realized on-card. This allows implementing firewalls between the applications, which provides a suitable security level for multiapplication environments. The application loader ensures the possibility of secure loading and deletion of card applications to and from the EEPROM, even during the utilization life stage of the card.

It is important to note that the software configuration and the file system loaded in the card are dependent on the life stage of the chip card. Table 3.1 presents each life stage of the card, along with the most important operations performed by a certain role in that stage.

3.2.2 Card file system and file referencing

The operating system of the chip card manages a file system that stores the data needed by each card application. ISO/IEC 7816-4 [5] supports two categories of files: dedicated files (DFs) and elementary files (EFs). They are organized in a hierarchical tree, with DF as branches and EF as leaves. A typical organization of the card's file system is schematized in Figure 3.3.

3.2.2.1 Master file and dedicated files

The highest DF in the hierarchy, which is the root of the tree, is also called the master file (MF), which is the only mandatory DF in the file organization. In the example presented in Figure 3.3, the MF contains one leaf, the elementary file EF1, and two branches, the dedicated files DF1 and DF2. Data that is used for all the applications in the card (e.g., administrative and general security information such as the ICC serial number, access control keys, card's general PIN, as well as data concerning the management of the card's life cycle) are stored in elementary files at the MF level. This information can be used by the operating system for creating another DF at the MF level.

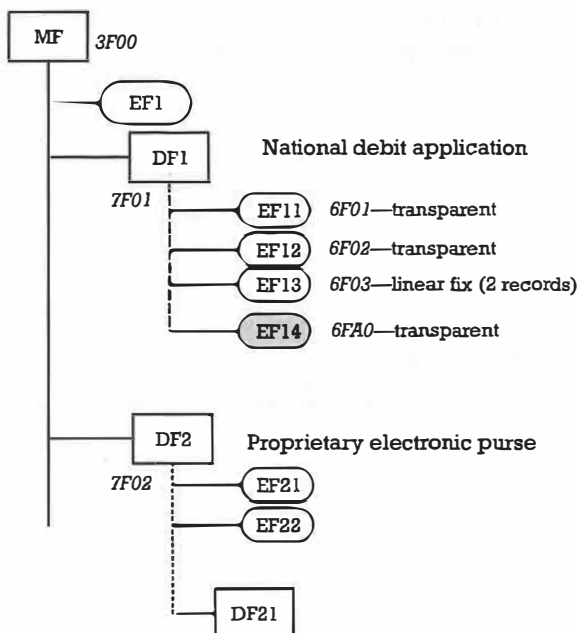
The dedicated file DF1 contains four leaves. The first three of them (EF11, EF12, and EF13) are working EFs, while EF14 is an internal EF. We will later see the difference between working and internal elementary files. The semantic of the information in DF1 and its elementary files will be explained in Section 3.3. The dedicated file DF2 contains only two leaves, which are the working elementary files EF21 and EF22. Each dedicated file can further contain other hierarchically inferior dedicated files. In Figure 3.3, DF2 contains one subdedicated file DF21.

Table 3.1
The Life Cycle of a Chip Card

Life Stage	Operation	Role Involved
IC fabrication	The integrated chip (IC) is produced, with the operating system in the ROM mask. For a proprietary card, this mask can contain the card application. For a Java card, the ROM contains the Java virtual machine. A unique ICC serial number is assigned to each chip.	IC manufacturer
ICC fabrication	The integrated chip is embedded in the plastic card.	Card manufacturer
Prepersonalization	The file system of the card is created. The data that is specific to the payment system and is common to all chip cards participating in the same scheme is also written during this stage. For a proprietary card, if the card application is resident in the EEPROM, the application software is loaded. For the Java card all the card applications that are foreseen in the standard configuration of the card are loaded.	Card manufacturer
Personalization	The data specific to each cardholder is filled in the appropriate files of the card.	Card issuer
Utilization	The card is operated according to the business goals of each application. For Java cards, card applications can be dynamically added in the EEPROM during the utilization stage. The only restriction is that the corresponding byte code originates from an application provider agreed upon by the card issuer and there is enough EEPROM space. Card applications can be also dynamically deleted from the EEPROM, according to the preferences of the cardholder.	Cardholder
End life	When the validity of the card expires, the card is disaffected by the card issuer, which can for example block the entire card.	Card issuer

A dedicated file can be seen as a container of data belonging to one card application. Several data elements of the card application that are semantically related are stored in the same elementary file. Application control information and cardholder's financial data are stored in the elementary

Figure 3.3
The organization
of the card's file
system.



files encompassed in the same DF. Each DF may contain cryptographic keys for implementing various security services, and each may have its own application PIN, which can be used to refine the access control mechanism of a multiapplication card.

The referencing of a DF in the card's file system, which corresponds to the possibility of selecting a card application from the terminal's side, can be performed in two distinct ways:

1. *Referencing with a fixed file identifier (FID), which consists of 2 bytes (4 hexadecimal digits).* For example, the MF always has the FID equal to 3F00, while DF1 has the FID equal to 7F01 and DF2 has the FID equal to 7F02, and so on. In order to be able to select a card application with its FID, the terminal application must know beforehand the file organization in that card. For example, in order to select DF21 starting from the MF level as the current directory, the terminal must first select DF2 with its FID, and only after this selection is successful can it select DF21 with its corresponding FID.
2. *Referencing with an application identifier (AID), which consists of up to 16 bytes.* The encoding of the AID is detailed in the ISO/IEC 7816-5 [6].

The AID comprises either the registered application provider identifier (RID), which optionally is concatenated with the proprietary application identifier extension (PIX), or the proprietary application identifier. The referencing of card applications with registered application provider identifiers has the advantage that the terminal does not have to know in advance the FID of the DF that stores the application or its position in the card's file system. Moreover, since the RID is unique worldwide, several applications can be stored in the card with no danger of referencing conflicts.

It will become obvious in Section 3.3 that referencing a DF with its FID is suitable for closed design proprietary card applications. In Section 3.4 we show that the open design interoperable card application uses DF referencing through the AID.

