EMV®*
Contactless Specifications for Payment Systems

Book C-2

Kernel 2 Specification

Version 2.1
March 2011

* EMV is a registered trademark in the U.S. and other countries and an unregistered trademark elsewhere. The EMV trademark is owned by EMVCo.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

EMV
Contactless Specifications for Payment Systems

Book C-2

Kernel 2 Specification

Version 2.1
March 2011

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of these Specifications is subject to the terms and conditions of the EMVCo Terms of Use agreement available at www.emvco.com. These Specifications are provided "AS IS" without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of these Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of these Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party's infringement of any intellectual property rights in connection with these Specifications.

Contents

1 Using This Manual 1
1.1 Purpose 1
1.2 Audience 1
1.3 Overview 2
1.4 Related Information 3
1.5 Terminology 5
1.5.1 Card 5
1.5.2 POS System 5
1.5.3 Reader 5
1.5.4 Terminal 6
1.5.5 Kernel 6
1.5.6 EMV Mode 6
1.5.7 Mag-Stripe Mode 6
1.5.8 Combination 7
1.5.9 Queue 7
1.5.10 Signal 8
1.5.11 Process 8
1.5.12 Configuration Option 8
1.5.13 Implementation Option 9
1.6 Notations 10
1.6.1 Application States 10
1.6.2 Requirements 11
1.6.3 Implementation Options 14
1.6.4 Hexadecimal Notation 15
1.6.5 Binary Notation 15
1.6.6 Decimal Notation 15
1.6.7 Data Object Notation 15
1.6.8 C-APDU Notational Convention 16
1.6.9 Other Notational Conventions 17

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

2 General Architecture 19
2.1 Introduction 19
2.2 POS System 21
2.2.1 Simple Payment Transaction 25
2.2.2 More Complex Transaction 26
2.3 Reader Processes 29
2.3.1 Process P 30
2.3.2 Process D 34
2.3.3 Process S 35
2.3.4 Process K 39
2.3.5 Process M 43
2.3.6 Inter-Process Communication 46
2.4 The Reader Database 47
3 Reader Process K — Kernel Processing 53
3.1 Introduction 53
3.2 Kernel Configuration and Implementation Options 55
3.2.1 Implementation Options 55
3.2.2 Configuration Options 58
3.3 The Kernel Database 60
3.4 Mag-Stripe Mode and EMV Mode 62
3.4.1 Overall Transaction Flow 62
3.4.2 Mag-Stripe Mode 63
3.4.3 EMV Mode 63
3.5 Data Exchange 64
3.5.1 Introduction 64
3.5.2 Sending Data 64
3.5.3 Requesting Data 65
3.6 Data Storage 67
3.6.1 Introduction 67
3.6.2 Standalone Data Storage 68
3.6.3 Integrated Data Storage 69
3.7 Torn Transaction Recovery 75
3.7.1 Introduction 75

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

3.7.2 Recovery Mechanism 75
3.7.3 Transaction Flow 76
3.8 Mobile Transactions 78
3.8.1 Introduction 78
3.8.2 Mobile Mag-Stripe Mode Transactions 78
3.8.3 Mobile EMV Mode Transactions 80
3.9 Balance Reading 81
3.9.1 Introduction 81
3.9.2 Reading 81
3.9.3 Display and Receipt 81
4 Data Organization 83
4.1 TLV Database 84
4.1.1 Principles 84
4.1.2 Access Conditions 85
4.1.3 Services 86
4.1.4 DOL Handling 90
4.2 Working Variables 91
4.3 List Handling 92
4.4 Torn Transaction Log 94
4.5 Configuration Data 96
4.5.1 Configuration Data – TLV Database 96
4.5.2 CA Public Key Database 98
4.5.3 Certification Revocation List 99
4.5.4 Phone Message Table 100
4.6 Lists of Data Objects in OUT 101
4.6.1 Data Record 101
4.6.2 Discretionary Data 104
4.7 Data Object Format 106
5 C-APDU Commands 109
5.1 Introduction 109
5.2 COMPUTE CRYPTOGRAPHIC CHECKSUM 111
5.2.1 Definition and Scope 111

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

5.2.2 Command Message 111
5.2.3 Data Field Returned in the Response Message 112
5.2.4 Status Bytes 112
5.3 GENERATE AC 113
5.3.1 Definition and Scope 113
5.3.2 Command Message 113
5.3.3 Data Field Returned in the Response Message 115
5.3.4 Status Bytes 117
5.4 GET DATA 118
5.4.1 Definition and Scope 118
5.4.2 Command Message 118
5.4.3 Data Field Returned in the Response Message 119
5.4.4 Status Bytes 119
5.5 GET PROCESSING OPTIONS 120
5.5.1 Definition and Scope 120
5.5.2 Command Message 120
5.5.3 Data Field Returned in the Response Message 120
5.5.4 Status Bytes 122
5.6 PUT DATA 123
5.6.1 Definition and Scope 123
5.6.2 Command Message 123
5.6.3 Data Field Returned in the Response Message 123
5.6.4 Status Bytes 124
5.7 READ RECORD 125
5.7.1 Definition and Scope 125
5.7.2 Command Message 125
5.7.3 Data Field Returned in the Response Message 126
5.7.4 Status Bytes 126
5.8 RECOVER AC 127
5.8.1 Definition and Scope 127
5.8.2 Command Message 127
5.8.3 Data Field Returned in the Response Message 128
5.8.4 Status Bytes 129

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

6 Kernel State Diagrams 131
6.1 Implementation Principles 132
6.2 Kernel Started 133
6.2.1 Local Variables 133
6.2.2 Flow Diagram 133
6.2.3 Processing 135
6.3 State 1 – Idle 137
6.3.1 Local Variables 137
6.3.2 Flow Diagram 137
6.3.3 Processing 142
6.4 State 2 – Waiting for PDOL Data 150
6.4.1 Local Variables 150
6.4.2 Flow Diagram 150
6.4.3 Processing 152
6.5 State 3 – Waiting For GPO Response 154
6.5.1 Local Variables 154
6.5.2 Flow Diagram 154
6.5.3 Processing 166
6.6 State 4 – Waiting for EMV Read Record Response 177
6.6.1 Local Variables 177
6.6.2 Flow Diagram 177
6.6.3 Processing 182
6.7 State 4' – Terminate on Next RA 188
6.7.1 Local Variables 188
6.7.2 Flow Diagram 188
6.7.3 Processing 190
6.8 State 5 – Waiting for Get Data Response 191
6.8.1 Local Variables 191
6.8.2 Flow Diagram 191
6.8.3 Processing 195
6.9 State 6 – Waiting for EMV Mode First Write Flag 198
6.9.1 Local Variables 198
6.9.2 Flow Diagram 198

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

6.9.3 Processing 201

6.10 States 4, 5, and 6 – Common Processing 203

6.10.1 Local Variables 203

6.10.2 Flow Diagram 203

6.10.3 Processing 212

6.11 State 7 – Waiting for Mag-stripe Read Record Response 222

6.11.1 Local Variables 222

6.11.2 Flow Diagram 222

6.11.3 Processing 226

6.12 State 8 – Waiting for Mag-stripe First Write Flag 232

6.12.1 Local Variables 232

6.12.2 Flow Diagram 232

6.12.3 Processing 234

6.13 States 7 and 8 – Common Processing 235

6.13.1 Local Variables 235

6.13.2 Flow Diagram 235

6.13.3 Processing 239

6.14 State 9 – Waiting for Generate AC Response - 1 243

6.14.1 Local Variables 243

6.14.2 Flow Diagram 243

6.14.3 Processing 248

6.15 State 10 – Waiting for Recover AC Response 254

6.15.1 Local Variables 254

6.15.2 Flow Diagram 254

6.15.3 Processing 258

6.16 States 9 and 10 – Common Processing 262

6.16.1 Local Variables 262

6.16.2 Flow Diagram 262

6.16.3 Processing 270

6.17 State 11 – Waiting for Generate AC Response - 2 281

6.17.1 Local Variables 281

6.17.2 Flow Diagram 281

6.17.3 Processing 294

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

6.18 State 12 – Waiting for Put Data Response Before Generate AC 311
6.18.1 Local Variables 311
6.18.2 Flow Diagram 311
6.18.3 Processing 314
6.19 State 13 – Waiting for CCC Response – 1 317
6.19.1 Local Variables 317
6.19.2 Flow Diagram 317
6.19.3 Processing 323
6.20 State 14 – Waiting for CCC Response - 2 329
6.20.1 Local Variables 329
6.20.2 Flow Diagram 329
6.20.3 Processing 335
6.21 State 15 – Waiting for Put Data Response After Generate AC 343
6.21.1 Local Variables 343
6.21.2 Flow Diagram 343
6.21.3 Processing 346

7 Procedures 349
7.1 Procedure – Pre-gen AC Balance Reading 349
7.1.1 Local Variables 349
7.1.2 Flow Diagram 349
7.1.3 Processing 351
7.2 State 16 – Waiting for Pre-gen AC Balance 352
7.2.1 Local Variables 352
7.2.2 Flow Diagram 352
7.2.3 Processing 354
7.3 Procedure – Post-gen AC Balance Reading 356
7.3.1 Local Variables 356
7.3.2 Flow Diagram 356
7.3.3 Processing 358
7.4 State 17 – Waiting for Post-gen AC Balance 359
7.4.1 Local Variables 359
7.4.2 Flow Diagram 359
7.4.3 Processing 361

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

7.5 Procedure – CVM Selection 362
7.5.1 Local Variables 362
7.5.2 Flow Diagram 362
7.5.3 Processing 367
7.6 Procedure – Prepare Generate AC Command 373
7.6.1 Local Variables 373
7.6.2 Flow Diagram 373
7.6.3 Processing 381
7.7 Procedure – Processing Restrictions 388
7.7.1 Local Variables 388
7.7.2 Flow Diagram 388
7.7.3 Processing 395
7.8 Procedure – Terminal Action Analysis 401
7.8.1 Local Variables 401
7.8.2 Flow Diagram 401
7.8.3 Processing 405
8 Security Algorithms 409
8.1 Unpredictable Number Generation 409
8.2 OWHF2 410
8.3 OWHF2AES 411
Annex A Data Dictionary 413
A.1 Data Objects by Name 413
A.1.1 Account Type 413
A.1.2 Acquirer Identifier 413
A.1.3 Active AFL 413
A.1.4 Active Tag 414
A.1.5 AC Type 414
A.1.6 Additional Terminal Capabilities 415
A.1.7 Amount, Authorized (Numeric) 416
A.1.8 Amount, Other (Numeric) 417
A.1.9 Application Capabilities Information 417
A.1.10 Application Cryptogram 418

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

A.1.11 Application Currency Code 419
A.1.12 Application Currency Exponent 419
A.1.13 Application Effective Date 419
A.1.14 Application Expiration Date 420
A.1.15 Application File Locator 420
A.1.16 Application Interchange Profile 421
A.1.17 Application Label 422
A.1.18 Application Preferred Name 422
A.1.19 Application PAN 422
A.1.20 Application PAN Sequence Number 422
A.1.21 Application Priority Indicator 423
A.1.22 Application Transaction Counter 423
A.1.23 Application Usage Control 424
A.1.24 Application Version Number (Card) 424
A.1.25 Application Version Number (Reader) 425
A.1.26 Balance Read Before Gen AC 425
A.1.27 Balance Read After Gen AC 425
A.1.28 CA Public Key Index (Card) 426
A.1.29 Card Data Input Capability 426
A.1.30 CDOL1 426
A.1.31 CDOL1 Related Data 427
A.1.32 Cryptogram Information Data 427
A.1.33 CVC3 (Track1) 427
A.1.34 CVC3 (Track2) 428
A.1.35 CVM Capability – CVM Required 428
A.1.36 CVM Capability – No CVM Required 429
A.1.37 CVM List 429
A.1.38 CVM Results 430
A.1.39 Data Needed 430
A.1.40 Data Record 430
A.1.41 Data To Send 431
A.1.42 DD Card (Track1) 431
A.1.43 DD Card (Track2) 431

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 13

A.1.44 Default UDOL 432
A.1.45 DF Name 432
A.1.46 Discretionary Data 432
A.1.47 DRDOL 433
A.1.48 DRDOL Related Data 433
A.1.49 DS AC Type 433
A.1.50 DS Digest H 434
A.1.51 DSDOL 434
A.1.52 DS ID 435
A.1.53 DS Input (Card) 435
A.1.54 DS Input (Term) 435
A.1.55 DS ODS Card 436
A.1.56 DS ODS Info 436
A.1.57 DS ODS Info For Reader 437
A.1.58 DS ODS Term 437
A.1.59 DS Requested Operator ID 438
A.1.60 DS Slot Availability 438
A.1.61 DS Slot Management Control 439
A.1.62 DS Summary 1 440
A.1.63 DS Summary 2 440
A.1.64 DS Summary 3 440
A.1.65 DS Summary Status 441
A.1.66 DS Unpredictable Number 441
A.1.67 DSVN Term 442
A.1.68 Error Indication 442
A.1.69 Failed MS Cntr 444
A.1.70 File Control Information Issuer Discretionary Data 444
A.1.71 File Control Information Proprietary Template 445
A.1.72 File Control Information Template 445
A.1.73 ICC Dynamic Number 445
A.1.74 ICC Public Key Certificate 445
A.1.75 ICC Public Key Exponent 446
A.1.76 ICC Public Key Remainder 446

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

A.1.77 IDS Status 446
A.1.78 Interface Device Serial Number 447
A.1.79 Issuer Action Code – Default 447
A.1.80 Issuer Action Code – Denial 447
A.1.81 Issuer Action Code – Online 447
A.1.82 Issuer Application Data 448
A.1.83 Issuer Code Table Index 448
A.1.84 Issuer Country Code 448
A.1.85 Issuer Public Key Certificate 448
A.1.86 Issuer Public Key Exponent 449
A.1.87 Issuer Public Key Remainder 449
A.1.88 Kernel Configuration 449
A.1.89 Kernel ID 450
A.1.90 Language Preference 450
A.1.91 Log Entry 450
A.1.92 Mag-stripe Application Version Number (Reader) 451
A.1.93 Mag-stripe CVM Capability – CVM Required 451
A.1.94 Mag-stripe CVM Capability – No CVM Required 452
A.1.95 Max Lifetime of Torn Transaction Log Record 452
A.1.96 Max Number of Torn Transaction Log Records 453
A.1.97 Merchant Category Code 453
A.1.98 Merchant Custom Data 453
A.1.99 Merchant Identifier 453
A.1.100 Merchant Name and Location 454
A.1.101 Message Hold Time 454
A.1.102 Mobile Support Indicator 454
A.1.103 NATC(Track1) 455
A.1.104 NATC(Track2) 455
A.1.105 Next Cmd 456
A.1.106 nUN 456
A.1.107 ODA Status 457
A.1.108 Offline Accumulator Balance 457
A.1.109 Outcome Parameter Set 458

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 15

A.1.110 PCVC3(Track1) 460
A.1.111 PCVC3(Track2) 460
A.1.112 PDOL 460
A.1.113 PDOL Related Data 461
A.1.114 POS Cardholder Interaction Information 461
A.1.115 Post-Gen AC Put Data Status 462
A.1.116 Pre-Gen AC Put Data Status 463
A.1.117 Proceed To First Write Flag 464
A.1.118 Protected Data Envelope 1 464
A.1.119 Protected Data Envelope 2 465
A.1.120 Protected Data Envelope 3 465
A.1.121 Protected Data Envelope 4 465
A.1.122 Protected Data Envelope 5 465
A.1.123 PUNATC(Track1) 466
A.1.124 PUNATC(Track2) 466
A.1.125 Reader Contactless Floor Limit 466
A.1.126 Reader Contactless Transaction Limit 467
A.1.127 Reader Contactless Transaction Limit (No On-device CVM) 467
A.1.128 Reader Contactless Transaction Limit (On-device CVM) 467
A.1.129 Reader CVM Required Limit 468
A.1.130 Reference Control Parameter 468
A.1.131 Response Message Template Format 1 469
A.1.132 Response Message Template Format 2 469
A.1.133 Security Capability 469
A.1.134 Service Code 470
A.1.135 Signed Dynamic Application Data 470
A.1.136 Static Data Authentication Tag List 470
A.1.137 Static Data To Be Authenticated 470
A.1.138 Tags To Read 471
A.1.139 Tags To Read Yet 471
A.1.140 Tags To Write After Gen AC 472
A.1.141 Tags To Write Before Gen AC 472
A.1.142 Tags To Write Yet After Gen AC 472

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 16

A.1.143 Tags To Write Yet Before Gen AC 473
A.1.144 Terminal Action Code – Default 473
A.1.145 Terminal Action Code – Denial 473
A.1.146 Terminal Action Code – Online 473
A.1.147 Terminal Capabilities 474
A.1.148 Terminal Country Code 475
A.1.149 Terminal Identification 475
A.1.150 Terminal Type 475
A.1.151 Terminal Verification Results 476
A.1.152 Third Party Data 477
A.1.153 Time Out Value 478
A.1.154 Torn Entry 478
A.1.155 Torn Record 478
A.1.156 Torn Temp Record 478
A.1.157 Track 1 Data 479
A.1.158 Track 1 Discretionary Data 479
A.1.159 Track 2 Data 480
A.1.160 Track 2 Discretionary Data 480
A.1.161 Track 2 Equivalent Data 481
A.1.162 Transaction Category Code 481
A.1.163 Transaction Currency Code 482
A.1.164 Transaction Currency Exponent 482
A.1.165 Transaction Date 482
A.1.166 Transaction Time 482
A.1.167 Transaction Type 483
A.1.168 UDOL 483
A.1.169 Unpredictable Number 483
A.1.170 Unpredictable Number (Numeric) 484
A.1.171 Unprotected Data Envelope 1 484
A.1.172 Unprotected Data Envelope 2 484
A.1.173 Unprotected Data Envelope 3 485
A.1.174 Unprotected Data Envelope 4 485
A.1.175 Unprotected Data Envelope 5 485

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

A.1.176 User Interface Request Data..... 486

A.2 Data Objects by Tag..... 488

Annex B Data Exchange..... 495

B.1 Introduction..... 495

B.2 Example 1 – Generic Data Exchange..... 495

B.3 Example 2 – Stand Alone Data Storage..... 497

B.4 Example 3 – Integrated Data Storage..... 500

Annex C Offline CAM Optimization..... 503

C.1 Introduction..... 503

C.2 Optimization Techniques..... 504

Annex D Glossary..... 511

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Figures

Figure 1.1—Symbols Used in Transaction Flow Diagrams 11
Figure 1.2—Example of Symbol Notation and Textual Description 13
Figure 1.3—Implementation Option 14
Figure 2.1—General Architecture 19
Figure 2.2—POS System Logical Architecture 21
Figure 2.3—Simple Payment Transaction 25
Figure 2.4—Complex Transaction 28
Figure 2.5—Reader Logical Architecture 29
Figure 2.6—Process P 30
Figure 2.7—Process D 34
Figure 2.8—Process S 35
Figure 2.9—Process K 40
Figure 2.10—Process M 45
Figure 2.11—Inter-Process Communication 46
Figure 2.12—Reader Database – Persistent Datasets 48
Figure 3.1—Example Implementation Option in Flow Charts – Symbols 56
Figure 3.2—Example Implementation Option in Flow Charts – Checks 57
Figure 3.3—Kernel Database 60
Figure 3.4—Summaries – Basic Principle 72
Figure 6.1—Kernel Started Flow Diagram 134
Figure 6.2—State 1 Flow Diagram 138
Figure 6.3—State 2 Flow Diagram 151
Figure 6.4—State 3 Flow Diagram 155
Figure 6.5—State 4 Flow Diagram 178
Figure 6.6—State 4' Flow Diagram 189
Figure 6.7—State 5 Flow Diagram 192
Figure 6.8—State 6 Flow Diagram 199
Figure 6.9—States 4, 5, and 6 – Common Processing – Flow Diagram 204
Figure 6.10—State 7 Flow Diagram 223
Figure 6.11—State 8 Flow Diagram 233
Figure 6.12—States 7 and 8 – Common Processing – Flow Diagram 236
Figure 6.13—State 9 Flow Diagram 244
Figure 6.14—State 10 Flow Diagram 255
Figure 6.15—States 9 and 10 – Common Processing – Flow Diagram 263
Figure 6.16—State 11 Flow Diagram 282
Figure 6.17—State 12 Flow Diagram 312
Figure 6.18—State 13 Flow Diagram 318

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Figure 6.19—State 14 Flow Diagram 330
Figure 6.20—State 15 Flow Diagram 344
Figure 7.1—Pre-gen AC Balance Reading Flow Diagram 350
Figure 7.2—State 16 Flow Diagram 353
Figure 7.3—Post-gen AC Balance Reading Flow Diagram 357
Figure 7.4—State 17 Flow Diagram 360
Figure 7.5—CVM Selection Flow Diagram 363
Figure 7.6—Prepare Generate AC Command Flow Diagram 374
Figure 7.7—Processing Restrictions Flow Diagram 389
Figure 7.8—Terminal Action Analysis Flow Diagram 402
Figure B.1—Data Exchange Example 496
Figure B.2—SDS Example 498
Figure B.3—IDS Example 500
Figure C.1—Do Background Crypto 505
Figure C.2—Finish Key Recovery 508

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 20

Tables

Table 1.1—Other Notational Conventions 17
Table 2.1—Terminal Functionality 23
Table 2.2—Reader Functionality 24
Table 2.3—Terminal-Reader Service Requests 26
Table 2.4—Responses from the Reader 27
Table 2.5—Reader Processes 29
Table 2.6—Services from Process P 31
Table 2.7—Responses from Process P 32
Table 2.8—Services from Process S 36
Table 2.9—Responses from Process S 36
Table 2.10—Select Response Message Data Field of a Card Application 38
Table 2.11—Status Bytes for Select Command 38
Table 2.12—Services from Process K 41
Table 2.13—Responses from Process K 42
Table 2.14—Reader Databases 49
Table 2.15—Persistent Dataset Process S (per Transaction Type) 50
Table 2.16—Persistent Dataset Kernel 2 51
Table 3.1—Kernel Functionality 53
Table 3.2—Kernel Implementation Options 55
Table 3.3—Options and Implementations 56
Table 3.4—Kernel Configuration Options 58
Table 3.5—Kernel Database Categories 61
Table 4.1—Access Conditions 85
Table 4.2—Torn Transaction Log Record 94
Table 4.3—Configuration Data in TLV Database that Require Default Value 96
Table 4.4—CA Public Key Related Data 98
Table 4.5—Certification Revocation List Related Data 99
Table 4.6—Phone Message Table 100
Table 4.7—*Data Record* Detail for EMV Mode Transaction 102
Table 4.8—*Data Record* Detail for Mag-Stripe Mode Transaction 103
Table 4.9—*Discretionary Data* for an EMV Mode Transaction 104
Table 4.10—*Discretionary Data* for a Mag-Stripe Mode Transaction 104
Table 5.1—Coding of the Instruction Byte 109
Table 5.2—Generic Status Bytes 110
Table 5.3—COMPUTE CRYPTOGRAPHIC CHECKSUM Command Message 111
Table 5.4—COMPUTE CRYPTOGRAPHIC CHECKSUM Response Message Data Field 112

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Table 5.5—Status Bytes for COMPUTE CRYPTOGRAPHIC CHECKSUM Command 112
Table 5.6—GENERATE AC Command Message 113
Table 5.7—GENERATE AC Reference Control Parameter 114
Table 5.8—GENERATE AC Response Message Data Field (Format 1) 115
Table 5.9—GENERATE AC Response Message Data Field (Format 2) – No CDA 116
Table 5.10—GENERATE AC Response Message Data Field (Format 2) – CDA 116
Table 5.11—Status Bytes for GENERATE AC Command 117
Table 5.12—GET DATA Command Message 118
Table 5.13—Supported P1 || P2 Values for GET DATA Command 119
Table 5.14—Status Bytes for GET DATA Command 119
Table 5.15—GET PROCESSING OPTIONS Command Message 120
Table 5.16—GET PROCESSING OPTIONS Response Message Data Field (Format 1) 121
Table 5.17—GET PROCESSING OPTIONS Response Message Data Field (Format 2) 121
Table 5.18—Status Bytes for GET PROCESSING OPTIONS Command 122
Table 5.19—PUT DATA Command Message 123
Table 5.20—Supported P1 || P2 values for PUT DATA Command 123
Table 5.21—Status Bytes for PUT DATA Command 124
Table 5.22—READ RECORD Command Message 125
Table 5.23—P2 of READ RECORD Command 125
Table 5.24—READ RECORD Response Message Data Field 126
Table 5.25—Status Bytes for READ RECORD Command 126
Table 5.26—RECOVER AC Command Message 127
Table 5.27—RECOVER AC Response Message Data Field – No CDA 128
Table 5.28—RECOVER AC Response Message Data Field – CDA 128
Table 5.29—Status Bytes for RECOVER AC Command 129
Table 6.1—Response Message Data Field 197
Table 6.2—Mandatory EMV Mode Data Objects 215
Table 6.3—Mandatory Card CDA Data Objects 218
Table 6.4—Mandatory Mag-stripe Mode Data Objects 229
Table 6.5—ICC Dynamic Data (IDS) 271
Table 6.6—ICC Dynamic Data (No IDS) 271
Table 6.7—ICC Dynamic Data (IDS) 300
Table 6.8—ICC Dynamic Data (No IDS) 300
Table 7.1—Response Message Data Field 355
Table 7.2—Response Message Data Field 361

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

1 Using This Manual

1.1 Purpose

This document, *EMV Contactless Specifications for Payment Systems, Book C-2 – Kernel 2 Specification*, should be read in conjunction with:

- *EMV Contactless Specifications for Payment Systems, Book A – Architecture and General Requirements*, hereafter referred to as [EMV Book A], and
- *EMV Contactless Specifications for Payment Systems, Book B – Entry Point Specification*, hereafter referred to as [EMV Book B].

This document defines the behaviour of the Kernel used in combination with cards supporting a MasterCard brand or cards having a Kernel Identifier indicating Kernel 2, as defined in [EMV Book B].

The Kernel requirements cover both EMV mode and mag-stripe mode contactless transactions.

1.2 Audience

This specification is intended for use by manufacturers of contactless readers and terminals. It may also be of interest to manufacturers of contactless cards and to financial institution staff responsible for implementing financial applications in contactless cards.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 23

1.3 Overview

This volume includes the following chapters and annexes.

- **Chapter 1** contains general information that helps the reader understand and use this specification.
- **Chapter 2** introduces the model that is the basis for the architecture of the POS System. It describes the two logical components, Terminal and Reader, and the interaction between the two. It focuses on the Reader functionality, which is modelled as the coexistence of different processes – the Kernel being one of these processes.
- **Chapter 3** gives an overview of the features supported by Kernel 2 as well as its configuration and implementation options.
- **Chapter 4** describes the organization of the Kernel data; it distinguishes between the TLV Database, working variables, and it defines the key terms used for describing the access to and manipulation of data.
- **Chapter 5** defines the commands and responses exchanged between the Kernel and the Card during the course of a transaction.
- **Chapters 6 and 7** describe the processing of the Kernel, represented by a series of state transformations and procedure calls.
- **Chapter 8** describes the security algorithms used during transaction processing.
- **Annex A** gives the dictionary of data objects supported by the Kernel.
- **Annex B** contains examples of Data Exchange, one of the new features of Kernel 2.
- **Annex C** describes techniques to optimize offline CAM operations.
- **Annex D** is the list of abbreviations used in this specification.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

1.4 Related Information

The following references are used in this document. The latest version applies unless a publication date is explicitly stated.

Reference	Document Title
[EMV Book 1]	*Integrated Circuit Card Specifications for Payment Systems – Book 1, Application Independent ICC to Terminal Interface Requirements*, Version 4.2, June 2008
[EMV Book 2]	*Integrated Circuit Card Specifications for Payment Systems – Book 2, Security and Key Management*, Version 4.2, June 2008
[EMV Book 3]	*Integrated Circuit Card Specifications for Payment Systems – Book 3, Application Specification*, Version 4.2, June 2008
[EMV Book 4]	*Integrated Circuit Card Specifications for Payment Systems – Book 4, Cardholder, Attendant, and Acquirer Interface Requirements*, Version 4.2, June 2008
[EMV Book A]	*EMV Contactless Specifications for Payment Systems, Book A – Architecture and General Requirements*, Version 2.1
[EMV Book B]	*EMV Contactless Specifications for Payment Systems, Book B – Entry Point Specification*, Version 2.1
[EMV Book D]	*EMV Contactless Specifications for Payment Systems, EMV Contactless Communication Protocol Specification*, Version 2.1
[ISO 639-1]	Codes for the representation of names of languages – Part 1: Alpha-2 Code
[ISO 3166-1]	Codes for the representation of names of countries and their subdivisions – Part 1: Country codes
[ISO 4217]	Codes for the representation of currencies and funds
[ISO/IEC 7813]	Information technology — Identification cards — Financial transaction cards

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 25

Reference	Document Title
[ISO/IEC 7816-4]	Identification cards — Integrated circuit(s) cards with contacts — Part 4: Organization, security and commands for interchange
[ISO/IEC 7816-5]	Registration of application providers
[ISO 8583:1993]	Financial transaction card originated messages – Interchange message specifications
[ISO/IEC 8825-1]	Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
[ISO/IEC 8859]	Information technology – 8-bit single-byte coded graphic character sets
[ISO 14443-4]	Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 4: Transmission protocol
[ISO 18031:2005]	Information technology – Security techniques – Random bit generation
[NIST SP800-22A]	A statistical test suite for random and pseudorandom number generators for cryptographic algorithms

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

1.5 Terminology

This section discusses the following terms, which have specialized meanings in this specification:

1.5.1 Card
1.5.2 POS System
1.5.3 Reader
1.5.4 Terminal
1.5.5 Kernel
1.5.6 EMV Mode
1.5.7 Mag-Stripe Mode
1.5.8 Combination
1.5.9 Queue
1.5.10 Signal
1.5.11 Process
1.5.12 Configuration Option
1.5.13 Implementation Option

1.5.1 Card

The Card, as used in these specifications, is a consumer device supporting contactless transactions.

1.5.2 POS System

The POS System is the collective term given to the payment infrastructure present at the merchant. It is made up of the Terminal and Reader.

1.5.3 Reader

The Reader is the device that supports the Kernel(s) and provides the contactless interface used by the Card. Although this can be an integral part of the POS System, it is considered in this specification as a separate logical entity.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 27

1.5.4 Terminal

The Terminal is the device that connects to the authorization and/or clearing network and that together with the Reader makes up the POS System. The Terminal and the Reader may exist in a single integrated device, but are considered separate logical entities in this document.

1.5.5 Kernel

The Kernel contains interface routines, security and control functions, and logic to manage a set of commands and responses to retrieve the necessary data from the Card to complete a transaction. The Kernel processing covers the interaction with the Card between the selection of the card application (excluded) and processing of the outcome of the transaction (excluded).

1.5.6 EMV Mode

“EMV mode” describes an operating mode of the POS System that indicates that this particular acceptance environment and acceptance rules support chip infrastructure. It is typically used in conjunction with the term “transaction” (i.e. EMV mode transaction) to indicate contactless payment using a full chip infrastructure carrying EMV minimum data.

1.5.7 Mag-Stripe Mode

“Mag-stripe mode” describes an operating mode of the POS System that indicates that this particular acceptance environment and acceptance rules support magnetic stripe infrastructure. It is typically used in conjunction with the term “transaction” (i.e. mag-stripe mode transaction) to indicate contactless payment based on Track 1 and/or Track 2 Data obtained from the Card.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

1.5.8 Combination

"Combination" can refer to any of the following (see [EMV Book B]):

For…	The combination of…
a card	an ADF (Application Definition File) Name a Kernel Identifier
a reader	an AID (Application Identifier) a Kernel ID
the Candidate List for final selection	an ADF Name a Kernel ID the Application Priority Indicator (if present) the Extended Selection (if present)

1.5.9 Queue

A Queue is a buffer that stores events to be processed. The events are stored in the order received.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 29

1.5.10 Signal

A Signal is an asynchronous event that is placed in a Queue and handled in a FIFO manner. A Signal can convey data as parameters, and the data provided in this way is used in the processing of the Signal.

If a Signal is timed – say with a timer value T – then there is a delay of (T x 100) milliseconds associated with the processing of the next Signal on the Queue. By default, Signals have a timer value of zero.

Processes generating events may have different priorities due to hardware or software constraints. As a result, the order in which events are put on the Queue of a Process may be different than the order in which the events were created.

In particular, Signals from Terminal-originated events may have lower priority and putting them on a Queue may be deferred until after the queuing of an expected Card-related Signal.

Low level processes that manage I/O and generate events have higher priority than high level processes (e.g. Process S and Process K). So if these low level processes have events pending, they will push these events on the Queue of high level processes before the high level processes can start processing and pushing events on the Queues of other (high level) processes.

Putting Signals on Queues cannot be postponed indefinitely, and no Signal must be lost. (Additional information is provided in section 6.1.)

1.5.11 Process

A Process is a logical component within a Reader that has one or more Queues to receive Signals. The processing of Signals, in combination with the data they carry, may then generate other Signals to be sent. Processing continues until all the Queues of a Process are empty, or until the Process terminates.

1.5.12 Configuration Option

A configuration option allows activation or deactivation of the Kernel software behind the option. The configuration option may change the execution path of the software but does not change the software itself. A configuration option is set in the Kernel database. The impact is therefore at the level of an AID and a transaction type; different AIDs may have a different setting for the same configuration option and hence have a different execution path.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 30

1.5.13 Implementation Option

An implementation option allows the vendor to select whether the software behind the option will be implemented in a particular installation. An implementation option, when chosen by the vendor, reduces the number of execution paths supported by the software, changes the software itself, and impacts all the AIDs that rely on this software.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 31

1.6 Notations

This section discusses notational conventions used in this specification:

1.6.1 Application States
1.6.2 Requirements
1.6.3 Implementation Options
1.6.4 Hexadecimal Notation
1.6.5 Binary Notation
1.6.6 Decimal Notation
1.6.7 Data Object Notation
1.6.8 C-APDU Notational Convention
1.6.9 Other Notational Conventions

1.6.1 Application States

This document specifies the Kernel processing as a state machine that is triggered by Signals that cause state transitions. The application states of the Kernel are written in a specific format to distinguish them from the text:

`state`

Example:

GOTO `s4 – waiting for EMV read record response`

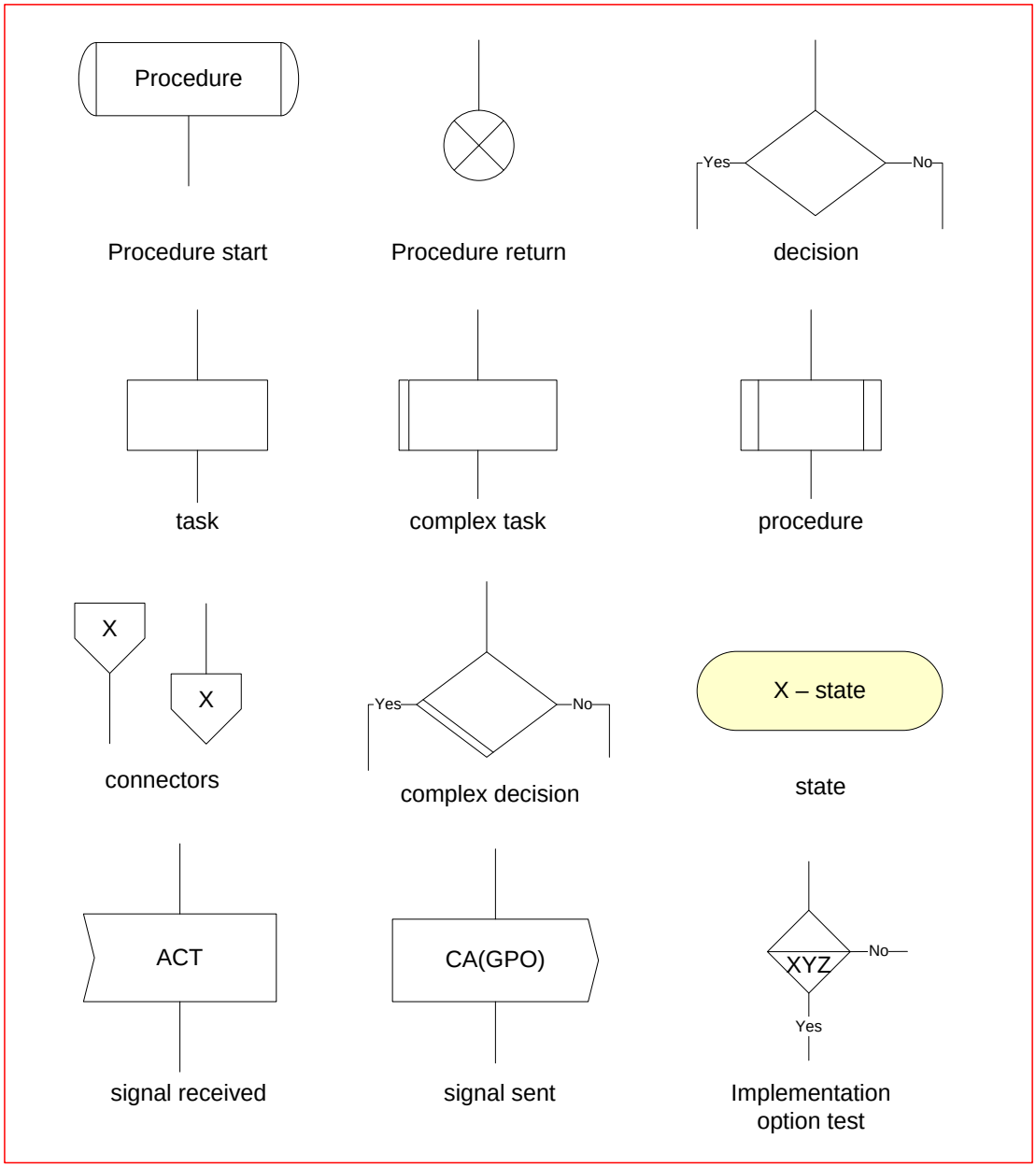
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 32

1.6.2 Requirements

To describe the state machine of the Kernel, this document uses a combination of flow diagrams and textual description.

Figure 1.1 shows the symbols used in the flow diagrams.

Figure 1.1—Symbols Used in Transaction Flow Diagrams



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 33

The combination of the flow diagrams and the corresponding textual descriptions constitute the requirements on the Kernel behaviour:

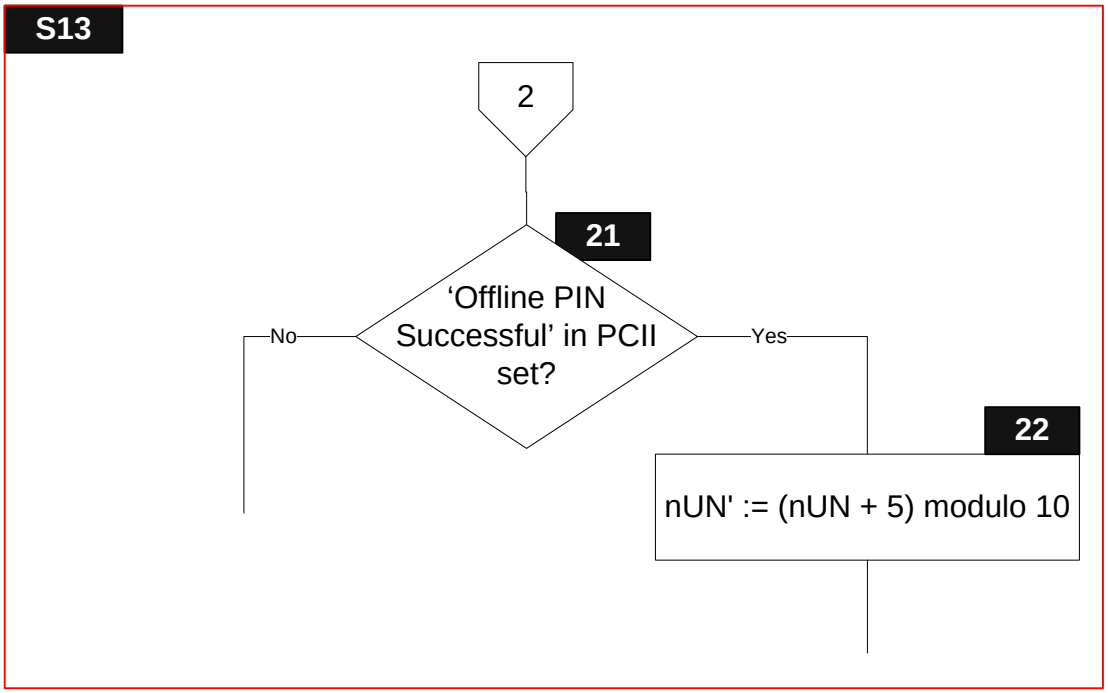
- Each diagram in this specification has a unique label.
- Each symbol in a diagram has a unique identifier that is the concatenation of the diagram label with the symbol number.
- The textual description corresponding to the symbol in a diagram starts with the identifier of the symbol.

The flow diagrams are read from top to bottom and define the order of execution of the processing steps. The textual description specifies the behaviour of the individual steps but bears no information on the order of execution.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Using the convention defined above, an example of a requirement is given in Figure 1.2 in combination with the textual description below:

Figure 1.2—Example of Symbol Notation and Textual Description



S13.22

nUN' := (*nUN* + 5) modulo 10

In this case:

- S13 is the label of the diagram.
- S13.22 is the unique identifier of one of the symbols.
- The textual description is that given following the symbol S13.22 and in this case it is *nUN'* := (*nUN* + 5) modulo 10.

The combination of the above constitutes a unique requirement that can be referred to as S13.22.