3.2.2.2 EFs

The data elements of a card application are encoded in elementary files. The elementary files of a card application can be further subdivided into working EF and internal EF:

- A working EF stores data that is not interpreted by the card application, but rather used by the terminal application exclusively during the execution of a protocol with the card.
- An internal EF stores data managed only by the card application for management and control purposes. Cryptographic parameters used for security services provided by the card as well as the cardholder's witness PIN or other cardholder verification codes (CHVs) are stored in internal EFs.

Two referencing methods for elementary files are used:

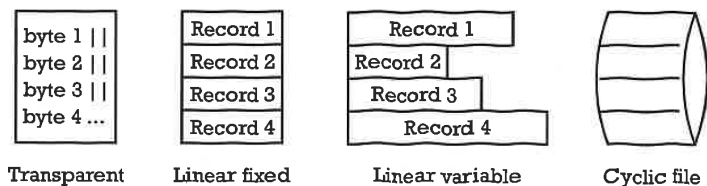
1. *Referencing with an FID, which consists of 2 bytes (4 hexadecimal digits).* The same FID referencing mechanism as that described for the DF can be also used for the EF. The disadvantage of this referencing mode is that before a file management command can be applied on an EF, the terminal must explicitly select this EF inside the DF corresponding to the card application. Examples of file management operations are the reading of some bytes from a transparent EF or the writing of a record in a linear fixed EF. Another disadvantage is that the terminal must know beforehand the FID of all the elementary

2. *Referencing with a Short File Identifier (SFI), which consists of a number between 1 and 30 that can be encoded on 5 bits.* This referencing method has the advantage that the SFI can be used as a file handler, which can be given as an input parameter to a file management command. This means that there is no need of executing an explicit selection of an EF inside the DF before calling a read/write command from/to an EF. Moreover, the SFI of all the working EFs in the card's application DF can be easily listed in a kind of DF table of contents. This helps the terminal learn by itself the publicly available working EF(s) existing in a DF. Therefore, this referencing method of EFs is preferred in open design interoperable card applications (see Section 3.4).

The structure of an EF depends on its intended use. As explained in ISO/IEC 7816-4 [5] one can distinguish among four basic types of EF structures. Transparent files consist of a sequence of bytes. A linear fixed file consists of a number of records, all having the same length. A linear variable file consists of a number of records, each with a variable length. The cyclic files contain records of fixed length organized in a ring structure. After all the records are written, the oldest entry in the file will be overwritten by the current entry to be stored. Figure 3.4 schematizes the four types of file structures.

The file header of each EF stores information about the type of EF file structure and the size of the file. It also stores the possible actions to be performed on the file (read, write, invalidate, rehabilitate, increase) as well as the access conditions under which a terminal application can perform that action (card's general PIN or application PIN, authentication with a symmetric key, access always permitted or access never permitted).

Figure 3.4
Four types of
elementary file
data structures.



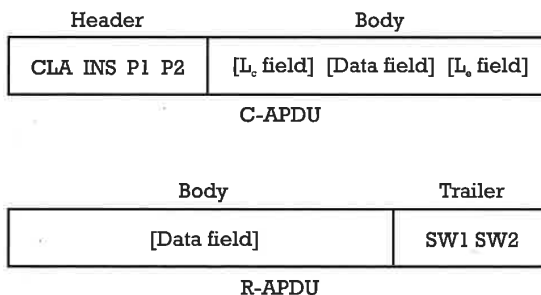
3.2.3 Command and response format

In accordance with the OSI 7-layer model, the information transaction exchanged between the card and the terminal can be divided into three protocol sections:

1. The physical layer protocol (layer 1) corresponds to the electrical signals on the I/O contact of the card.
2. Data transmission protocols (layer 2) correspond to T = 0 and T = 1 protocols [4]. They are both asynchronous, half-duplex protocols. T = 0 is a byte-oriented transmission protocol of the first-generation chip cards when the computing power and the RAM on the chip was limited. It does not allow the transmission of data both in the command and in the response. T = 1 is a block-oriented protocol, which better respects the OSI reference model and allows transmission of data both in the command and in the response. The data is handled in blocks and the error checking is carried out on an entire block of data rather than on 1 byte.
3. Application protocols for command and response data (layer 7). A step in an application protocol consists of sending a command application protocol data unit (C-APDU) from the terminal application to the card application. The latter processes it and sends back the response application protocol data unit (R-APDU) to the terminal application. A schematized picture of a C-APDU/R-APDU pair is given in Figure 3.5.

The C-APDU consists of a mandatory header of 4 bytes and an optional body of a variable length. The header includes the class of instructions to which the command belongs (CLA), the instruction code (INS) determining the command inside of a class, and the parameters of the instruction

Figure 3.5
Command/
response pair
(C-APDU/
R-APDU).



in the data field of the command.

- *Data field*: This field (of L_c bytes) contains the string to be sent as input data to the card application.
- L_c : This field (of a variable length up to 3 bytes) can contain the maximum number of bytes expected in the data field of the response returned by the card.

The R-APDU contains a conditional body of variable length L , that can be less than or equal to L_c . The R-APDU includes the trailer, which is a mandatory field of 2 bytes containing the status words (SW1, SW2). The status words inform the terminal application about the result of executing the command in the card application.

The ISO/IEC 7816-4 [5] standard defines only the basic commands. They can be grouped in file selection, read data, modify/delete data, generate data, compare data, and authenticate through cryptographic functions. In addition to these standardized commands, each card operating system or each card application defines private use commands. For example, the EMV™ debit/credit card application defines its own commands beside those in ISO/7816-4 (e.g., the GET PROCESSING OPTIONS, and GENERATE AC). The commands for the creation and personalization of files in the card's file system and the commands for blocking either an application or the entire card are further examples of private use commands. One can understand why even if chip card operating systems have been implemented according to the ISO/IEC 7816 standard, it does not necessarily mean that they are compatible with each other [11].

3.2.4 Card application and terminal application

The terminal at the point of service is a card acceptor device (CAD) equipped with a chip card reader, which is often referred to as the Interface Device (IFD). The terminal interacts with an ICC according to the client server model.

- A client application runs in the terminal. This client application is referred to as the terminal application.

- A server application runs in the ICC. This server application is referred to as the card application.

The terminal application sends commands as a client to the card application, which responds as a server.

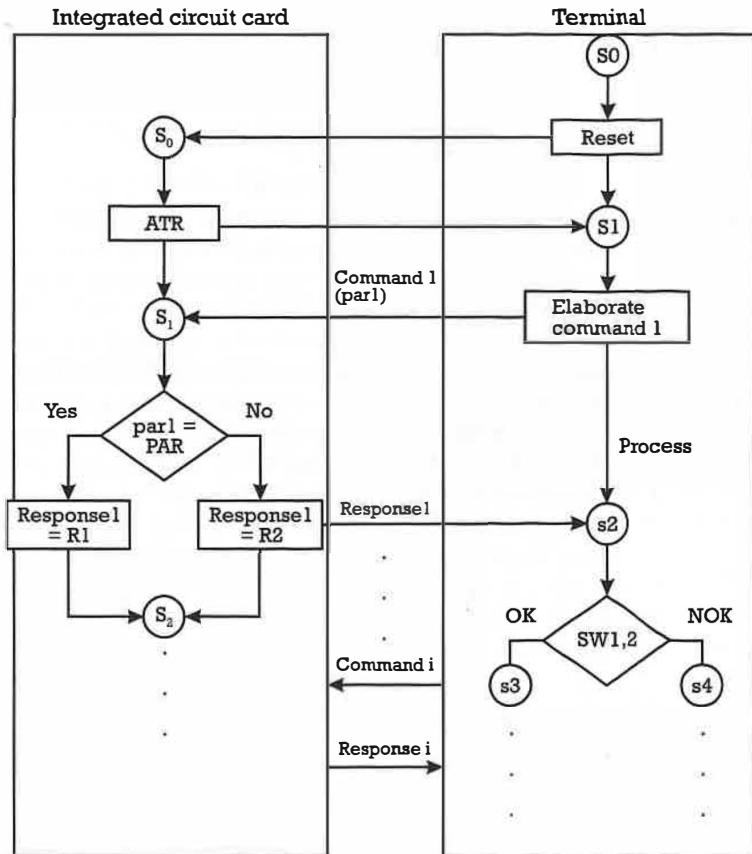
To easily explain the processing performed by both the terminal application and the card application, as well as their interaction in the client server model, it is convenient to represent them as algorithmic state machines (ASM). For an ASM the next state depends on the actual state and the event that triggers the transition from one state to another. The state of the ICC consists of a set of data elements and cryptographic parameters. The cryptographic parameters are organized in the card's file system. From this point of view the ICC can be seen as a permanent storage medium. Compared to a magnetic stripe card, which was a passive storage medium, the ICC has computational power provided by its own microprocessor. The event that triggers the transition of the ICC from one state to another is a command with parameters received from the terminal. As a consequence of this transition the ICC performs an action. First, the action consists of updating the value of the data elements and cryptographic parameters stored in the ICC (i.e., the state of the ICC), according to the requirements of the command and the accompanying parameters. Second, the action computes a response that is returned to the terminal. The response can contain the value of one or more data elements stored or computed in the card and a status word, which describes whether or not the command was successfully completed in the card. In case of failure, the status word indicates the source of error. The response received from the card represents the event that triggers the transition of the terminal from one state to another.

The terminal brings the ICC from an initial state S_0 to an operational state S_1 through an electrical reset. The action performed by the card following the reset is to prepare and send back to the terminal an answer to reset (ATR) response, which contains enough information to allow the communication subsystem of the terminal to synchronize with the communication subsystem of the card. Once this initial handshake is performed, the terminal can send C-APDU to and receive R-APDU from the card. The client-server relationship established between the terminal and the card is presented in Figure 3.6.

The set of commands and responses exchanged between the terminal and the ICC in the framework of a transaction is called a transaction profile.

A card application in the ICC contains a set of data elements that can be accessed by the terminal after a successful selection of the application. A data element is the smallest information unit that can be identified by a

Figure 3.6
Terminal
application and
card application
in a client server
configuration.



name, a description of its logical content, and a format. Data elements are mapped onto data objects, which are encoded according to a certain format (e.g., fixed length format, BER-TLV format, and others).

The terminal application consists of the sequence of commands, which are launched by the terminal to trigger the transition of the card application from one state to another. This determines the processing in the card according to the functionality of the card application. The terminal application also processes the data objects received in the responses from the card and the status words reported at the end of each command.

Several issues are identified in relation to the design of a card and terminal application, indifferent to whether it is a proprietary or interoperable solution:

1. One has to define the encoding of data elements into data objects in the card and the terminal application.
2. The organization of these data objects in a file system stored in the card and the referencing of files in this system must be defined. If several card applications reside in the same card, it is necessary to specify the separation of files corresponding to each application.
3. The set of commands supported by the card and the possible responses elaborated towards the terminal must be also defined.
4. The underlying cryptographic technology used for implementing the necessary security protections in both the card and the terminal must be chosen.

The possible solutions to these issues are restricted to the framework provided by the ISO/IEC 7816. The next two sections show how the aforementioned issues are solved in the case of a proprietary and closed payment application and in the case of an interoperable and open payment application, respectively.

3.3 Proprietary payment application

The approach described in this section outlines an oversimplified proprietary design solution, which can be adopted by payment system operators migrating from magnetic stripe cards to integrated circuit cards. The purpose is to show the shortcoming of this approach, in case open design and interoperability are business requirements.

Let us assume that a payment system operator provides a proprietary payment application, which consists of both a card and a terminal application. It is intended for the purpose of a dedicated business goal—for example, a national debit scheme for POS payments. The proprietary application is completely controlled by the payment system operator, who has designed and specified it according to its business requirements.