The requirements relate to the behaviour of the Kernel but leave flexibility in the actual implementation. The implementation must behave in a way that is indistinguishable from the behaviour specified in this document. Indistinguishable means that it creates the output as predicted by this specification for a given input. There is no requirement that the implementation realize the behaviour through a state machine as described in this document.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 35

1.6.3 Implementation Options

The specification allows for implementation options resulting in different possible implementations. Whether an option is implemented or not conditions the presence of certain data objects and symbols in flow diagrams.

The conventions for indicating these data objects and symbols are the following:

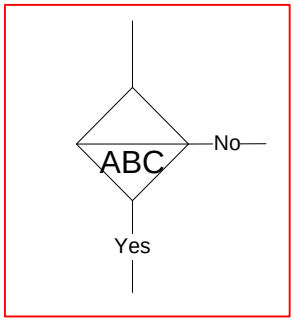
- In the data dictionary (Annex A), a data object that is linked to certain implementations is labelled as such in the “Implementations” field of the data object.
- In the flow diagrams, a symbol that is linked to an implementation option is labelled as such.
  - A symbol that is always present has a label consisting of all numeric digits.
  - A symbol that is linked to an implementation option (e.g. ABC) has a label consisting of numeric digits preceded by the first character of the reference to the implementation option.
  - A symbol that is linked to a combination of implementation options (e.g. ABC and XYZ) has a label consisting of numeric digits preceded by the first characters of the reference of all the implementation options.

  For example, for the implementation options ABC and XYZ, a symbol with a value 42 would have as its reference AX42.
- In the flow diagrams, if a decision diamond is used to test whether an implementation option is supported or not, the diamond has a label that refers to the implementation option.

  Example:

  A diamond that tests whether implementation ABC is supported is labelled ABC, as illustrated in Figure 1.3.

Figure 1.3—Implementation Option



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 36

1.6.4 Hexadecimal Notation

Values expressed in hexadecimal form are enclosed in straight single quotes.

For example, 27509 decimal is expressed in hexadecimal as '6B75'.

1.6.5 Binary Notation

Values expressed in binary form are followed by the letter b.

For example, '08' hexadecimal is expressed in binary as 00001000b.

1.6.6 Decimal Notation

Values expressed in decimal form are not enclosed in single quotes.

For example, '0B' hexadecimal is expressed in decimal as 11.

1.6.7 Data Object Notation

Data objects used for this specification are written in a specific font to distinguish them from the text:

Data Object Name

Example:

Application File Locator

Pre-Gen AC Put Data Status

To refer to a sub-element of a data object (i.e. a specific bit, set of bits, or byte of a multi-byte data object), the following notational convention is used:

- If the sub-element is defined in the data dictionary (Annex A), with each possible value of the sub-element having a name, then the following conventions apply:
  - The reference to the sub-element is 'Name of Sub-element' in *Data Object Name*.
  - The reference to the value is VALUE OF SUB-ELEMENT.

  Examples:
  - 'Offline PIN verification successful' in *POS Cardholder Interaction Information* refers to bit 5 of byte 2 in *POS Cardholder Interaction Information*.
  - 'CVM' in *Outcome Parameter Set* := ONLINE PIN means the same as “bits 4 to 1 of byte 4 of *Outcome Parameter Set* are set to 0010b”.
- Alternatively, an index may be used to identify a sub-element of a data object. In this case the following notational conventions apply:

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 37

- To refer to a specific byte of a multi-byte data object, a byte index is used within brackets (i.e. [ ]).

  For example, *Terminal Verification Results*[2] represents byte 2 of *Terminal Verification Results*. The first byte (leftmost or most significant) of a data object has index 1.

- To refer to a specific bit of a single byte multi-bit data object, a bit index is used within brackets [ ].

  For example, *Cryptogram Information Data*[7] represents the 7th bit of the *Cryptogram Information Data*. The first bit (rightmost or least significant) of a data object has index 1.

- To refer to a specific bit of a multi-byte data object, a byte index and a bit index are used within brackets (i.e. [ ][ ]).

  For example, *Terminal Verification Results*[2][4] represents bit 4 of byte 2 of the *Terminal Verification Results*.

- Ranges of bytes are expressed with the x : y notational convention:

  For example, *Terminal Verification Results*[1 : 4] represents bytes 1, 2, 3, and 4 of the *Terminal Verification Results*.

- Ranges of bits are expressed with the y : x notational convention:

  For example, *Cryptogram Information Data*[5 : 1] represents bits 5, 4, 3, 2, and 1 of the *Cryptogram Information Data*.

1.6.8 C-APDU Notational Convention

C-APDUs are written in a specific format to distinguish them from the text:

COMMAND

Example:

GET PROCESSING OPTIONS

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

1.6.9 Other Notational Conventions

Notations for processing data and managing memory are described in Table 1.1.

Table 1.1—Other Notational Conventions

Symbol	Meaning	Example
SET	A specific bit in a data object is set to the value 1b	SET 'CDA failed' in *Terminal Verification Results*
CLEAR	A specific bit in a data object is set to the value 0b	CLEAR 'Cardholder verification was not successful' in *Terminal Verification Results*
:=	A specific value is assigned to a data object or to a sub-element of a data object	'Status' in *Outcome Parameter Set* := END APPLICATION
OR	This notation is used for both the logical and bitwise OR operation. Its meaning is therefore context-specific.	Bitwise AND and OR: IF [((*Terminal Action Code – Online* OR *Issuer Action Code – Online*) AND *Terminal Verification Results*) = '0000000000']
AND	This notation is used for both the logical and bitwise AND operation. Its meaning is therefore context-specific.	Logical AND: IF [IsNotEmptyList(*Data To Send*) AND IsEmptyList(*Tags To Read Yet*)]
NOT	This notation is used for the logical negation operation.	IF [NOT ParseAndStoreCardResponse(TLV)]
||	Two binary data objects are concatenated.	A := 'AB34' B := A || 'FFFF' means that B is assigned the value 'AB34FFFF'

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 39

Symbol	Meaning	Example
IF THEN ELSE	This textual description is used to specify decision logic, using the following syntax: IF T THEN GOTO X ELSE GOTO Y ENDIF where T is a statement resulting in true or false and X and Y are symbol identifiers.	IF *Amount, Authorized (Numeric)* > *Reader CVM Required Limit* THEN GOTO S456.E25 ELSE GOTO S456.E26 ENDIF
GOTO	A GOTO statement is used to indicate the next step in the following two instances: • A decision diamond containing a test whose outcome determines subsequent processing • An off-page reference to another flow diagram	

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 40

2 General Architecture

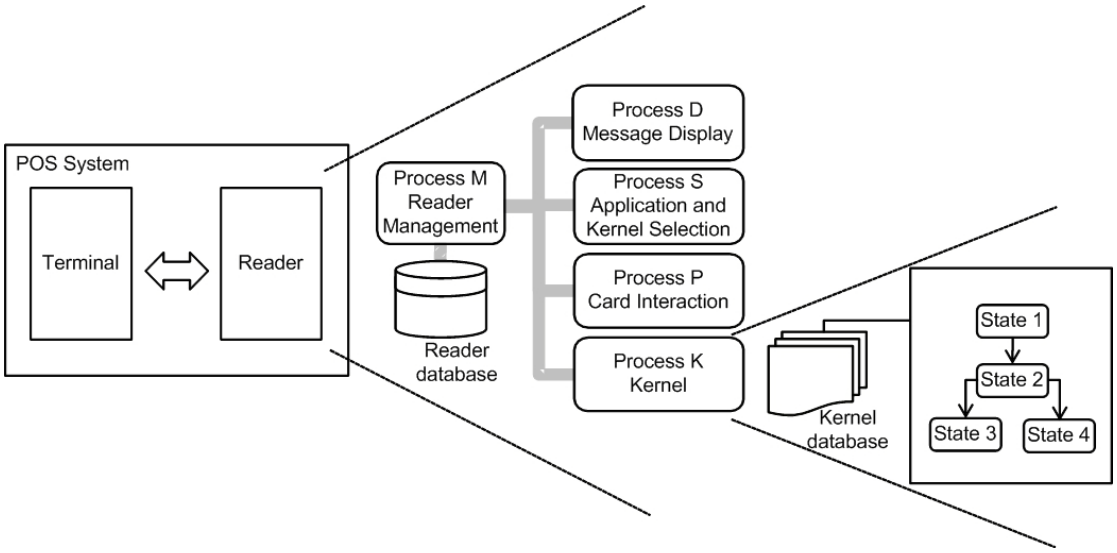
2.1 Introduction

As described in [EMV Book A], the general architecture of a POS System consists of a Terminal and a Reader, where the terms Terminal and Reader refer to a separation in responsibility and functionality between two logical entities.

This document starts from this general architecture, as illustrated in the left hand side of Figure 2.1, then zooms in on the Reader. Figure 2.1 shows how the Reader functionality is allocated to different processes: Process M(ain), Process D(isplay), Process S(elect), Process P(CD), and Process K(ernel).

Zooming in further on Process K, Figure 2.1 illustrates the two components of the Kernel: the Kernel software, modelled as a state machine, and the Kernel database, consisting of a number of separate datasets.

Figure 2.1—General Architecture



The Reader model presented in this document is slightly different from the model that is described in [EMV Book A] and [EMV Book B], as functionality is partitioned differently.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 41

[EMV Book A] partitions the functionality between the POS System, the Entry Point, and the Kernel in a specific manner but the partitioning is not prescriptive. It is easy to see how the Kernel maps onto Process K and how the other processes can be mapped into the POS System and the Entry Point. The difference lies mainly in the functionality that is allocated to the Entry Point.

In [EMV Book B], the Entry Point has some but not complete control of the electromagnetic field and handles the outcome of the Kernel. This functionality falls under Process M in this document.

This difference in partitioning has no impact on the Kernel requirements – which is the purpose of this document – and has no impact on the implementation of Reader, Terminal, or POS System.

There is no requirement to create devices that use the architecture and the partitioning as laid out in this document, as equally there is no requirement in [EMV Book A] on the partitioning.

The only requirements in this document apply to the Kernel and these requirements define the externally-observable behaviour, independent of the internal organization of the Reader.

Section 2.2 describes one way of partitioning the functionality between Terminal and Reader, and the Terminal-Reader interaction that results from such a partitioning. This interaction is described as a set of services that the Terminal can request from the Reader and vice versa. Service requests are modelled as Signals.

Section 2.3 describes how the Reader functionality is allocated to five processes that together ensure the Reader functionality. Each process has its own Queue(s) and communicates with the other processes through Signals.

Section 2.4 describes how each of the processes is configured and controlled and describes the role of the Reader database, consisting of multiple datasets for the different processes.

Chapter 3 and onwards then focuses on the Kernel as one of the processes, modelled to run independently of the other processes (concurrent operation) and described as a state machine. It sources its data from the Kernel database, consisting of a number of separate datasets.

None of the sections in Chapter 2 or Chapter 3 contains requirements on the Kernel (or the POS System); the information in these sections is relevant for understanding the different steps of a transaction and the services that may be requested from the Kernel.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 42

2.2 POS System

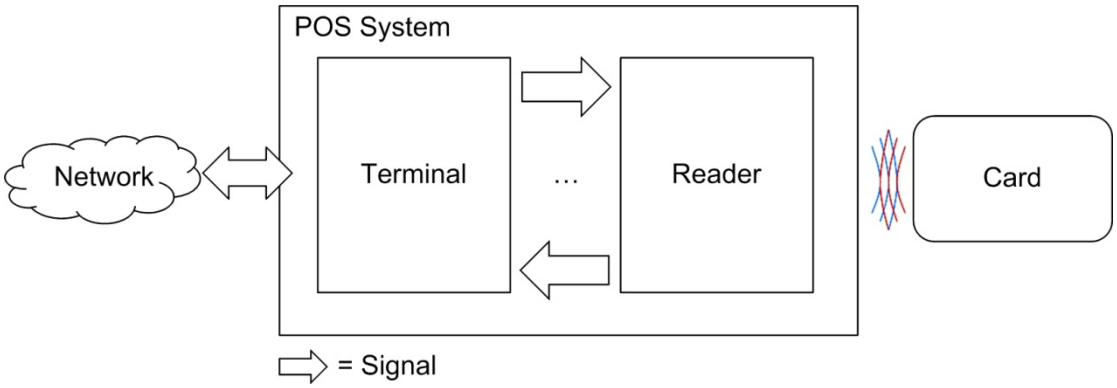
The physical architecture of the POS System can be any of the following:

- Fully integrated terminal: All functionality is included in a single device.
- Intelligent card reader: The reader handles most of the contactless transaction processing, passing the results for completion by the Terminal.
- Combination of terminal and transparent card reader: The Reader provides communication with the Card, whilst Kernels and other processes are in the Terminal.

The design described in this document is based on a physical architecture that is along the lines of an intelligent card reader; however it is not intended to be prescriptive.

The logical partitioning of the overall functionality of the POS System between Terminal and Reader is illustrated in Figure 2.2. The dialogue between Terminal and Reader is modelled as service requests, with Signals being used as vehicle for communicating these requests.

Figure 2.2—POS System Logical Architecture



The combination of Table 2.1 and Table 2.2 describes the overall functionality of the POS System: Table 2.1 lists functionality covered by the Terminal and Table 2.2 lists the functionality allocated to the Reader. The distribution of responsibility between Terminal and Reader laid out in Table 2.1 and Table 2.2 is in line with the physical architecture described in this document.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 43

The distribution of functionality between Terminal and Reader described in this specification is not intended to be prescriptive nor is the coding of the Signals prescriptive. The following rules however should be observed in regard of the specification:

- Whenever the Terminal – Reader interface uses a tagged data object of which the tag is coded on three bytes (for example 'DF8106' – Data Needed), this data object may be coded and conveyed by the actual communications mechanism in any appropriate manner. There are no requirements on the format or coding of such data object and any format or coding that achieves the same overall result is acceptable.
- When the Terminal – Reader interface uses a tagged data object of which the tag is coded on a single byte (for example '5A' – Application PAN) or is coded on two bytes (for example '9F02' – Amount, Authorized (Numeric)), this data object must be exchanged unaltered between the Terminal and the Reader. Neither its coding nor its format can be changed.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Table 2.1—Terminal Functionality

Functionality	Explanation
Business logic to determine the transaction amount and transaction type	In most cases, the transaction amount is determined prior to the transaction or is a fixed amount. In some cases the transaction amount may be determined or changed during the course of the transaction, based on information recovered from the Card.
Online authorization and transaction logging	The transaction may need to be authorized online. The terminal sends the online authorization request to the issuer. Upon completion of the transaction, it stores the clearing record and prepares the batch file for submission to the acquirer. The authorization request and clearing record include different data depending on whether the transaction was completed in mag-stripe mode or EMV mode.
Data storage logic to analyze the content of the data read from the Card and update it	This logic includes the security checks to verify the integrity and authenticity of the data stored on the Card as well as controlling access to the data. The detail of the content of the data to be stored on the Card is outside the scope of this document and will vary from one operator to the other. This document places no specific requirements on the structure of the data, and the Card and Reader are completely unaware of and unaffected by its structure.
Service provisioning or goods dispensing	The customer receives a service or physical goods in exchange of payment.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 45

Table 2.2—Reader Functionality

Functionality	Explanation
Communication with the Card	This includes the protocol for the contactless interface as defined by [EMV Book D] and the exchange of APDUs as defined in [ISO/IEC 7816-4].
User Interface	This includes the displaying of a message, a (LED/Audio) status, and optionally a language indicator and the duration for which the message should be shown. The message may include an amount or balance and currency code or currency symbol.
Selection of the Card application and identification of the Kernel	This functionality includes: • building the candidate list and identifying the application with the highest priority from the candidate list • selecting this application and identifying which Kernel should process it
Collection of (authenticated) payment data from the Card to populate an authorization and/or clearing record	Having completed the interaction with the Card, the Reader returns the necessary data for the Terminal to create an authorization or clearing message.
Management of Data Exchanges between Kernel and Terminal	Data Exchange provides a flexible communication mechanism between Terminal and Kernel. It allows the Kernel to send tagged data to and request data from the Terminal. It allows the Terminal to exercise a level of control on the Kernel by virtue of its ability to: • update the current transaction data • request tagged data from the Reader and Card • have tagged data written on the Card
Processing of the outcome provided by the Kernel	The Kernel indicates whether a transaction is approved offline, declined offline, authorized online, or if another action is required.

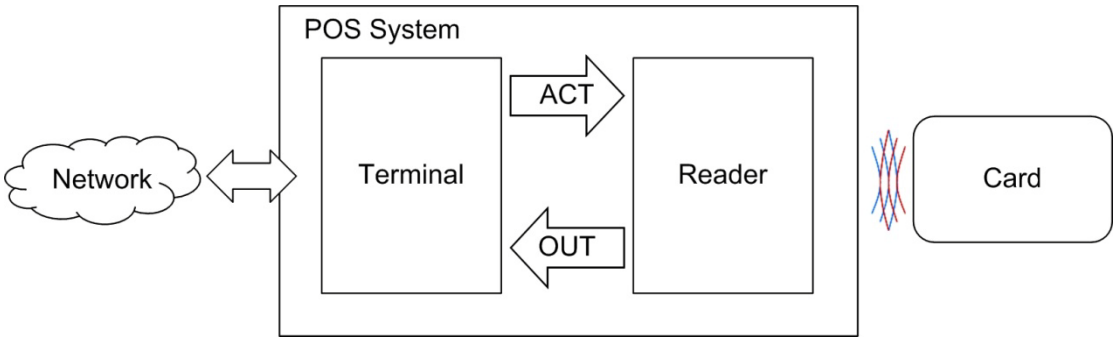
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 46

Functionality	Explanation
Configuration and control of the above	The different processing blocks within the Reader need to be configured, activated, and deactivated as a function of the transaction type, the AID, and the Kernel that has been selected.

2.2.1 Simple Payment Transaction

For the logical partitioning described in this document, a simple payment transaction requires only the exchange of two Signals between Terminal and Reader, as illustrated in Figure 2.3. These Signals are referred to as an ACT(ivate) and OUT(come).

Figure 2.3—Simple Payment Transaction



- The ACT Signal is used to activate the Reader and contains parameters such as the transaction amount and the transaction type. In some cases, the ACT Signal may not be needed and the Reader may be configured such that a contactless transaction starts automatically after the previous transaction has completed. This configuration parameter is referred to as “Autorun” and it can have value “Yes” or “No”:
  - If the value of Autorun is “No”, then the Reader activates the field and starts polling for a card upon receipt of the ACT Signal.
  - If the value of Autorun is “Yes”, then the Reader attempts a transaction as soon as the previous transaction is completed and the Card is removed from the field. The transaction starts when a Card is detected in the field.
- The OUT Signal indicates the outcome of the transaction. It contains a subset of the Outcome from the Kernel. The notions of Outcome and the *Outcome Parameter Set* are described in [EMV Book A]. From the *Outcome Parameter Set*, the relevant information for the Terminal is the following:

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 47

- The status of the transaction (Approved, Online Request, Declined, or End Application)
- The CVM option to be applied by the Terminal (Online PIN, Confirmation Code Verified, Obtain Signature, No CVM, or N/A)
- The need for printing a receipt (Yes or N/A)
- The presence of a data record used for authorization and/or clearing (Yes or No)
- The presence of discretionary data (Yes or No)

2.2.2 More Complex Transaction

Figure 2.3 show only the basic service that a Terminal can request from a Reader and the two key Signals that go with it. In reality, the list of services can be more elaborate and Table 2.3 provides a more comprehensive (but not necessarily exhaustive) list. For each of the services, a corresponding Signal is indicated in the column on the right.

Table 2.3—Terminal-Reader Service Requests

Terminal-to-Reader Interaction	Corresponding Signal
Update the Reader's TLV Database	UPD(ate)
Query the Reader's TLV Database	QUERY
Start a transaction	ACT(ivate)
Stop a transaction	STOP
Abort a transaction in case of error or anomaly	ABORT
Display a message	MSG
Provide data needed for a transaction in progress and indicate to the Reader to continue processing the transaction or request additional data from the Reader	DET

The UPD and the QUERY Signal include a mechanism to uniquely identify the database being accessed, as the Reader may have several TLV datasets for managing different Kernels, different AIDs, and different transaction types. One way of doing this is by including a database identifier.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 48

When relevant, the Reader provides data back to the Terminal or simply acknowledges the Signal. For each Signal containing a service request, the corresponding Signal – if there is one – is indicated in Table 2.4.

Table 2.4—Responses from the Reader

Terminal Signal	Corresponding Reader Signal	Comment
UPD	None	
QUERY	QUERY_REPLY	Contains the TLV encoded data object requested.
ACT	OUT	Contains the result of the transaction, including the transaction status, data record, and potentially discretionary data.
STOP	STOP_ACK	
ABORT	None	May trigger the OUT Signal linked to the ACT Signal
MSG	None	
DET	None	May trigger a subsequent DEK Signal

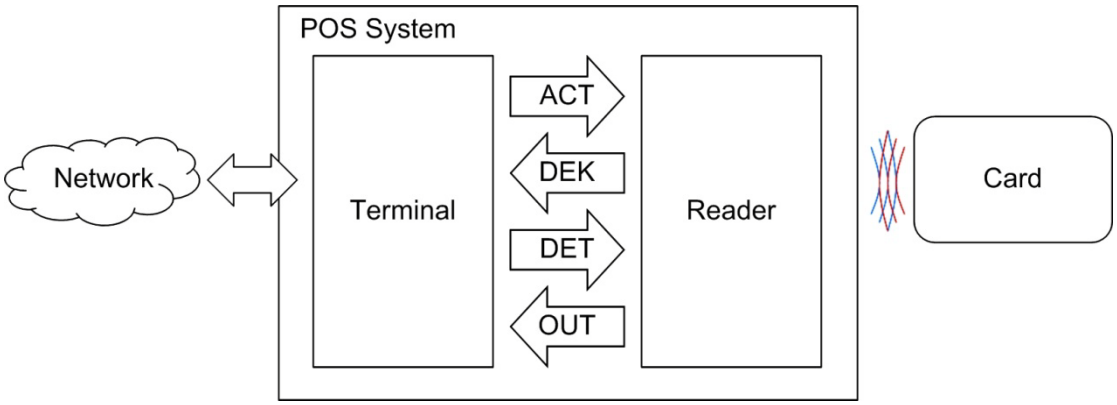
More complex transactions, for example transactions involving data storage, may use the Data Exchange (DE) mechanism as a flexible means of exchanging information between the Terminal and the Reader. A Data Exchange Signal sent by the Reader is referred to as DEK (= Data Exchange Kernel); a Data Exchange Signal from the Terminal is referred to as DET (= Data Exchange Terminal).

This version of the specification focuses on configurations that require no more than one DEK and one DET Signal between Terminal and Reader, as illustrated in Figure 2.4.

Annex B contains some use cases of what can be supported using a single DEK/DET exchange.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 49

Figure 2.4—Complex Transaction



Using the Data Exchange mechanism, the Reader (and the Kernel in particular) can request a service from the Terminal (e.g. if extra data are needed to complete a transaction) by sending a Data Exchange from Reader (DEK) Signal. If the Terminal is able to service the request, it returns a Data Exchange from Terminal (DET) Signal with the requested data.

The DEK Signal has to identify the database being used and needs a means of managing the session. One means of doing so is to use a database identifier and a session identifier:

- Including a database identifier gives the semantic meaning to the tags as the meaning of tags can vary with the Kernel and Kernel database that is used for a particular transaction.
- A session identifier ensures that each DET Signal refers back to the DEK Signal that initiated the session. The session identifier can be managed as part of the underlying communications methods used by an implementation.

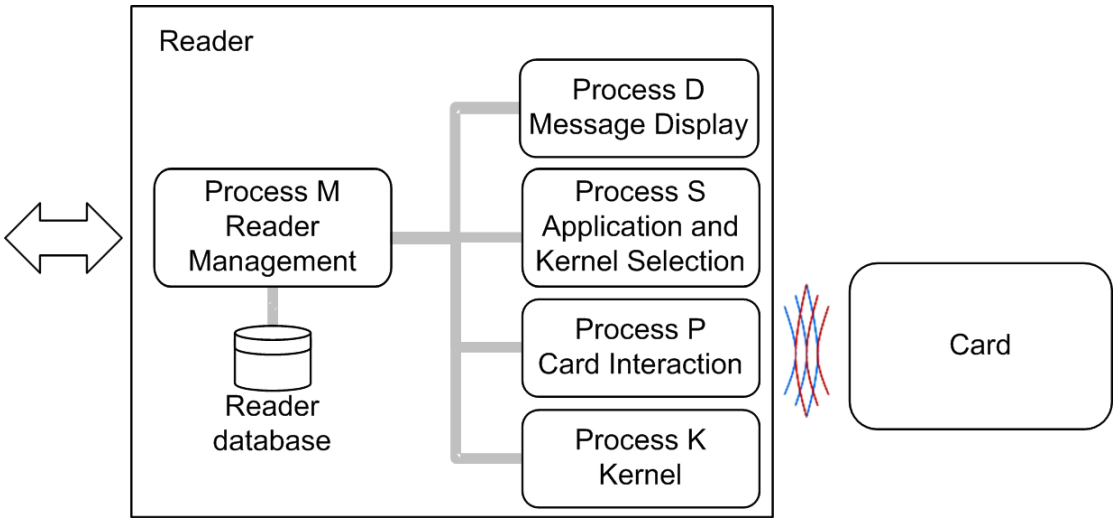
For similar reasons, the Terminal should include both the database identifier and the session identifier or their equivalent in the DET Signal so that the Reader (and the Kernel in particular) can check that the database identifier and session identifier in the DET Signal match those sent in the DEK Signal and ignore the DET Signal if this is not the case.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 50

2.3 Reader Processes

As illustrated in Figure 2.5, the Reader is modelled as a set of Processes and each Process runs independently of the other processes. The role of the Reader database is explained in section 2.4.

Figure 2.5—Reader Logical Architecture



The different processes are listed in Table 2.5.

Table 2.5—Reader Processes

Process	Responsibility
Process P(CD)	Management of the contactless interface
Process D(isplay)	Management of the user interface
Process S(election)	Selection of the Card Application and Kernel
Process K(ernel)	Interaction with the Card once the application has been selected, covering the payment and data storage transaction flow specific to Kernel 2
Process M(ain)	Overall control and sequencing of the different processes. As part of this role, it is also responsible for the configuration and activation of the Kernel and the processing of its outcome. Process M is also responsible for initiating the housekeeping within the Kernel.

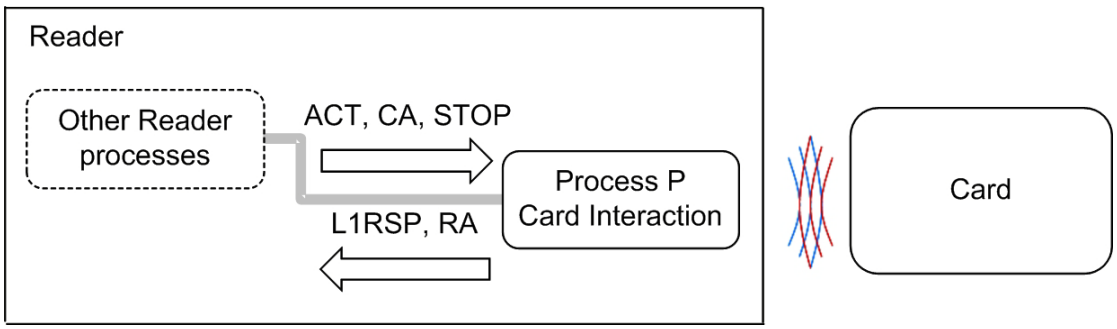
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 51

The remainder of this section introduces the functionality and configuration of the different processes.

2.3.1 Process P

Process P implements the functionality described in [EMV Book D] and [ISO 7816-4] and manages the access to the Card as illustrated in Figure 2.6.

Figure 2.6—Process P



Process P provides the services listed in Table 2.6 to the other processes. The column on the right indicates the corresponding Signal to call the service. Process P may require a different set of configuration data (in the ACT Signal) to select the polling loop, if more than one polling loop is supported.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 52

Table 2.6—Services from Process P

Services	Corresponding Signal
Generate a reset before putting on the field and then start the main loop as described in [EMV Book D]. The process then executes the main loop until one or more Cards are found and notifies the requester of the result.	ACT
Send a C-APDU to the Card and return either an R-APDU or an error indication. The parameter to the Signal is the command to be sent to the Card	CA(C-APDU)
Manage the card removal, either by removing the field immediately or going through the removal sequence with or without a message prompt to the customer. Unless instructed to remove the field immediately, report back when the Card has been removed. The different options are listed below:	
• Remove the field immediately	STOP(Abort)
• Perform card removal as described in [EMV Book D] and indicate when the Card has been removed.	STOP(CloseSession)
• Perform card removal as described in [EMV Book D], request the cardholder to remove the Card if it is still in the field, and indicate when the Card has been removed.	STOP(CloseSessionCardCheck)

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 53

Process P responds to the service requests as indicated in Table 2.7.

Table 2.7—Responses from Process P

Signal In	Signal Out	Comment
ACT	L1RSP(code)	L1 response, with code as one of the following: • Collision detected, if more than one Card has been found • Card detected, if a single Card has been found
CA	RA(R-APDU)	If there is no L1 error, the RA Signal contains the R-APDU sent back in response to a C-APDU.
	L1RSP(code)	If there is an L1 error, L1RSP is returned with code as one of the following: • Error – Timeout; an L1 timeout has occurred • Error – Protocol; an L1 protocol error has occurred • Error – Transmission; any other error
STOP	L1RSP(code)	L1 response, with code as “Card removed”, where the STOP was one of the CloseSession options listed in Table 2.6.

As can be seen in Table 2.7, the functionality described in [EMV Book D] is supported through the Signals ACT, STOP, and L1RSP; the [ISO 7816-4] protocol is supported through the Signals CA and RA.

Activation and closure of the card communications is performed by Process M and is done by sending of the Signals ACT and STOP respectively:

- The ACT Signal causes Process P to put the field on, and start the polling sequence and the card activation as described in [EMV Book A]. If the field was already on when the ACT Signal was received, it is reset first and any communication that was in progress is terminated. Once the field is on again, Process P continues to search for a Card until one or more are found, unless stopped by a STOP (or another ACT) Signal.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 54

- The STOP Signal may have one of the following as a parameter: “Abort”, “CloseSession”, or “CloseSessionCardCheck”:
  - “Abort” makes Process P drop the field and stop current processing.
  - “CloseSession” starts the removal sequence and returns a Signal L1RSP(Card Removed) when the Card has been removed.
  - “CloseSessionCardCheck” includes a request to check for Card presence. If the Card is still present, then it causes a “Please Remove Card” message to be displayed as part of the removal sequence and returns L1RSP(Card Removed) when the Card has been removed. If the Card has been removed already, then no message is displayed and an L1RSP(Card Removed) is returned immediately.

Process P sends the C-APDU included in the CA Signal to the Card and responds with either:

- an RA Signal containing the R-APDU or SW12 returned by the Card, or
- an L1RSP Signal that includes an L1 event such as a timeout, transmission error, or protocol error.

As part of processing L1 events, Process P hides some of the low level processing from the other processes by adding context to the low level information. A timeout in the half-duplex protocol is reported (in an L1RSP Signal) as an error, i.e. “Error-timeout”, but a timeout that occurs after the removal sequence has been initiated is reported as “Card removed”.

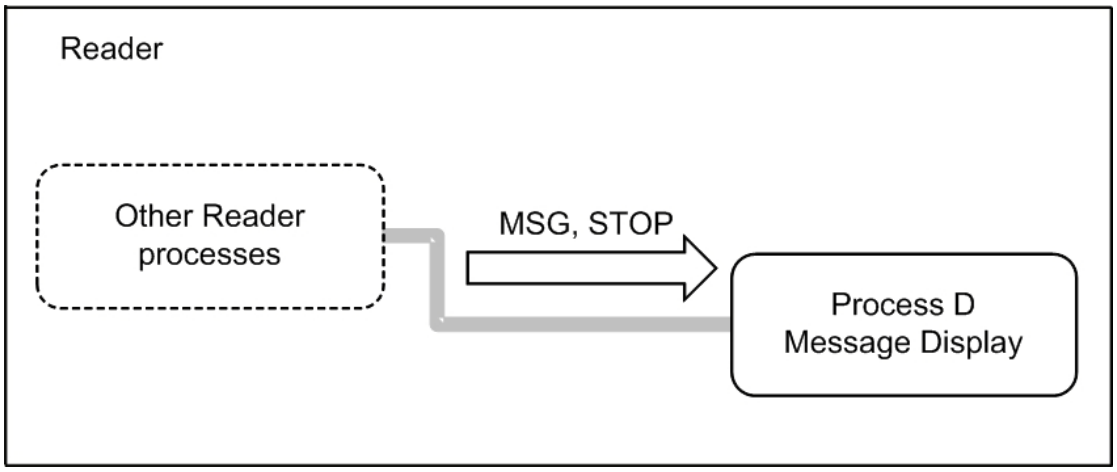
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 55

2.3.2 Process D

Process D manages the User Interface Requests as defined in [EMV Book A] and displays a message and/or a status.

As illustrated in Figure 2.7, a MSG Signal is used as a carrier of the *User Interface Request Data*. Process D may receive MSG Signals from any other Process.

Figure 2.7—Process D



The STOP Signal clears the display immediately and flushes all pending messages.

The MSG and STOP Signals are not acknowledged.

The *User Interface Request Data* can include a message identifier, a status, a hold time, a language preference, and a balance or amount to be displayed.

For more information on the *User Interface Request Data*, please refer to section 7.1 of [EMV Book A].

For displaying messages and/or indicating status, Process D needs the following configuration data:

- default language
- the currency symbol to display for each currency code and the number of minor units for that currency code
- a number of message strings in the default language and potentially other languages
- a number of status identifiers (and the corresponding audio and LED signals)

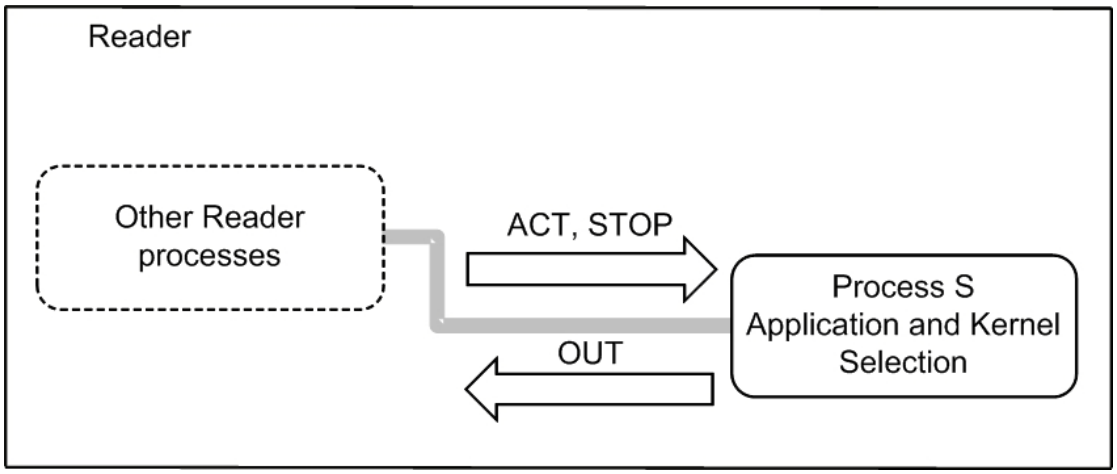
The status identifiers and message identifiers are defined in section 9.2 and section 9.4 respectively of [EMV Book A].

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 56

2.3.3 Process S

Process S manages the application and Kernel selection as described in [EMV Book B]. Upon activation, it returns the selected application and Kernel (in the form of the AID and the Kernel ID respectively) in an OUT Signal, as illustrated in Figure 2.8.

Figure 2.8—Process S



Process S provides the services listed in Table 2.8, with the corresponding Signal to call each service in the right column. For each transaction, Process S is initialized (by Process M) with a list of Combinations {AID – *Kernel ID*}.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 57

Table 2.8—Services from Process S

Services	Corresponding Signal
Build the candidate list (by sending a SELECT PPSE), sort the entries by priority, and select the application with the highest priority from this list (by sending a SELECT AID).	ACT(A) or ACT(B)[1]
Remove the top level entry from the candidate list and, if there is still an eligible entry in the candidate list (i.e. the candidate list is not empty), select the (new) top level entry (by sending a SELECT AID).	ACT(C)[1]
Stop processing.	STOP

Process S responds to the service requests as indicated in Table 2.9.

Table 2.9—Responses from Process S

Signal In	Signal Out	Comment
ACT	OUT	Includes the selected Combination {AID – Kernel ID}, the *File Control Information Template* of the selected *DF Name*, and the SW12 returned by the Card.
STOP	OUT	

[1] The parameters A, B, and C refer to Start A, Start B, and Start C in [EMV Book B], Chapter 3; as Kernel 2 does not use the results of the pre-processing, Start A and Start B – or ACT(A) and ACT(B) – are equivalent.

Kernel 2 does not use Start D.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 58

Some features from [EMV Book B] are not relevant for Kernel 2.

Kernel 2 does not use the results of pre-processing as described in Chapter 3 of [EMV Book B]. The checks of the *Amount, Authorized (Numeric)* against the different limits[2] are delegated to Kernel 2. Therefore, the following two points should be observed:

- For every reader Combination {AID – Kernel ID} with Kernel ID indicating Kernel 2, Entry Point Configuration Data, as defined in Table 5.2 of [EMV Book A], must not be present.
- As a result, Entry Point Pre-Processing Indicators as described in Table 5.3 of [EMV Book A] contain no meaningful information and shall not be part of the Kernel database. In particular the copy of TTQ (see Table 5-3 and Table 5-4 of [EMV Book A]) shall not be part of the Kernel database as tag '9F66' has a different meaning for Kernel 2. For more information on the Kernel database, see section 3.3.

As a side effect, AIDs running on Kernel 2 may be included in the candidate list and be selected anticipating a high value transaction (i.e. above the *Reader CVM Required Limit*) while the cardholder device only allows low value transactions (i.e. below or equal to the *Reader CVM Required Limit*). This condition is picked up by Kernel 2, which then requests the next AID from the candidate list to be selected by means of a “Select Next” Outcome.

Table 2.10 gives the *File Control Information Template* expected in response to a successful selection of a Card application matching Kernel 2. It contains application-specific information such as *Application Label*, *Application Preferred Name*, etc. and can contain payment system tags such as *Third Party Data*.

[2] *Reader Contactless Transaction Limit*, *Reader CVM Required Limit,* and *Terminal Floor Limit*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 59

Table 2.10—Select Response Message Data Field of a Card Application

Tag	Value				Presence
'6F'	*File Control Information Template*				M
	'84'	*DF Name* (AID)			M
	'A5'	*File Control Information Proprietary Template*			M[3]
		'50'	*Application Label*		O
		'87'	*Application Priority Indicator*		O
		'5F2D'	*Language Preference*		O
		'9F38'	*PDOL*		O
		'9F11'	*Issuer Code Table Index*		O
		'9F12'	*Application Preferred Name*		O
		'BF0C'	*File Control Information Issuer Discretionary Data*		O
			'9F6E'	*Third Party Data*	O
			'XXXX'	One or more additional data objects from application provider, Issuer, or ICC supplier	O

The expected Status Words returned by the Card application for the SELECT command are listed in Table 2.11.

Table 2.11—Status Bytes for Select Command

SW1	SW2	Meaning
'62'	'83'	Selected file invalidated
'67'	'00'	Wrong length
'6A'	'81'	Function not supported
'6A'	'82'	File not found
'6A'	'86'	Incorrect parameters P1-P2
'90'	'00'	Normal processing

[3] The *File Control Information Proprietary Template* may be empty. In this case the length must be set to zero.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 60

2.3.4 Process K

The Reader may support multiple Kernels but only one Kernel will execute at a time. The Kernel that is activated depends on the information returned by Process S, which may in turn depend on data retrieved from the Card.

For each transaction, Process K is initialized with a Kernel-specific dataset. Within the different available datasets, the value of the data objects may vary depending on the selected AID and the transaction type. More information on the initialization of the Kernel-specific dataset is provided in section 2.4.

The database for each Kernel can be different and the data items are specific to the Kernel; a payment system or private tag can have a different meaning for different Kernels.

Once the Kernel is selected and configured, it executes as Process K. Using the services of Process P as an intermediary, Process K manages the interaction with the Card application beyond application selection. Upon completion, Process K sends its results to Process M in an OUT Signal and then terminates.

For the remainder of the document, it is assumed that Kernel 2 is selected. More detail on the configuration and initialization of Kernel 2 is provided in section 3.2.

As part of its interaction with the Card, Kernel 2:

- checks the compatibility between the Kernel settings and the Card settings; these checks include both business (for example transaction type, domestic or international acceptance) and technical (for example versioning) aspects,
- reads and writes the necessary payment and non-payment related data,
- determines the need for cardholder verification and the method to be used,
- performs risk management, resulting in the decision to approve/decline the transaction offline or seek online authorization,
- requests messages to be displayed depending on the details of the transaction,
- authenticates data, if and when relevant,
- informs Process M of the transaction outcome through the OUT Signal.