The card application is instantiated in chip cards of cardholders who are clients of an issuer. The issuer has established a business agreement for implementing the card application, which is provided by the payment system operator. The issuer has no freedom to customize the card application to its specific business needs. An ICC carrying the proprietary card application is accepted with terminals managed by an acquirer that has also established a business relationship with the payment system operator. The

CHAPTER

6

Contents

- 6.1 Overview of the EMV™ debit/credit transaction
- 6.2 Initiate application processing
- 6.3 Read application data
- 6.4 Off-line data authentication
- 6.5 Processing restrictions
- 6.6 Cardholder verification
- 6.7 Terminal risk management
- 6.8 Terminal action analysis
- 6.9 On-line processing and issuer authentication
- 6.10 Issuer scripts

Debit and Credit with EMV™

The goal of this chapter is to offer a tutorial presentation on the transaction profile of the EMV™ debit and credit payment application. This presentation should allow someone who has no time to read the entire specification to more easily understand the EMV™ payment method. It should be regarded as accompanying material that helps the reader get accustomed with this payment technology as quickly as possible. The chapter *does not aim to replace the specifications*, which should always be kept as the ultimate reference for any details. The presentation emphasizes the analysis of the protocol between the ICC and the terminal, rather than discussing the card and the terminal separately.

The remainder of the chapter is organized as follows. In Section 6.1 we give a high level description of the transaction flow for EMV™ debit/credit payment applications. Starting with Section 6.2, we describe in considerable detail the messages exchanged between the chip card and the terminal during each stage, the processing performed by the card, and the processing performed by the terminal. Matching the EMV™ transaction stages to the chapter sections is as follows: initiate application processing (Section 6.2), read application data (Section 6.3), card authentication (Section 6.4), processing restrictions (Section 6.5), cardholder verification (Section 6.6), and terminal risk management (Section 6.7). Section 6.8, corresponding to the terminal action analysis stage, considers the possible actions proposed by the terminal concerning the finalization of an EMV™ transaction at the point of service. Section 6.9 analyzes the on-line processing and issuer

authentication stage. Section 6.10 describes the prescriptions of the EMV™ standard concerning the possibility of updating financial data in the card application through the issuer script mechanism after the card enters its utilization life stage.

6.1 Overview of the EMV™ debit/credit transaction

In this section we give an overall picture of the interchange between the ICC and the terminal (Figure 6.1) and between the terminal and the IH (Figure 6.2) in order to complete the protocol of an EMV™ debit/credit payment application. The reader can compare the processing in these two diagrams with the processing performed for the magnetic stripe cards, as schematized in Figures 2.1 and 2.2 (see Section 2.1). It can be noticed that the complexity of the protocol between the card and the terminal increases in the case of an EMV™ transaction, while the processing performed by the payment network and back-office is comparable.

We refer now to Figure 6.1. All the processing described below is performed if the EMV™ debit/credit application is correctly selected during the application selection mechanism (see Section 4.4).

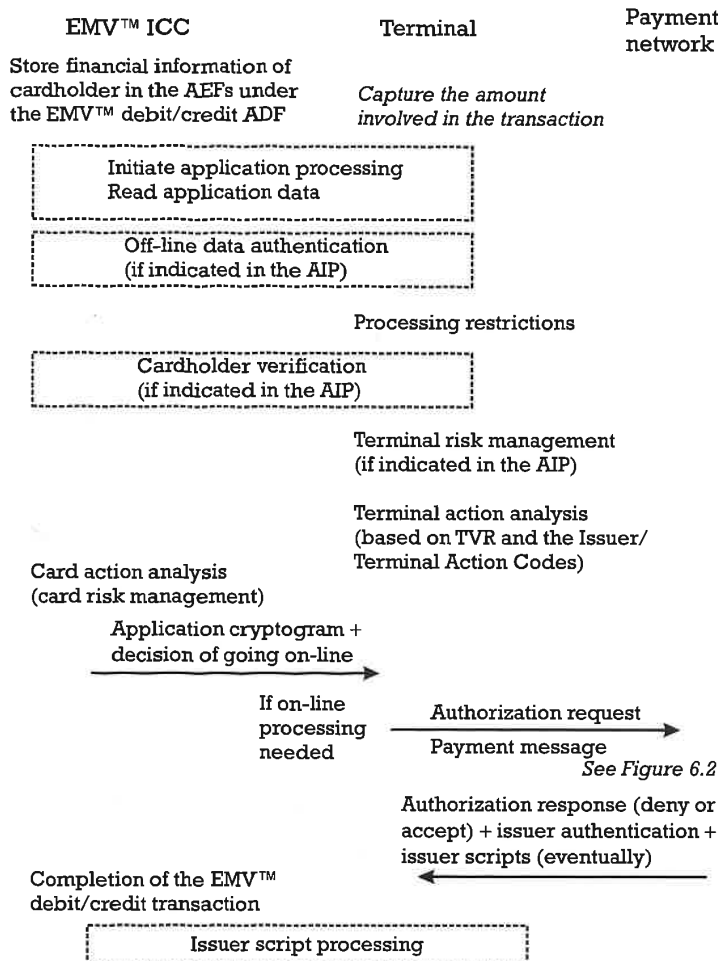
First, the terminal performs the initiate application processing function. The terminal informs the ICC about the business environment at the point of service, which helps the card tuning its strategy according to the conditions of the current transaction. The card provides a roadmap on its public AEF(s) in the AFL (see Section 3.4.2), and a profile of the transaction interchange in the AIP (see Section 3.4.3).

Based on the information contained in the AFL, the terminal is able to perform the read application data function. With this function all the public information contained in the AEF(s) attached to the EMV™ debit/credit card application is read by the terminal. After reading the content of the card, the terminal decides whether it has all the necessary data objects to continue the transaction.

Then, depending on the information contained in the AIP, the terminal decides whether off-line data authentication must be performed to enforce the card authentication security service. If the answer is positive, the terminal determines whether off-line SDA (see Appendix D, Section D.6.2) or off-line DDA (see Appendix D, Section D.7.2) is the cryptographic mechanism to be used for implementing this security service.

Furthermore, the terminal performs the processing restrictions function, which evaluates whether the ICC is entitled to ask for a certain financial service at the point of service (cash withdrawal, payment at a POS). If the

Figure 6.1
Interchange
between the
ICC and the
terminal for
an EMV™
debit/credit
transaction.



card is eligible for the kind of service required in the allowed environment (domestic or international), then the transaction interchange is continued or otherwise is aborted.

The terminal performs the cardholder verification stage if the ICC has proposed this function in the AIP. If this function is supported by the card application, the terminal determines a cardholder verification method that is mutually supported by the ICC and the terminal. Then, using this method, the terminal and the card establish the link between the user of the card and the legitimate cardholder.

The terminal also performs the terminal risk management function. This function allows the terminal to establish (from the point of view of the acquirer) whether the amount involved in the transaction is between some acceptable lower and upper floors. The terminal may also decide whether the transaction should be forwarded on-line following a large number of transactions that were completed off-line by the card.

At this stage of the processing, the terminal has a set of results from the completion of the previous functions. These results are accumulated in a 5-byte register called the Terminal Verification Results (TVR). The terminal compares this witness register against the Issuer Action Codes and the terminal action codes, which indicate the action to be performed by the terminal in case one stage in the transaction interchange did not complete as expected. Following this analysis, which is called the terminal action analysis, the terminal either denies or accepts the transaction. In the latter case, the terminal has the choice of completing the transaction locally (off-line) if no major risks are detected, or it can ask for the on-line assistance of the issuer. In either case, the card application is required to produce an application cryptogram, or even a digital signature, which will serve the dual purpose of authenticating on-line the card to the issuer or acting as a non-repudiation proof of the card's participation in a certain transaction.

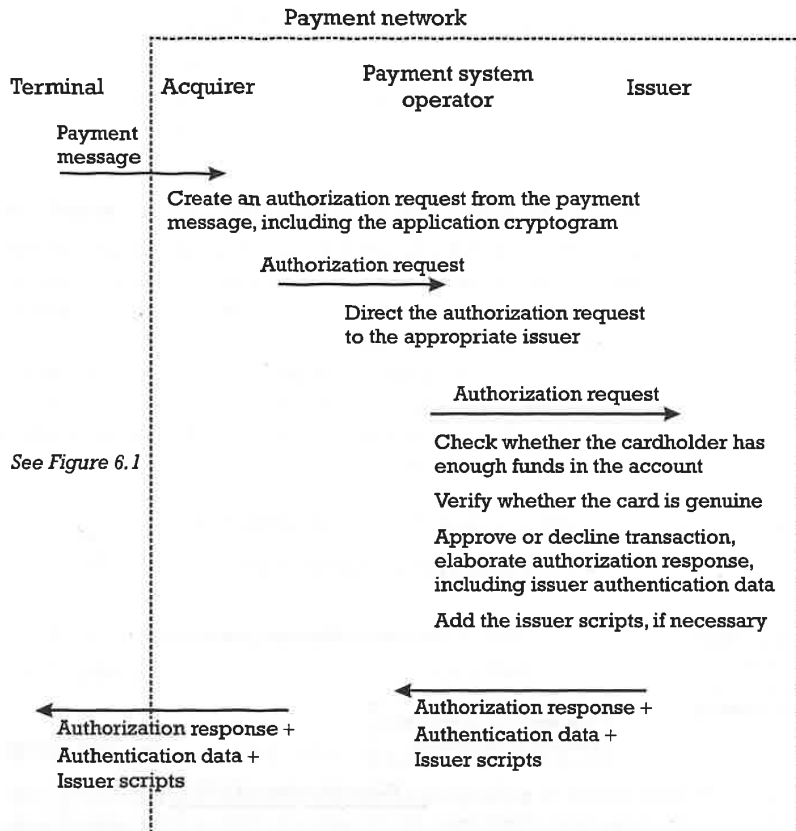
The ICC performs its own risk management procedure, according to the security policy of the issuer, within the function called card action analysis. Following the card risk management, the ICC could agree with the type of transaction finalization proposed by the terminal or it could ask for another type of finalization. For example, instead of accepting the off-line completion of the transaction, the card may ask the terminal to perform an on-line authorization request to the issuer. After performing the card action analysis, the decision about the off-line or on-line completion of the EMV™ debit/credit transaction is taken.

In case the terminal must contact the issuer on-line, the payment network has to perform its part of processing, as schematized in Figure 6.2.

The processing is similar to that performed in the case of magnetic stripe cards, with the observation that the payment message sent to the acquirer contains the supplementary information gathered by the terminal from the card during the processing explained in Figure 6.1. After receiving this information, the issuer can better assess whether the card is genuine or not, since the Application Cryptogram produced by the card is "fresh" for every new transaction. The authorization response that is sent back by the IH allows the terminal and the ICC either to deny or to accept the transaction, according to the indication transmitted by the issuer. To this end, the issuer

Figure 6.2

Payment network processing of an EMV™ debit/credit transaction.



must authenticate itself to the ICC, such that the ICC trusts that the finalization decision for the current EMV™ debit/credit transaction comes from the genuine IH.

It is important to note that the authorization response may also contain a sequence of commands to be transmitted by the terminal to the card, which were requested by the issuer. This is the issuer script processing function, through which the issuer can update the card application's parameters in the ICC during its lifetime, after the personalization stage is completed. In order to perform the issuer script processing, the authentication of the issuer must have been successful, such that the ICC can trust that the genuine issuer sends this sequence of commands through the terminal at the point of service.

- ATC and the Unpredictable Number;
- All the data objects necessary to reproduce the transaction claim.

If the ICC performs a combined DDA/AC generation, the terminal can verify off-line that the card is genuine by verifying the correctness of the Signed Dynamic Application Data returned in the R-APDU of the GENERATE AC command.

To this end the terminal first obtains an authentic copy of the ICC public key ($n_{icr} e_{ic}$), applying the procedure described in Section 6.4.3.2. Using this key, the terminal verifies the Signed Dynamic Application Data by applying a slightly modified version of the processing described in Section 6.4.3.4. The entire procedure can be found in Section 6.6.2 of Book 2 [2]. If the verification passes, the terminal accepts the fact that the card is genuine. If, however, the Signed Dynamic Application Data is stored (e.g., by the payment system operator), the issuer would have a supplementary possibility to prove in a court of law the participation of an EMV™ card in a certain transaction. This facility, however, has significant costs in terms of storage space, and should be utilized only for transactions involving important values of the parameter Amount, Authorized.