From the viewpoint of the Reader and depending on the implementation and configuration options chosen, Kernel 2 can provide three services (see Figure 2.9):

- Through its interaction with the Card, it creates a transaction record for authorization and/or clearing.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 61

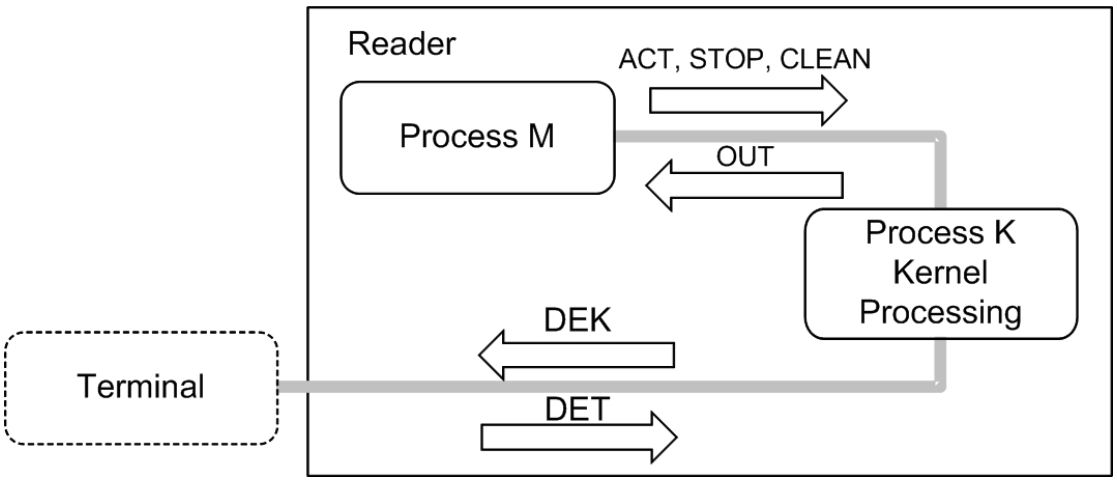
- It performs house-keeping by removing torn transactions from the Torn Transaction Log that have aged off without having been recovered. The Torn Transaction Log is the repository in which the Kernel stores information on torn transactions. More information on torn transactions and the Torn Transaction Log is provided in section 3.7.
- It can interact with the Terminal directly for Data Exchange.

In addition, the Kernel may be instructed to cancel a transaction in progress.

Seen from the Terminal (and again depending on the implementation and configuration options chosen), Kernel 2 allows reading and writing data from and to the Card.

Figure 2.9 illustrates the different services provided by Kernel 2 and separates the Signals exchanged between the Kernel and the other Reader processes from the Signals exchanged with the Terminal.

Figure 2.9—Process K



The different services are listed in Table 2.12, with the corresponding Signal to call the service indicated in the right column. Only Process M or the Terminal request these services from Process K.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 62

Table 2.12—Services from Process K

Services	Corresponding Signal
Return an authorization or clearing record.	ACT(Data) As a minimum, "Data" includes the *File Control Information Template* received from the Card in the response to the SELECT command.
Stop processing	STOP
Clean up the Torn Transaction Log by removing torn transaction that were not recovered and that have been aged off the log.	CLEAN
Return data from the Kernel database or from the Card. Write data to the Kernel database or to the Card.	DET(Data)

Process K responds to the incoming service request with an outgoing Signal as described in Table 2.13.

The CLEAN Signal indicates to the Kernel that housekeeping must be performed. As a result of the housekeeping, aged-off transactions are sent to the Terminal.

The CLEAN Signal is always acknowledged with one or more OUT Signals. Each OUT Signal, except for the last, includes in the *Discretionary Data* a torn transaction that was aged off the log – if there is any. For the last OUT Signal in response to the CLEAN Signal, the *Discretionary Data* is always empty to indicate the sequence of OUT Signals is finished.

The situation for the DET Signal is somewhat different. Within a transaction, the Terminal can only send one or more DET Signals after receiving a DEK Signal[4]. So a DET Signal is as much a response to a DEK Signal as it is a request to the Kernel.

The DEK Signal is sent only if the Kernel has data for the Terminal or needs data from the Terminal. The DEK and DET Signal are exchanged as part of the Data Exchange mechanism.

[4] As it needs to have received (an equivalent of) the database identifier and the session identifier

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 63

Table 2.13—Responses from Process K

Signal In	Signal Out	Comment
ACT	OUT	The OUT Signal includes • the Outcome, including the *Outcome Parameter Set* • *Data Record* – if any • *Discretionary Data* • *User Interface Request Data* – if any
STOP	OUT	
CLEAN	OUT	Includes the aged off transactions in the *Discretionary Data*, if there are any.
DET	DEK or n/a	The DEK Signal can be used to request additional data to be provided in a subsequent DET Signal, as well as to provide data that was requested via a configuration setting or a previous DET Signal. The DEK Signal contains • the *Data Needed* data object, which is the list of tags of data items that the Kernel needs from the Terminal • the *Data To Send* data object, which is the list of tags with data values that the Terminal has requested
n/a	DEK	

The list of Outcomes and the corresponding *Outcome Parameter Set* is defined in [EMV Book A]. The Kernel 2 specific instantiation of the Outcomes and the corresponding *Outcome Parameter Set* are defined in the data dictionary (Annex A).

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 64

2.3.5 Process M

Process M is responsible for coordinating the other processes. Process M has two different roles:

- It coordinates the processes to perform a transaction.
- It gives Process K the opportunity to perform housekeeping on a regular basis when it is not performing a transaction.

  The housekeeping sequence is initiated in Kernel 2 by a CLEAN Signal (instead of an ACT Signal) immediately after start-up. The configuration of Kernel 2 is not relevant during housekeeping. If the OUT Signal from the Kernel includes any torn transactions that were aged off the log, then Process M sends these to the Terminal in an OUT Signal. The Terminal can then log these transactions for customer care – as customers may complain that their card was debited and that they did not receive the service. The information can also be used to build statistics and monitor suspicious transactions, where a torn transaction was provoked intentionally with the intention of committing fraud and the Card was then not represented.

The remainder of this section focuses on the coordination that is needed to perform a transaction. The overall process is illustrated in Figure 2.10:

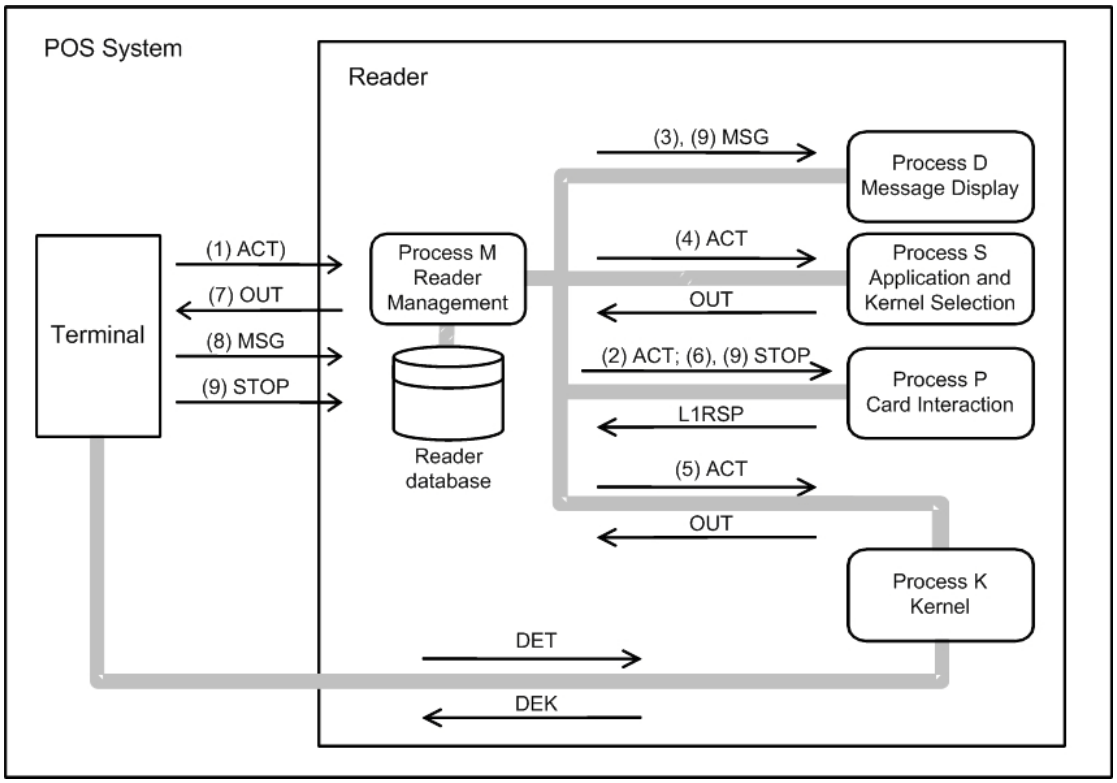
1. Process M receives the ACT Signal from the Terminal.
2. Process M starts Process P by sending it an ACT Signal to start the main loop as described in [EMV Book D].
3. Process M requests Process D to display the READY message (through a MSG Signal).
4. Upon receipt of the Signal L1RSP(Card detected) from Process P, Process M activates Process S by sending it an ACT(A or B) Signal, to indicate that this is the first attempt at the transaction. When Process S completes successfully, it responds with an OUT Signal with the selected Combination {AID – Kernel ID}, the *File Control Information Template* of the selected *DF Name*, and the SW12 returned by the Card.
5. Based on this information, Process M then configures Process K for the specific *Transaction Type* and AID, using a Kernel-specific dataset, and sends it an ACT Signal containing transactional data (such as the *Amount, Authorized (Numeric)* and the *File Control Information Template* received in the response to the SELECT command). When Process K completes, it returns an OUT Signal to Process M, including the *Outcome Parameter Set*, *Discretionary Data*, and *Data Record*, if any.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 65

6. Process M analyzes the 'Status' in *Outcome Parameter Set* and executes the instructions encoded in the other fields of the *Outcome Parameter Set*. As required, Process M instructs Process P with the Signal STOP(CloseSession) to perform the removal sequence. It may also use a STOP(CloseSessionCardCheck) Signal to prompt the cardholder to remove the Card if it is still in the field. Alternatively it may send an ACT signal to Process S to select the next application on the Card.
7. Process M passes a subset of the *Outcome Parameter Set*, the *Data Record*, and the *Discretionary Data* to the Terminal in the OUT Signal.
8. If the transaction is processed online, the Reader should receive a MSG Signal from the Terminal to indicate whether the transaction was approved or declined.
9. Optionally, upon receipt of the STOP Signal, Process M ensures that the Card is removed from the Reader. It sends a STOP(CloseSessionCardCheck) Signal to Process P. When Process P returns an L1RSP(Card Removed) Signal, Process M acknowledges the STOP Signal from the Terminal by sending it a STOP_ACK Signal.
10. If the Reader is configured in Autorun mode, Process M then reactivates the polling sequence (through an ACT Signal to Process P) and displays the READY message by going back to step 2 above. Alternatively, it displays the IDLE message by signalling Process D and goes dormant until it receives an ACT Signal again (see step 1).

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 66

Figure 2.10—Process M



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 67

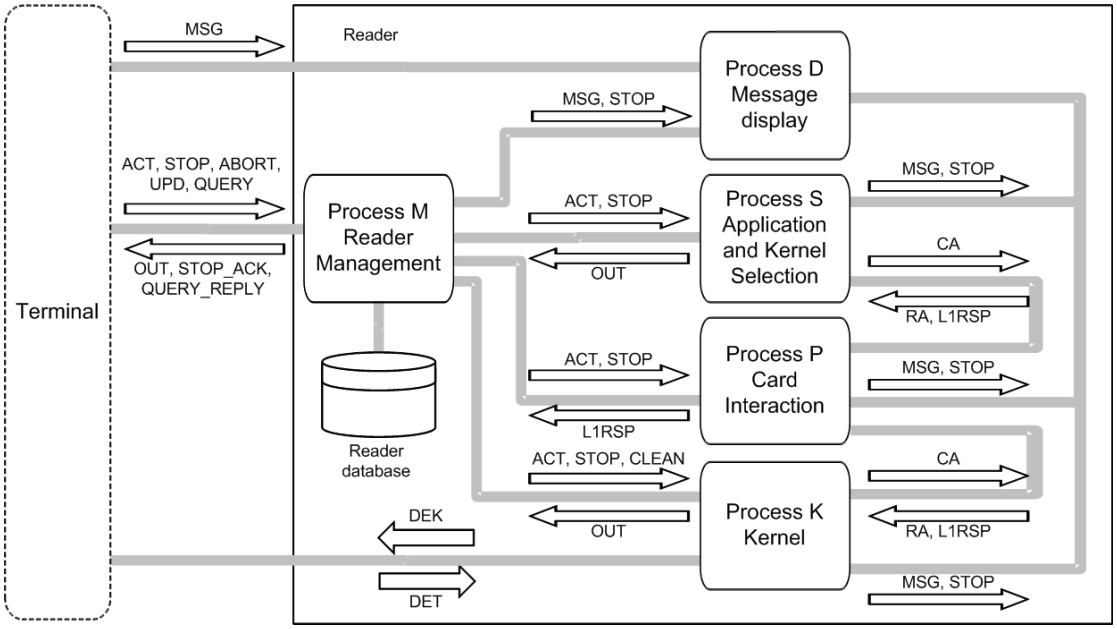
2.3.6 Inter-Process Communication

Not illustrated in Figure 2.10 is the communication between the different processes. As an example:

- Through CA Signals, Process S and Process K request Process P to pass commands (C-APDUs) to the Card and get the Card response (R-APDU) back in an RA Signal. If no response is received from the Card or if the response contains an error, Process P returns an L1RSP Signal, with an indication of the error.
- Through a MSG Signal, Process K requests Process D to update the display.

The inter-process communication is shown on the right hand side of Figure 2.11.

Figure 2.11—Inter-Process Communication



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 68

2.4 The Reader Database

As indicated in Figure 2.5 on page 29, the Reader maintains a database that is divided into datasets.

A dataset can contain either persistent data or transient data:

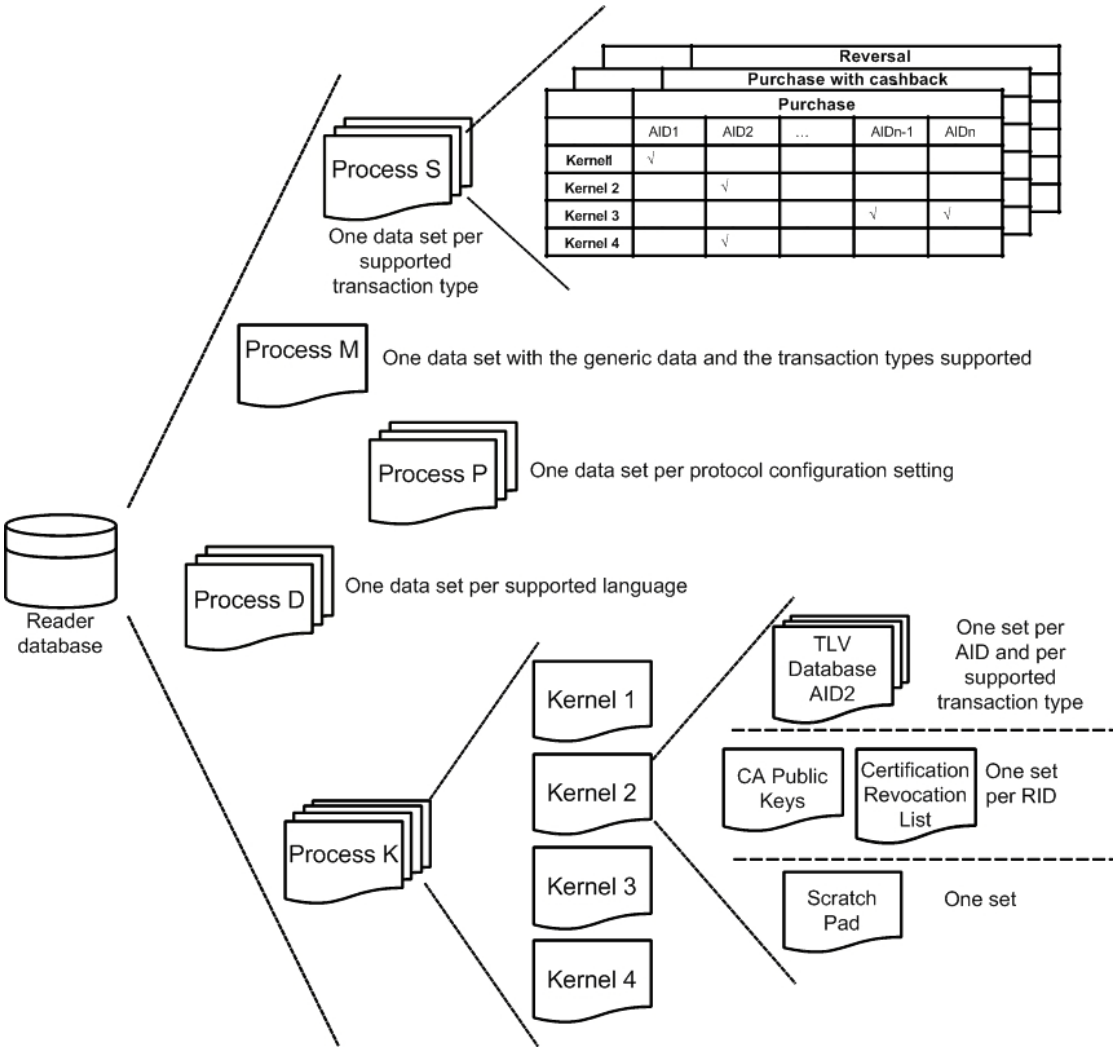
- For a dataset that contains persistent data, its content persists over several transactions.
- A dataset that contains transient data is created at the beginning of a transaction as a copy of a dataset with persistent data and populated with transaction-specific data. Its content
  - is used to initialize one of the processes,
  - can be updated as part of transaction processing by the process or as a result of an ACT or DET[5] Signal,
  - does not persist beyond the transaction in progress.

An overview of the different persistent datasets is given in Figure 2.12, with additional details in Table 2.14.

[5] Only for Process K

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 69

Figure 2.12—Reader Database – Persistent Datasets



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 70

Table 2.14—Reader Databases

Process	Persistent	Transient
Process M	One dataset, including generic data and the different transaction types supported. Examples of generic data are *Interface Device Serial Number*, *Terminal Country Code*, *Transaction Currency Code*, and *Transaction Currency Exponent.* Transaction types can be purchase, purchase with cashback, or refund.	
Process P	One or more datasets, one for each protocol configuration setting. Each dataset contains (part of) the configuration settings as defined in Annex A of [EMV Book D].	A copy of one of the datasets, once the polling loop has been decided.
Process D	Multiple datasets for Process D, one for each supported language. Each dataset contains the message strings behind the message identifiers.	A copy of one of the datasets, once the language has been selected.
Process S	Multiple datasets for Process S, one dataset per transaction type. Each dataset contains a list of Combinations {AID – *Kernel ID*} – see Table 2.15.	A copy of the list of Combinations relevant for the selected transaction type.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 71

Process	Persistent	Transient
Process K	Multiple Kernel-specific datasets for Process K. Each Kernel-specific dataset includes different subsets. For Kernel 2, see Table 2.16.	A copy of (one or more subsets of) one of the persistent Kernel-specific datasets, relevant for the selected transaction type and AID. This copy, in combination with one or more persistent data subsets, then constitutes the Kernel database. More information on the Kernel database is provided in section 3.3.

For Process S, a persistent dataset with the list of Combinations relevant for a specific transaction type can be represented as in Table 2.15. For this particular example, the list of Combinations would be: {AID1 – Kernel 1}, {AID2 – Kernel 2}, {AID2 – Kernel 4}, …, {AID*n*-1 – Kernel 3}, and {AID*n* – Kernel 3}.

Table 2.15—Persistent Dataset Process S (per Transaction Type)

	Transaction Type				
	AID1	**AID2**	**…**	**AIDn-1**	**AIDn**
Kernel 1	√				
Kernel 2		√			
Kernel 3				√	√
Kernel 4		√			

For each entry marked '√' in Table 2.15 (and per *Transaction Type*), there is a Kernel-specific persistent dataset with values that differ per AID and *Transaction Type*.

For Kernel 2, the persistent dataset consists of the subsets given in Table 2.16. Updates to the datasets are exceptional and, except for the scratch pad, happen outside transaction processing.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 72

Table 2.16—Persistent Dataset Kernel 2

Subset	Purpose
The TLV Database	Contains the TLV-encoded data objects relevant to a transaction. The values of the TLV-encoded data objects do not vary per transaction.
The list of CA public keys	Information linked to the CA public keys, including the index, modulus, and exponent. CA public keys can be shared between AIDs that have the same RID and sharing can be done across Kernels. The Reader should be able to store the information for at least six keys per RID.
The Certification Revocation List	A list of Issuer Public Key Certificates that payment systems have revoked for each RID supported by the Kernel. Note that as for the list of CA public keys, entries in the Certification Revocation List may be shared between Kernels where Kernels support the same RID.
The Phone Message Table	Defines the message and status identifiers as a function of the *POS Cardholder Interaction Information*.
The scratch pad	This piece of memory can be used by the Kernel to store and retrieve information across different transactions. The organization of this memory is Kernel-specific and the role of Process M is limited to making the memory available to Process K. It does not need to be non-volatile memory (i.e. memory that holds its content without power being applied) and data of the scratch pad may be lost in case of power failure of the Reader. Kernel 2 may use it to store the Torn Transaction Log or to keep track of the number of (consecutive) torn transactions. When used for this purpose, the torn transactions from cards with different AIDs can be grouped in a single Torn Transaction Log.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 73

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

3 Reader Process K — Kernel Processing

3.1 Introduction

This chapter zooms in on the different features of Kernel 2.

Section 3.2 describes the implementation and configuration options of Kernel 2 and how these are indicated in the flow diagrams.

Section 3.3 gives an overview of the Kernel 2 database.

Sections 3.4 through 3.9 provide more details on the functionality of Kernel 2, as summarized in Table 3.1.

Table 3.1—Kernel Functionality

Function	Comment	Section
Transaction modes	The Kernel supports two transaction modes: • Mag-stripe mode, resulting in mag-stripe–like data to be submitted for authorization • EMV mode, resulting in EMV-like data to be submitted for authorization and/or clearing	Section 3.4
Data Exchange	The Kernel uses the Data Exchange mechanism as a means of communicating directly with the Terminal. It allows the Kernel to send tagged data to and request data from the Terminal through the DEK Signal. It also allows the Terminal to exercise a level of control on the Reader through the DET Signal by virtue of its ability to: • update the Kernel database • request tagged data from the Kernel database • have tagged data written to the Card	Section 3.5

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 75

Function	Comment	Section
Data storage	Data storage is an extension of the regular transaction flow such that the Card can be used as a scratch pad or mini data store with simple write and read functionality. Two types of data storage are supported for EMV mode transactions: • Standalone Data Storage (SDS) • Integrated Data Storage (IDS) Data storage does not apply for mag-stripe mode transactions.	Section 3.6
Recovery of torn transactions	The customer may remove the Card from the field of a Reader before the transaction has completed. If the Card is presented again, the Kernel supports a mechanism to retrieve the missing data and provide a data record for authorization and/or clearing. Torn transaction recovery does not apply for mag-stripe mode transactions.	Section 3.7
Mobile CVM	Transactions involving mobile phones are different from standard card transactions as the phone can be used to authenticate the cardholder. For this purpose, the Kernel distinguishes between a cardholder device that delegates the CVM processing to the Terminal and a cardholder device that can perform cardholder verification itself. For the latter, the Kernel applies different limits and it delegates the CVM processing to the cardholder device.	Section 3.8
Card balance reading	The Kernel is capable of recognizing a Card that offers access to its (offline) balance and can read it before the transaction is completed, after the transaction is completed, or both. The results are then made available to the Terminal and put on display.	Section 3.9

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 76

3.2 Kernel Configuration and Implementation Options

Not all the features listed in Table 3.1 have to be present and/or activated in each implementation of Kernel 2. Within this context, it is important to distinguish "configuration options" from "implementation options" (as defined in section 1.5).

Kernel 2 supports two implementation options and five configuration options.

3.2.1 Implementation Options

The two implementation options are listed in Table 3.2.

Table 3.2—Kernel Implementation Options

Implementation Options	Description
EMV mode transaction flow	If the EMV mode transaction flow is not supported, then Kernel 2 only supports the mag-stripe mode transaction flow, resulting in mag-stripe–like data to be submitted for authorization. The following features cannot be supported if the EMV mode transaction flow is not implemented: • Data storage • Recovery of torn transactions • Card balance reading
Data Exchange	If the Data Exchange (DE) mechanism is not supported, then the functionality linked to the DEK and DET Signals is not supported. The Kernel cannot send data to or request data from the Terminal during transaction processing and vice versa for the Terminal. The data storage feature is not supported.

These two implementation options give rise to four possible valid Kernel implementations, supporting the functionality as described in Table 3.3.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 77

Table 3.3—Options and Implementations

	DE not supported	DE supported
EMV mode not supported	Mag-stripe mode "MS"	Mag-stripe mode + DE "DE"
EMV mode supported	Mag-stripe mode + EMV mode "EMV"	Mag-stripe mode + EMV mode + DE "EMV/DE"

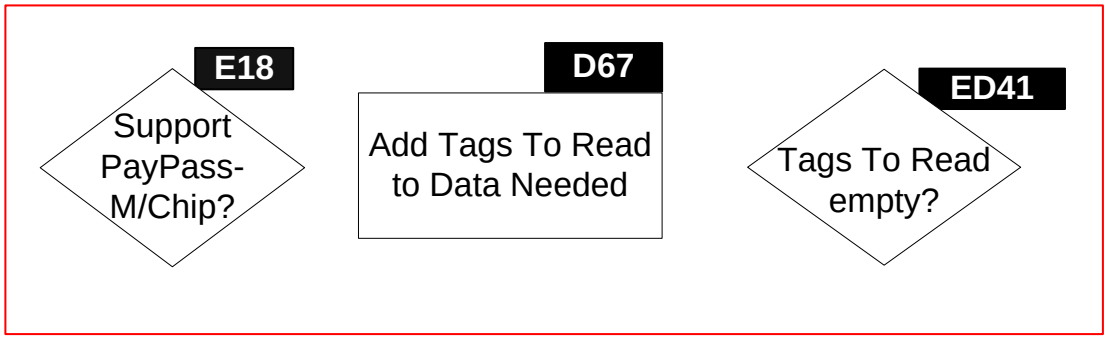
To refer to the two implementation options, the labels "EMV" and "DE" are used to indicate the EMV mode transaction flow and Data Exchange respectively.

To refer to the four implementations, the labels "MS", "EMV", "DE", and "EMV/DE" are used as indicated in Table 3.3. Later in the document "MS" is often replaced by "Always" as the mag-stripe mode is supported by all four implementations.

Using the conventions defined in section 1.6.3:

- In the data dictionary, the Implementations field lists the implementations for which the data object must be present. Data objects that must be implemented for the four possible valid Kernel implementations are labelled "Always" (instead of MS, EMV, DE, EMV/DE).
- In the flow charts, symbols that are specific to the EMV mode transaction flow are labelled with "E" followed by numeric digits. Symbols that are specific to Data Exchange are labelled "D" followed by numeric digits. Symbols that are specific to the support of both the EMV mode transaction flow and Data Exchange are labelled "ED". The three options are illustrated in Figure 3.1.

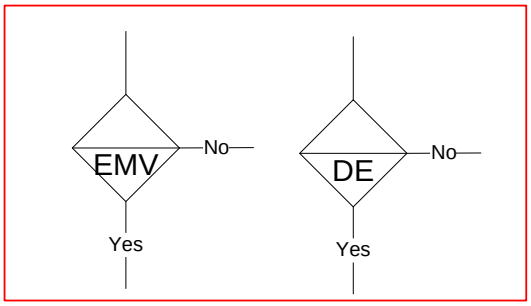
Figure 3.1—Example Implementation Option in Flow Charts – Symbols



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 78

- The diamond that checks on whether the EMV mode transaction flow is implemented gets the label “EMV”. In a similar manner, the diamond checking on the support of Data Exchange gets a label DE. The two options are illustrated in Figure 3.2.

Figure 3.2—Example Implementation Option in Flow Charts – Checks



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 79

3.2.2 Configuration Options

The different configuration options are listed in Table 3.4, as well as the method to activate a particular option. If the condition for activation is not satisfied, the option is de-activated.

Table 3.4—Kernel Configuration Options

Configuration Options	Description	Activation	Precondition
IDS	The Kernel supports IDS.	Through the data object *DS Requested Operator ID* and *DSVN Term* If *DS Requested Operator ID* is present (even with a length of zero) and *DSVN Term* is present with a length different from zero, then IDS is supported.	The EMV mode transaction flow and DE are supported as implementation options.
EMV mode only	The Kernel only supports the EMV mode transaction flow and does not support mag-stripe mode flow.	Through the setting of 'Only EMV mode transactions supported' in *Kernel Configuration*	The EMV mode transaction flow is supported as implementation option.
Mag-stripe mode only	The Reader only supports the mag-stripe mode transaction flow and does not support the EMV mode transaction flow.	Through the setting of 'Only mag-stripe mode transactions supported' in *Kernel Configuration*	

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 80

Configuration Options	Description	Activation	Precondition
Balance reading and display	If the Card indicates support for balance reading, then the Kernel may read the balance before or after the GENERATE AC and send the information to Process D for display.	Through the data objects *Balance Read Before Gen AC* and *Balance Read After Gen AC*. If one or both of these tags is present (with a length of zero), then the specified balance reading is supported.	The EMV mode transaction flow is supported as implementation option.
Torn transaction recovery	The Kernel tracks torn transactions and tries to recover them if transaction recovery is supported by the Card.	Through the number of entries possible in the torn transaction log, indicated by the value of data object *Max Number of Torn Transaction Log Records* If *Max Number of Torn Transaction Log Records* is present and set to a value different from zero, then torn transaction recovery is supported.	The EMV mode transaction flow is supported as implementation option.

All the above configuration options for the Kernel are set at the level of the AID and the transaction type and are part of the TLV Database in the persistent dataset of Kernel 2.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 81

3.3 The Kernel Database

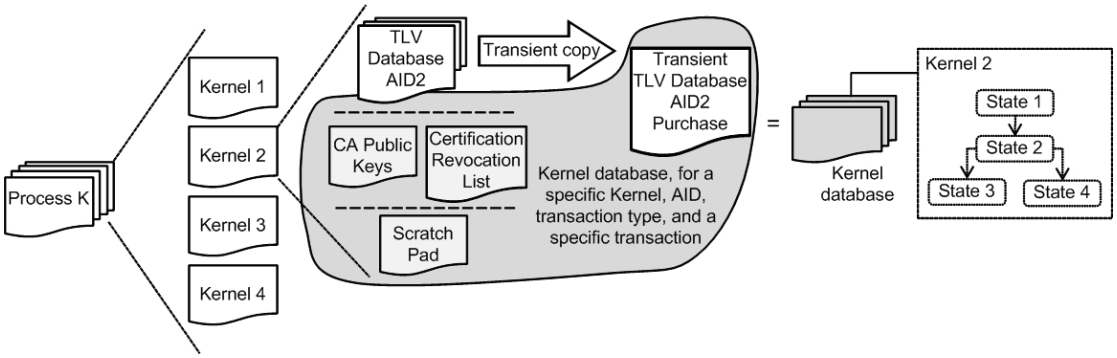
The Kernel database as introduced in section 2.4 is the list of data items used by the Kernel during the processing of a transaction. Part of it may be held in volatile memory as its lifetime is limited to a single transaction.

When the Kernel processing starts, the Kernel database is already initiated with:

- The portion of the persistent dataset of Kernel 2 for a specific AID (or RID) that includes the list of CA public keys, the Certification Revocation List, and the scratch pad (see Table 2.16);
- A transient copy of the TLV Database for a specific AID and transaction type (see Table 2.16). Within the TLV Database, entries may exist with zero length.

Figure 3.3 illustrates how the Kernel database that drives the state machine is constructed from the persistent dataset and a transient copy of the TLV Database.

Figure 3.3—Kernel Database



Note that the Kernel database as it is initialized by Process M does not include internal data objects of the Kernel, such as *CVM Results* or *Terminal Verification Results*. These data objects are initialized by the Kernel itself.

In addition to the Kernel database, the Kernel receives transaction data items in the ACT Signal. These data items originate from the (Terminal) ACT Signal and from the OUT Signal of the application and Kernel selection process (Process S). These data items with their volatile values are added to the database as well.

During transaction processing, the Kernel may receive events from Process M, the Card, and the Terminal. This input, together with the Kernel's progression through the transaction processing, causes further updates to the Kernel database.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 82

While performing a transaction, the Kernel ensures that updates to the Kernel database are done only by the authorized 'source' (origin) of the data item. For this purpose, data items are put in different categories and the category determines the Signal – and therefore source – that can update data objects within a category.

The different categories and corresponding Signals are illustrated in Table 3.5.

Table 3.5—Kernel Database Categories

Data Category	Signal
Terminal sourced data object – configuration data	n/a
Terminal sourced data object – transaction data	DET, ACT
Kernel defined value or internal data object	n/a Value can only be changed as part of Kernel processing
Card sourced data object	RA[6]

[6] The *File Control Information Template* is received in an ACT signal but is treated as an RA as that is how it was delivered to Process S.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 83

3.4 Mag-Stripe Mode and EMV Mode

3.4.1 Overall Transaction Flow

Upon receipt of an ACT Signal, the Kernel initiates the transaction on the Card through a GET PROCESSING OPTIONS command.

Based on the response from the Card, in particular the *Application Interchange Profile*, the Kernel continues with either a mag-stripe mode or an EMV mode transaction. In both cases, the Kernel reads data record(s) from the Card (through one or more READ RECORD commands). Then the Kernel requests the Card to generate a cryptogram, which is then included in the *Data Record*.

Once all the data from the Card, including the cryptogram, are retrieved, the Kernel indicates that the Card can be removed.

The Kernel completes the transaction by preparing the remainder of the *Data Record*, the *Outcome Parameter Set* information, and *Discretionary Data* (as defined in [EMV Book A]). For an EMV mode transaction, the *Data Record* contains EMV-like data; for a mag-stripe mode transaction, it contains mag-stripe–like data. The Kernel returns the above data to the main process (Process M) and this concludes the transaction for the Kernel, which then terminates execution.

The remainder of this section highlights the difference in transaction flow between mag-stripe mode and EMV mode transactions.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

3.4.2 Mag-Stripe Mode

For a mag-stripe mode transaction, after the GET PROCESSING OPTIONS command, the Kernel continues with the following steps:

1. It reads the data records from the Card, containing *Track 1 Data* and *Track 2 Data*, together with instruction on how to populate the discretionary data.
2. It issues the COMPUTE CRYPTOGRAPHIC CHECKSUM command, including *Unpredictable Number (Numeric)* to the Card, requesting the Card to return a CVC3 cryptogram, calculated over *Unpredictable Number (Numeric)*.
3. It populates the *Track 2 Data* with the *Unpredictable Number (Numeric)*, the *Application Transaction Counter*, and *CVC3 (Track2)*.
4. If *Track 1 Data* is present, it populates the *Track 1 Data* with the *Unpredictable Number (Numeric)*, the *Application Transaction Counter*, and *CVC3 (Track1)*.
5. It sets *nUN* equal to the (meaningful) length of the *Unpredictable Number (Numeric)* and populates *Track 2 Data* and (if present) *Track 1 Data* with this value.
6. It requests the transaction to be sent online.

3.4.3 EMV Mode

For an EMV mode transaction, after the GET PROCESSING OPTIONS command, the Kernel continues with the following steps:

1. It determines which form of Offline Data Authentication to perform.
2. It reads the data records of the Card (using READ RECORD commands). If the same transaction involving the same Card is recognized in the Kernel's internal log of torn transactions, then an attempt is made to recover the transaction – see section 3.7.
3. It performs Terminal Risk Management and Terminal Action Analysis, and selects a cardholder verification method for the transaction.
4. It requests an *Application Cryptogram* from the Card by issuing a GENERATE AC command. If a response is not received from the Card, the Kernel considers the transaction as "torn", and stores the transaction details in its internal log of torn transactions, before terminating – see section 3.7.
5. It performs Offline Data Authentication as appropriate.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 85

3.5 Data Exchange

3.5.1 Introduction

Terminal and Kernel can communicate through the Data Exchange mechanism.

The Kernel can send tagged data to and request data from the Terminal through the DEK Signal.

The Terminal can control the Kernel through the DET Signal by virtue of its ability to:

- update the current transaction database of the Kernel
- request tagged data from the Kernel or from the Card
- manage the transaction flow pace by withholding necessary data (so that the Kernel asks for it) or providing these data earlier than needed to avoid delays.

3.5.2 Sending Data

As part of its configuration or through an ACT or DET Signal, the Kernel has a data object (*Tags To Read*) containing the tags (and lengths) of the data objects to be sent to the Terminal. If a tag refers to card data, this data is retrieved through READ RECORD commands – as part of reading the records listed in the *Application File Locator* – or through a GET DATA command[7]. Note that this list excludes the IDS data which is sent automatically if IDS is activated in the Kernel.

When the Kernel has completed the (currently outstanding) requests from the Terminal, it sends the data to the Terminal via a DEK Signal.

The information in the DEK Signal may trigger the Terminal to send another list of data to read (DET Signal). This list is then appended to the original list and may result in another set of GET DATA commands if the request includes tags referring to card data.

The Kernel uses a buffer, called *Tags To Read Yet*, to accumulate the different read requests included in *Tags To Read*.

Data To Send is another buffer, accumulating the multiple data that the Kernel has for the Terminal. It is populated with TLV data retrieved in response to *Tags To Read Yet* processing.

The process continues until all records have been read and there are no more data objects in the list that need to be read using a GET DATA command.

[7] The kernel has a list of data objects that are read using GET DATA; all other data objects are read using READ RECORD commands. Note that no files or records other than those listed in the *Application File Locator* are read.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 86

3.5.3 Requesting Data

If one of the following data objects is present in the Kernel database with the length of the value field set to zero, then the Kernel sends a DEK Signal to request the value of the data object:

- *Tags To Read*
- *Tags To Write Before Gen AC*
- *Tags To Write After Gen AC*
- *Proceed To First Write Flag*

The last three data objects are relevant for data storage and are discussed in section 3.6.

In more general terms, the Kernel applies the following rules for Terminal sourced data objects (as opposed to Kernel and Card sourced data objects):

1. If the Kernel database contains a Terminal sourced data object that has length of zero and if this data object is needed during the transaction, then the Kernel requests this data object in a DEK Signal by including its tag in *Data Needed*.

   The data object can be needed during the transaction for two reasons:

   - The Kernel needs it for its own processing, e.g. *Amount, Authorized (Numeric)*.
   - The Card requests it in a DOL, e.g. *Merchant Customer Data*.

2. If the data object is not present in the Kernel database, it is not requested from the Terminal. This condition applies only if the Kernel does not need this data object for its own processing. When this data object is requested by the Card in a DOL, it is zero filled in the data of the corresponding command.

3. If the data object is present with length different from zero, it is not requested from the Terminal. It is sent to the Card when requested in a DOL and normal padding and truncation rules apply.

By putting a Terminal sourced data object or one or more of the data objects listed above in the database with a zero length, the Terminal deliberately withholds the data so that the Kernel specifically asks for it, thereby giving the Terminal the ability to pace the transaction flow and change the value of transaction data, based on information received during the transaction flow.

As indicated above, the Kernel uses a buffer, called *Data Needed*, to accumulate tags that the Kernel needs from the Terminal. It is populated with TL data. In a similar manner, if IDS is being used, the Kernel uses DEK Signals to request the data that it needs to complete the transaction.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 87

The Terminal may send multiple DET Signals, each DET Signal containing a *Tags To Write Before Gen AC* or *Tags To Write After Gen AC* data object. The Kernel manages these DET Signals through two buffers: *Tags To Write Yet Before Gen AC* and *Tags To Write Yet After Gen AC*. These buffers are used to accumulate the TLV data objects included in *Tags To Write Before Gen AC* tag and *Tags To Write After Gen AC* tag respectively.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 88

3.6 Data Storage

3.6.1 Introduction

Data storage is an extension of the regular EMV mode transaction flow such that the Card can be used as a scratch pad or mini data store with simple write and read functionality. Data storage does not apply for mag-stripe mode transactions.

Two types of data storage are possible: Standalone Data Storage (SDS) or Integrated Data Storage (IDS).

The following characteristics are common to both types of data storage:

1. They rely on the Data Exchange mechanism as described in section 3.5 and without this mechanism, data storage cannot be supported.
2. All data are read from the Card before any data are written. To make sure the reading process is completed and that the Terminal has received all required data, the Kernel checks whether it can move to the writing stage.

   This check is referred to as the "chokepoint" and uses the *Proceed To First Write Flag* data object, as introduced in section 3.5.3.

   The *Proceed To First Write Flag* may take one of the following values:

   - When *Proceed To First Write Flag* is absent, the Kernel can move to the writing phase of the transaction.
   - When *Proceed To First Write Flag* has length zero, the Kernel requests a value for the *Proceed To First Write Flag* from the Terminal. It waits until the Terminal provides this value before moving to the writing phase.
   - When *Proceed To First Write Flag* has value zero, the Kernel waits until the Terminal provides a value different from zero before moving to the writing phase.
   - When *Proceed To First Write Flag* has a value different from zero, the Kernel can move to the writing phase of the transaction.

The Kernel may support one or both data storage methods and is configured accordingly. However, the use of data storage by the Kernel in a given transaction is conditional on the Card's indication of support for data storage. The Card support for SDS and IDS is indicated in the response to the SELECT AID command. The *File Control Information Template* may contain the *Application Capabilities Information* data object which, if present, indicates the support provided for SDS and IDS.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 89

3.6.2 Standalone Data Storage

SDS uses dedicated commands (GET DATA, PUT DATA) for explicit reading and writing of data. It introduces a range of payment system tags ('9F70' to '9F79') for the reading and writing of non-payment data, so that they can be included in *Tags To Read*, *Tags To Write Before Gen AC*, or *Tags To Write After Gen AC* (see section 3.5). The whole range is freely readable using the GET DATA command.