6.9 On-line processing and issuer authentication

During the EMV™ transaction processing, the terminal may send an authorization request message (1100) to the acquirer because of at least one of the following reasons:

- The terminal is an “on-line-only” type, which requires always the authorization of the issuer.
- The attendant explicitly triggers the authorization request message, since there are some suspicions about the cardholder at the point of service.
- The terminal risk management stage has chosen the current transaction at random for on-line authorization. The terminal could reach the same decision following the velocity checking, which revealed a high number of transactions that were performed off-line.
- The terminal requires on-line authorization following the terminal action analysis stage, when the TVR was compared against the terminal/Issuer Action Code-On-line. The terminal issues the first

GENERATE AC command with the reference control parameter (P1) set up to ARQC. The card analyzes the proposal of the terminal according to its own card risk management procedure, which reflects the security and availability policies of the issuer, and agrees with an on-line authorization by the issuer.

- The terminal requires off-line completion following the terminal action analysis. The terminal issues the first GENERATE AC command with the reference control parameter (P1) set up to TC. The card analyzes the proposal of the terminal according to its own card risk management procedure and decreases the decision level proposed by the terminal from TC to ARQC.

Whenever the card answers with an ARQC in the Cryptogram Information Data, the terminal starts up the on-line processing function. This means that the card and the terminal judged the current transaction outside the limits of risk for an off-line completion, as defined by the issuer, payment system, and acquirer. Therefore, the issuer is required to analyze the actual EMV™ transaction at the point of service, based on the information it receives from the terminal and the account information it stores in the IH in connection with the card. This guarantees that the issuer can review the conditions of the transaction and can approve or reject it.

6.9.1 Authorization request and response with chip data

The terminal generates an authorization request message 1100. The data fields included in this message are those explained in Section 2.9 for the processing of transactions using magnetic stripe cards. In addition, the authorization request message includes the field ICC system-related data, corresponding to bit 55 in the bitmap (see Section 2.8.1) of the data elements present in the 1100 message, according to ISO 8583:1993. This field contains all the data objects from the point of service, which allows the IH to verify the correctness of the ARQC produced by the card. Among these data objects one can mention:

- Application Cryptogram (ARQC);
- Account identification data of the card (PAN and PAN Sequence Number);
- ATC and the unpredictable number;

- Issuer Application Data (IAD), which may include the key indicator and the algorithm version number to uniquely specify a certain version of the IMK_{Ac} and of the MAC algorithm used to compute the ARQC in the card. The IAD may also include the witness values of the SDA or DDA processing by the terminal (i.e., the Data Authentication Code or the ICC Dynamic Number, respectively). In addition, the card risk management verification results can also be included in the IAD.
- All the data objects necessary to reproduce the transaction claim.

After receiving the authorization request message 1100, the IH first verifies that the card at the point of service is genuine. To this end, the cryptographic module of the IH must verify the correctness of the ARQC, according to the following steps:

- Use the key indicator present in the IAD to retrieve the correct version of the IMK_{Ac} .
- Use the account identification data to derive the key MK_{Ac} corresponding to the card participating in the transaction.
- Derive the actual session key SSK_{Ac} from MK_{Ac} , using the ATC as diversifier information in the keys-tree.
- Use the algorithm version number in the IAD to call the same algorithm for the computation of the MAC as that used in the card.
- Use SSK_{Ac} to produce a MAC on the same data objects that were considered by the card in the transaction claim.
- Compare the computed MAC against the ARQC received in the authorization request message 1100. If the two values are equal, the ARQC is accepted as valid evidence that the card at the point of service is authentic and genuine.

The IH identifies the cardholder's account using the account identification data received in the authorization request message. The account information kept in connection with the card, including the history of the most recent transactions, allows the issuer to run an improved risk management procedure. The IH can perform additional verifications that are not possible at the point of service. Thus, the issuer can identify transactions performed with cards that were reported stolen, or that have already spent the limit of the account balance. Correspondingly, the issuer may approve or reject the transaction, indicating also the reason of the rejection.

The issuer prepares the Issuer Authentication Data with tag 91, which consists of two items: the authorization response cryptogram (ARPC) on 8 bytes, and the Authorization Response Code (ARC) on 2 bytes.

The cryptographic module of the IH computes the ARPC as a triple DES cryptogram in ECB mode on the result of an XOR operation between the ARQC and the ARC padded at right with 6 bytes 00h. The ARQC is the value received in the 1100 message and serves as a challenge for countering replay attacks. This cryptogram is produced with the same session key SSK_{Ac} , which was already produced by the cryptographic module of the IH, since the verification of the ARQC. The ARPC allows the card to authenticate the issuer and to check that the Authorization Response Code represents the decision of the issuer concerning the finalization of the current EMV™ transaction.

The issuer may also consider the status of the EMV™ transaction, as the terminal reports it in the TVR and as the card reports it in the card risk management verification results, which was a data object included in the IAD. These two data objects are recommended for inclusion in the authorization request message. Interpreting this status information of the EMV™ transaction or considering some other post-issuance management operations, the issuer may decide whether it is necessary to correct the operation of the card. Examples of such corrections include:

- The extension of an expiration date, which would allow a cardholder who forgot to renew his or her card to operate, for a limited period of time, with an expired card;
- The unblocking of an application, following a PIN try limit exceeded;
- The modification of some parameters of the card risk management, like amount spending limits per day, per week, or per month, depending on the cardholder's availability of funds in the corresponding account;
- The updating of some cryptographic parameters, which are periodically changed for security reasons.

All of these corrections decided by the IH are translated in a sequence of post-issuance commands, which are encapsulated in the so-called Issuer Script Template 1 with tag 71 or in the Issuer Script Template 2 with tag 72.

After all this processing, the IH creates the authorization response message 1110. This message includes the data fields already mentioned in Section 2.9, in connection with the processing of transactions carried out with magnetic stripe cards. In addition, the authorization response message

optionally includes the ICC system-related data field, corresponding to bit 55 in the bitmap of the data elements present in the 1110 message [4]. When the issuer authentication is performed, this field contains the Issuer Authentication Data, which consists of the ARPC and the ARC. When post-issuance management of the ICC is needed, the issuer includes in this field the Issuer Script Template 1 or the Issuer Script Template 2.

If the terminal does not receive the authorization response message, or it receives it too late, or with an invalid syntax, then the terminal shall process the transaction as being unable to go on-line (see Section 6.8.5).

If the authorization response message is received by the terminal but it does not contain the Issuer Authentication Data, the terminal shall not execute the `EXTERNAL AUTHENTICATE` command and shall set bit 5, "Issuer authentication was performed," in the TSI to 0. Otherwise, the card performs the issuer authentication as explained below.

6.9.2 Issuer Authentication

If the Issuer Authentication Data (tag 91) is received in the authorization response message 1110, the terminal checks the content of bit 3, "Issuer Authentication is supported," in the AIP.

When this bit is set to 1, the card supports issuer authentication through the `EXTERNAL AUTHENTICATE` command. In this case, the terminal prepares the C-APDU corresponding to this command according to Table 6.9.

The terminal may send this command only once in a transaction. The ICC should return the `SW1SW2 = 6985` in the R-APDU, if this restriction is not respected. Regardless of the status words in the R-APDU, the terminal

Table 6.9

C-APDU of the `EXTERNAL AUTHENTICATE` Command

Code	Value
CLA	00 (interindustry command)
INS	82
P1	00
P2	00
Lc	8 to 16 bytes
Data	<i>Issuer Authentication Data</i> —mandatory first 8 bytes contains the ARPC, while the following 1 to 8 bytes, if any, have a proprietary structure. Usually, when the ARC is considered in the computation of the ARPC, the bytes 9 and 10 include the ARC
Le	Not present

shall set to 1 bit 5, "Issuer authentication was performed," in the TSI. In case the SW1SW2 are different than 9000, the terminal shall set bit 7, "Issuer authentication was unsuccessful," in byte 5 of the TVR.

When bit 3, "Issuer Authentication is supported," in the AIP is set to 0, the card may combine the issuer authentication with the second GENERATE AC command. In this case, the issuer should have listed the data object Issuer Authentication Data (tag 91) in the CDOL2.

Regardless of whether EXTERNAL AUTHENTICATE or GENERATE AC is used to perform issuer authentication, the processing performed by the card is the same. The ICC decrypts the ARPC with the session key SSK_{AC} , which was previously derived for the computation of the ARQC. After performing the XOR operation with the value of the ARQC, the ARC can be obtained from the leftmost 2 bytes of the result. This computed value of the ARC is compared against the value received in the Issuer Authentication Data. If the two values are identical, the issuer authentication has passed.

6.10 Issuer scripts

It was mentioned in Section 6.9.1 that the authorization response message 1110 received from the IH can include post-issuance commands to be delivered to the ICC via the terminal. These commands are not relevant for the current EMV™ transaction, but they are used for updating the application data in the card during its utilization lifetime, or for switching an application in the card or even the entire card between the "unblocked" and "blocked" states. EMV™ does not make any provisions for updating the application code in the card. The format of the post-issuance commands can be proprietary to the issuer. They are not meaningful for the terminal, which should only dispatch them from the authorization response message and send them sequentially to the ICC that has to be updated.

6.10.1 Processing of issuer script templates

The IH can group the post-issuance commands in two types of templates:

1. Issuer Script Template 1 (tag 71) groups proprietary post-issuance commands to be transmitted to the ICC *before* sending the 2nd GENERATE AC to the ICC.
2. Issuer Script Template 2 (tag 72) groups proprietary post-issuance commands to be transmitted to the ICC after sending the second GENERATE AC to the ICC.

Each issuer script template, regardless of whether it is of the type 1 or type 2, can include the following data objects:

- Issuer Script Identifier (tag 9F18), which is represented on 4 bytes in binary format. This identifier is optional and is not interpreted by the terminal. The Issuer Script Identifier allows the issuer to distinguish among several issuer script templates that can be included in the same authorization response message.
- Issuer Script Command APDU (tag 86) has a variable number of bytes depending on the type of the C-APDU to be sent to the card. Several Issuer Script Command APDU(s) can be included in an issuer script template.

The issuer can send more than one issuer script template in the same authorization response message. The only restriction is that the total length of the issuer script templates is less than or equal to 128 bytes.

After the reception of the authorization response message, the terminal processes each issuer script template in the sequence it appears in field 55 of the authorization response message. For each template the terminal performs the following processing:

- Create a new data structure issuer script results of 5 bytes (see Appendix A5 in Book 4 [3]), which will store the results concerning the processing of the commands contained in the current issuer script template. Add this data structure at the end of a byte string containing the data structures corresponding to other templates that were already processed.
- Reset the command counter that keeps the number of Issuer Script Command APDU(s) identified in the current issuer script template.
- Parse the value field of the current template.
 - Check whether the Issuer Script Identifier (tag 9F18) is present. In the affirmative case, copy the value field of this data object into bytes 2 to 5 in the issuer script results. The Issuer Script Identifier is meaningful to the issuer when interpreting the issuer script results reported by the terminal after sending of the post-issuance commands to the ICC.
 - Create a first-in-first-out stack (FIFO), where each element contains the value field of one Issuer Script Command APDU data object (tag

86) separated from the value field of the template. Each new element added in the stack increments the command counter.

- The processing sequence described below is performed before the second GENERATE AC, if the current template was identified with tag 71, or after the second GENERATE AC, in case the current template was identified with tag 72. Repeat the following steps a number of times equal to the command counter:
 - Pop the C-APDU kept in the current element of the FIFO stack indicated by the stack pointer.
 - Deliver this C-APDU to the ICC.
 - Examine only the status word SW1 in the R-APDU delivered by the ICC.
 - If SW1 indicates normal processing (SW1 = 90) or warning (SW1 = 62 or 63), the processing continues with the next Issuer Script Command APDU stored in the stack. If the command counter indicates that this is the first C-APDU that is processed, set to 1 bit 3, "Script processing was performed," in byte 1 of the TSI.
 - If SW1 indicates an error condition, the processing does not continue with other C-APDU(s) in the stack. The terminal shall set the first nibble of the first byte in the issuer script results to 1, "Script processing failed." The terminal shall write the sequence number of the Issuer Script Command APDU that reported the error in the second nibble of the first byte in the issuer script results. This sequence number equals the value of the command counter, when less than E, or otherwise the sequence number is set to F. The terminal sets to 1 bit 6, "Script processing failed before final GENERATE AC," in the byte 5 of the TVR, if the current template is encoded with tag 71. The terminal sets to 1 bit 5, "Script processing failed after final GENERATE AC," in byte 5 of the TVR, if the current template is encoded with tag 72.
- When the processing of the entire sequence of Issuer Script Command APDU(s) is successfully performed, the terminal sets up the first nibble of the first byte in the issuer script results to 2, "Script processing successful." The terminal shall write the value 0 in the second nibble of the first byte in the issuer script results.