Writing is done in one of two ways:

- Using a PUT DATA command with secure messaging, for tags '9F70' to '9F74'.
- Using a PUT DATA command without secure messaging, for tags '9F75' to '9F79'.

The length of the data is variable. The maximum length is implementation specific, and is between 32 and 192 bytes. If present, the *Application Capabilities Information* from the Card indicates the configuration of the SDS tags. The relevant coding is described in the data dictionary (Annex A).

Writing can be done before and after the GENERATE AC, hence two lists to distinguish between data objects to be written to the Card before and those to be written afterwards. This distinction is indicated by the list names: *Tags To Write Before Gen AC* and *Tags To Write After Gen AC*.

Each list is TLV coded, containing Tag, Length as well as Value of the data to write. The lists may be part of the Kernel configuration, or may be communicated to the Kernel during the transaction using Data Exchanges, via a DET Signal.

Once the Kernel has the go-ahead to move to writing, it may send one or more PUT DATA commands to the Card, each command containing one data object from the first list (*Tags To Write Before Gen AC*) and in the order as they are in this list. Once all data from this first list are sent to the Card, the Kernel sends the GENERATE AC command.

After the GENERATE AC command, the Kernel then repeats this process for the second list (*Tags To Write After Gen AC*).

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 90

3.6.3 Integrated Data Storage

IDS builds the reading and writing functions into existing payment commands (GET PROCESSING OPTIONS and GENERATE AC). The command-response sequence exchanged between the Card and Kernel is therefore unchanged from a normal purchase transaction. It also addresses the security mechanisms of those exchanges.

This section describes the overall transaction flow and the security design.

IDS: Overall Transaction Flow

Support for IDS in the transaction flow can be summarized as follows:

1. Process S selects the application. If the Card supports IDS, this is indicated in the Card's response and the *PDOL* includes the tag of the operator identifier. The Card's response is included in the ACT Signal activating the Kernel, and is therefore part of the current transaction database of the Kernel.
2. The operator's slot is selected through the inclusion of the operator identifier in the GET PROCESSING OPTIONS command data as part of the *PDOL Related Data*.
3. If a slot is currently present for this identifier, the Card returns the contents of the slot in its response to the GET PROCESSING OPTIONS command together with slot management data. If it is not present, the Card indicates whether a new slot is available for allocation to this identifier. As well as the normal *Application Interchange Profile* and *Application File Locator* data objects, the GET PROCESSING OPTIONS response (using Format 2) returns[8], if available, the following:
   - the non-payment data (*DS ODS Card*)
   - the type of data (*DS Slot Management Control*)
   - a hash of the transaction context calculated by the Card when data was written to the Card in a previous transaction (*DS Summary 1*)
   - an indication of which type of data (volatile or permanent) may be stored (*DS Slot Availability*)
4. The information on the slot data is passed to the Terminal (DEK Signal), which can then decide to update the data or allocate a new slot, as appropriate for the particular transaction. The Terminal passes this information to the Kernel (DET Signal) and the Kernel sends the new data to the Card appended to the end of the *CDOL1* data in the GENERATE AC command.

[8] Although not relevant to the reading of the data, note that the GET PROCESSING OPTIONS response also includes a card challenge (*DS Unpredictable Number*). This is part of the IDS security mechanism.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 91

For this purpose, the Card supports a (single) *DSDOL*, applicable for the GENERATE AC command. *DSDOL* is read through the READ RECORD command, in a record present in the *Application File Locator*. The Kernel appends the (non-payment data) data objects listed in the *DSDOL* in the order as indicated in the *DSDOL* and with the lengths as indicated in *DSDOL* (except for the last element which may be shorter). Except for the last tag in *DSDOL*, all tags are handled according to the rules specified in section 5.4 of [EMV Book 3]. The last tag indicated in *DSDOL* is appended with the length defined in the TLV Database and no padding is applied if the length specified in the *DSDOL* entry is greater than the actual length of the data object in the Kernel database.

The data objects that are included in the *DSDOL* tags list are:

- The type of data (*DS ODS Info*)
- The result of a one-way function, to set a new access control (*DS Digest H*)
- The input to a one-way function, to get access control (*DS Input (Card)*)
- The non-payment data envelope (*DS ODS Term*)

5. Including the additional data in the GENERATE AC command may influence the outcome of the transaction and does not automatically result in a data update or a slot allocation. Whether data will be written to the Card and the outcome of the transaction depends on four elements:
   - The type of application cryptogram (i.e. TC, ARQC, or AAC) proposed by the Terminal in the *DS AC Type*
   - The type of application cryptogram resulting from the Kernel (terminal) risk management and action analysis, indicated in *AC Type*
   - The settings in *DS ODS Info For Reader* sent by the Terminal
   - The type of *Application Cryptogram* generated by the Card, as reported in *Cryptogram Information Data* – see step 6

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 92

The algorithm is described below and assumes there is an order amongst the different application cryptograms TC, ARQC, and AAC, with TC being the highest and AAC being the lowest (i.e. TC > ARQC > AAC). The algorithm is as follows:

The Kernel compares its *AC Type* to the Terminal's *DS AC Type*.

- If the Kernel *AC Type* is higher, then the Kernel sets its *AC Type* equal to the *DS AC Type*. For example, if the Terminal requests an ARQC in *DS AC Type* and the Kernel's risk management decision results in a TC in *AC Type*, then the Kernel sets its *AC Type* to ARQC, which is lower.
- If the Kernel *AC Type* is lower, then:
  - If *DS ODS Info For Reader* indicates that the IDS data can be used for *AC Type*, then the Kernel includes the IDS data in the GENERATE AC command data.
  - Otherwise:
    - If *DS ODS Info For Reader* indicates that the transaction may continue without IDS data in the GENERATE AC command data then the Kernel sends the GENERATE AC without IDS data.
    - Otherwise, the Kernel terminates the transaction and returns an OUT Signal.

6. If the IDS data are included in GENERATE AC command data, then the Card may or may not write the data. If the data is written, then the Card confirms to the Kernel that the slot has been allocated and that the new data has been updated. If there is an error with the data or if the type of *Application Cryptogram* generated by the Card is different from that requested by the Kernel, then the Card does not store the data. In any case, the Card response includes an authenticated hash of the transaction context of the initial data read (*DS Summary 2*) as well as a hash of the transaction context of the resulting data (*DS Summary 3*).
7. If the response to the GENERATE AC command indicates that the data were not written, the Kernel checks *DS ODS Info For Reader* on whether the transaction should be continued or not.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 93

IDS: Security Design

The security design is based on the following assumptions and mechanisms.

Assumptions

The service is provided based on the data read (*DS ODS Card*) and is conditional on the data being authentic. If the data cannot be authenticated, then the service will not be provided.

The Terminal has a cryptographic method to add a MAC to the data that it stores in the data written to the Card to ensure that a third party has not tampered with the data.

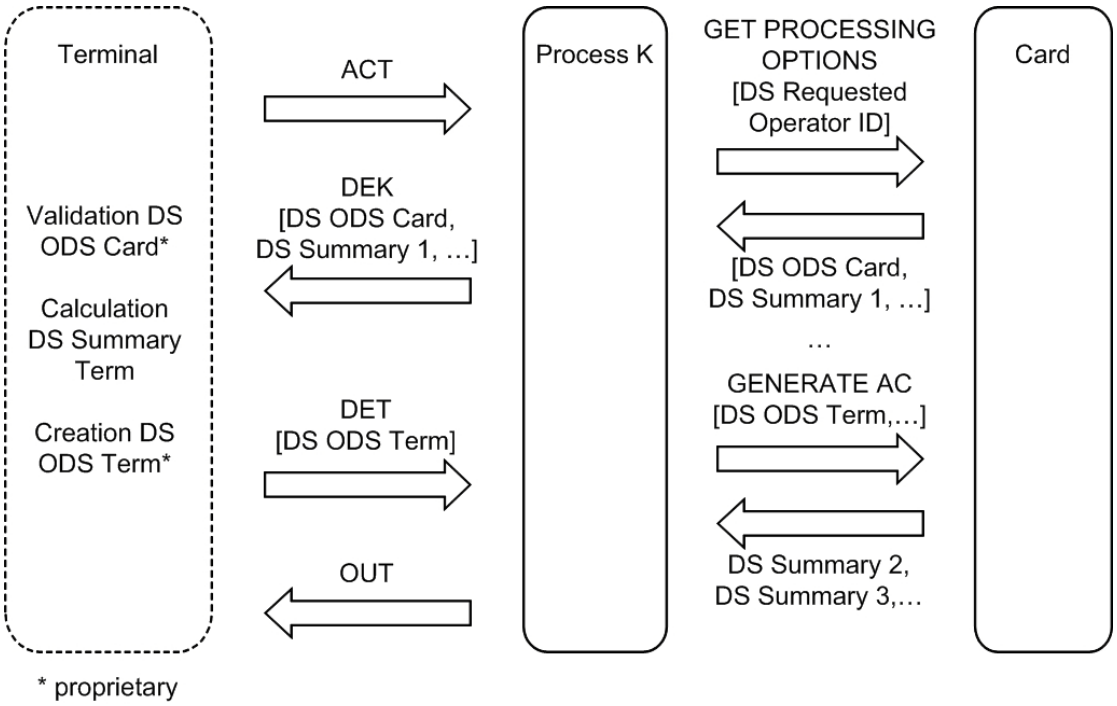
If the Terminal wants to protect the data against skimming and replay, the operator uses the security mechanisms as proposed in this specification.

Mechanisms

The security is built on a combination of the proprietary mechanisms implemented in the Terminal, hashes over the transaction data – called Summaries – and strong offline card authentication using public key cryptography.

The basic principle behind the Summaries is illustrated in Figure 3.4.

Figure 3.4—Summaries – Basic Principle



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 94

There is a Summary for the data read and for the data written.

The Summary is a data item that is:

- Computed independently by both the Card (= *DS Summary 3*) and the Terminal at each write operation
- Computed as a one-way function on the identity of the Card and transaction critical data
- Used by the Terminal as input into the (proprietary) security mechanism for protecting its data (= *DS ODS Term*)
- Returned by the Card to the Terminal next time the data is read (= *DS ODS Card*, *DS Summary 1*)
- Included in the CDA signature of the transaction to authenticate the Summaries

Because *DS Summary 1* is returned outside of the CDA signature (and therefore not authenticated), the Card returns the data object in the CDA signature as well, where it is then referred to as *DS Summary 2*.

DS ODS Card and *DS Summary 1* (as well as other data) are returned by the Card and passed to the Terminal. The Terminal validates the authenticity and integrity of *DS ODS Card*, using a proprietary mechanism in combination with *DS Summary 1*.

Assuming that *DS Summary 1* is authentic (which will be confirmed through *DS Summary 2*), the Terminal calculates a Summary over the new transaction data and updates *DS ODS Card*, which then becomes *DS ODS Term*. *DS ODS Term* is sent to the Kernel, which passes it on to the Card.

If the Card updates the slot data with *DS ODS Term*, it calculates a new Summary, taking the existing Summary as input, and stores this new Summary with the slot data. If for some reason the slot data are not updated, no new Summary is calculated and the Summary stored with the slot data does not change.

The Summary stored with the slot data is returned by the Card as *DS Summary 3*.

For the Kernel it is simple to see whether the slot update was successful or not: If the value of *DS Summary 3* is different from the value of *DS Summary 1* (and hence *DS Summary 2*), then the slot data has been updated.

Wedge attacks are detected as both the Card and Reader independently hash critical data into these Summaries. Both of the Summaries calculated by the Card (*DS Summary 2* and *DS Summary 3*) are included in the CDA signature as part of the ICC Dynamic Data. The Kernel will detect tampering with the communication between Terminal and Card when it compares *DS Summary 2* with *DS Summary 1* and *DS Summary 3* with *DS Summary 2*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 95

Copying and cloning is prevented through inclusion of an authenticated Card identifier (*DS ID*) and a Card challenge (*DS Unpredictable Number*) in the Summary, in combination with the operator's proprietary mechanism for generating a MAC from the data.

For write control, the security is built on a one-way function. At personalization, the Card stores the result of a one-way function over the *DS Input (Card)* data item, which must match the digest that protects the write access to the slot in the Card. Together with the new data, the Terminal provides a new digest (*DS Digest H*) to fit the newly written data (*DS ODS Term*).

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 96

3.7 Torn Transaction Recovery

3.7.1 Introduction

The customer may remove the Card from the field of a Reader before the transaction has completed. The generic term used for this is “tearing”, resulting in a “torn transaction”. In case of a torn transaction, the Kernel invites the cardholder to present the Card again.

If the Card maintains an offline balance (for example if the Card implements a prepaid or preauthorized product), this balance may have been decremented and a second presentment should not decrement the balance again.

In a similar manner, data read from the Card may have been updated and written to the Card. Presenting the Card again should not cause another update to occur.

For this reason, a new mechanism has been specified that allows the data from a torn transaction to be recovered without impacting the counters on the Card or the data written to the Card.

3.7.2 Recovery Mechanism

The principle of transaction recovery is simple; if the Kernel failed to receive a response to a GENERATE AC command, it may ask for it again with the RECOVER AC command. If the Card had not advanced so far in its transaction as to update its counters and create the response, then it responds by telling the Kernel that it cannot recover (there is nothing to recover) and a new transaction may safely be performed. This new transaction does not require starting the complete transaction again; the Kernel may continue with the GENERATE AC command.

In order to perform transaction recovery, the Kernel maintains:

- a Torn Transaction Log (for each AID or set of AIDs), stored in the scratch pad (see section 2.4), and
- an indication of the depth of the log file (*Max Number of Torn Transaction Log Records*) provided by the Reader.

In combination with Process M, the Kernel implements specific functionality to maintain the Torn Transaction Log, including protection against unauthorized access and periodic house-keeping to purge old entries.

Support for transaction recovery by the Kernel is indicated by *Max Number of Torn Transaction Log Records*. In most cases, the *Max Number of Torn Transaction Log Records* can be set to one; for specialized, high-throughput readers, it can be set to a small number such as two or three.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 97

Support for transaction recovery by the Card is indicated by the presence of the *DRDOL*. Absence of the *DRDOL* indicates that the Card does not support the RECOVER AC command. In this case, an entry in the Torn Transaction Log is not created and if the Card is presented again, the Kernel continues as if it were a new transaction.

3.7.3 Transaction Flow

The normal transaction flow is modified in three ways:

- logging a torn transaction,
- identifying a torn transaction, and
- recovering a torn transaction.

Logging a Torn Transaction

The starting state is an empty list of torn transactions. There are two conditions to be fulfilled for a torn transaction to be logged:

- The Card data includes the *DRDOL*.
- A tear occurs during the GENERATE AC command.

If the transaction fails due to a timeout, transmission, or protocol error in the GENERATE AC command and the Card data includes a *DRDOL*, then a new record is added to the Torn Transaction Log. This record includes the *Application PAN* and the *Application PAN Sequence Number*, as well as other transaction data including that indicated by *CDOL1* and *DRDOL*. If adding this new record to the log means that an old record is displaced, then the old record is sent to the Terminal (as part of the *Discretionary Data* in an OUT Signal).

Identifying a Torn Transaction

When the records have been read from the Card and the *Application PAN* and *Application PAN Sequence Number* are known, the Kernel checks the Torn Transaction Log for a matching entry (i.e. an entry with the same PAN and PAN Sequence Number).

If there is a matching entry, transaction recovery is attempted by sending a RECOVER AC command to the Card. Otherwise, the Kernel simply continues with normal transaction processing.

If recovery of a previous transaction was attempted but failed, then the Kernel continues with normal transaction processing at the same point.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 98

Recovering a Torn Transaction

The Kernel populates the RECOVER AC command data with the data identified by the Card in its *DRDOL*, following the rules that apply for any DOL.

Recovery is done using the following steps:

1. From the Torn Transaction Log, together with the other data listed in *DRDOL*, the Kernel recovers the *DRDOL Related Data* for the torn transaction recovery attempt and sends a RECOVER AC command to the Card.
2. If the RECOVER AC command times out or if there is a protocol error, then another recovery may be attempted.
3. If however a response is obtained with SW1SW2 = '9000', this is then a confirmation that the Card had processed the GENERATE AC command in the torn transaction. In this case, the Kernel restores the transaction context from the Torn Transaction Log and processing continues as per a response to a GENERATE AC command, with the additional step of removing the entry from the Torn Transaction Log.
4. A response with SW1SW2 ≠ '9000' indicates that the Card had not processed the GENERATE AC command in the torn transaction. The Kernel sends a GENERATE AC command. If there is a valid response to this GENERATE AC command (other than timeout or protocol error) then the entry is removed from the Torn Transaction Log and a consistency check[9] is performed. If the consistency check fails, or if the response to the GENERATE AC command is not valid, then no new entry is created in the Torn Transaction Log.

[9] If the value of *DS Summary 1* of the torn transaction (i.e. *DS Summary 1* received in the GET PROCESSING OPTIONS of the torn transaction) does not match *DS Summary 1* of the current transaction, then this is an error.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 99

3.8 Mobile Transactions

3.8.1 Introduction

Transactions involving mobile phones are different from standard card transactions as the phone can be used to authenticate the cardholder.

For this purpose, the Kernel distinguishes between a cardholder device that delegates the CVM processing to the Terminal and a cardholder device that can perform cardholder verification itself. For ease of reference, the latter are often referred to as phones, as this is the most common form factor that supports this functionality. Yet, the distinction between the two types of devices is independent of the form factor and is based on the *Application Interchange Profile*.

For cards, the Kernel performs CVM processing based on the *CVM List* for an EMV mode transaction, and delegates the CVM processing to the Terminal for a mag-stripe mode transaction.

For phones, the Kernel delegates the CVM processing to the phone and ignores the *CVM List*, if present. The Kernel proceeds as follows:

- It sets the *Reader Contactless Transaction Limit* to the applicable limit for phones.
- If the transaction amount exceeds the *Reader CVM Required Limit*, then:
  - It informs the phone that the transaction amount exceeds the *Reader CVM Required Limit*, expecting the phone to perform CVM processing.
  - It then checks whether the phone has performed CVM processing or not.

3.8.2 Mobile Mag-Stripe Mode Transactions

For the support of mobile mag-stripe mode transactions, the Kernel has two mobile specific data objects:

1. The *Mobile Support Indicator*, indicating that the Kernel supports mobile and that a particular transaction requires CVM
2. A *Reader Contactless Transaction Limit (On–device CVM)* for phones (as opposed to a *Reader Contactless Transaction Limit (No On–device CVM)* for cards)

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 100

The Kernel also recognizes one additional Card data object, the *POS Cardholder Interaction Information*. When returned by the Card, the *POS Cardholder Interaction Information* indicates whether:

- Offline PIN has been completed successfully.
- The context is conflicting, meaning the cardholder device detected a discrepancy between the data used for a first tap and the data used for a second tap, the first and second tap being both part of the same transaction.
- The application is activated, and if not, how to remedy this and activate the application.
- A button push or a PIN entry is required.
- The limits are exceeded or not.

The Kernel checks the *Amount, Authorized (Numeric)* against the *Reader Contactless Transaction Limit* and returns an OUT Signal if the transaction amount is greater than this limit. The OUT Signal includes a Status value “Select Next”, to request that the next AID from the candidate list should be selected.

The Kernel then checks whether the transaction amount exceeds the *Reader CVM Required Limit* and, if so, updates the *Mobile Support Indicator* accordingly.

The *Mobile Support Indicator* is then included in the data of the COMPUTE CRYPTOGRAPHIC CHECKSUM command, as part of the data requested in the *UDOL*.

The response to the COMPUTE CRYPTOGRAPHIC CHECKSUM command includes dynamic *CVC3 (Track2)* and the *POS Cardholder Interaction Information* indicates that CVM has been performed.

For step 5 of section 3.4.2, the Kernel uses a different value for *nUN*. The Kernel offsets the (meaningful) length of the *Unpredictable Number (Numeric)* by 5 (i.e. nUN + 5) and uses it to populate *Track 2 Data* and (if present) *Track 1 Data*.

Offsetting *nUN* informs the issuer that CVM was required for this transaction and that CVM processing was delegated to the phone. The issuer verifies whether the CVM processing was correct by checking the correctness of the CVC3 data.

If the COMPUTE CRYPTOGRAPHIC CHECKSUM does not return the *CVC3 (Track2)* data object, the transaction should be declined and the Reader can use *POS Cardholder Interaction Information* to inform the customer of the corrective action to take.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 101

3.8.3 Mobile EMV Mode Transactions

For the support of mobile EMV mode transactions, the Kernel uses the *Reader Contactless Transaction Limit* for phones and the *POS Cardholder Interaction Information* as they were introduced in section 3.8.2.

As for a mobile mag-stripe mode transaction, the Kernel checks the *Application Interchange Profile* and sets the *Reader Contactless Transaction Limit* either equal to the value for phones or to the value used for cards.

If a device identifies itself as one that defers cardholder verification to the device, then CDA is to be used in the GENERATE AC command to avoid fraud.

The Kernel checks the *Amount, Authorized (Numeric)* against the *Reader Contactless Transaction Limit* and returns an OUT Signal if the transaction amount is greater than this limit. The OUT Signal includes a Status value “Select Next”, to request that the next AID from the candidate list should be selected.

The Kernel then checks the transaction amount against the *Reader CVM Required Limit*.

If the transaction amount is equal to or below the *Reader CVM Required Limit*, then cardholder verification is not required. If the transaction amount is greater than the limit, then the Kernel sets the *CVM Results* to indicate that offline (plaintext) PIN was performed (by the ICC) successfully.

The *CVM Results* are included in the GENERATE AC command, as part of the data requested by *CDOL1*.

Once the response to the GENERATE AC command has been received, the Kernel performs offline card authentication.

The response to the GENERATE AC may include the *POS Cardholder Interaction Information*. The Kernel uses this to check that cardholder verification has been performed successfully or, in the case of a decline, to inform the customer to take corrective action.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 102

3.9 Balance Reading

3.9.1 Introduction

A Card may have an offline balance, and some products require the balance to be read and made available to the customer, either on a receipt or on a display. Not all cards support balance reading and those that do explicitly indicate it in the *Application Capabilities Information*.

3.9.2 Reading

If balance reading is required as a configuration option then *Balance Read Before Gen AC* or *Balance Read After Gen AC* or both are present in the Kernel database with a zero length. These tags may also be included on a per transaction basis as part of the Kernel activation (ACT Signal) or using the Data Exchange mechanism (DEK/DET).

If balance reading is not required, both tags are absent from the Kernel database for the duration of the transaction.

3.9.3 Display and Receipt

If the *Balance Read After Gen AC* is successfully read and the transaction is approved offline, then it is shown on the display by including it in the *User Interface Request Data* that the Kernel sends to Process D.

If both *Balance Read Before Gen AC* and *Balance Read After Gen AC* are present in the Kernel database, then both data objects will be included in the *Discretionary Data* but only one balance will be displayed and this will be *Balance Read After Gen AC*, assuming that it was read without error.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 103

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

4 Data Organization

This chapter defines the data organization of the Kernel. The following topics are included:

4.1 TLV Database

4.2 Working Variables

4.3 List Handling

4.4 Torn Transaction Log

4.5 Configuration Data

4.6 Lists of Data Objects in OUT

4.7 Data Object Format

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 105

4.1 TLV Database

4.1.1 Principles

The Kernel maintains a TLV Database to store all the TLV encoded data objects.

This TLV Database is instantiated at the time of instantiation of the Kernel with an initial set of data objects. This is a copy of the persistent Kernel-specific dataset that is relevant for the selected transaction type and AID. It will be updated during the processing of the transaction.

The TLV Database is updated using information received from a number of sources: at start-up from the Reader, with data from the Card, with data from the Terminal, and with data that results from the Kernel's own processing.

A data object is known by the Kernel if its tag is listed in the data dictionary of Annex A. Other data objects with proprietary tags not listed in the data dictionary may be present in the database at the time of instantiation.

A data object is considered to be present if its tag appears in the TLV Database (length may be zero).

A data object is empty if it is present and its length is zero. A data object is not empty if it is present and its length is greater than zero.

Data objects in the TLV Database have a name, a tag, a length, and a value; for example:

Name:	*Amount, Authorized (Numeric)*
Tag:	'9F02'
Length:	6
Value:	'000000002345'

The index to access data objects in the TLV Database is the tag. The list of tags known by the Kernel is fixed and is defined by the tags of the TLV encoded data objects in the data dictionary.

The name of the TLV encoded data object is also used to represent the value field. The following example initializes the value field of the *Terminal Verification Results* to zero:

Terminal Verification Results := '0000000000'

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 106

4.1.2 Access Conditions

Data objects in the TLV Database are assigned access conditions as described in Table 4.1.

Table 4.1—Access Conditions

Access Condition	Description
ACT/DET	These data objects are transaction related data objects sent to the Kernel by the Terminal with the ACT and DET Signals. They may also be present in the TLV Database when the Kernel is instantiated. Proprietary data objects (i.e. data objects with tags not listed in the data dictionary of Annex A) can be updated with the ACT and DET Signals if their length at instantiation is different from zero.
RA	These data objects are transaction related data objects sent to the Kernel by the Card with the RA Signal. Proprietary data objects can be updated with the RA Signal if their length at instantiation is equal to zero. An exception is data objects contained in the *File Control Information Template* which are passed to the Kernel with the ACT Signal, but which have the RA access condition assigned.
K	All data objects in the TLV Database can be updated by the Kernel. Every data object has the K (Kernel) access condition assigned.

All data objects can be read by the Card (via a DOL) and by the Terminal (via *Tags To Read*).

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 107

4.1.3 Services

Services available to interrogate and manipulate the TLV Database are the following:

Boolean IsKnown(T)

Returns TRUE if tag T is defined in the data dictionary of the Kernel as defined in Annex A.

Boolean IsPresent(T)

Returns TRUE if the TLV Database includes a data object with tag T.

Note that the length of the data object may be zero.

Note also that proprietary data objects that are not known can be present if they have been provided in the TLV Database at Kernel instantiation. In this case the IsKnown() service returns FALSE and the IsPresent() service returns TRUE.

Boolean IsNotPresent(T)

Returns TRUE if the TLV Database does not include a data object with tag T.

Boolean IsNotEmpty(T)

Returns TRUE if all of the following are true:

- The TLV Database includes a data object with tag T.
- The length of the data object is different from zero.

Boolean IsEmpty(T)

Returns TRUE if all the following are true:

- The TLV Database includes a data object with tag T.
- The length of the data object is zero.

T TagOf(DataObjectName)

Returns the tag of the data object with name DataObjectName.

Initialize(T)

Initializes the data object with tag T with a zero length. After initialization the data object is present in the TLV Database.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 108

DataObject GetTLV(T)

Retrieves the TLV encoded data object with tag T from the TLV Database. Returns NULL if the TLV Database does not include a data object with tag T.

Length GetLength(T)

Retrieves from the TLV Database the length of the data object with tag T. Returns NULL if the TLV Database does not include a data object with tag T.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 109

Boolean ParseAndStoreCardResponse(TLV String)

TLV Encoding Error := FALSE

Parse TLV String according the Basic Encoding Rules in [ISO/IEC 8825-1] and set TLV Encoding Error to TRUE if parsing error.

```
IF [TLV Encoding Error]
THEN
    Return FALSE
ELSE
    FOR every primitive TLV in TLV String
    {
        IF [IsKnown(T)]
        THEN
            IF      [(IsNotPresent(T) OR IsEmpty(T)) AND update conditions of
                    T include RA Signal]
            THEN
                    Store LV in the TLV Database for tag T
            ELSE
                    Return FALSE
            ENDIF
        ELSE
            IF      [IsPresent(T)]
            THEN
                IF      [IsEmpty(T) AND update conditions of T include RA
                        Signal]
                THEN
                        Store LV in the TLV Database for tag T
                ELSE
                        Return FALSE
                ENDIF
            ENDIF
        ENDIF
    }
    Return TRUE
ENDIF
```


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 110

UpdateWithDetData(Terminal Sent Data)

Copies all incoming data (Terminal Sent Data) to the Kernel TLV Database if update conditions allow.

```
FOR every TLV in Terminal Sent Data
{
    IF      [(IsKnown(T) OR IsPresent(T)) AND
            update conditions of T include DET Signal]
    THEN
            Store LV in the TLV Database for tag T
    ENDIF
}
IF  [Terminal Sent Data includes Tags To Read]
THEN
    AddListToList(Tags To Read, Tags To Read Yet)
ENDIF
IF  [Terminal Sent Data includes Tags To Write Before Gen AC]
THEN
    AddListToList(Tags To Write Before Gen AC, Tags To Write Yet Before Gen
    AC)
ENDIF
IF  [Terminal Sent Data includes Tags To Write After Gen AC]
THEN
    AddListToList(Tags To Write After Gen AC, Tags To Write Yet After Gen AC)
ENDIF
```


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 111

4.1.4 DOL Handling

TLV encoded data objects moved from the Kernel to the Card are identified by a DOL sent to the Kernel by the Card.

DOLs used in this specification are processed as follows:

- *DRDOL*, *CDOL1*, *PDOL*, and *UDOL*

  DOL handling must be performed according to the rules specified in section 5.4 of [EMV Book 3].

- *DSDOL*

  All entries except the last must be handled according to the rules specified in section 5.4 of [EMV Book 3].

  The last entry in DSDOL must be handled according to the rules specified in section 5.4 of [EMV Book 3], unless the length specified in this entry is greater than the actual length of the data object in the TLV Database. In this case, no padding must be applied and the value must be appended with the length defined in the TLV Database.

Note that if DE is supported, tags in a DOL that exist in the TLV Database with zero length are still handled according the rules specified in section 5.4 of [EMV Book 3], but in addition any such data objects get requested from the Terminal before the chokepoint so that the terminal is afforded the opportunity to furnish a value for these data objects.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 112

4.2 Working Variables

The Kernel makes use of a number of working variables that are not stored in the TLV Database. They are managed by the Kernel in an implementation specific way.

Working variables can be:

- Local

  The lifetime of local working variables is limited to the state transition process or procedure in which they are defined. These data objects do not appear in the data dictionary.

- Global

  The lifetime of global working variables is the same as the lifetime of the Kernel process. Global working variables are listed in the data dictionary without a tag.

  These data objects are managed by the Kernel itself.

  Global working variables can only be read and written by internal processing of the Kernel.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 113

4.3 List Handling

Data is passed between the Kernel and other entities within Signals. The data within the Signals contain a list of tags, in order to request data, or a list of data objects in response to a request.

Each list has a unique name, and acts as a container for a collection of ListItems. A ListItem is a single element in a List. A ListItem is a tag in a list of tags or a data object in a list of data objects.

The following lists of tags are supported:

- *Tags To Read*
- *Tags To Read Yet*
- *Data Needed*

The following lists of TLV encoded data objects are supported:

- *Tags To Write After Gen AC*
- *Tags To Write Before Gen AC*
- *Tags To Write Yet After Gen AC*
- *Tags To Write Yet Before Gen AC*
- *Data To Send*
- *Data Record*
- *Discretionary Data*
- *Torn Record*

The following methods are used to manipulate lists.

Initialize(List)

Initializes a List. This creates the List structure if it does not exist, and initializes its contents to be empty, i.e. the List contains no ListItems. This method can be called at any time during the operation of the Kernel in order to clear and reset a list.

AddToList(ListItem, List)

If ListItem is not included in List, then adds ListItem to the end of List.

Updates ListItem if it is already included in the List.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 114

RemoveFromList(ListItem, List)

Removes ListItem from the List if ListItem is present in List. Ignores otherwise.

AddListToList(List1, List2)

Adds the ListItems in List1 that are not yet included in List2 to the end of List2.

Updates ListItems that are already included in List2.

ListItem GetAndRemoveFromList(List)

Removes and returns the first ListItem from List. Returns NULL if List is empty.

T GetNextGetDataTagFromList(List)

Removes and returns the first tag from a list of tags that is categorized as being available from the Card using a GET DATA command.

If no tag is found, NULL is returned.

Boolean IsEmptyList(List)

Returns TRUE if List contains no ListItems.

Boolean IsNotEmptyList(List)

Returns TRUE if List contains ListItems.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 115

4.4 Torn Transaction Log

The Torn Transaction Log is a log of the latest torn transactions. The maximum number of records in the Torn Transaction Log is implementation specific and is defined by *Max Number of Torn Transaction Log Records*. If *Max Number of Torn Transaction Log Records* is zero, then transaction recovery is not supported.

A record in the Torn Transaction Log is a list of data objects. Every record in the Torn Transaction Log is a constructed TLV encoded data object with tag 'FF8101' and contains the primitive data objects as shown in Table 4.2, if they are present and not empty in the transaction.

Table 4.2—Torn Transaction Log Record

Data Object	Implementations
Amount, Authorized (Numeric)	EMV, EMV/DE
Amount, Other (Numeric)	EMV, EMV/DE
Application PAN	EMV, EMV/DE
Application PAN Sequence Number	EMV, EMV/DE
Balance Read Before Gen AC	EMV, EMV/DE
CDOL1 Related Data	EMV, EMV/DE
CVM Results	EMV, EMV/DE
DRDOL Related Data	EMV, EMV/DE
DS Summary 1	EMV/DE
IDS Status	EMV/DE
Interface Device Serial Number	EMV, EMV/DE
PDOL Related Data	EMV, EMV/DE
Reference Control Parameter	EMV, EMV/DE
Terminal Capabilities	EMV, EMV/DE
Terminal Country Code	EMV, EMV/DE
Terminal Type	EMV, EMV/DE
Terminal Verification Results	EMV, EMV/DE
Transaction Category Code	EMV, EMV/DE
Transaction Currency Code	EMV, EMV/DE

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 116

Data Object	Implementations
Transaction Date	EMV, EMV/DE
Transaction Time	EMV, EMV/DE
Transaction Type	EMV, EMV/DE
Unpredictable Number	EMV, EMV/DE

A Torn Transaction Log record includes the data objects included in the Data Record as well as data objects requested by DOLs. It is likely that this will lead to duplication. Memory usage can be optimised by only storing the DOL-related data that is not already stored, provided that the DOL-related data is reconstructed correctly when required.

The Torn Transaction Log is located in the scratch pad provided to the Kernel at instantiation and is managed by the Kernel. Depending on the implementation, it may be that the Torn Transaction Log does not exist the first time the Kernel is executed. In this case, an empty Torn Transaction Log must be created.

If the Torn Transaction Log already contains *Max Number of Torn Transaction Log Records* records and a new record is added, then the oldest record must be overwritten.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 117

4.5 Configuration Data

At the time of instantiation of the Kernel the data objects listed in this section are initialized.

4.5.1 Configuration Data – TLV Database

Configuration data objects in the TLV Database should receive a value at instantiation of the Kernel.

The data objects listed in Table 4.3 are the configuration data objects that must have a value for the Kernel to work properly. If these data objects do not receive a value at instantiation, a default value must be stored in the TLV Database.

Table 4.3—Configuration Data in TLV Database that Require Default Value

Data Object Name	Implementations	Default Value
Additional Terminal Capabilities	EMV, EMV/DE	'0000000000'
Application Version Number (Reader)	EMV, EMV/DE	'0002'
Card Data Input Capability	EMV, EMV/DE	'00'
CVM Capability – CVM Required	EMV, EMV/DE	'00'
CVM Capability – No CVM Required	EMV, EMV/DE	'00'
Default UDOL	Always	'9F6A04'
Kernel Configuration	EMV, EMV/DE	'00'
Kernel ID	Always	'02'
Mag-stripe Application Version Number (Reader)	Always	'0001'
Mag-stripe CVM Capability – CVM Required	Always	'F0'
Mag-stripe CVM Capability – No CVM Required	Always	'F0'

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 118

Data Object Name	Implementations	Default Value
Max Lifetime of Torn Transaction Log Record	EMV, EMV/DE	'012C'
Max Number of Torn Transaction Log Records	EMV, EMV/DE	'00'
Message Hold Time	Always	'000013'
Reader Contactless Floor Limit	EMV, EMV/DE	'000000000000'
Reader Contactless Transaction Limit (No On-device CVM)	Always	'000000000000'
Reader Contactless Transaction Limit (On-device CVM)	Always	'000000000000'
Reader CVM Required Limit	Always	'000000000000'
Security Capability	EMV, EMV/DE	'00'
Terminal Action Code – Default	EMV, EMV/DE	'CC00000000'
Terminal Action Code – Denial	EMV, EMV/DE	'00000000000'
Terminal Action Code – Online	EMV, EMV/DE	'CC00000000'
Terminal Country Code	Always	'0000'
Terminal Type	EMV, EMV/DE	'00'
Time Out Value	DE, EMV/DE	'01F4'

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 119

4.5.2 CA Public Key Database

The Kernel implementing the EMV mode implementation option has access to a CA Public Key Database containing the CA Public Keys applicable for the RID of the selected AID. This CA Public Key Database is made available to the Kernel and is read-only.

The CA Public Key Index uniquely identifies the CA Public Key in the CA Public Key Database.

Table 4.4 lists the set of data objects that must be available in the CA Public Key Database for each CA Public Key.

Table 4.4—CA Public Key Related Data

Field Name	Length	Description	Format
CA Public Key Index	1	Identifies the CA Public Key in conjunction with the RID	b
CA Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
CA Public Key Algorithm Indicator	1	Identifies the digital signature algorithm to be used with the CA Public Key	b
CA Public Key Modulus	var. (max 248)	Value of the modulus part of the CA Public Key	b
CA Public Key Exponent	1 or 3	Value of the exponent part of the CA Public Key, equal to 3 or $2^{16} + 1$	b
CA Public Key Check Sum (Only necessary if used to verify the integrity of the CA Public Key)	20	A check value calculated on the concatenation of all parts of the CA Public Key (RID, CA Public Key Index, CA Public Key Modulus, CA Public Key Exponent) using SHA-1	b

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 120

4.5.3 Certification Revocation List

The Kernel implementing the EMV mode implementation option has access to a CRL applicable for the RID of the selected AID. This CRL is made available to the Kernel and is read-only.

Table 4.5 lists the set of data objects that must be available in the CRL for each revoked certificate. If, during CDA, a concatenation of the *CA Public Key Index (Card)* and the Certificate Serial Number recovered from the *Issuer Public Key Certificate* is on this list, then CDA fails.

Table 4.5—Certification Revocation List Related Data

Field Name	Length	Description	Format
CA Public Key Index	1	Identifies the CA Public Key in conjunction with the RID	b
Certificate Serial Number	3	Number unique to this certificate assigned by the certification authority	b
Additional Data	var.	Optional terminal proprietary data, such as the date the certificate was added to the revocation list	b

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 121

4.5.4 Phone Message Table

The Kernel has access to a Phone Message Table applicable for the selected AID. It defines the message and status identifiers as a function of the *POS Cardholder Interaction Information*.

The message and status identifiers for a given *POS Cardholder Interaction Information* are defined as follows:

```
FOR every entry in the Phone Message Table
{
    IF  [(PCII MASK[ID] AND POS Cardholder Interaction Information) =
        PCII VALUE[ID]]
    THEN
        MESSAGE[ID] is the message identifier to be used
        STATUS[ID] is the status identifier to be used
        EXIT loop
    ENDIF
}
```

Table 4.6 gives an example of a Phone Message Table for the current definition of the *POS Cardholder Interaction Information*.

Table 4.6—Phone Message Table

ID	PCII MASK	PCII VALUE	MESSAGE	STATUS
1	'000800'	'000800'	SEE PHONE	NOT READY
2	'000400'	'000400'	SEE PHONE	NOT READY
3	'000100'	'000100'	SEE PHONE	NOT READY
4	'000200'	'000200'	SEE PHONE	NOT READY
5	'000000'	'000000'	DECLINED	NOT READY

Note that the last entry in the Phone Message Table must always have PCII MASK and PCII VALUE set to '000000'.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 122

4.6 Lists of Data Objects in OUT

This section specifies the lists of data objects included in the OUT Signal: *Data Record* and *Discretionary Data*.

4.6.1 Data Record

If the 'Status' in *Outcome Parameter Set* is DECLINED, ONLINE REQUEST, or APPROVED, the Kernel provides the Terminal with an OUT Signal including a *Data Record* that contains the necessary data objects for authorization and clearing. The *Data Record* is a list of data objects. Its content depends on the transaction profile.

The *Data Record* for an EMV mode transaction is as shown in Table 4.7. The *Data Record* for a mag-stripe mode transaction is as shown in Table 4.8.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 123

Table 4.7—*Data Record* Detail for EMV Mode Transaction

Data Object
Amount, Authorized (Numeric)
Amount, Other (Numeric)
Application Cryptogram
Application Expiration Date
Application Interchange Profile
Application Label
Application PAN
Application PAN Sequence Number
Application Preferred Name
Application Transaction Counter
Application Version Number (Reader)
Cryptogram Information Data
CVM Results
DF Name
Interface Device Serial Number
Issuer Application Data
Issuer Code Table Index
Terminal Capabilities
Terminal Country Code
Terminal Type
Terminal Verification Results
Track 2 Equivalent Data
Transaction Category Code
Transaction Currency Code
Transaction Date
Transaction Type
Unpredictable Number

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 124

Table 4.8—*Data Record* Detail for Mag-Stripe Mode Transaction

Data Object
Application Label
Application Preferred Name
DF Name
Interface Device Serial Number
Issuer Code Table Index
Mag-stripe Application Version Number (Reader)
Track 1 Data
Track 2 Data

The following methods are used to create the *Data Record*:

```
CreateEMVDataRecord ()
    Initialize(Data Record)
    FOR every Data Object in Table 4.7
    {
        IF      [IsPresent(TagOf(Data Object))]
        THEN
                AddToList(GetTLV(TagOf(Data Object)), Data Record)
        ENDIF
    }

CreateMSDataRecord ()
    Initialize(Data Record)
    FOR every Data Object in Table 4.8
    {
        IF      [IsPresent(TagOf(Data Object))]
        THEN
                AddToList(GetTLV(TagOf(Data Object)), Data Record)
        ENDIF
    }
```


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 125

4.6.2 Discretionary Data

The Kernel always includes *Discretionary Data* in the OUT Signal. The *Discretionary Data* is a list of data objects. Its content depends on the transaction profile.