6.10.2 Post-Issuance Commands

The post-issuance commands can be divided into two groups:

1. Commands that change the status of an application or of the entire card, including APPLICATION BLOCK, APPLICATION UNBLOCK, and CARD BLOCK (see Sections 2.5.1, 2.5.2, and 2.5.3 in Book 3 [1]);
2. Commands that change the values of some internal parameters, like the status of a PIN, of some cryptographic keys, or a PIN value, or the values of the data elements associated with the EMV™ application that participates in the card risk management processing. This category includes the EMV™ post-issuance command PIN CHANGE/UNBLOCK (see Section 2.5.10 in Book 3 [1]). Payment systems and issuers may define supplementary commands tailored to their specific needs.

The post-issuance commands use secure messaging:

- The integrity and authenticity of the issuer is achieved using a MAC.
- The confidentiality of cryptographic keys or of a PIN value to be updated in the EMV™ application is achieved through symmetric key encryption.

References

- [1] EMVCo, *EMV 2000 Integrated Circuit Card Specification for Payment Systems, BOOK 3—Application Specification*, Version 4.0, December 2000, <http://www.emvco.com/specifications.cfm>.
- [2] EMVCo, *EMV 2000 Integrated Circuit Card Specification for Payment Systems, BOOK 2—Security and Key Management*, Version 4.0, December 2000, <http://www.emvco.com/specifications.cfm>.
- [3] EMVCo, *EMV 2000 Integrated Circuit Card Specification for Payment Systems, BOOK 4—Cardholder, Attendant, and Acquirer Interface Requirements*, Version 4.0, December 2000, <http://www.emvco.com/specifications.cfm>.
- [4] ISO/IEC 8583:1993, "Financial Transaction Card Originated Messages—Interchange Message Specifications," 1995.
- [5] ISO/IEC 3166, "Codes for the Representation of Names of Countries," 1997.

the rightmost position of M . Then add the smallest number of bytes 00h, which makes the length of the padded message $M' = M \parallel '80' \parallel '00' \parallel \dots \parallel '00'$ a multiple of 8 bytes. The message M' is then divided in t blocks P_i , $i = 1, \dots, t$.

2. The MAC session key K_s consists of only a leftmost key of 8 bytes $K_s = K_{sl}$, or of the concatenation of a leftmost key K_{sl} and a rightmost key K_{sr} such that $K_s = K_{sl} \parallel K_{sr}$, which is 16 bytes in total.
3. Use the 64 bit-length block cipher DES with the key K_{sl} in the CBC mode on the blocks P_i , $i = 1, \dots, t$ to obtain H_i , $i = 1, \dots, t$ as $H_i = \text{DES}(K_{sl})[P_i \text{ XOR } H_{i-1}]$, $i = 1, \dots, t$. The initial vector IV is equal to $H_0 = '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00'$.
4. Compute an 8-byte final block H_{t+1} , either according to the main process of ISO/IEC 9797 (i.e., $H_{t+1} = H_t$), or according to the Optional Process 1 of ISO/IEC 9797 (i.e., $H_{t+1} = \text{DES}(K_{sl})[\text{DES}^{-1}(K_{sr})[H_t]]$).
5. The MAC is then computed as the most significant s bytes of H_{t+1} .

E.5 Key derivation

This appendix deals with the process of key derivation, which is useful at least in the following two situations, which are typical for the design of electronic payment systems.

1. Since tamper-resistant devices cannot be completely trusted, the issuer stores in the EEPROM of the device cryptographic material that is unique to the device but is derived from some master key of the issuer. Thus, the secret key that is specific to the device is derived as $K_d = F_1(MK)[\text{Diversification_Info}]$. According to its key management policies, the issuer establishes the content of the diversification information *Diversification_Info*. It may contain items like the serial number of the device, the identifier of the master key used for key derivation, and/or other information specific either to the device or to the issuer. The specific form of the derivation function F_1 and the features of the master key MK are also particular choices of the issuer.
2. Since the repeated use of one single key for performing CBC encryption can lead to the leakage of information about the plaintext (see Section E.2), it is highly recommended that the confidentiality

service for each communication session be implemented with a session key. The value of the session key K_s is derived from the same permanent key K_p , which is known to both parties, using as diversification information a time-variant byte sequence, like a random number or the value of a counter, which is incremented for each new transaction performed. Thus, the session key is computed as $K_s = F_2(K_p)$ [*time_variant*]. The communicating parties jointly convene the algorithm describing the function F_2 and the features of the permanent key K_p .

Specific details on these algorithms are provided in the descriptions of some business cases presented in the book.

References

- [1] NIST FIPS PUB 46, *Data Encryption Standard*, January 1977 (revised as FIPS PUB 46-1:1988 and FIPS PUB 46-2:1993).
- [2] EMVCo, *EMV 2000 Integrated Circuit Card Specification for Payment Systems, BOOK 2—Security and Key Management*, Version 4.0, December 2000, <http://www.emvco.com/specifications.cfm>.
- [3] NIST FIPS PUB Draft, *Advanced Encryption Standard*, <http://csrc/nist/gov/publications/drafts/dfips-AES.pdf>.
- [4] NIST FIPS PUB 81, *DES Modes of Operation*, December 1980.
- [5] Menezes, A. J., P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, Boca Raton, FL: CRC Press, 1996.
- [6] Kocher, P. C., "Breaking DES," *RSA Laboratories' Cryptobytes*, Vol. 5, No. 2, 1999, also at <http://www.rsa.com/rsalabs/pubs/crypto>.
- [7] ISO/IEC 9797, "Information Technology—Data Cryptographic Techniques—Data Integrity Mechanisms Using a Cryptographic Check Function Employing a Block Cipher Algorithm," 1994.