The *Discretionary Data* for an EMV mode transaction is as shown in Table 4.9. The *Discretionary Data* for a mag-stripe mode transaction is as shown in Table 4.10.

Table 4.9—*Discretionary Data* for an EMV Mode Transaction

Data Object
Balance Read After Gen AC
Balance Read Before Gen AC
DS Summary 3
DS Summary Status
Error Indication
Post-Gen AC Put Data Status
Pre-Gen AC Put Data Status
Third Party Data
Torn Record

Table 4.10—*Discretionary Data* for a Mag-Stripe Mode Transaction

Data Object
DD Card (Track1)
DD Card (Track2)
Error Indication
Third Party Data

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 126

The following methods are used to create the *Discretionary Data*:

```
CreateEMVDiscretionaryData ()
    Initialize(Discretionary Data)
    FOR every Data Object in Table 4.9
    {
        IF      [IsPresent(TagOf(Data Object))]
        THEN
                AddToList(GetTLV(TagOf(Data Object)), Discretionary Data)
        ENDIF
    }

CreateMSDiscretionaryData ()
    Initialize(Discretionary Data)
    FOR every Data Object in Table 4.10
    {
        IF      [IsPresent(TagOf(Data Object))]
        THEN
                AddToList(GetTLV(TagOf(Data Object)), Discretionary Data)
        ENDIF
    }
```


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 127

4.7 Data Object Format

Data objects that have the numeric (n) format are BCD encoded, right justified with leading hexadecimal zeros. Data objects that have the compressed numeric (cn) format are BCD encoded, left justified, and padded with trailing 'F's. Note that the length indicator in the numeric and compressed numeric format notations (e.g. n 4) specifies the number of digits and not the number of bytes.

Data objects that have the alphanumeric (an) or alphanumeric special (ans) format are ASCII encoded, left justified, and padded with trailing hexadecimal zeros.

When moving data from one entity to another (for example Card to Reader) or when concatenating data, the data must always be passed in decreasing order, regardless of how it is stored internally. The leftmost byte (byte 1) is the most significant byte.

Bytes or bits specified as Reserved for Future Use (RFU) must be set to the value indicated, or to zero if no value is given.

An entity receiving data specified as RFU must not examine or depend upon the coding of these bytes or bits.

Data objects are TLV encoded in the following cases:

- Data objects sent from the Card to the Kernel (RA Signal)
- Data objects sent to the Kernel at instantiation or with the ACT and DET Signals
- Data objects sent to the Terminal included in *Data To Send*
- Data objects included in the MSG and OUT Signals
- Data objects included in the records of the Torn Transaction Log

It is the responsibility of the issuer to ensure that data in the Card is of the correct format. No format checking other than that specifically defined is mandated for the Kernel.

However, if during normal processing it is recognized that data read from the Card is incorrectly formatted, the Kernel must perform the processing described in this section.

Other than exceptions specifically defined in this document, data object formatting that does not comply with the requirements in section 12.2.5 of [EMV Book 1] and sections 7.5 and 10.5 of [EMV Book 3] can be considered as a format error.

If a format error is detected, the Kernel must update the *Error Indication* data object as follows:

'L2' in *Error Indication* := CARD DATA ERROR

The Kernel must then process the exception according to the state in which it occurs, as described here.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 128

States 3, 4, 5, 6, 7, and 8

The Kernel must prepare the *Outcome Parameter Set* and send an OUT Signal (as shown below):

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)),
GetTLV(TagOf(*Discretionary Data*))) signal

The Kernel must then exit.

States 9 and 10

The Kernel must process the error as “Invalid Response - 1”, as described under connector C in Figure 6.15.

State 11

The Kernel must process the error as “Invalid Response - 1”, as described under connector C in Figure 6.16.

State 13

The Kernel must process the error as “Invalid Response”, as described under connector A in Figure 6.18.

State 14

The Kernel must process the error as “Invalid Response”, as described under connector A in Figure 6.19.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 129

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

5 C-APDU Commands

This chapter defines the commands and responses supported by the Kernel:

5.1 Introduction
5.2 COMPUTE CRYPTOGRAPHIC CHECKSUM
5.3 GENERATE AC
5.4 GET DATA
5.5 GET PROCESSING OPTIONS
5.6 PUT DATA
5.7 READ RECORD
5.8 RECOVER AC

5.1 Introduction

The INS byte of the C-APDU is structured according to [EMV Book 1]. The coding of INS and its relationship to CLA are shown in Table 5.1. The last column indicates the implementation option for which this C-APDU has to be implemented.

Table 5.1—Coding of the Instruction Byte

CLA	INS	Meaning	Implementations
'80'	'2A'	COMPUTE CRYPTOGRAPHIC CHECKSUM	Always
'80'	'AE'	GENERATE AC	EMV, EMV/DE
'80'	'CA'	GET DATA	EMV, EMV/DE
'80'	'A8'	GET PROCESSING OPTIONS	Always
'80'	'DA'	PUT DATA	EMV/DE
'00'	'B2'	READ RECORD	Always
'80'	'D0'	RECOVER AC	EMV, EMV/DE

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 131

The status bytes returned by the Card are coded as specified in section 6.3.5 of [EMV Book 3]. In addition to the status bytes specific to each command, the Card may return the status bytes shown in Table 5.2.

Table 5.2—Generic Status Bytes

SW1	SW2	Meaning
'6D'	'00'	Instruction code not supported or invalid
'6E'	'00'	Class not supported
'6F'	'00'	No precise diagnosis

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

5.2 COMPUTE CRYPTOGRAPHIC CHECKSUM

5.2.1 Definition and Scope

The COMPUTE CRYPTOGRAPHIC CHECKSUM command initiates the computation of the dynamic CVC3 on the Card.

5.2.2 Command Message

The COMPUTE CRYPTOGRAPHIC CHECKSUM command message is coded according to Table 5.3.

Table 5.3—COMPUTE CRYPTOGRAPHIC CHECKSUM Command Message

Code	Value
CLA	'80'
INS	'2A'
P1	'8E'
P2	'80'
Lc	var.
Data	*UDOL* related data
Le	'00'

The data field of the command message is coded according to the *UDOL* following the rules defined in section 4.1.4. If the Card does not have a *UDOL*, the Kernel uses the *Default UDOL*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 133

5.2.3 Data Field Returned in the Response Message

The data field of the response message is a constructed data object with tag '77' (Response Message Template) as shown in Table 5.4. The value field may include several TLV coded data objects, but always includes the *Application Transaction Counter*. The value field may also include the *CVC3 (Track1)*, *CVC3 (Track2)*, and *POS Cardholder Interaction Information*.

Data objects in *Response Message Template Format 2* may appear in any order.

Table 5.4—COMPUTE CRYPTOGRAPHIC CHECKSUM Response Message Data Field

Tag	Value	Presence
'77'	*Response Message Template Format 2*	M
'9F36'	*Application Transaction Counter*	M
'9F60'	*CVC3 (Track1)*	O
'9F61'	*CVC3 (Track2)*	O
'DF4B'	*POS Cardholder Interaction Information*	O

5.2.4 Status Bytes

The status bytes that may be sent in response to the COMPUTE CRYPTOGRAPHIC CHECKSUM command are listed in Table 5.5.

Table 5.5—Status Bytes for COMPUTE CRYPTOGRAPHIC CHECKSUM Command

SW1	SW2	Meaning
'67'	'00'	Wrong length
'69'	'85'	Conditions of use not satisfied
'6A'	'86'	Incorrect parameters P1-P2
'90'	'00'	Normal processing

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 134

5.3 GENERATE AC

5.3.1 Definition and Scope

The GENERATE AC command sends transaction-related data to the Card, which then computes and returns an *Application Cryptogram*. Depending on the risk management in the Card, the cryptogram returned by the Card may differ from that requested in the command message. The Card may return an AAC (transaction declined), an ARQC (online authorization request), or a TC (transaction approved).

5.3.2 Command Message

The GENERATE AC command message is coded according to Table 5.6.

Table 5.6—GENERATE AC Command Message

Code	Value
CLA	'80'
INS	'AE'
P1	Reference Control Parameter (see Table 5.7)
P2	'00'
Lc	var.
Data	*CDOL1 Related Data* || *DSDOL* related data (conditional (if IDS write performed))
Le	'00'

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 135

Table 5.7—GENERATE AC Reference Control Parameter

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							AAC
0	1							TC
1	0							ARQC
1	1							RFU
		x						RFU
		0						Other values RFU
			0					CDA not requested
			1					CDA requested
				x	x	x	x	RFU
				0	0	0	0	Other values RFU

The data field of the command message contains *CDOL1 Related Data* coded according to *CDOL1* following the rules defined in section 4.1.4.

In the case of IDS data writing, the data field of the command message is a concatenation of *CDOL1 Related Data* followed by *DSDOL* related data coded according to *DSDOL* following the rules defined in section 4.1.4.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 136

5.3.3 Data Field Returned in the Response Message

The data field in the response message to the GENERATE AC command is coded according to either format 1 or format 2, as follows.

Format 1

In the case of format 1, the data object returned in the response message is a primitive data object *Response Message Template Format 1* with tag equal to '80'. The value field consists of the concatenation without delimiters (tag and length) of the value fields of the data objects specified in Table 5.8.

Format 1 is not used if CDA is performed.

Table 5.8—GENERATE AC Response Message Data Field (Format 1)

Value	Presence
Cryptogram Information Data	M
Application Transaction Counter	M
Application Cryptogram	M
Issuer Application Data	O

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 137

Format 2

In the case of format 2, the data object returned in the response message varies depending on whether CDA was performed or not.

CDA Not Performed

If CDA is not performed, the data object returned in the response message is a constructed data object with tag equal to '77' (*Response Message Template Format 2*), as specified in Table 5.9.

Data objects in *Response Message Template Format 2* may appear in any order.

Table 5.9—GENERATE AC Response Message Data Field (Format 2) – No CDA

Tag	Value	Presence
'77'	*Response Message Template Format 2*	M
'9F27'	*Cryptogram Information Data*	M
'9F36'	*Application Transaction Counter*	M
'9F26'	*Application Cryptogram*	M
'9F10'	*Issuer Application Data*	O
'DF4B'	*POS Cardholder Interaction Information*	O

CDA Performed

If CDA is performed, the data object returned in the response message is a constructed data object with tag equal to '77' (*Response Message Template Format 2*). It contains at least the three mandatory data objects specified in Table 5.10, and optionally the *Issuer Application Data*.

Data objects in *Response Message Template Format 2* may appear in any order.

Table 5.10—GENERATE AC Response Message Data Field (Format 2) – CDA

Tag	Value	Presence
'77'	*Response Message Template Format 2*	M
'9F27'	*Cryptogram Information Data*	M
'9F36'	*Application Transaction Counter*	M
'9F4B'	*Signed Dynamic Application Data*	M
'9F10'	*Issuer Application Data*	O
'DF4B'	*POS Cardholder Interaction Information*	O

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 138

5.3.4 Status Bytes

The status bytes that may be sent in response to the GENERATE AC command are listed in Table 5.11.

Table 5.11—Status Bytes for GENERATE AC Command

SW1	SW2	Meaning
'67'	'00'	Wrong length
'69'	'85'	Conditions of use not satisfied
'6A'	'86'	Incorrect parameters P1-P2
'90'	'00'	Normal processing

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 139

5.4 GET DATA

5.4.1 Definition and Scope

The GET DATA command is used to retrieve a primitive data object from the Card not encapsulated in a record.

5.4.2 Command Message

The GET DATA command message is coded according to Table 5.12.

Table 5.12—GET DATA Command Message

Code	Value
CLA	'80'
INS	'CA'
P1 || P2	Tag
Lc	Not present
Data	Not present
Le	'00'

Single byte tags are preceded with a leading '00' byte to fill P1 || P2. Table 5.13 lists the tag values supported for the GET DATA command.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 140

Table 5.13—Supported P1 || P2 Values for GET DATA Command

P1 || P2	Data Object	Implementations
'9F50'	*Offline Accumulator Balance*	EMV, EMV/DE
'9F70'	*Protected Data Envelope 1*	EMV/DE
'9F71'	*Protected Data Envelope 2*	EMV/DE
'9F72'	*Protected Data Envelope 3*	EMV/DE
'9F73'	*Protected Data Envelope 4*	EMV/DE
'9F74'	*Protected Data Envelope 5*	EMV/DE
'9F75'	*Unprotected Data Envelope 1*	EMV/DE
'9F76'	*Unprotected Data Envelope 2*	EMV/DE
'9F77'	*Unprotected Data Envelope 3*	EMV/DE
'9F78'	*Unprotected Data Envelope 4*	EMV/DE
'9F79'	*Unprotected Data Envelope 5*	EMV/DE

5.4.3 Data Field Returned in the Response Message

The data field of the response message contains the primitive data object referred to in P1 || P2 of the command message (in other words, including its tag and its length).

5.4.4 Status Bytes

The status bytes that may be sent in response to the GET DATA command are listed in Table 5.14.

Table 5.14—Status Bytes for GET DATA Command

SW1	SW2	Meaning
'69'	'85'	Conditions of use not satisfied
'6A'	'81'	Wrong parameter(s) P1 || P2; function not supported
'6A'	'88'	Referenced data (data object) not found
'90'	'00'	Normal processing

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 141

5.5 GET PROCESSING OPTIONS

5.5.1 Definition and Scope

The GET PROCESSING OPTIONS command initiates the transaction within the Card.

5.5.2 Command Message

The GET PROCESSING OPTIONS command message is coded according to Table 5.15.

Table 5.15—GET PROCESSING OPTIONS Command Message

Code	Value
CLA	'80'
INS	'A8'
P1	'00'
P2	'00'
Lc	var.
Data	*PDOL Related Data*
Le	'00'

The data field of the command message is the Command Template with tag '83' and with a value field coded according to the *PDOL* provided by the Card in the response to the SELECT command. If the *PDOL* is not provided by the Card, the length field of the template is set to zero. Otherwise the length field is the total length of the value fields of the data objects transmitted to the Card. The value fields are concatenated according to the rules defined in section 4.1.4.

5.5.3 Data Field Returned in the Response Message

The data field in the response message to the GET PROCESSING OPTIONS command is coded according to either format 1 or format 2, as follows.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Format 1

In the case of format 1, the data object returned in the response message is a primitive data object with tag equal to '80'. The value field consists of the concatenation without delimiters (tag and length) of the value fields of the *Application Interchange Profile* and the *Application File Locator*, as shown in Table 5.16.

Table 5.16—GET PROCESSING OPTIONS Response Message Data Field (Format 1)

Value	Presence
Application Interchange Profile	M
Application File Locator	M

Format 2

In the case of format 2, the data object returned in the response message is a constructed data object with tag '77' (*Response Message Template Format 2*). The value field may include several TLV coded objects, but always includes the *Application Interchange Profile* and *Application File Locator*, as shown in Table 5.17. If IDS is supported by both Card and Kernel, then also the IDS related data objects shown in Table 5.17 may be included in the *Response Message Template Format 2*.

Data objects in *Response Message Template Format 2* may appear in any order.

Table 5.17—GET PROCESSING OPTIONS Response Message Data Field (Format 2)

Tag	Value	Presence
'77'	*Response Message Template Format 2*	M
'82'	*Application Interchange Profile*	M
'94'	*Application File Locator*	M
'9F6F'	*DS Slot Management Control*	O
'9F5F'	*DS Slot Availability*	O
'9F7F'	*DS Unpredictable Number*	O
'9F7D'	*DS Summary 1*	O
'9F54'	*DS ODS Card*	O

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 143

5.5.4 Status Bytes

The status bytes that may be sent in response to the GET PROCESSING OPTIONS command are listed in Table 5.18.

Table 5.18—Status Bytes for GET PROCESSING OPTIONS Command

SW1	SW2	Meaning
'67'	'00'	Wrong length
'69'	'85'	Conditions of use not satisfied
'6A'	'86'	Incorrect parameters P1-P2
'90'	'00'	Normal processing

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

5.6 PUT DATA

5.6.1 Definition and Scope

The PUT DATA command is used to store a primitive data object not encapsulated in a record in the Card.

5.6.2 Command Message

The PUT DATA command message is coded according to Table 5.19.

Table 5.19—PUT DATA Command Message

Code	Value
CLA	'80'
INS	'DA'
P1 || P2	Tag
Lc	var.
Data	New data value
Le	Not present

Single byte tags are preceded with a leading '00' byte to fill P1 || P2. Table 5.20 lists the tag values supported for the PUT DATA command.

Table 5.20—Supported P1 || P2 values for PUT DATA Command

P1 || P2	Data Object
'9F75'	*Unprotected Data Envelope 1*
'9F76'	*Unprotected Data Envelope 2*
'9F77'	*Unprotected Data Envelope 3*
'9F78'	*Unprotected Data Envelope 4*
'9F79'	*Unprotected Data Envelope 5*

5.6.3 Data Field Returned in the Response Message

There is no data field in the response message of the PUT DATA command.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 145

5.6.4 Status Bytes

The status bytes that may be sent in response to the PUT DATA command are listed in Table 5.21.

Table 5.21—Status Bytes for PUT DATA Command

SW1	SW2	Meaning
'67'	'00'	Wrong length
'6A'	'88'	Referenced data (data object) not found
'90'	'00'	Normal processing

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

5.7 READ RECORD

5.7.1 Definition and Scope

The READ RECORD command reads a file record in a linear file. The response of the Card consists of returning the record.

5.7.2 Command Message

The READ RECORD command message is coded according to Table 5.22.

Table 5.22—READ RECORD Command Message

Code	Value
CLA	'00'
INS	'B2'
P1	Record number
P2	See Table 5.23
Lc	Not present
Data	Not present
Le	'00'

Table 5.23 specifies the coding of P2 of the READ RECORD command.

Table 5.23—P2 of READ RECORD Command

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x	x				SFI
					1	0	0	P1 is a record number

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 147

5.7.3 Data Field Returned in the Response Message

The data field in the Card response contains the record requested by the command. For SFIs in the range 1-10, the record is a TLV constructed data object with tag '70' as shown in Table 5.24.

Table 5.24—READ RECORD Response Message Data Field

'70'	Length	Record Template

5.7.4 Status Bytes

The status bytes that may be sent in response to the READ RECORD command are listed in Table 5.25.

Table 5.25—Status Bytes for READ RECORD Command

SW1	SW2	Meaning
'69'	'85'	Conditions of use not satisfied
'6A'	'82'	Wrong parameters P1 P2; file not found
'6A'	'83'	Wrong parameters P1 P2; record not found
'6A'	'86'	Incorrect parameters P1 P2
'90'	'00'	Normal processing

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 148

5.8 RECOVER AC

5.8.1 Definition and Scope

The RECOVER AC command recovers from the Card the last transaction that was completed by this Card.

5.8.2 Command Message

The RECOVER AC command message is coded according to Table 5.26.

Table 5.26—RECOVER AC Command Message

Code	Value
CLA	'80'
INS	'D0'
P1	'00'
P2	'00'
Lc	var.
Data	*DRDOL Related Data*
Le	'00'

The data field of the command message contains *DRDOL Related Data*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 149

5.8.3 Data Field Returned in the Response Message

The data object returned in the response message varies depending on whether CDA was performed or not.

CDA Not Performed

If CDA is not performed, the data object returned in the response message is a constructed data object with tag equal to '77', as specified in Table 5.27.

Table 5.27—RECOVER AC Response Message Data Field – No CDA

Tag	Value	Presence
'77'	*Response Message Template Format 2*	M
'9F27'	*Cryptogram Information Data*	M
'9F36'	*Application Transaction Counter*	M
'9F26'	*Application Cryptogram*	M
'9F10'	*Issuer Application Data*	O
'DF4B'	*POS Cardholder Interaction Information*	O

CDA Performed

If CDA is performed, the data object returned in the response message is a constructed data object with tag equal to '77'. It contains at least the three mandatory data objects specified in Table 5.28, and optionally the *Issuer Application Data*.

Table 5.28—RECOVER AC Response Message Data Field – CDA

Tag	Value	Presence
'77'	*Response Message Template Format 2*	M
'9F27'	*Cryptogram Information Data*	M
'9F36'	*Application Transaction Counter*	M
'9F4B'	*Signed Dynamic Application Data*	M
'9F10'	*Issuer Application Data*	O
'DF4B'	*POS Cardholder Interaction Information*	O

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 150

5.8.4 Status Bytes

The status bytes that may be sent in response to the RECOVER AC command are listed in Table 5.29.

Table 5.29—Status Bytes for RECOVER AC Command

SW1	SW2	Meaning
'67'	'00'	Wrong length
'69'	'85'	Conditions of use not satisfied
'6A'	'86'	Incorrect parameters P1-P2
'6A'	'88'	Transaction cannot be recovered
'90'	'00'	Normal processing

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 151

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

6 Kernel State Diagrams

This chapter describes the transaction processing of the Kernel after it has been initiated by Process M.

Additional functionality that is not specified in this chapter (and the procedures it invokes) can be considered optional for the implementation, provided that the principles contained in [EMV Book 3] and [EMV Book 4] are respected, and that the functionality specified here is not compromised.

While other transaction types may be supported, this chapter focuses on the interaction between the Card and the Reader for a purchase and refund transaction.

6.1 Implementation Principles
6.2 Kernel Started
6.3 State 1 – Idle
6.4 State 2 – Waiting for PDOL Data
6.5 State 3 – Waiting For GPO Response
6.6 State 4 – Waiting for EMV Read Record Response
6.7 State 4' – Terminate on Next RA
6.8 State 5 – Waiting for Get Data Response
6.9 State 6 – Waiting for EMV Mode First Write Flag
6.10 States 4, 5, and 6 – Common Processing
6.11 State 7 – Waiting for Mag-stripe Read Record Response
6.12 State 8 – Waiting for Mag-stripe First Write Flag
6.13 States 7 and 8 – Common Processing
6.14 State 9 – Waiting for Generate AC Response - 1
6.15 State 10 – Waiting for Recover AC Response
6.16 States 9 and 10 – Common Processing
6.17 State 11 – Waiting for Generate AC Response - 2
6.18 State 12 – Waiting for Put Data Response Before Generate AC
6.19 State 13 – Waiting for CCC Response – 1
6.20 State 14 – Waiting for CCC Response - 2
6.21 State 15 – Waiting for Put Data Response After Generate AC

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 153

6.1 Implementation Principles

The transaction processing is specified as a state machine that is triggered by external Signals that cause state transitions.

These principles are used in order to present the application concepts. The same principles do not have to be followed in the actual implementation. However, the implementation must behave in a way that is indistinguishable from the behavior specified in this chapter.

If there is a difference in priority between processes that generate events (see section 1.5.10), then pushing the STOP or DET Signal on the Queue of the Kernel may be deferred until after the next Signal from Process P (i.e. a Signal that carries either an R-APDU or a Level 1 error in response to a C-APDU) is pushed on the same Queue.

This implies that it may not be possible for the Terminal to force termination of a transaction via a STOP Signal if the Card erroneously requests more wait time whilst never giving a response. It also means that a STOP Signal sent by the Terminal after the Kernel has sent the final READ RECORD command (and therefore before procedures such as Terminal Action Analysis) will be ignored.

Similarly, if the queuing of a DET Signal is postponed, then in addition to the time penalty – the time spent waiting for a Card response could have been used for the processing of the DET Signal – the updates to the TLV Database linked to the DET Signal will be postponed or ignored.

A pending STOP Signal may not be put on the Queue of the Kernel immediately but it must be put on the Queue if there are no pending Signals from Process P, and will therefore be processed in the next state before the next response from Process P.

In a similar manner, a DET Signal can only remain pending until there are no outstanding events from Process P.

As an alternative to processing a deferred Signal in the next state, an implementation may check whether there is an outstanding DET or STOP Signal on the Queue and process it within the current state, immediately after the sending of each CA Signal to the Card.

For most use cases, this approach will give a reader behaviour as if signals were not deferred. More importantly, it does not suffer from a time penalty as the time spent waiting for the Card response can still be used for the processing of the DET signal.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 154

6.2 Kernel Started

6.2.1 Local Variables

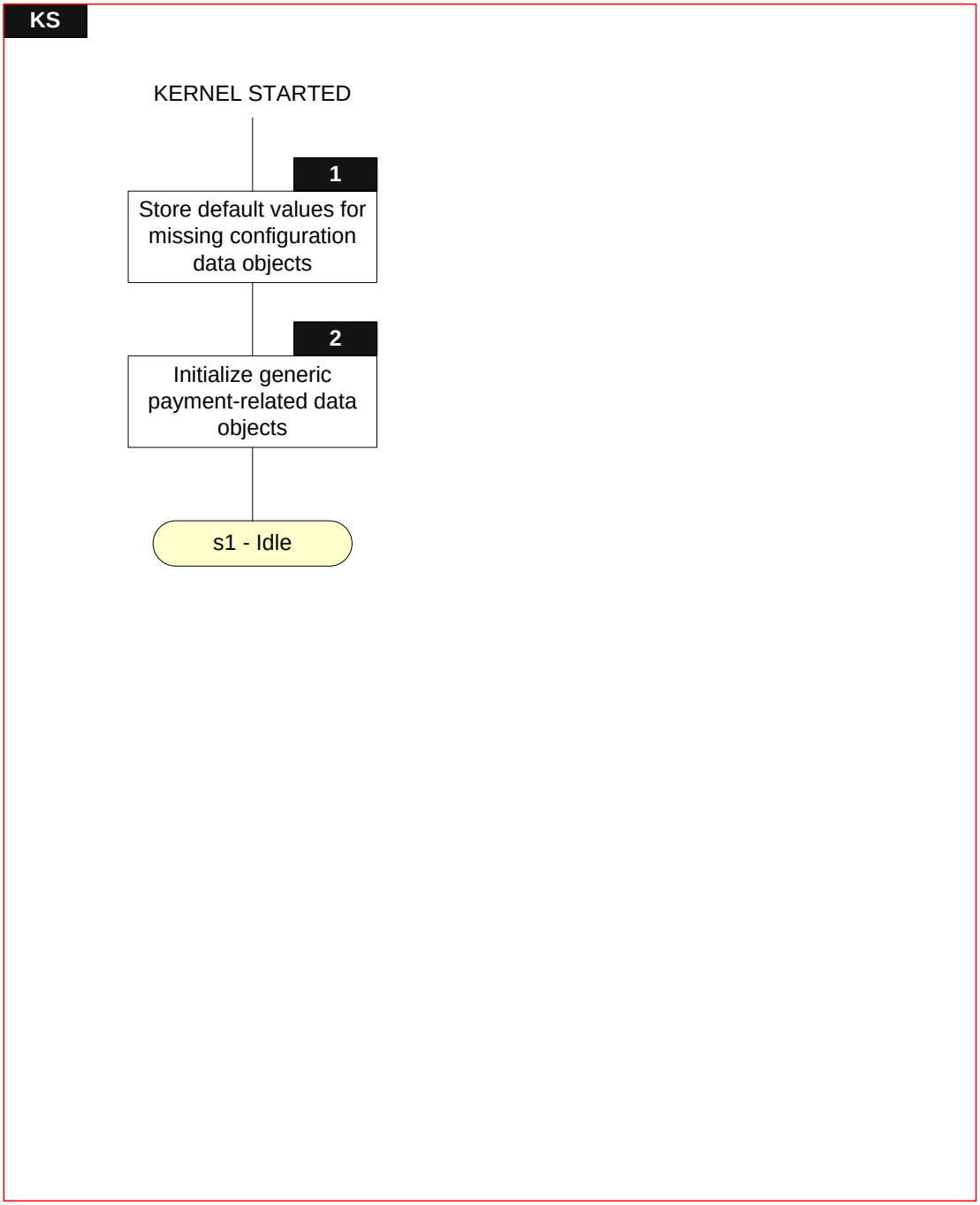
Name	Length	Format	Description
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 253	b	Value of TLV encoded string

6.2.2 Flow Diagram

Figure 6.1 shows the flow diagram of startup of the Kernel. Symbols in this diagram are labeled KS.X.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 155

Figure 6.1—Kernel Started Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 156

6.2.3 Processing

KS.1

FOR every T for which a default value is defined in Table 4.3
{
    IF [IsNotPresent(T)]
    THEN
        Store LV as per Table 4.3 in the TLV Database for tag T
    ENDIF
}

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 157

KS.2

Mobile Support Indicator := '01'

Initialize *Outcome Parameter Set* as follows:

Outcome Parameter Set := '0000 … 00'
'Status' in *Outcome Parameter Set* := N/A
'Start' in *Outcome Parameter Set* := N/A
'CVM' in *Outcome Parameter Set* := N/A
CLEAR 'UI Request on Outcome Present' in *Outcome Parameter Set*
CLEAR 'UI Request on Restart Present' in *Outcome Parameter Set*
CLEAR 'Data Record Present' in *Outcome Parameter Set*
SET 'Discretionary Data Present' in *Outcome Parameter Set*
'Receipt' in *Outcome Parameter Set* := N/A
'Alternate Interface Preference' in *Outcome Parameter Set* := N/A
'Field Off Request' in *Outcome Parameter Set* := 0
'Removal Timeout' in *Outcome Parameter Set* := 0
'Online Response Data' in *Outcome Parameter Set* := N/A

Initialize *User Interface Request Data* as follows:

User Interface Request Data := '0000 … 00'
'Message Identifier' in *User Interface Request Data* := N/A
'Status' in *User Interface Request Data* := N/A
'Hold Time' in *User Interface Request Data* := *Message Hold Time*
'Language Preference' in *User Interface Request Data* := '0000000000000000'
'Value Qualifier' in *User Interface Request Data* := NONE
'Value' in *User Interface Request Data* := '000000000000'
'Currency Code' in *User Interface Request Data* := '0000'

Initialize *Error Indication* as follows:

Error Indication := '0000 … 00'
'L1' in *Error Indication* := OK
'L2' in *Error Indication* := OK
'L3' in *Error Indication* := OK
'SW12' in *Error Indication* := '0000'
'Msg On Error' in *Error Indication* := N/A

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 158

6.3 State 1 – Idle

6.3.1 Local Variables

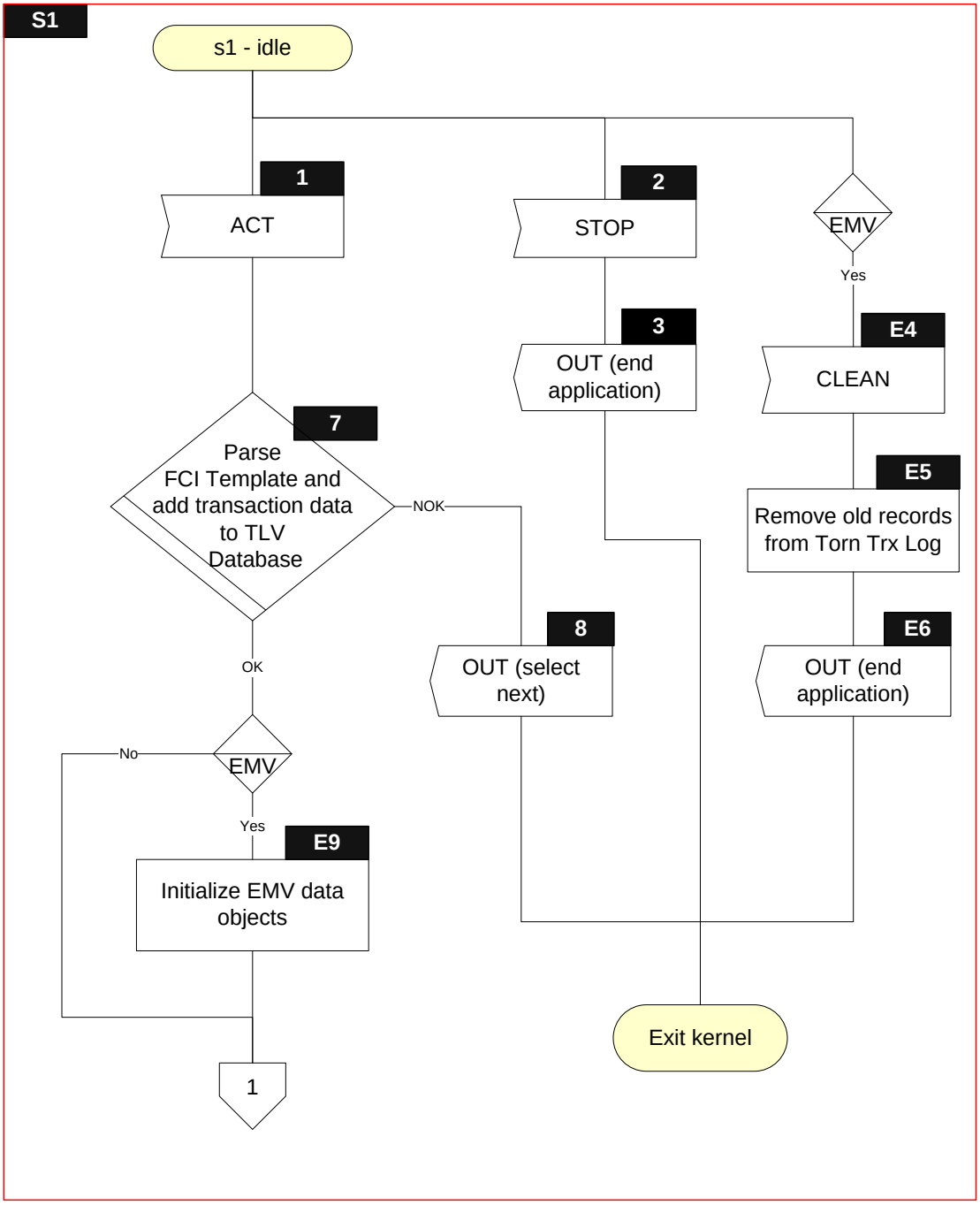
Name	Length	Format	Description
Sync Data	var.	b	List of data objects returned with ACT signal
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
Missing PDOL Data Flag	1	b	Boolean used to indicate if data referenced in *PDOL* is not present in the TLV Database. (Used only when DE implementation option is implemented.)

6.3.2 Flow Diagram

Figure 6.2 shows the flow diagram of `s1 – idle`. Symbols in this diagram are labeled S1.X.

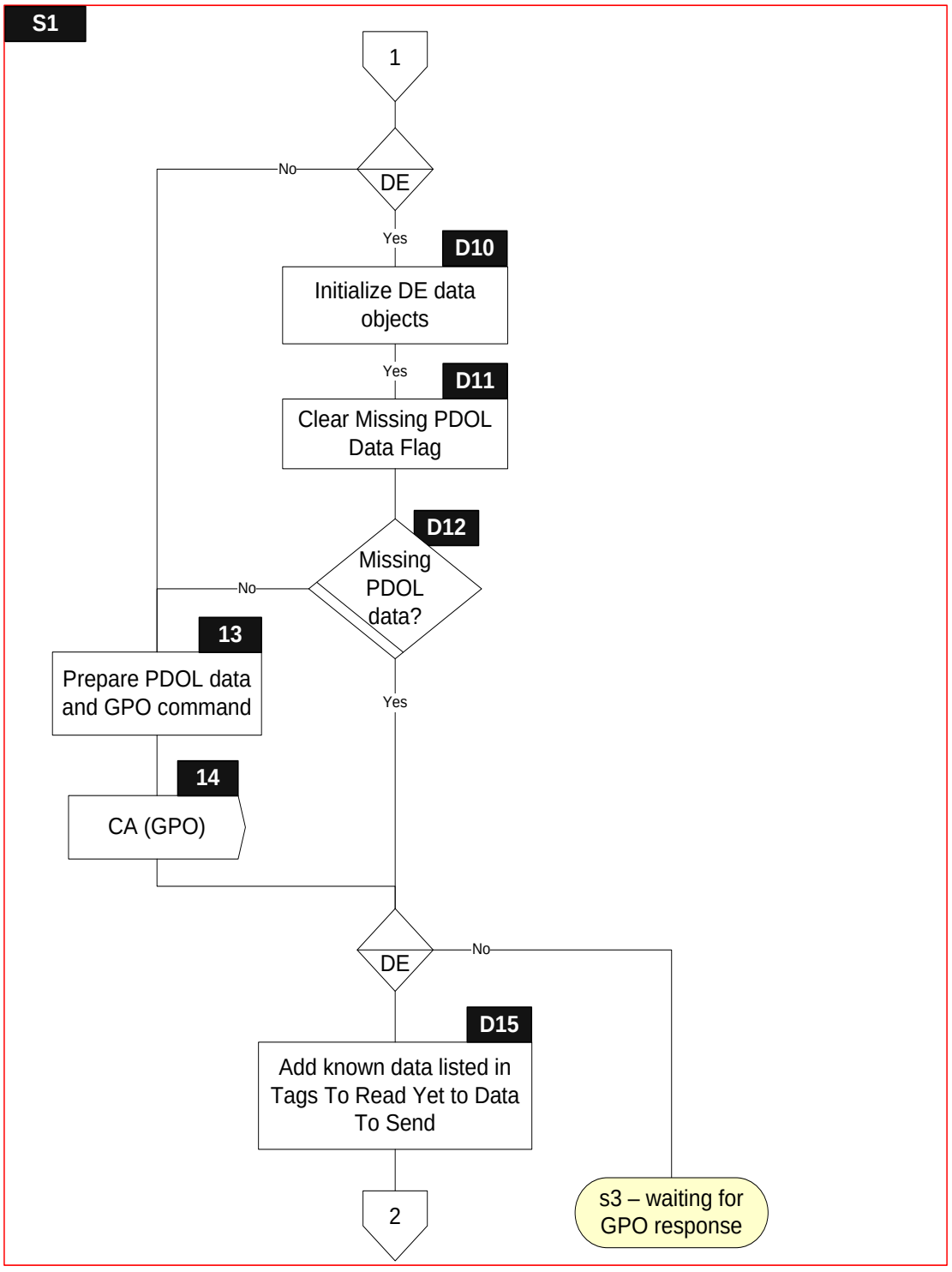
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 159

Figure 6.2—State 1 Flow Diagram



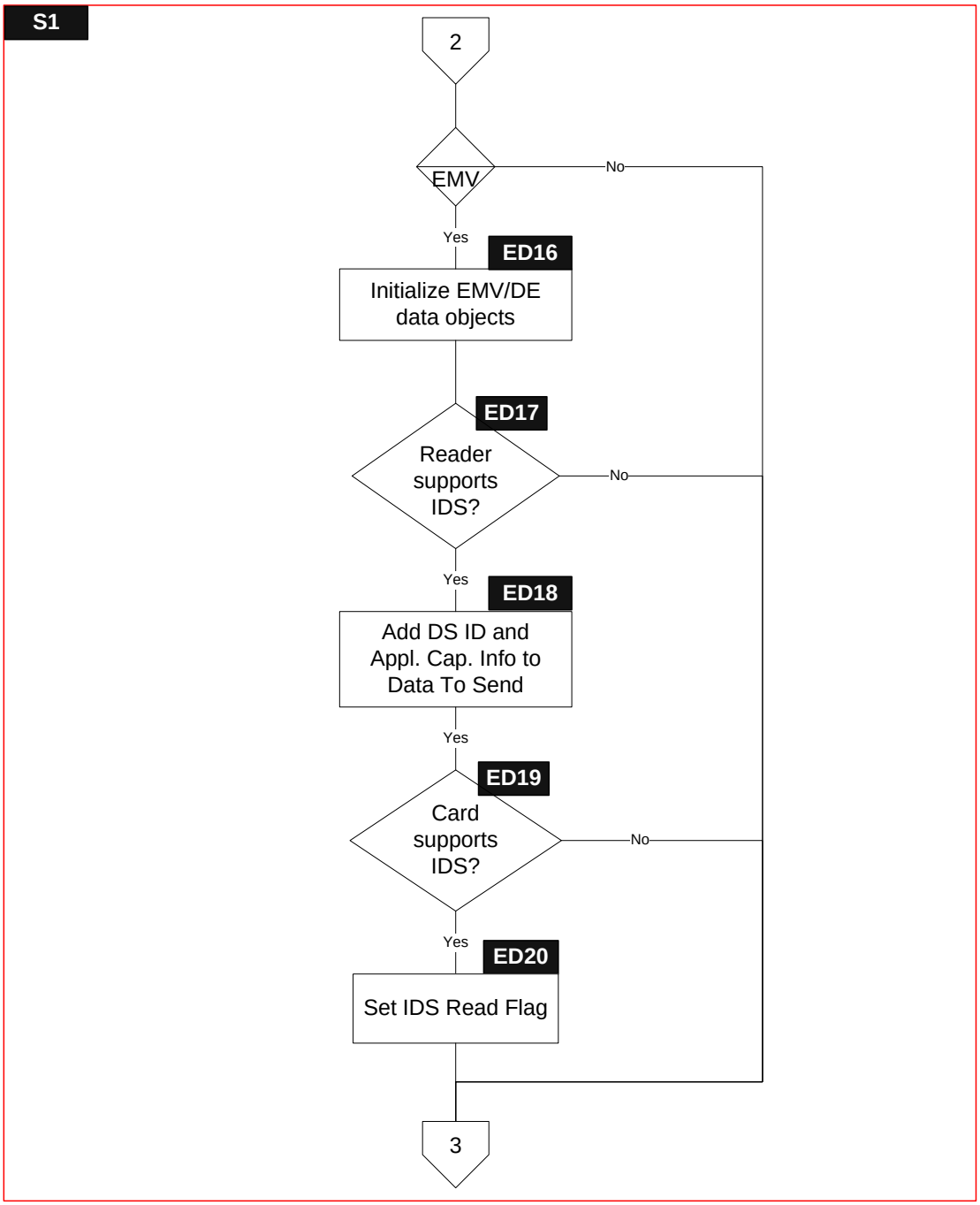
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 160

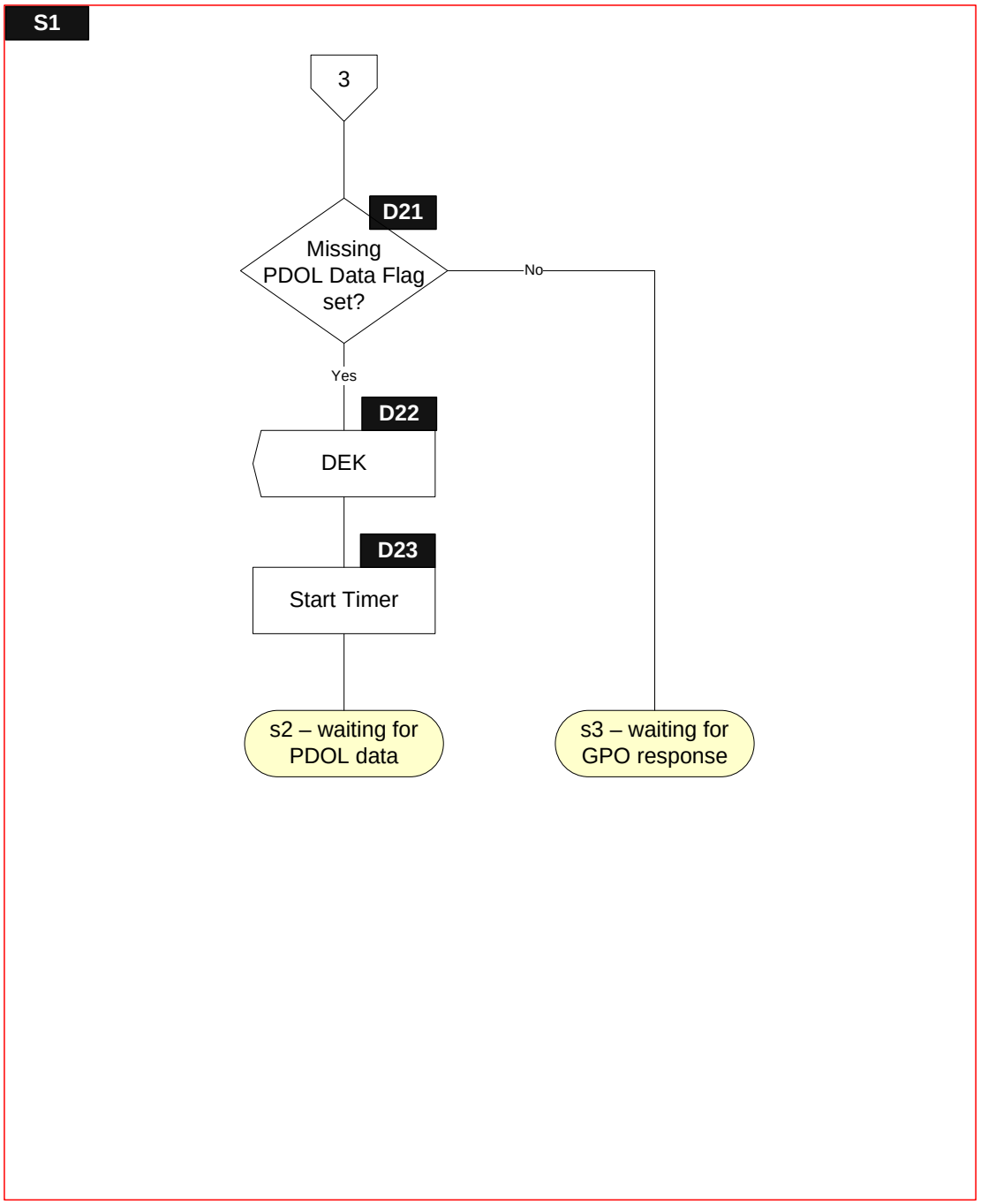


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 161



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 162



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 163

6.3.3 Processing

S1.1

Receive ACT signal with Sync Data

S1.2

Receive STOP signal

S1.3

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

Initialize(*Discretionary Data*)

AddToList(GetTLV(TagOf(*Error Indication*)), *Discretionary Data*)

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S1.E4

Receive CLEAN signal with Sync Data

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

S1.E5

FOR every TLV in Sync Data
{
IF [(IsKnown(T) OR IsPresent(T)) AND update conditions of T include ACT signal]
THEN
Store LV in the TLV Database for tag T
ENDIF
}
'Status' in *Outcome Parameter Set* := END APPLICATION

Remove old records from Torn Transaction Log as follows:
FOR every Record in Torn Transaction Log
{
IF [Difference between *Transaction Date* and *Transaction Time* in Record and *Transaction Date* and *Transaction Time* in TLV Database is greater than *Max Lifetime of Torn Transaction Log Record*]
THEN
Initialize(*Discretionary Data*)
AddToList(Record, *Discretionary Data*)
Remove Record from Torn Transaction Log
Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)),
GetTLV(TagOf(*Discretionary Data*))) signal
ENDIF
}

S1.E6

Initialize(*Discretionary Data*)
Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 165

S1.7

Add the transaction data provided in the ACT signal to the TLV Database

Parse and store the *File Control Information Template* if included in Sync Data

```
FOR every TLV in Sync Data
{
        IF      [T = TagOf(File Control Information Template)]
        THEN
                IF      [NOT ParseAndStoreCardResponse(TLV)]
                THEN
                        'L2' in Error Indication := PARSING ERROR
                        GOTO S1.8
                ENDIF
        ELSE
                IF      [(IsKnown(T) OR IsPresent(T)) AND update conditions of T
                        include ACT signal]
                THEN
                        Store LV in the TLV Database for tag T
                ENDIF
        ENDIF
}
```

If the *Language Preference* is returned from the Card, then copy it to 'Language Preference' in *User Interface Request Data*:

```
IF      [IsNotEmpty(TagOf(Language Preference))]
THEN
        'Language Preference' in User Interface Request Data := Language Preference
ENDIF

IF      [IsNotPresent(TagOf(DF Name)) OR IsEmpty(TagOf(DF Name))]
THEN
        'L2' in Error Indication := CARD DATA MISSING
        GOTO S1.8
ENDIF
```

GOTO S1.E9 or S1.D10 or S1.13

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 166

S1.8

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

Initialize(*Discretionary Data*)

AddToList(GetTLV(TagOf(*Error Indication*)), *Discretionary Data*)

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S1.E9

CVM Results := '000000'

'AC type' in *AC Type* := TC

Terminal Verification Results := '0000000000'

ODA Status := '00'

Terminal Capabilities[1] := *Card Data Input Capability*

Terminal Capabilities[2] := '00'

Terminal Capabilities[3] := *Security Capability*

Initialize(*Static Data To Be Authenticated*)

S1.D10

Initialize(*Data Needed*)

Initialize(*Data To Send*)

Initialize(*Tags To Read Yet*)

IF [IsNotEmpty(TagOf(*Tags To Read*))]

THEN

AddListToList(*Tags To Read*, *Tags To Read Yet*)

ENDIF

IF [IsEmpty(TagOf(*Tags To Read*))]

THEN

AddToList(TagOf(*Tags To Read*), *Data Needed*))

ENDIF

S1.D11

CLEAR Missing PDOL Data Flag

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 167

S1.D12

FOR every TL entry in the *PDOL*
{
    IF [IsEmpty(T)]
    THEN
        SET Missing PDOL Data Flag
        AddToList(T, *Data Needed*)
    ENDIF
}
IF [Missing PDOL Data Flag]
THEN
    GOTO S1.D15
ELSE
    GOTO S1.13
ENDIF

S1.13

Prepare GET PROCESSING OPTIONS command as specified in section 5.5. Use *PDOL* to create *PDOL Related Data* as a concatenated list of data objects without tags or lengths following the rules specified in section 4.1.4.

S1.14

Send CA(GET PROCESSING OPTIONS) signal

S1.D15

FOR every T in *Tags To Read Yet*
{
    IF [IsNotEmpty(T)]
    THEN
        AddToList(GetTLV(T), *Data To Send*)
        RemoveFromList(T, *Tags To Read Yet*)
    ENDIF
}

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 168

S1.ED16

IDS Status := '00'

DS Summary Status := '00'

Post-Gen AC Put Data Status := '00'

Pre-Gen AC Put Data Status := '00'

DS Digest H := '0000000000000000'

Initialize(*Tags To Write Yet After Gen AC*)

Initialize(*Tags To Write Yet Before Gen AC*)

IF [IsNotEmpty(TagOf(*Tags To Write Before Gen AC*))]

THEN

AddListToList(*Tags To Write Before Gen AC*, *Tags To Write Yet Before Gen AC*)

ENDIF

IF [IsNotEmpty(TagOf(*Tags To Write After Gen AC*))]

THEN

AddListToList(*Tags To Write After Gen AC*, *Tags To Write Yet After Gen AC*)

ENDIF

IF [IsEmpty(TagOf(*Tags To Write Before Gen AC*))]

THEN

AddToList(TagOf(*Tags To Write Before Gen AC*), *Data Needed*))

ENDIF

IF [IsEmpty(TagOf(*Tags To Write After Gen AC*))]

THEN

AddToList(TagOf(*Tags To Write After Gen AC*), *Data Needed*))

ENDIF

S1.ED17

IF [IsNotEmpty(TagOf(*DSVN Term*))
AND IsPresent(TagOf(*DS Requested Operator ID*))]

THEN

GOTO S1.ED18

ELSE

GOTO S1

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 169

S1.ED18

IF [IsPresent(TagOf(*DS ID*))]

THEN

AddToList(GetTLV(TagOf(*DS ID*)), *Data To Send*)

ELSE

Add empty *DS ID* to *Data To Send*:

AddToList(TagOf(*DS ID*) || '00', *Data To Send*)

ENDIF

IF [IsPresent(TagOf(*Application Capabilities Information*))]

THEN

AddToList(GetTLV(TagOf(*Application Capabilities Information*)), *Data To Send*)

ELSE

Add empty *Application Capabilities Information* to *Data To Send*:

AddToList(TagOf(*Application Capabilities Information*) || '00', *Data To Send*)

ENDIF

S1.ED19

IF [IsNotEmpty (TagOf (*Application Capabilities Information*)) AND

(('Data Storage Version Number' in *Application Capabilities Information* = VERSION 1)
OR
('Data Storage Version Number' in *Application Capabilities Information* = VERSION 2))

AND IsNotEmpty (TagOf (*DS ID*))]

THEN

GOTO S1.ED20

ELSE

GOTO S1

ENDIF

S1.ED20

SET 'Read' in *IDS Status*

S1.D21

IF [Missing PDOL Data Flag is set]

THEN

GOTO S1.D22

ELSE

GOTO `s3 – waiting for GPO response`

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 170

S1.D22

Send DEK(*Data To Send*, *Data Needed*) signal

Initialize(*Data To Send*)

Initialize(*Data Needed*)

S1.D23

Start Timer (*Time Out Value*)

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 171

6.4 State 2 – Waiting for PDOL Data

6.4.1 Local Variables

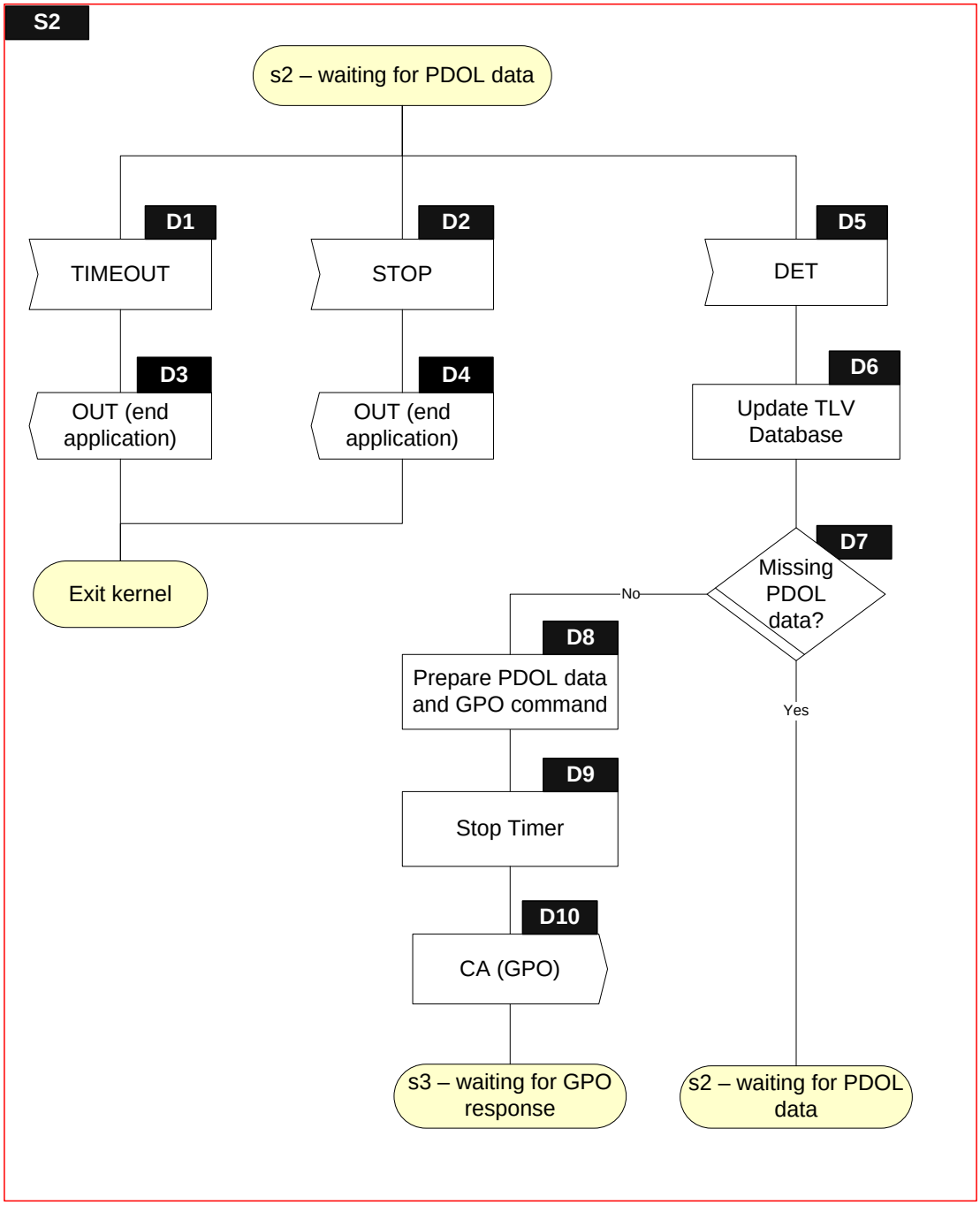
Name	Length	Format	Description
Sync Data	var.	b	List of data objects returned with DET signal
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
Missing PDOL Data Flag	1	b	Boolean used to indicate if data referenced in *PDOL* is not present in the TLV Database.

6.4.2 Flow Diagram

Figure 6.3 shows the flow diagram of `s2 – waiting for PDOL data`. Symbols in this diagram are labeled S2.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 172

Figure 6.3—State 2 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 173

6.4.3 Processing

S2.D1

Receive TIMEOUT signal

S2.D2

Receive STOP signal

S2.D3

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := TIME OUT

Initialize(*Discretionary Data*)

AddToList(GetTLV(TagOf(*Error Indication*)), *Discretionary Data*)

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S2.D4

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

Initialize(*Discretionary Data*)

AddToList(GetTLV(TagOf(*Error Indication*)), *Discretionary Data*)

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S2.D5

Receive DET signal with Sync Data

S2.D6

UpdateWithDetData(Sync Data)

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 174

S2.D7

CLEAR Missing PDOL Data Flag
FOR every TL entry in *PDOL*
{
    IF [IsEmpty(T)]
    THEN
        SET Missing PDOL Data Flag
    ENDIF
}
IF [Missing PDOL Data Flag]
THEN
    GOTO `s2 – waiting for PDOL data`
ELSE
    GOTO S2.D8
ENDIF

S2.D8

Prepare GET PROCESSING OPTIONS command as specified in section 5.5. Use *PDOL* to create *PDOL Related Data* as a concatenated list of data objects without tags or lengths following the rules specified in section 4.1.4.

S2.D9

Stop Timer

S2.D10

Send CA(GET PROCESSING OPTIONS) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 175

6.5 State 3 – Waiting For GPO Response

6.5.1 Local Variables

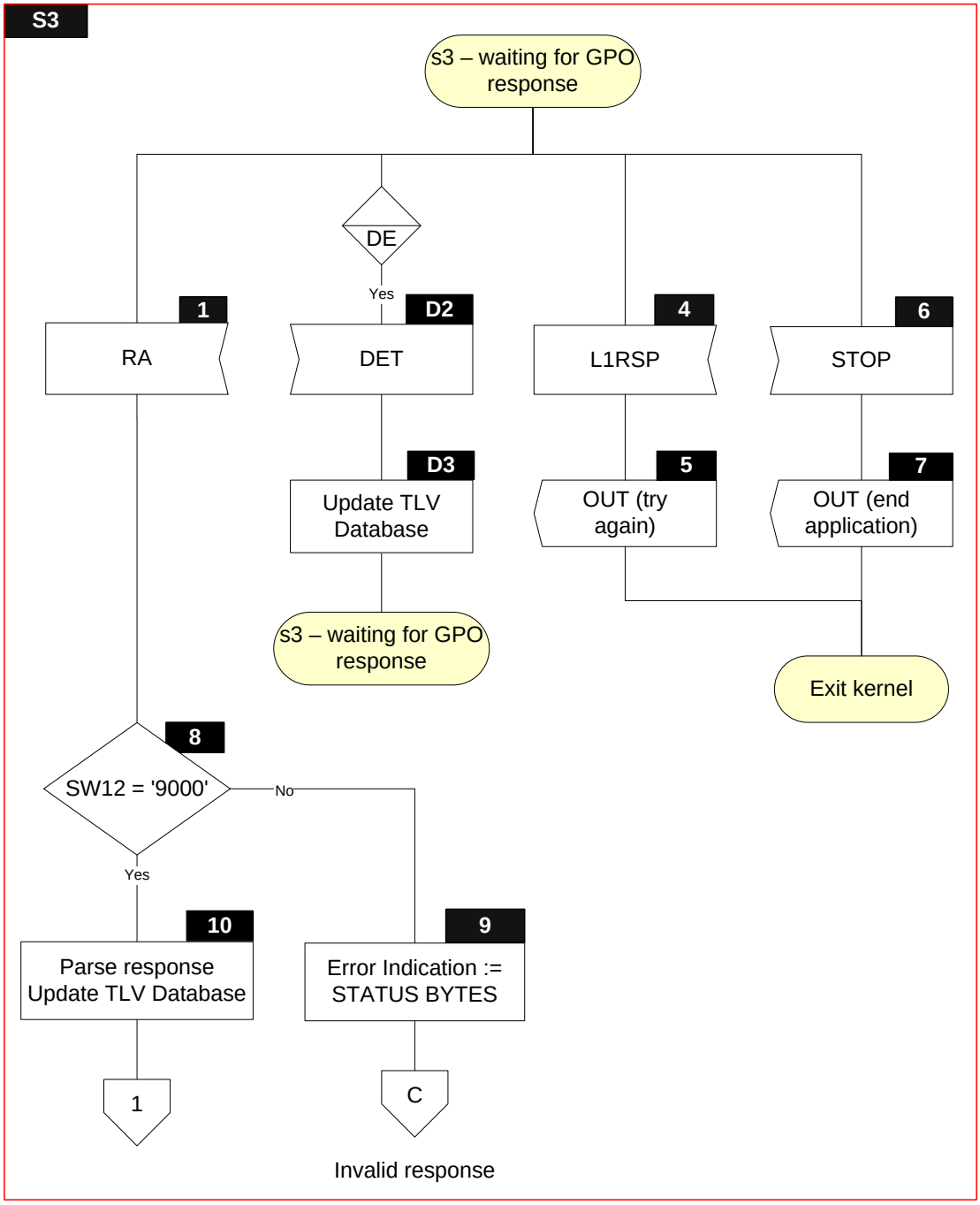
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Sync Data	var.	b	List of data objects returned with DET signal
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of GET PROCESSING OPTIONS
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 253	b	Value of TLV encoded string

6.5.2 Flow Diagram

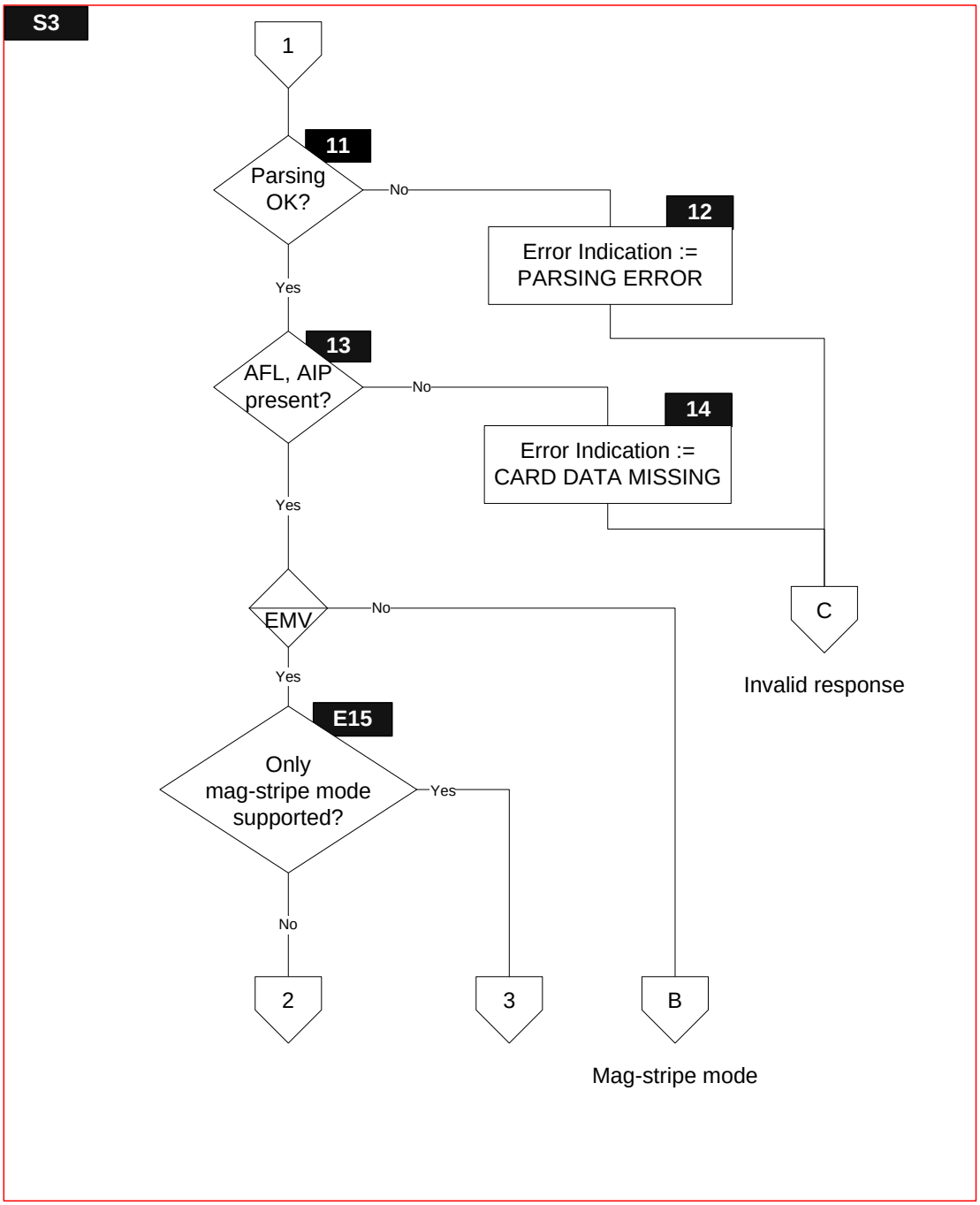
Figure 6.4 shows the flow diagram of `s3 – waiting for GPO response`. Symbols in this diagram are labeled S3.X.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 176

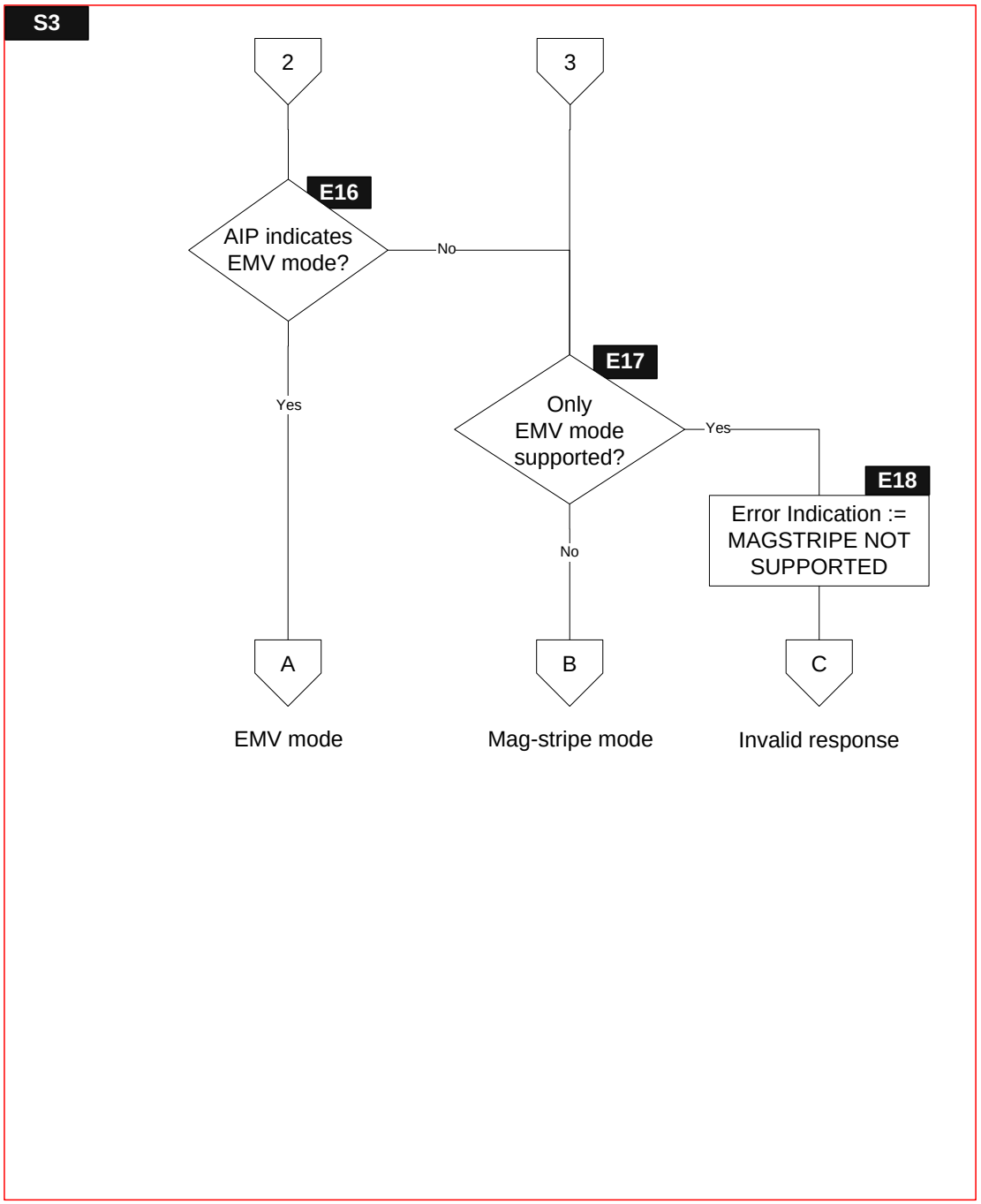
Figure 6.4—State 3 Flow Diagram



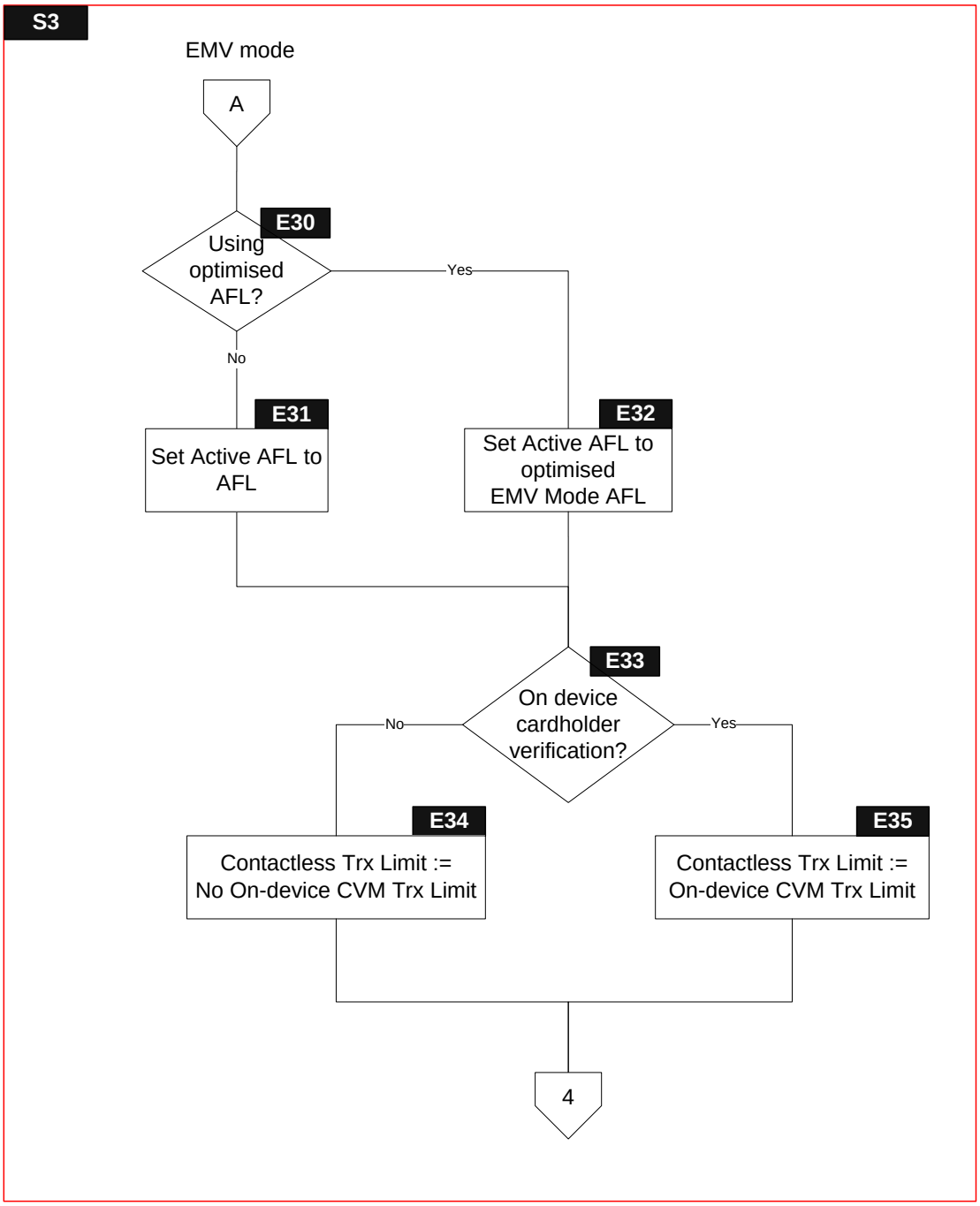
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 177



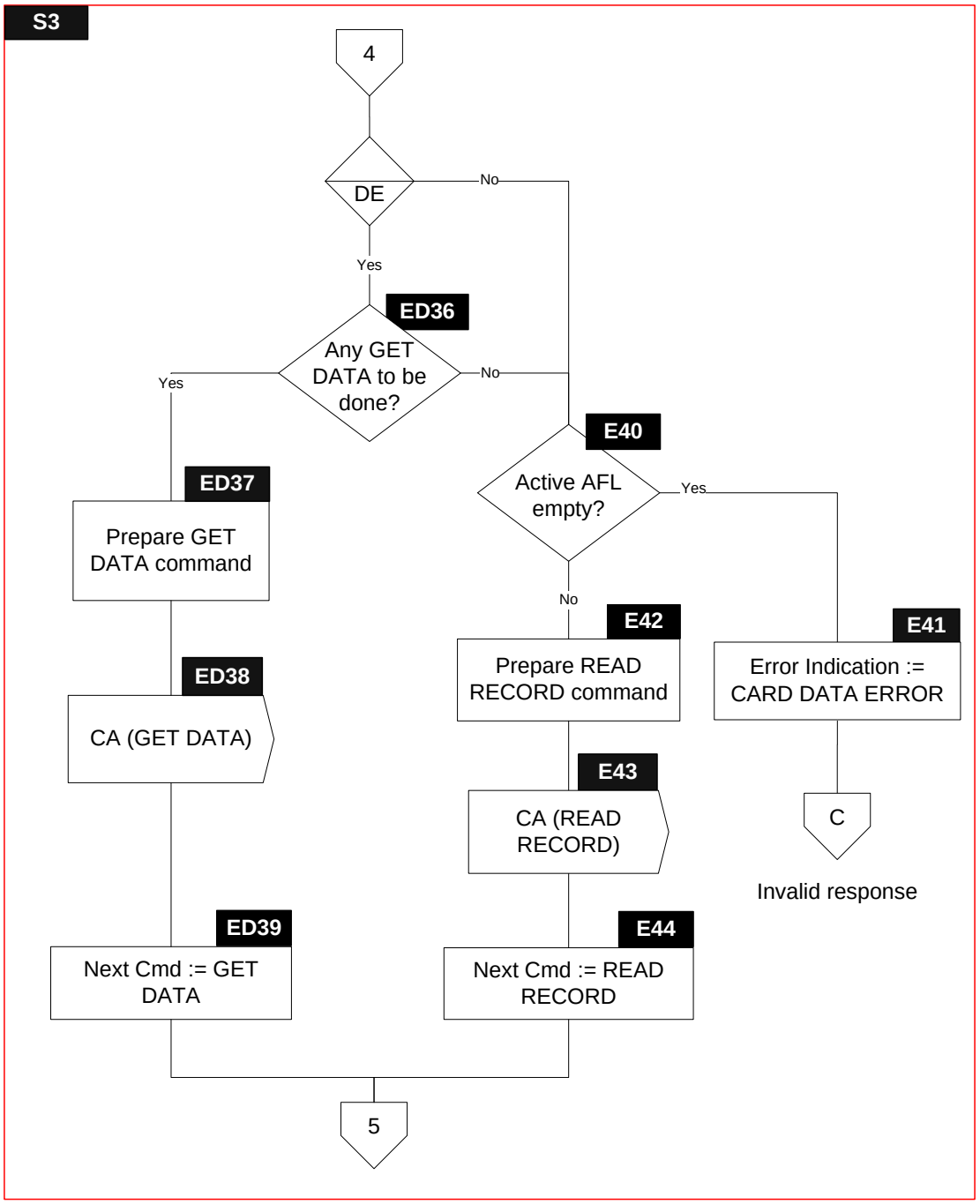
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 178



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 179

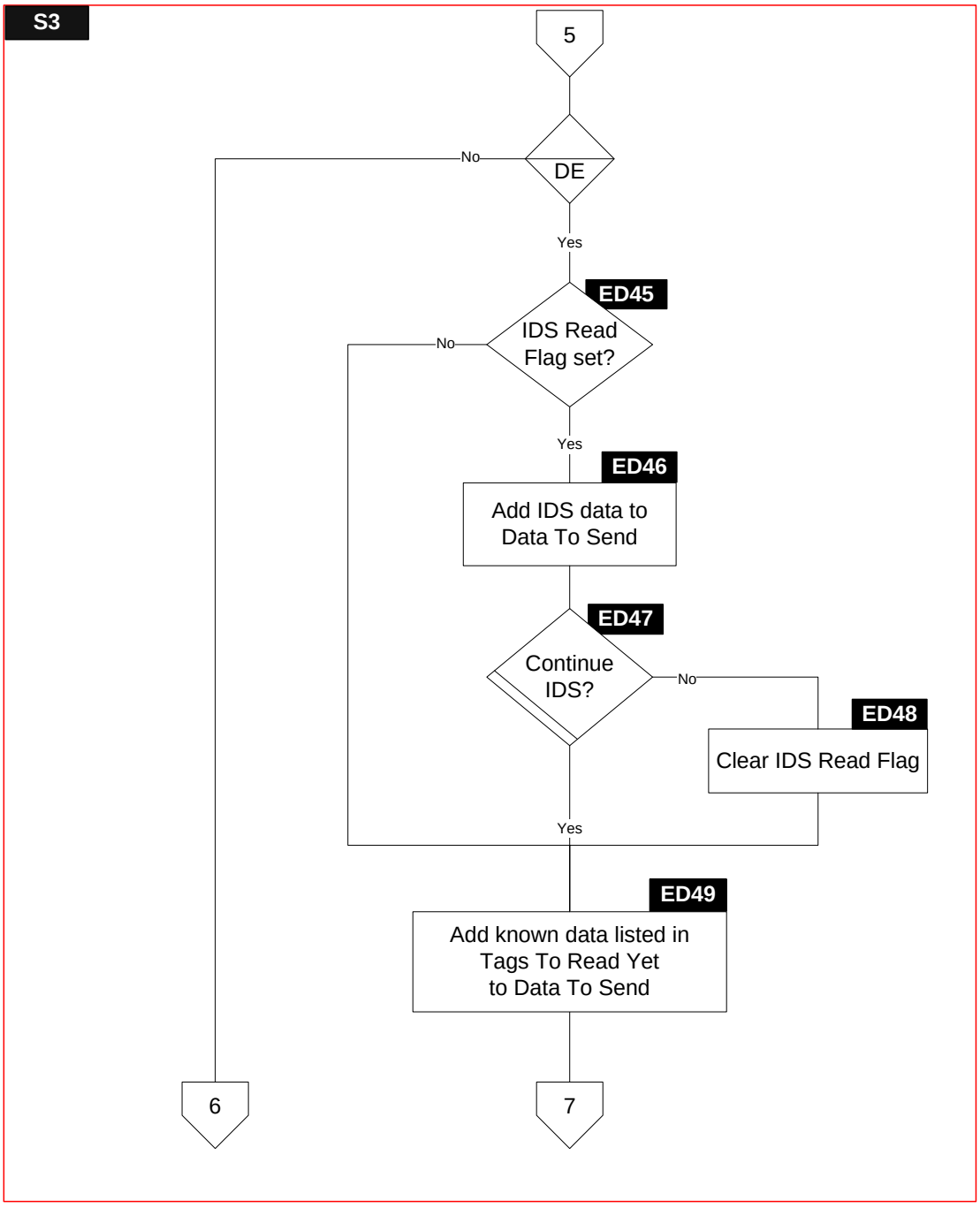


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 180

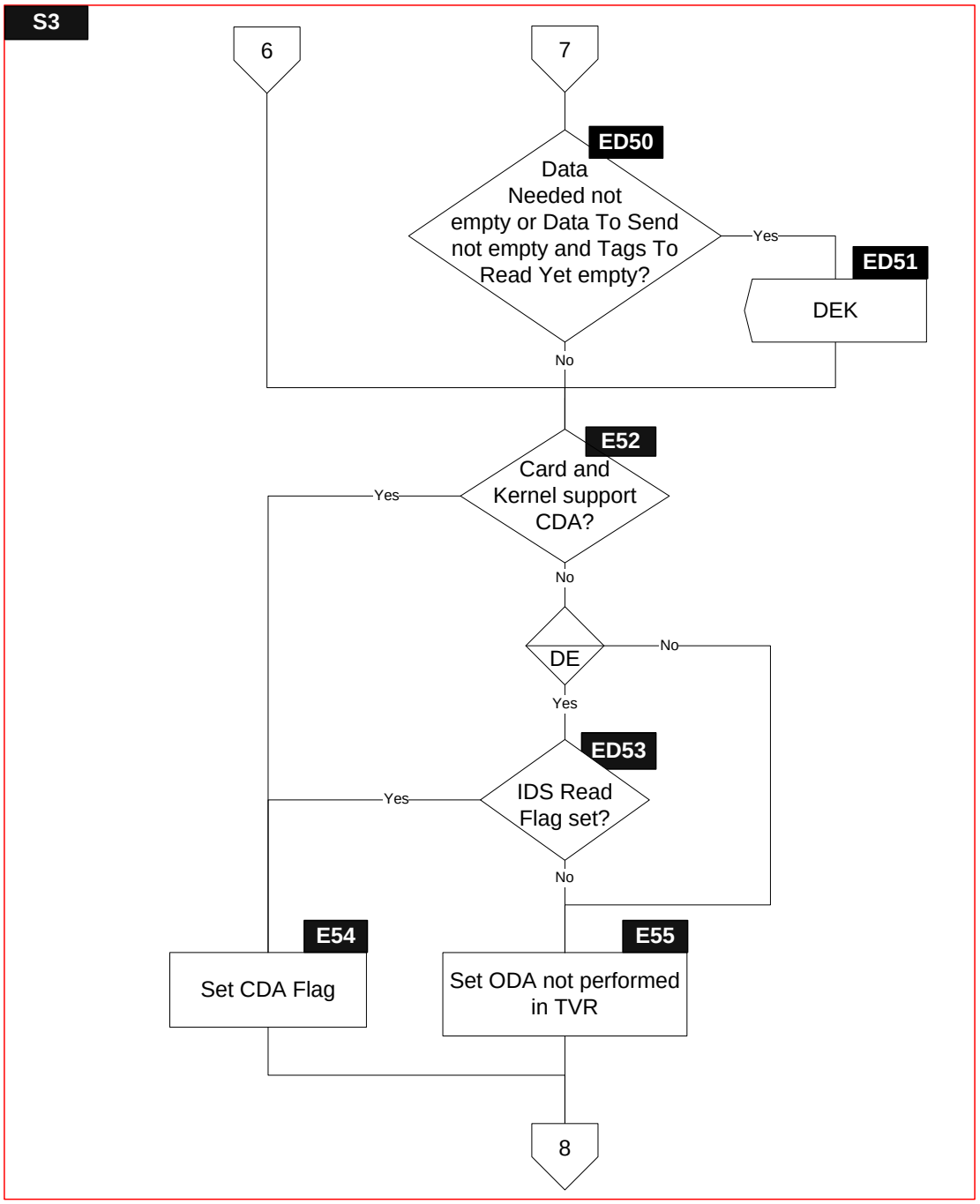


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 181

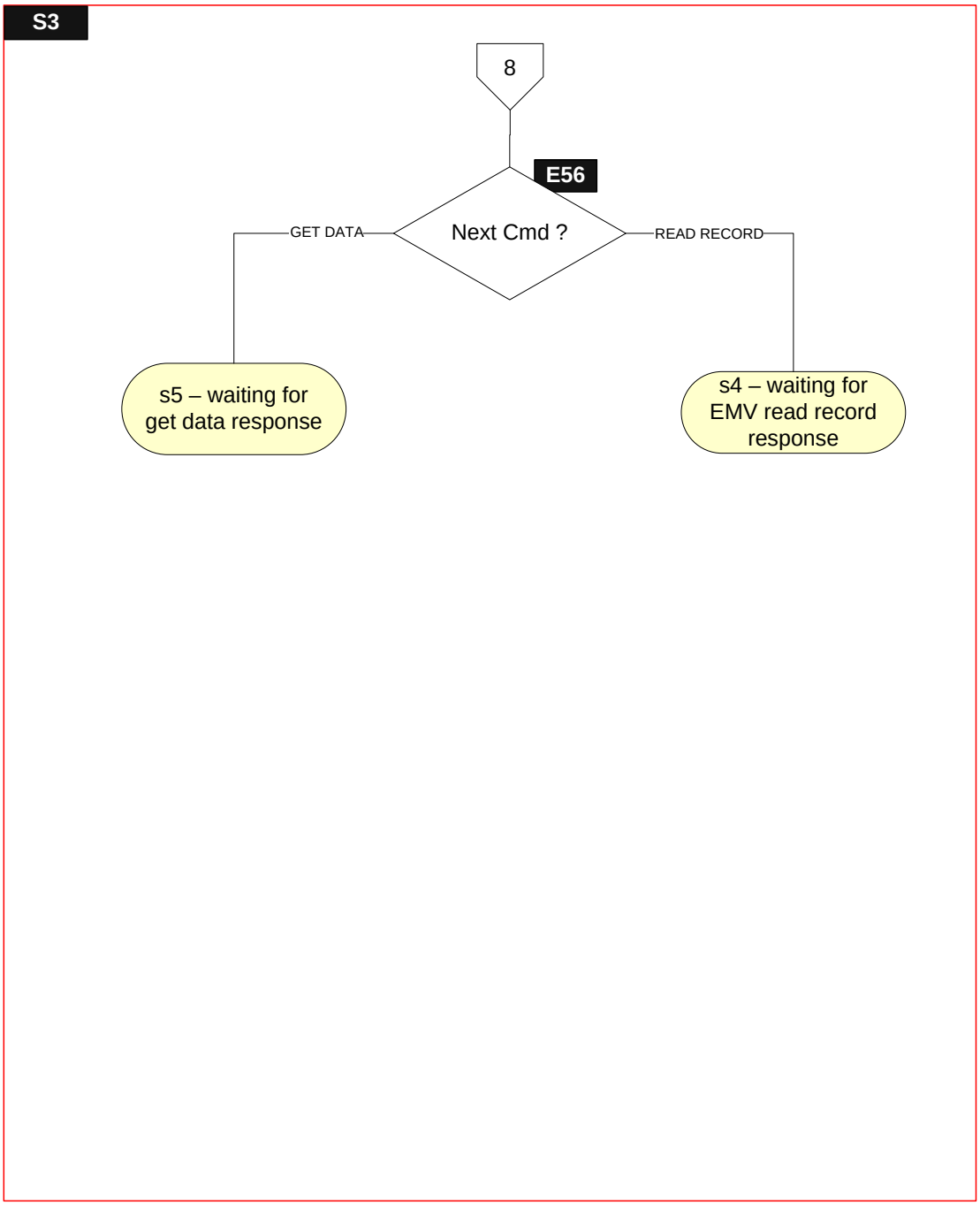


© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 182

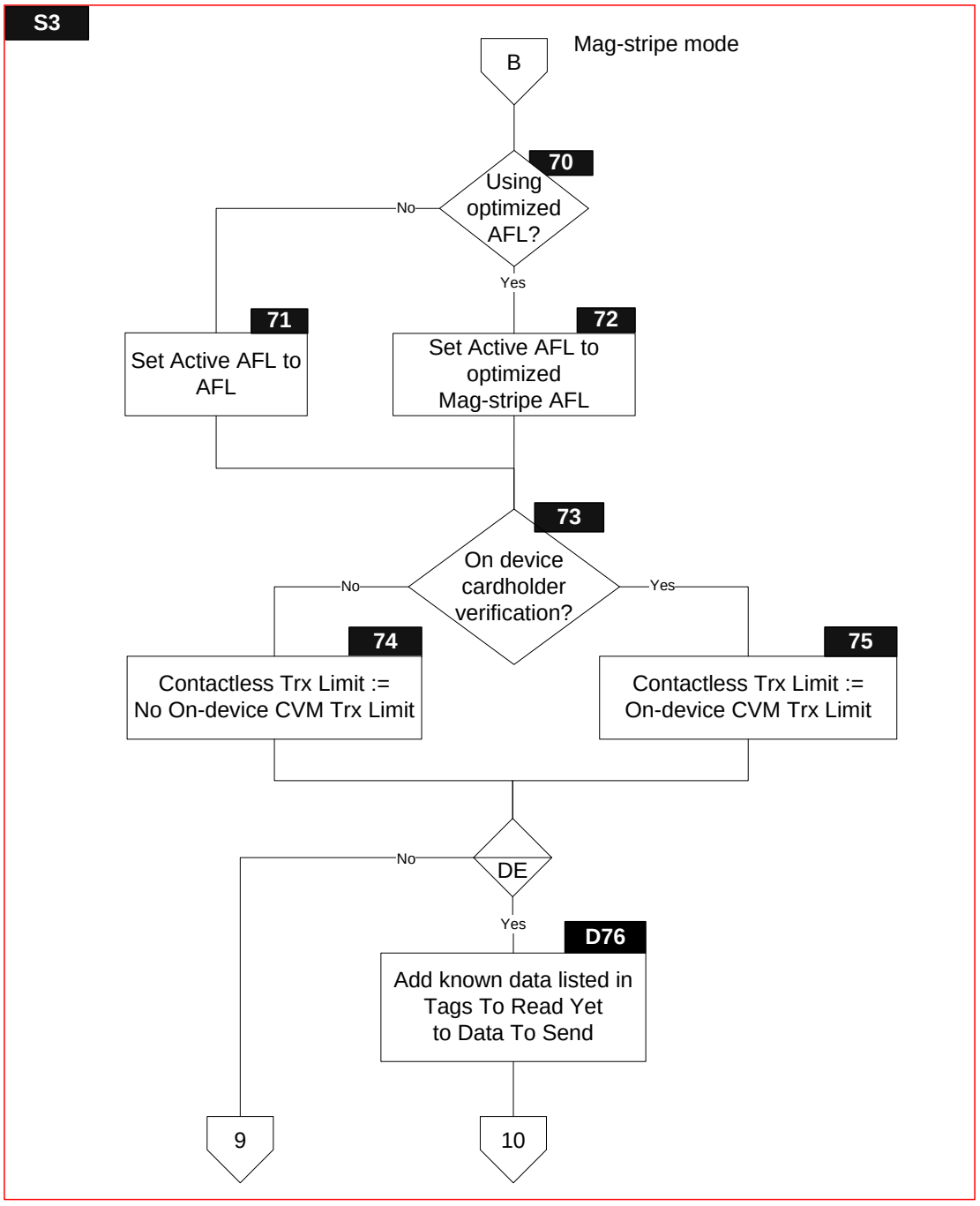


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 183

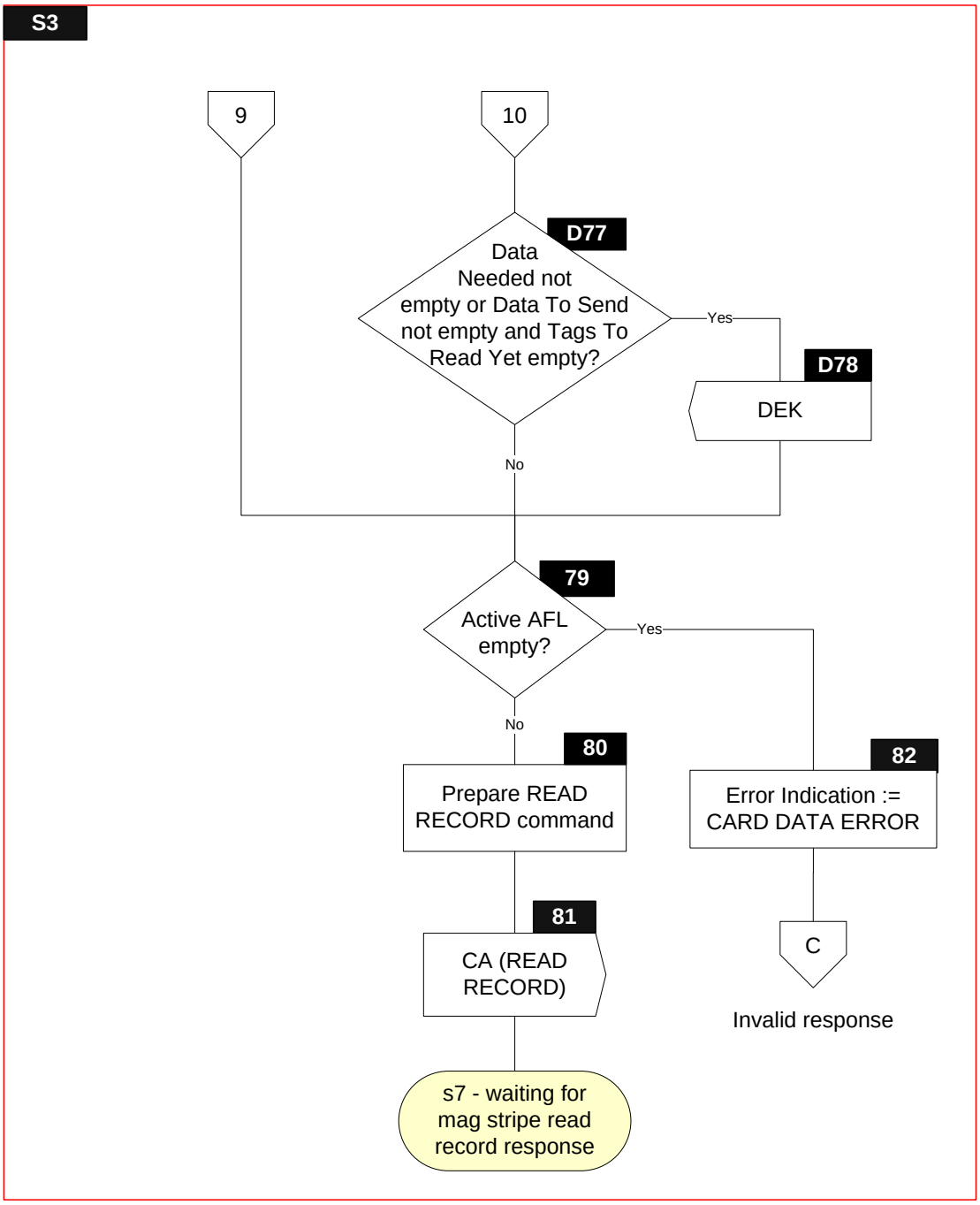


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 184

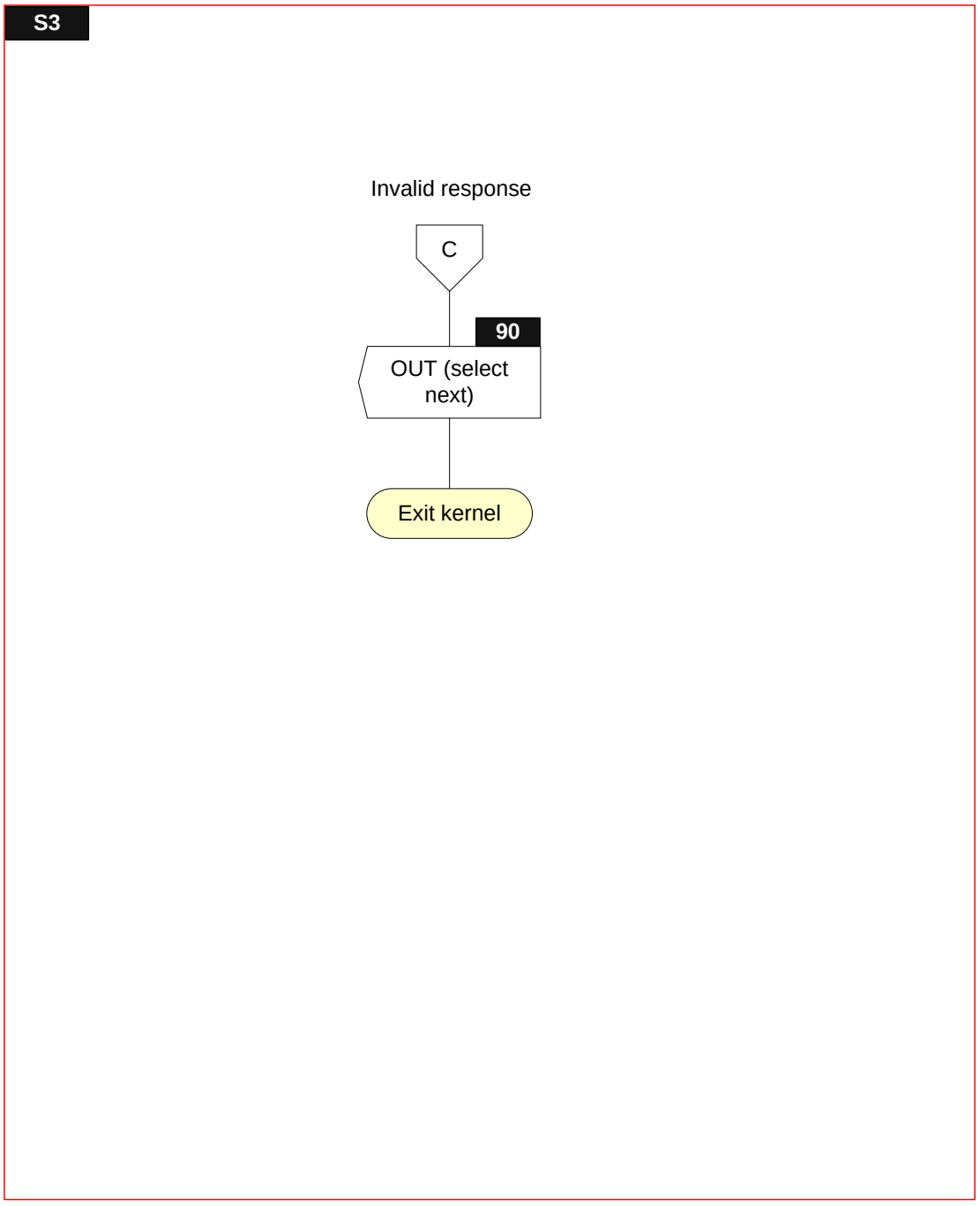


© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 185



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 186



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 187

6.5.3 Processing

S3.1

Receive RA signal with Response Message Data Field and SW12

S3.D2

Receive DET signal with Sync Data

S3.D3

UpdateWithDetData(Sync Data)

S3.4

Receive L1RSP signal with Return Code

S3.5

'Status' in *Outcome Parameter Set* := TRY AGAIN

'Start' in *Outcome Parameter Set* := B

'L1' in *Error Indication* := Return Code

Initialize(*Discretionary Data*)

AddToList(GetTLV(TagOf(*Error Indication*)), *Discretionary Data*)

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S3.6

Receive STOP signal

S3.7

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

Initialize(*Discretionary Data*)

AddToList(GetTLV(TagOf(*Error Indication*)), *Discretionary Data*)

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S3.8

IF [SW12 = '9000']

THEN

GOTO S3.10

ELSE

GOTO S3.9

ENDIF

S3.9

'L2' in *Error Indication* := STATUS BYTES

'SW12' in *Error Indication* := SW12

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 188

S3.10

Parsing Result := FALSE

IF [(Length of Response Message Data Field > 0) AND (Response Message Data Field[1] = '77')]

THEN

Parsing Result := ParseAndStoreCardResponse(Response Message Data Field)

ELSE

IF [(Length of Response Message Data Field > 0) AND (Response Message Data Field[1] = '80')]

THEN

Retrieve *Application Interchange Profile* and *Application File Locator* from Response Message Data Field according to section 5.5.3

Store *Application Interchange Profile* and *Application File Locator* in TLV Database

Set Parsing Result to TRUE if successful

ENDIF

ENDIF

S3.11

IF [Parsing Result]

THEN

GOTO S3.13

ELSE

GOTO S3.12

ENDIF

S3.12

'L2' in *Error Indication* := PARSING ERROR

S3.13

IF [IsNotEmpty(TagOf(*Application File Locator*)) AND IsNotEmpty(TagOf(*Application Interchange Profile*))]

THEN

GOTO S3.E15 or S3.70

ELSE

GOTO S3.14

ENDIF

S3.14

'L2' in *Error Indication* := CARD DATA MISSING

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 189

S3.E15

IF ['Only mag-stripe mode transactions supported' in *Kernel Configuration* is set]
THEN
GOTO S3.E17
ELSE
GOTO S3.E16
ENDIF

S3.E16

IF ['EMV mode is supported' in *Application Interchange Profile* is set]
THEN
GOTO S3.E30
ELSE
GOTO S3.E17
ENDIF

S3.E17

IF ['Only EMV mode transactions supported' in *Kernel Configuration* is set]
THEN
GOTO S3.E18
ELSE
GOTO S3.70
ENDIF

S3.E18

'L2' in *Error Indication* := MAGSTRIPE NOT SUPPORTED

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 190

EMV Mode

S3.E30

IF [(GetLength(TagOf(*Application File Locator*)) ≥ 4) AND (*Application File Locator*[1:4] = '08010100')]

THEN

GOTO S3.E32

ELSE

GOTO S3.E31

ENDIF

S3.E31

Active AFL := *Application File Locator*

S3.E32

Active AFL := *Application File Locator*[5:n], where n = GetLength(TagOf(*Application File Locator*))

S3.E33

IF ['On device cardholder verification is supported' in *Application Interchange Profile* is set]

THEN

GOTO S3.E35

ELSE

GOTO S3.E34

ENDIF

S3.E34

Reader Contactless Transaction Limit := *Reader Contactless Transaction Limit (No On-device CVM)*

S3.E35

Reader Contactless Transaction Limit := *Reader Contactless Transaction Limit (On-device CVM)*

S3.ED36

Active Tag := GetNextGetDataTagFromList(*Tags To Read Yet*)

IF [*Active Tag* = NULL]

THEN

GOTO S3.E40

ELSE

GOTO S3.ED37

ENDIF

S3.ED37

Build GET DATA command for *Active Tag* as defined in section 5.4

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 191

S3.ED38

Send CA(GET DATA) signal

S3.ED39

'Next Cmd' in *Next Cmd* := GET DATA

S3.E40

IF [*Active AFL* is empty]
THEN
GOTO S3.E41
ELSE
GOTO S3.E42
ENDIF

S3.E41

'L2' in *Error Indication* := CARD DATA ERROR

S3.E42

Build READ RECORD command for the first record indicated by *Active AFL* as defined in section 5.7

S3.E43

Send CA(READ RECORD) signal

S3.E44

'Next Cmd' in *Next Cmd* := READ RECORD

S3.ED45

IF ['Read' in *IDS Status* is set]
THEN
GOTO S3.ED46
ELSE
GOTO S3.ED49
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 192

S3.ED46

IF [IsNotEmpty(TagOf(*DS Slot Availability*))]
THEN
AddToList(GetTLV(TagOf(*DS Slot Availability*)), *Data To Send*)
ENDIF
IF [IsNotEmpty(TagOf(*DS Summary 1*))]
THEN
AddToList(GetTLV(TagOf(*DS Summary 1*)), *Data To Send*)
ENDIF
IF [IsNotEmpty(TagOf(*DS Unpredictable Number*))]
THEN
AddToList(GetTLV(TagOf(*DS Unpredictable Number*)), *Data To Send*)
ENDIF
IF [IsNotEmpty(TagOf(*DS Slot Management Control*))]
THEN
AddToList(GetTLV(TagOf(*DS Slot Management Control*)), *Data To Send*)
ENDIF
IF [IsPresent(TagOf(*DS ODS Card*))]
THEN
AddToList(GetTLV(TagOf(*DS ODS Card*)), *Data To Send*)
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 193

S3.ED47

Continue with IDS when:

- *DS Requested Operator ID* is not known by the Card, but all the necessary data objects are returned by the Card to perform an IDS write, or
- *DS Requested Operator ID* is known by the Card

This is done as follows:

```
IF      [(IsNotEmpty(TagOf(DS Slot Availability)) AND
          IsNotEmpty(TagOf(DS Summary 1)) AND
          IsNotEmpty(TagOf(DS Unpredictable Number)) AND
          IsNotPresent(TagOf (DS ODS Card)))
        OR
         (IsNotEmpty(TagOf(DS Summary 1)) AND
          IsPresent(TagOf (DS ODS Card))) ]
THEN
        GOTO S3.ED49
ELSE
        GOTO S3.ED48
ENDIF
```

S3.ED48

CLEAR 'Read' in *IDS Status*

S3.ED49

```
FOR every entry T in Tags To Read Yet
{
        IF      [IsNotEmpty(T)]
        THEN
                AddToList(GetTLV(T), Data To Send)
                RemoveFromList(T, Tags To Read Yet)
        ENDIF
}
```

S3.ED50

```
IF      [IsNotEmptyList(Data Needed) OR
         (IsNotEmptyList(Data To Send) AND IsEmptyList(Tags To Read Yet))]
THEN
        GOTO S3.ED51
ELSE
        GOTO S3.E52
ENDIF
```


© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 194

S3.ED51

Send DEK(*Data To Send*, *Data Needed*) signal

Initialize(*Data To Send*)

Initialize(*Data Needed*)

S3.E52

IF ['CDA supported' in *Application Interchange Profile* is set
AND 'CDA' in *Terminal Capabilities* is set)]

THEN

GOTO S3.E54

ELSE

GOTO S3.ED53 or GOTO S3.E55

ENDIF

S3.ED53

IF ['Read' in *IDS Status* is set]

THEN

GOTO S3.E54

ELSE

GOTO S3.E55

ENDIF

S3.E54

SET 'CDA' in *ODA Status*

S3.E55

SET 'Offline data authentication was not performed' in *Terminal Verification Results*

S3.E56

IF ['Next Cmd' in *Next Cmd* = READ RECORD]

THEN

GOTO `s4 – waiting for EMV read record response`

ELSE

GOTO `s5 – waiting for get data response`

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 195

Mag-stripe Mode

S3.70

IF [(GetLength(TagOf(*Application File Locator*)) ≥ 4) AND
(*Application File Locator*[1:4] = '08010100')]

THEN

GOTO S3.72

ELSE

GOTO S3.71

ENDIF

S3.71

Active AFL := *Application File Locator*

S3.72

Active AFL := *Application File Locator*[1:4]

S3.73

IF ['On device cardholder verification is supported' in *Application Interchange Profile* is set]

THEN

GOTO S3.75

ELSE

GOTO S3.74

ENDIF

S3.74

Reader Contactless Transaction Limit := *Reader Contactless Transaction Limit (No On-device CVM)*

S3.75

Reader Contactless Transaction Limit := *Reader Contactless Transaction Limit (On-device CVM)*

S3.D76

FOR every entry T in *Tags To Read Yet*

{

IF [IsNotEmpty(T)]

THEN

AddToList(GetTLV(T), *Data To Send*)

RemoveFromList(T, *Tags To Read Yet*)

ENDIF

}

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 196

S3.D77

IF [IsNotEmptyList(*Data Needed*) OR
(IsNotEmptyList(*Data To Send*) AND IsEmptyList(*Tags To Read Yet*))]

THEN

GOTO S3.D78

ELSE

GOTO S3.79

ENDIF

S3.D78

Send DEK(*Data To Send*, *Data Needed*) signal

Initialize(*Data To Send*)

Initialize(*Data Needed*)

S3.79

IF [*Active AFL* is empty]

THEN

GOTO S3.82

ELSE

GOTO S3.80

ENDIF

S3.80

Build command data for READ RECORD for the first record indicated by *Active AFL* as defined in section 5.7

S3.81

Send CA(READ RECORD) signal

S3.82

'L2' in *Error Indication* := CARD DATA ERROR

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 197

Invalid Response

S3.90

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

Initialize(*Discretionary Data*)

AddToList(GetTLV(TagOf(*Error Indication*)), *Discretionary Data*)

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 198

6.6 State 4 – Waiting for EMV Read Record Response

6.6.1 Local Variables

Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Sync Data	var.	b	List of data objects returned with DET signal
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Record	var. up to 256	b	Response Message Data Field of the R-APDU of READ RECORD
Signed Flag	1	b	Boolean used to indicate if current record is signed
Sfi	1	b	SFI of current record
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 253	b	Value of TLV encoded string

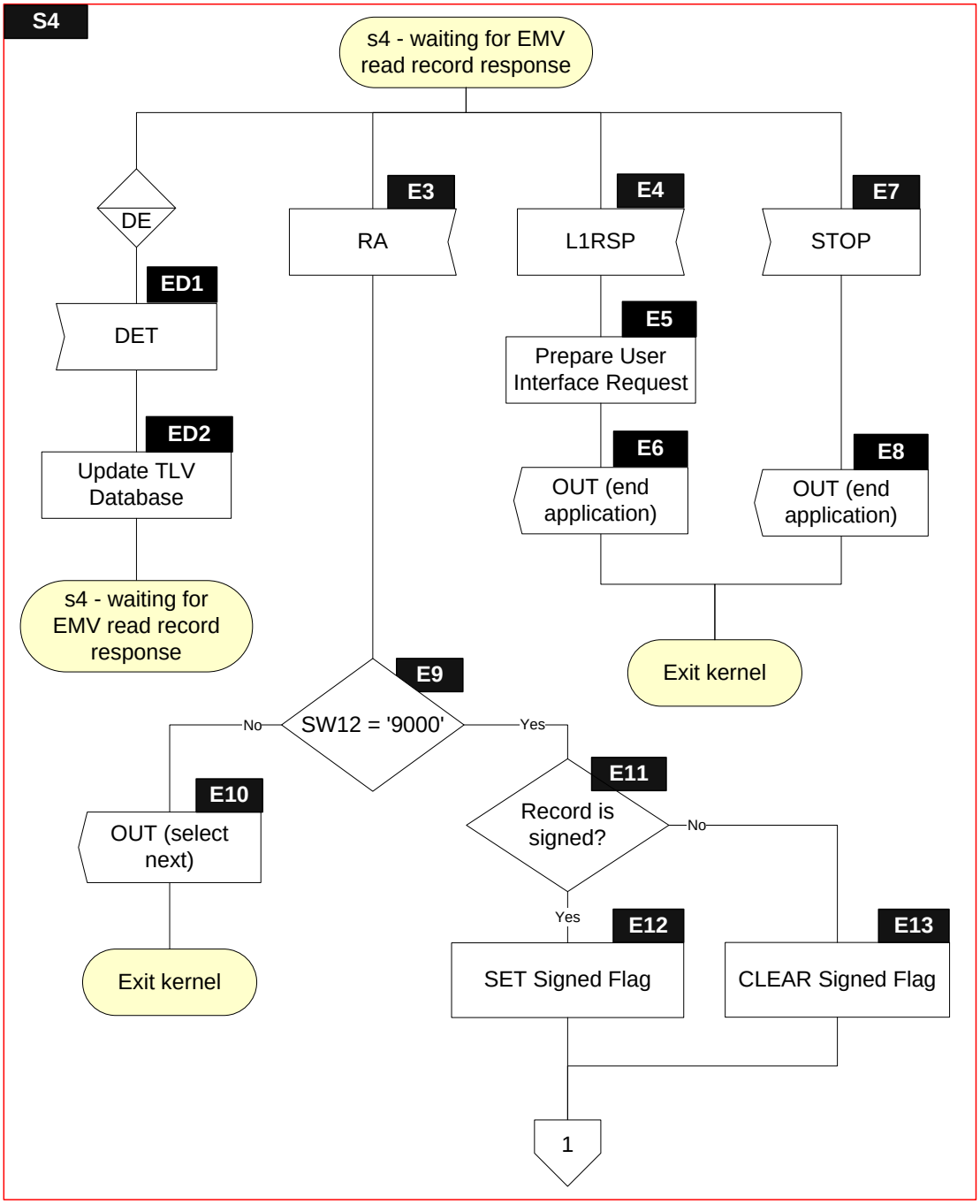
6.6.2 Flow Diagram

Figure 6.5 shows the flow diagram of `s4 – waiting for EMV read record response`. Symbols in this diagram are labeled S4.X.

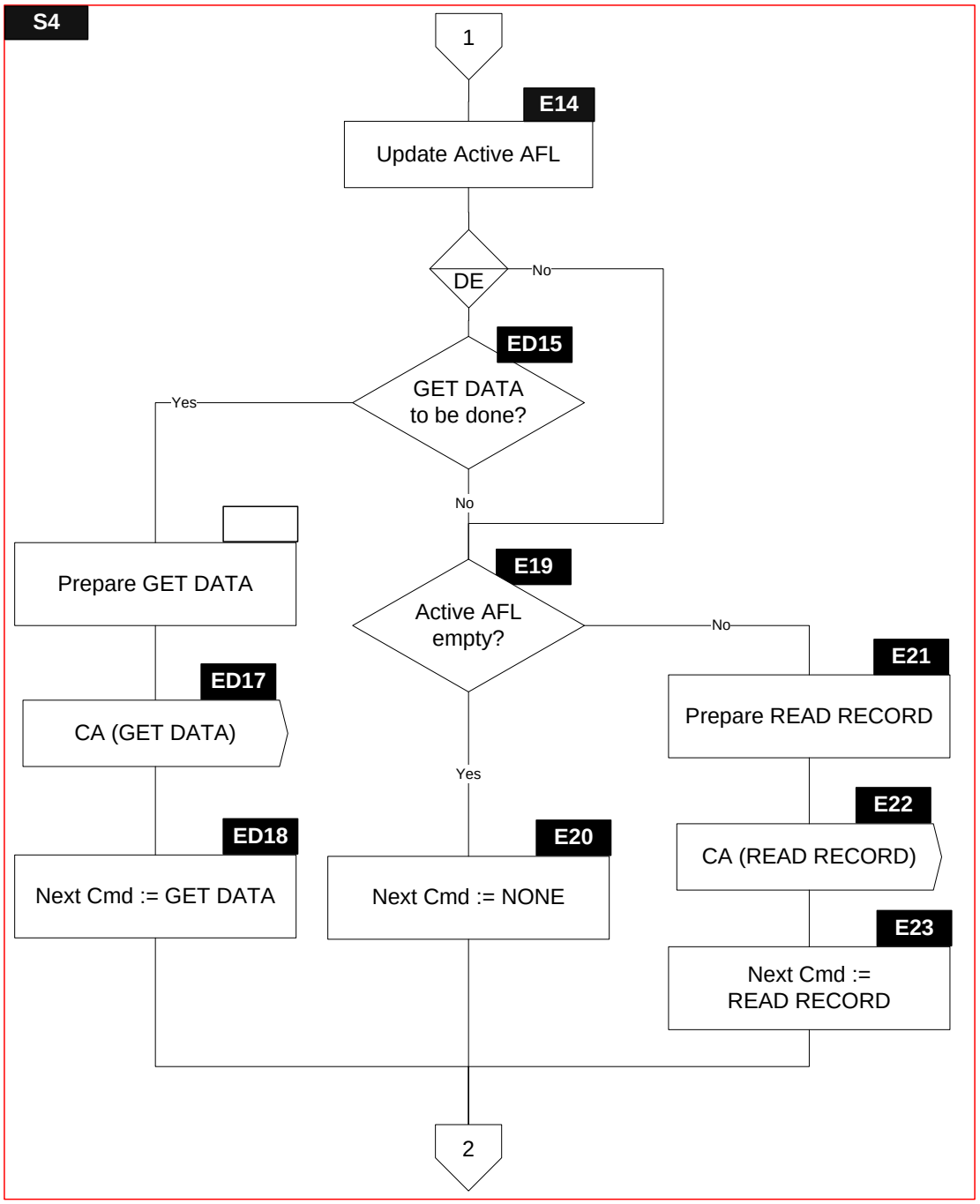
Note: The preparation of data read from the Card for offline data authentication begins in the following flow diagram. While the implementer may follow the steps described here, it is also possible to optimize the process as described in Annex C.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 199

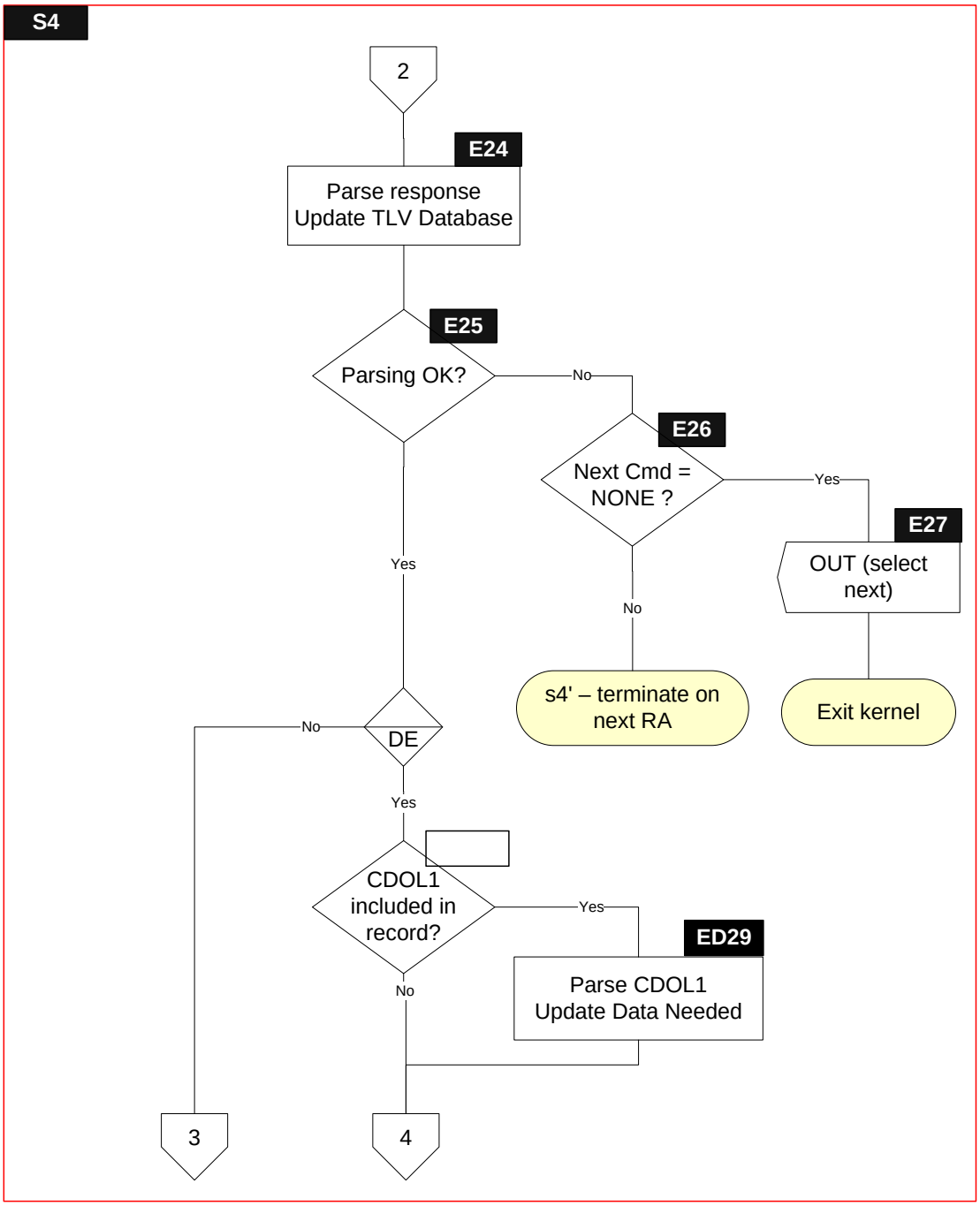
Figure 6.5—State 4 Flow Diagram



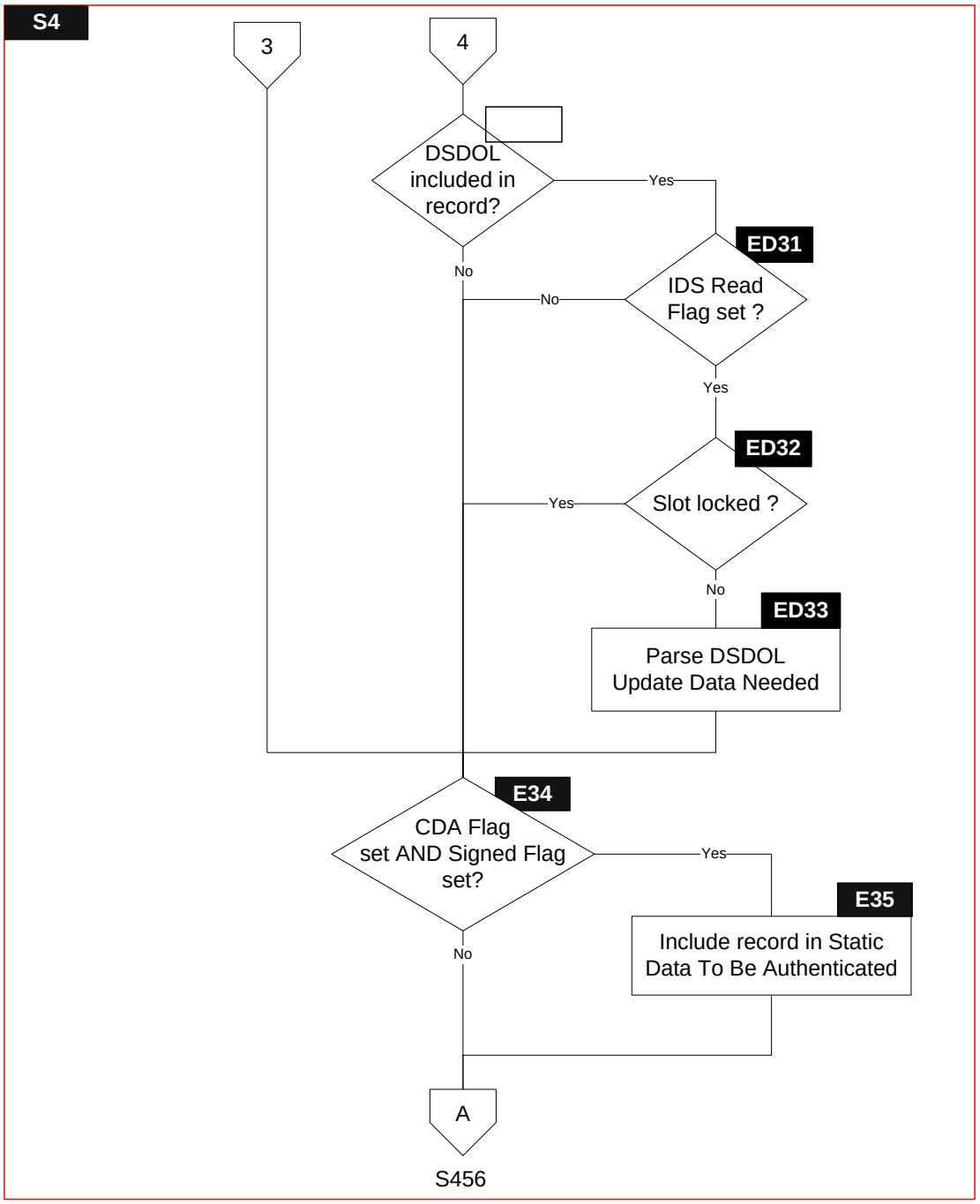
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 200



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 201



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 202



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 203

6.6.3 Processing

S4.ED1

Receive DET signal with Sync Data

S4.ED2

UpdateWithDetData(Sync Data)

S4.E3

Receive RA signal with Record and SW12

S4.E4

Receive L1RSP signal with Return Code

S4.E5

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S4.E6

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S4.E7

Receive STOP signal

S4.E8

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S4.E9

IF [SW12 = '9000']

THEN

GOTO S4.E11

ELSE

GOTO S4.E10

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 204

S4.E10

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := STATUS BYTES

'SW12' in *Error Indication* := SW12

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S4.E11

IF [*Active AFL* indicates first record (i.e. current record) is signed]

THEN

GOTO S4.E12

ELSE

GOTO S4.E13

ENDIF

S4.E12

SET Signed Flag

S4.E13

CLEAR Signed Flag

S4.E14

Sfi := SFI of first record in *Active AFL*

Remove first record from *Active AFL*

S4.ED15

Active Tag := GetNextGetDataTagFromList (*Tags To Read Yet*)

IF [*Active Tag* is not NULL]

THEN

GOTO S4.ED16

ELSE

GOTO S4.E19

ENDIF

S4.ED16

Prepare GET DATA command for *Active Tag* as specified in section 5.4

S4.ED17

Send CA(GET DATA command) signal

S4.ED18

'Next Cmd' in *Next Cmd* := GET DATA

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 205

S4.E19

IF [*Active AFL* is empty]
THEN
GOTO S4.E20
ELSE
GOTO S4.E21
ENDIF

S4.E20

'Next Cmd' in *Next Cmd* := NONE

S4.E21

Prepare READ RECORD command for first record in *Active AFL* as specified in section 5.7

S4.E22

Send CA(READ RECORD command) signal

S4.E23

'Next Cmd' in *Next Cmd* := READ RECORD

S4.E24

IF [Sfi ≤ 10]
THEN
IF [(Length of Record > 0) AND (Record[1] = '70')]
THEN
Parsing Result := ParseAndStoreCardResponse(Record)
ELSE
Parsing Result := FALSE
ENDIF
ELSE
Processing of records in proprietary files is beyond the scope of this specification
ENDIF

S4.E25

IF [Parsing Result]
THEN
GOTO S4.ED28 or S4.E34
ELSE
GOTO S4.E26
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 206

S4.E26

IF ['Next Cmd' in *Next Cmd* = NONE]

THEN

GOTO S4.E27

ELSE

GOTO `s4' – terminate on next RA`

ENDIF

S4.E27

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := PARSING ERROR

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S4.ED28

IF [Record includes data object with tag equal to TagOf(*CDOL1*)]

THEN

GOTO S4.ED29

ELSE

GOTO S4.ED30

ENDIF

S4.ED29

FOR every TL in *CDOL1*

{

IF [IsEmpty(T)]

THEN

AddToList(T, *Data Needed*)

ENDIF

}

S4.ED30

IF [Record includes data object with tag equal to TagOf(*DSDOL*)]

THEN

GOTO S4.ED31

ELSE

GOTO S4.E34

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 207

S4.ED31

IF ['Read' in *IDS Status* is set]
THEN
    GOTO S4.ED32
ELSE
    GOTO S4.E34
ENDIF

S4.ED32

IF [IsNotEmpty(*DS Slot Management Control*) AND
  'Locked slot' in *DS Slot Management Control* is set]
THEN
    GOTO S4.E34
ELSE
    GOTO S4.ED33
ENDIF

S4.ED33

FOR every TL in *DSDOL*
{
    IF [IsEmpty(T)]
    THEN
        AddToList(T, *Data Needed*)
    ENDIF
}

S4.E34

IF [Signed Flag AND 'CDA' in *ODA Status* is set]
THEN
    GOTO S4.E35
ELSE
    GOTO S456.E1
ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 208

S4.E35

IF [Sfi ≤ 10]

THEN

IF [Enough space left in *Static Data To Be Authenticated* to append Record (without tag '70' and length)]

THEN

Append Record (excluding tag '70' and length) at the end of *Static Data To Be Authenticated*

ELSE

SET 'CDA failed' in *Terminal Verification Results*

ENDIF

ELSE

IF [(Record[1] = '70') AND
Record is TLV encoded AND
Enough space left in *Static Data To Be Authenticated* to append Record]

THEN

Append Record (including tag '70' and length) at the end of *Static Data To Be Authenticated*

ELSE

SET 'CDA failed' in *Terminal Verification Results*

ENDIF

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 209

6.7 State 4' – Terminate on Next RA

6.7.1 Local Variables

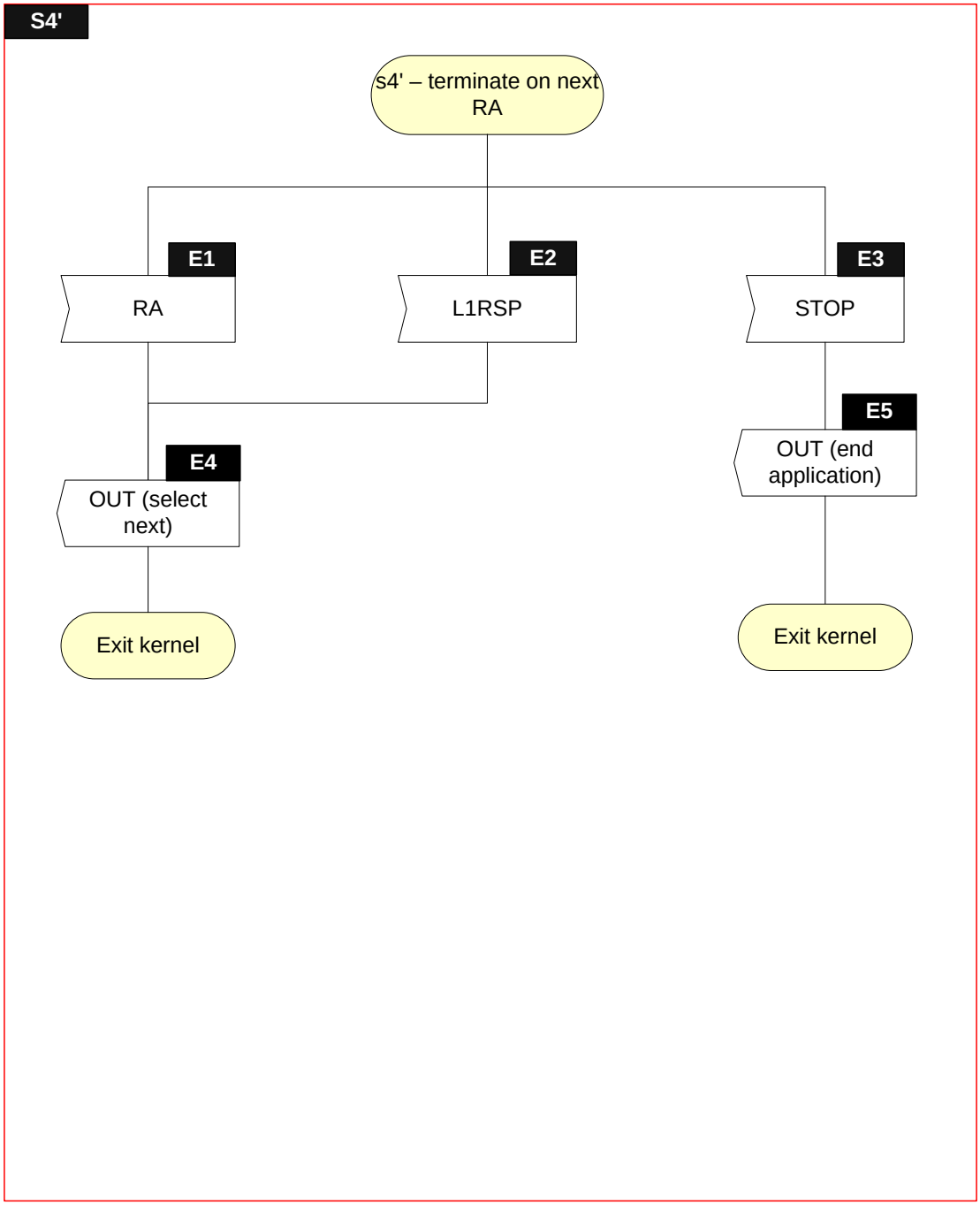
None

6.7.2 Flow Diagram

Figure 6.6 shows the flow diagram of `s4' – terminate on next RA`. Symbols in this diagram are labeled S4'.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 210

Figure 6.6—State 4' Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 211

6.7.3 Processing

S4'.E1

Receive RA signal

S4'.E2

Receive L1RSP signal

S4'.E3

Receive STOP signal

S4'.E4

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := PARSING ERROR

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S4'.E5

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 212

6.8 State 5 – Waiting for Get Data Response

6.8.1 Local Variables

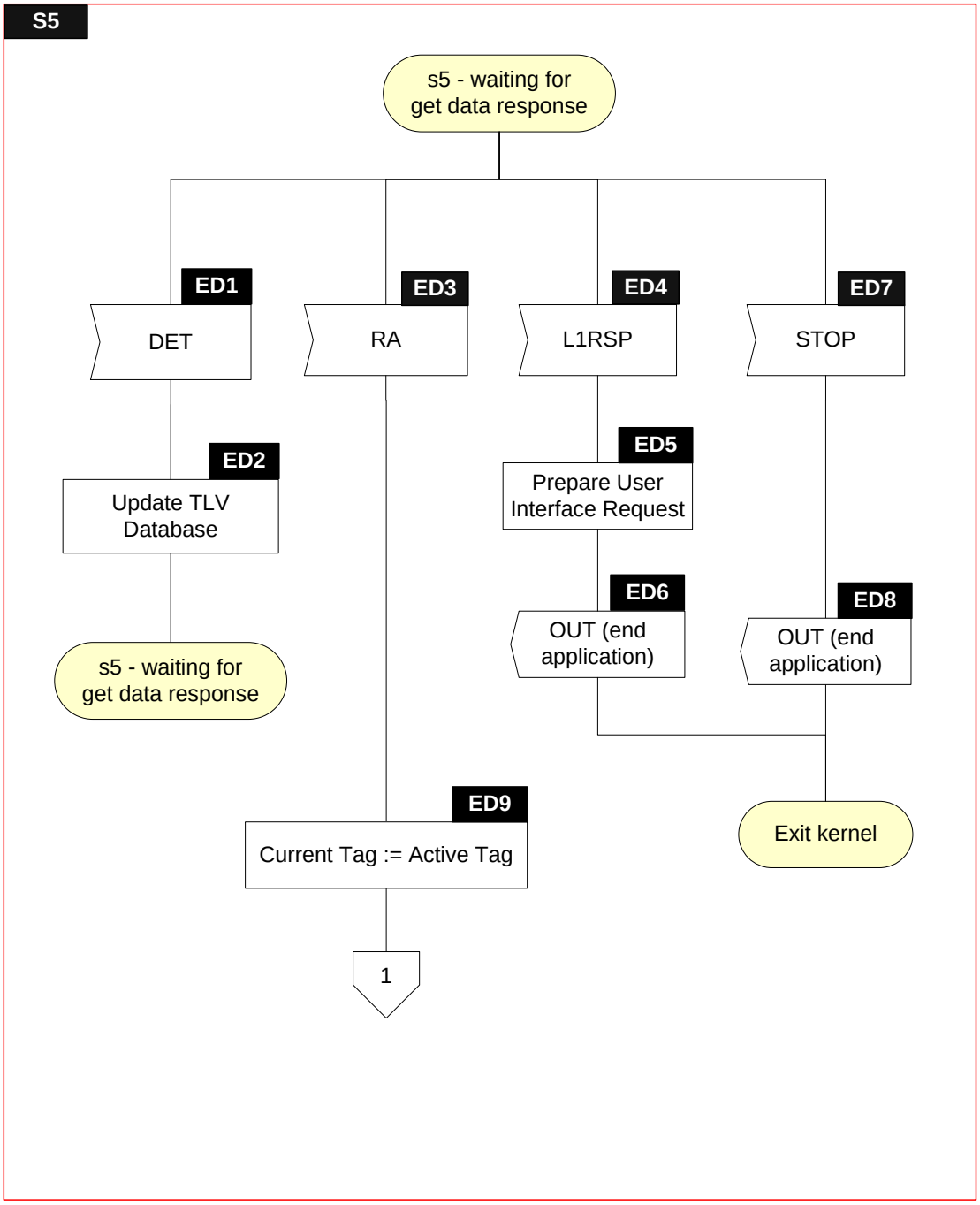
sName	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Sync Data	var.	b	List of data objects returned with DET signal
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of GET DATA
Current Tag	var.	b	Tag indicating the tag of the current GET DATA
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 252	b	Value of TLV encoded string

6.8.2 Flow Diagram

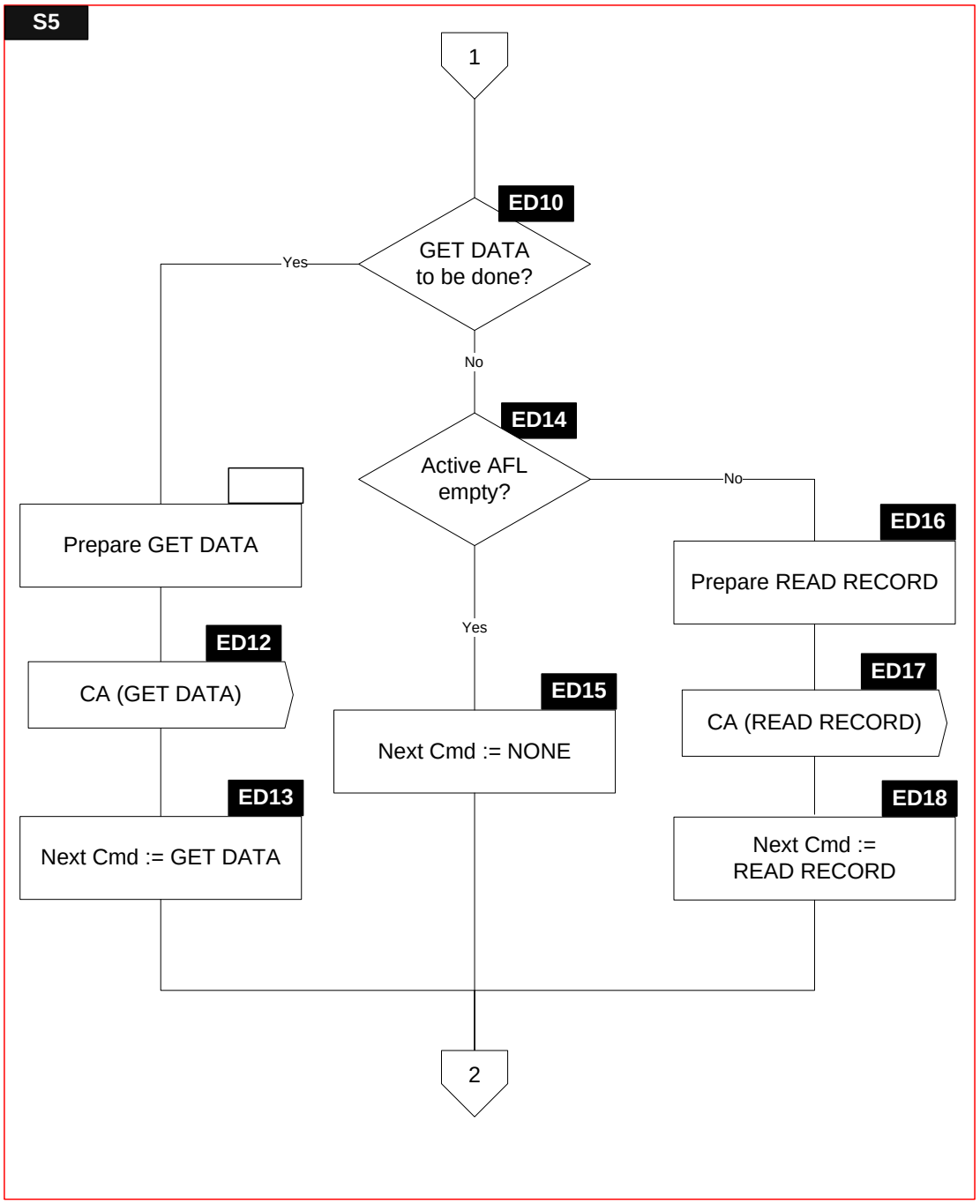
Figure 6.7 shows the flow diagram of `s5 – waiting for get data response`. Symbols in this diagram are labeled S5.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 213

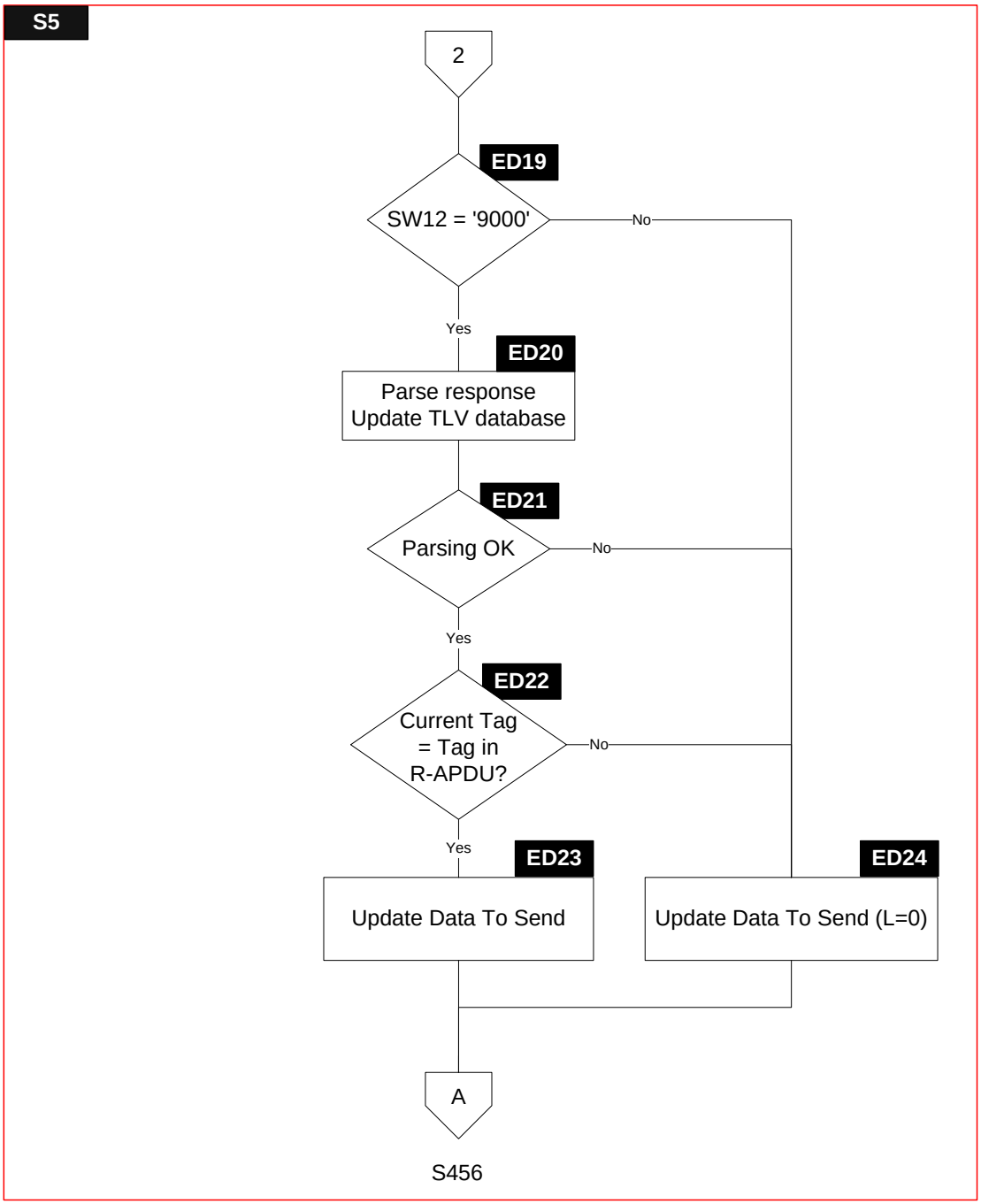
Figure 6.7—State 5 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 214



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 215



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 216

6.8.3 Processing

S5.ED1

Receive DET signal with Sync Data

S5.ED2

UpdateWithDetData(Sync Data)

S5.ED3

Receive RA signal with Response Message Data Field and SW12

S5.ED4

Receive L1RSP signal with Return Code

S5.ED5

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S5.ED6

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S5.ED7

Receive STOP signal

S5.ED8

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S5.ED9

Current Tag := *Active Tag*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 217

S5.ED10

Active Tag := GetNextGetDataTagFromList (*Tags To Read Yet*)

IF [*Active Tag* is not NULL]

THEN

GOTO S5.ED11

ELSE

GOTO S5.ED14

ENDIF

S5.ED11

Prepare GET DATA command for *Active Tag* as specified in section 5.4

S5.ED12

Send CA(GET DATA command) signal

S5.ED13

'Next Cmd' in *Next Cmd* := GET DATA

S5.ED14

IF [*Active AFL* is empty]

THEN

GOTO S5.ED15

ELSE

GOTO S5.ED16

ENDIF

S5.ED15

'Next Cmd' in *Next Cmd* := NONE

S5.ED16

Prepare READ RECORD command for first record in *Active AFL* as specified in section 5.7

S5.ED17

Send CA(READ RECORD command) signal

S5.ED18

'Next Cmd' in *Next Cmd* := READ RECORD

S5.ED19

IF [SW12 = '9000']

THEN

GOTO S5.ED20

ELSE

GOTO S5.ED24

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 218

S5.ED20

Parsing Result := ParseAndStoreCardResponse(Response Message Data Field)

Retrieve T, L and V from Response Message Data Field

Table 6.1—Response Message Data Field

T	L	V

S5.ED21

IF [Parsing Result]
THEN
GOTO S5.ED22
ELSE
GOTO S5.ED24
ENDIF

S5.ED22

IF [Current Tag = T]
THEN
GOTO S5.ED23
ELSE
GOTO S5.ED24
ENDIF

S5.ED23

AddToList(TLV, *Data To Send*)

S5.ED24

Add Current Tag with zero length to *Data To Send*:
AddToList(Current Tag || '00', *Data To Send*)

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 219

6.9 State 6 – Waiting for EMV Mode First Write Flag

6.9.1 Local Variables

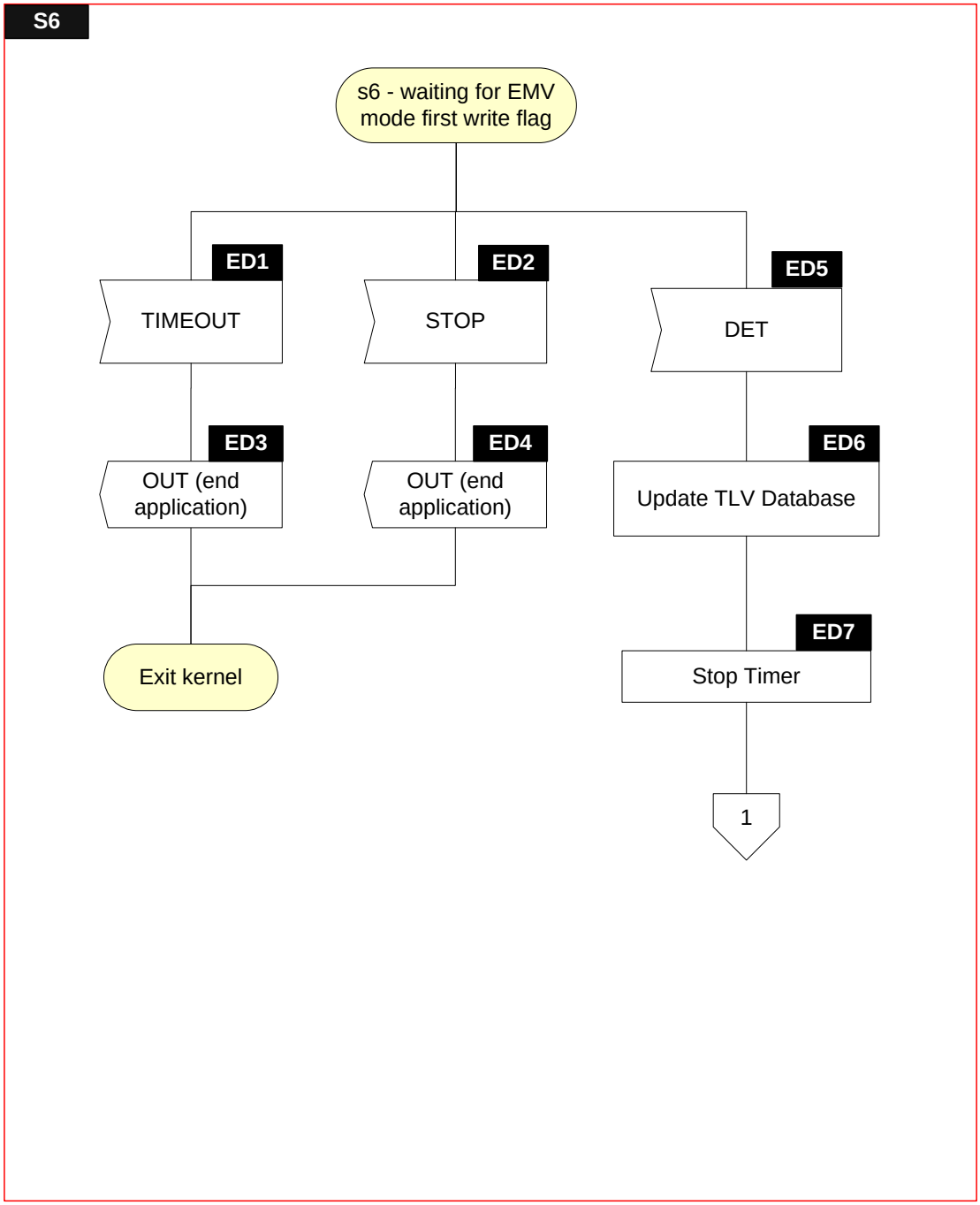
Name	Length	Type	Description
Sync Data	var.	b	List of data objects returned with DET signal
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 252	b	Value of TLV encoded string

6.9.2 Flow Diagram

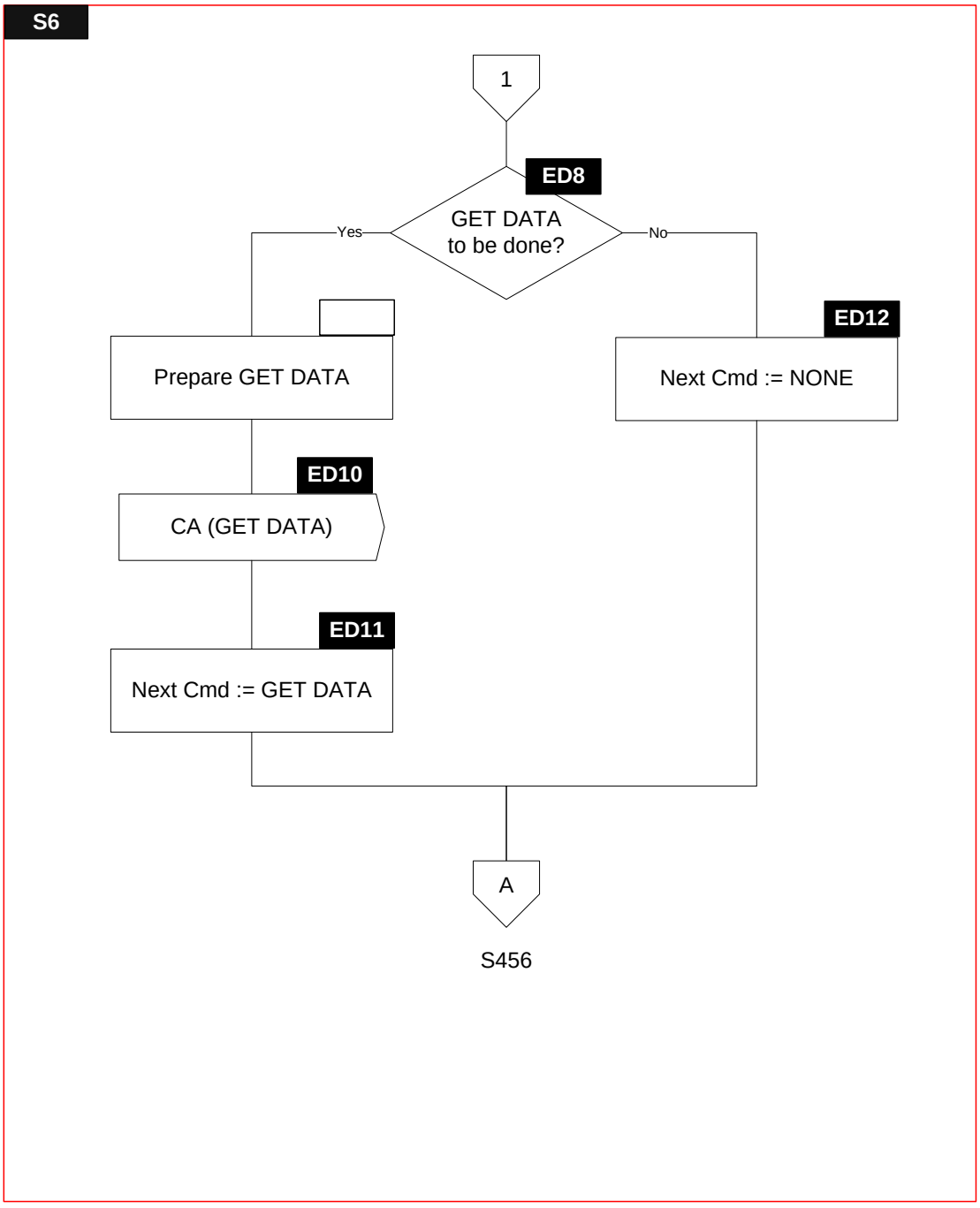
Figure 6.8 shows the flow diagram of `s6 – waiting for EMV mode first write flag`. Symbols in this diagram are labeled S6.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 220

Figure 6.8—State 6 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 221



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 222

6.9.3 Processing

S6.ED1

Receive TIMEOUT signal

S6.ED2

Receive STOP signal

S6.ED3

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := TIME OUT

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S6.ED4

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S6.ED5

Receive DET signal with Sync Data

S6.ED6

UpdateWithDetData(Sync Data)

S6.ED7

Stop Timer

S6.ED8

Active Tag := GetNextGetDataTagFromList (*Tags To Read Yet*)

IF [*Active Tag* is not NULL]

THEN

GOTO S6.ED9

ELSE

GOTO S6.ED12

ENDIF

S6.ED9

Prepare GET DATA command for *Active Tag* as specified in section 5.4

S6.ED10

Send CA(GET DATA command) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 223

S6.ED11

'Next Cmd' in *Next Cmd* := GET DATA

S6.ED12

'Next Cmd' in *Next Cmd* := NONE

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 224

6.10 States 4, 5, and 6 – Common Processing

6.10.1 Local Variables

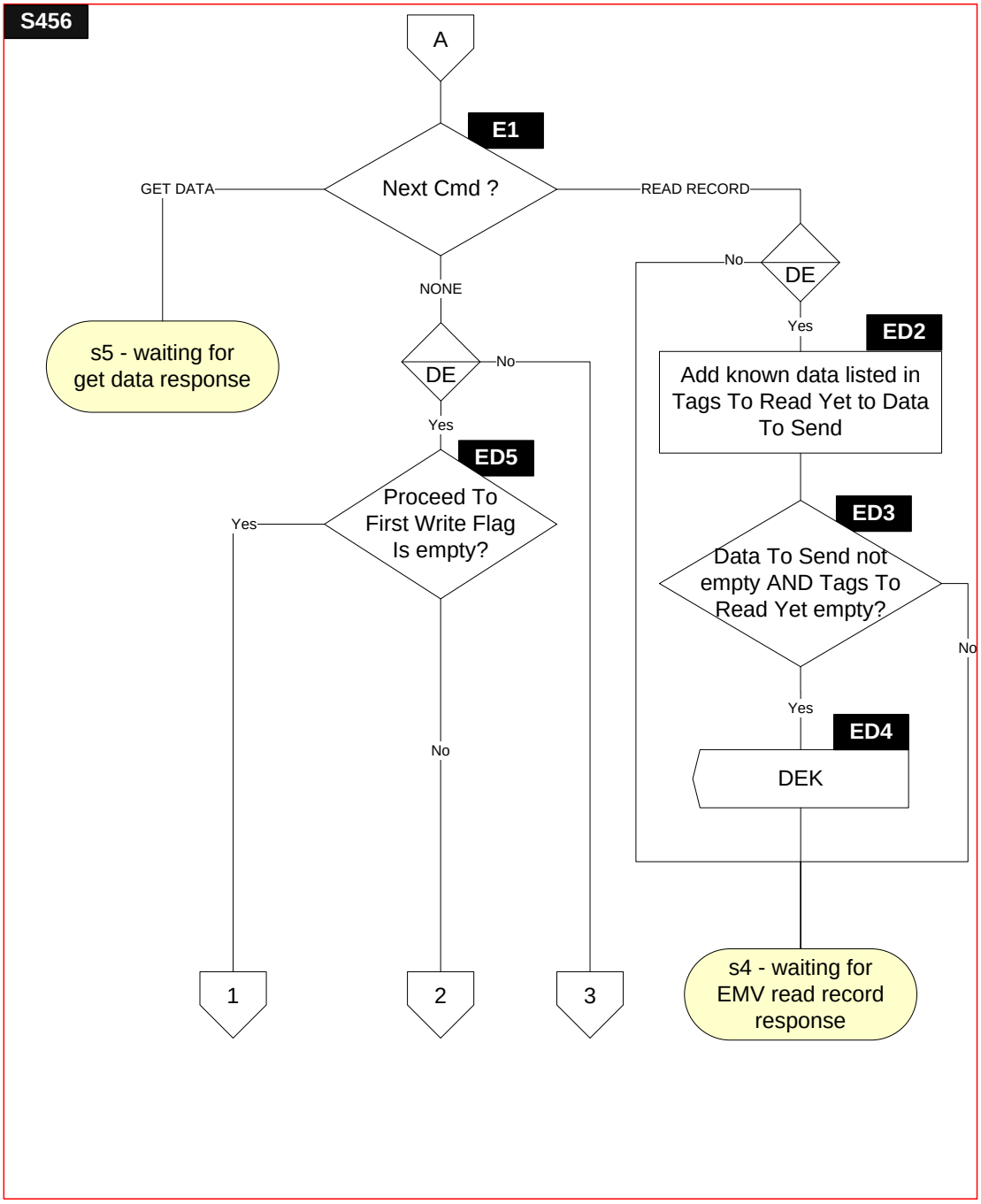
Local variables for common processing are defined in states 4, 5, and 6.

6.10.2 Flow Diagram

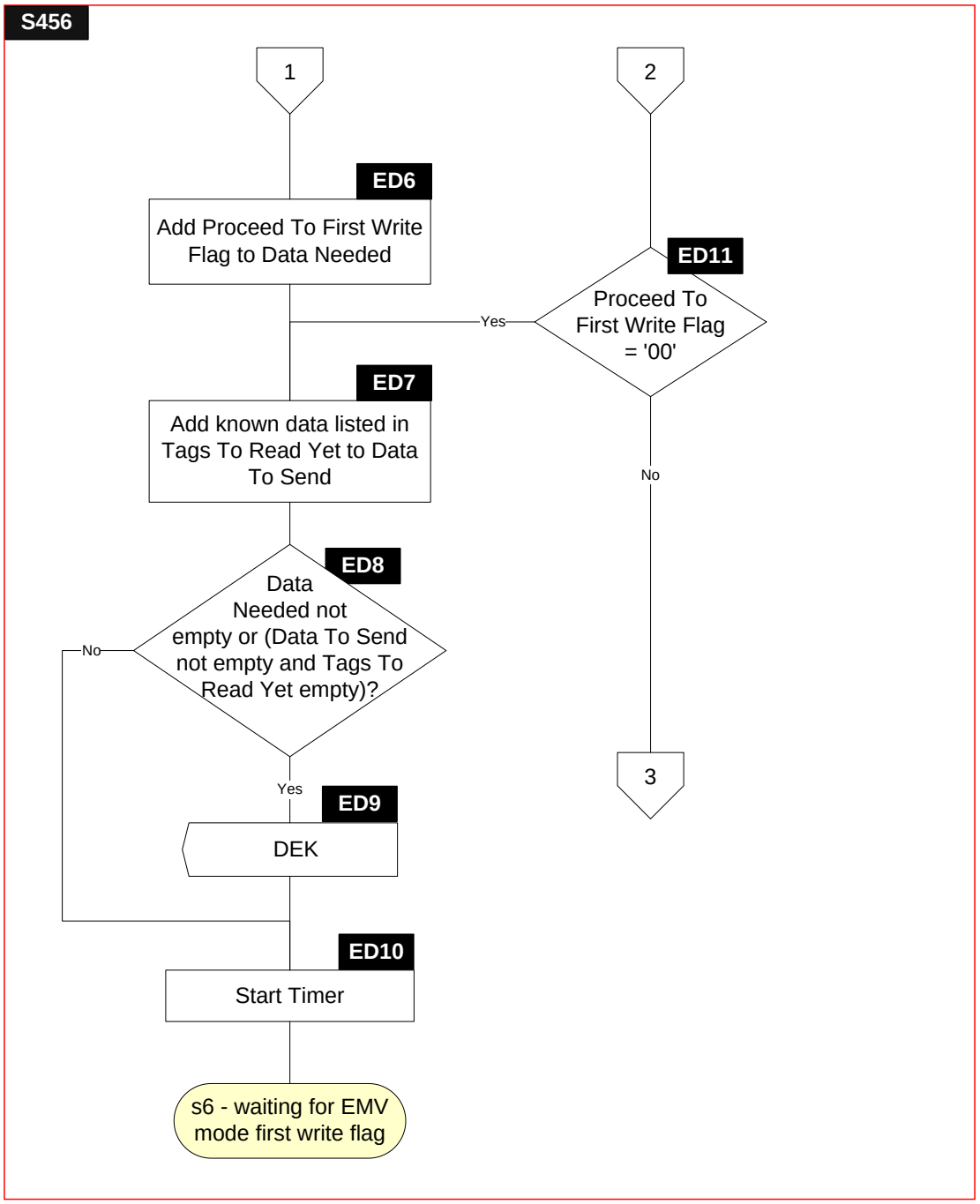
Figure 6.9 shows the flow diagram for common processing between states 4, 5, and 6. Symbols in this diagram are labeled S456.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 225

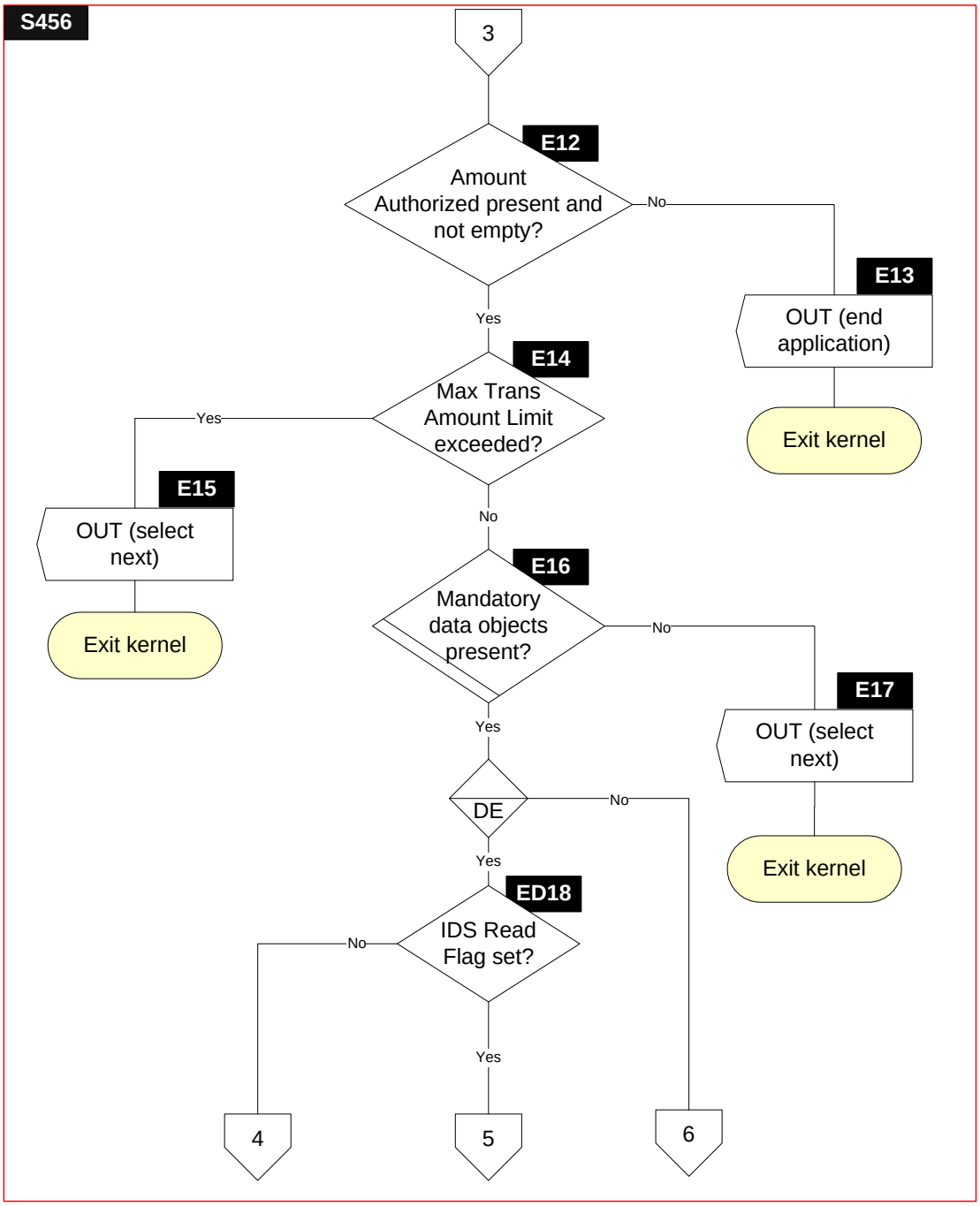
Figure 6.9—States 4, 5, and 6 – Common Processing – Flow Diagram



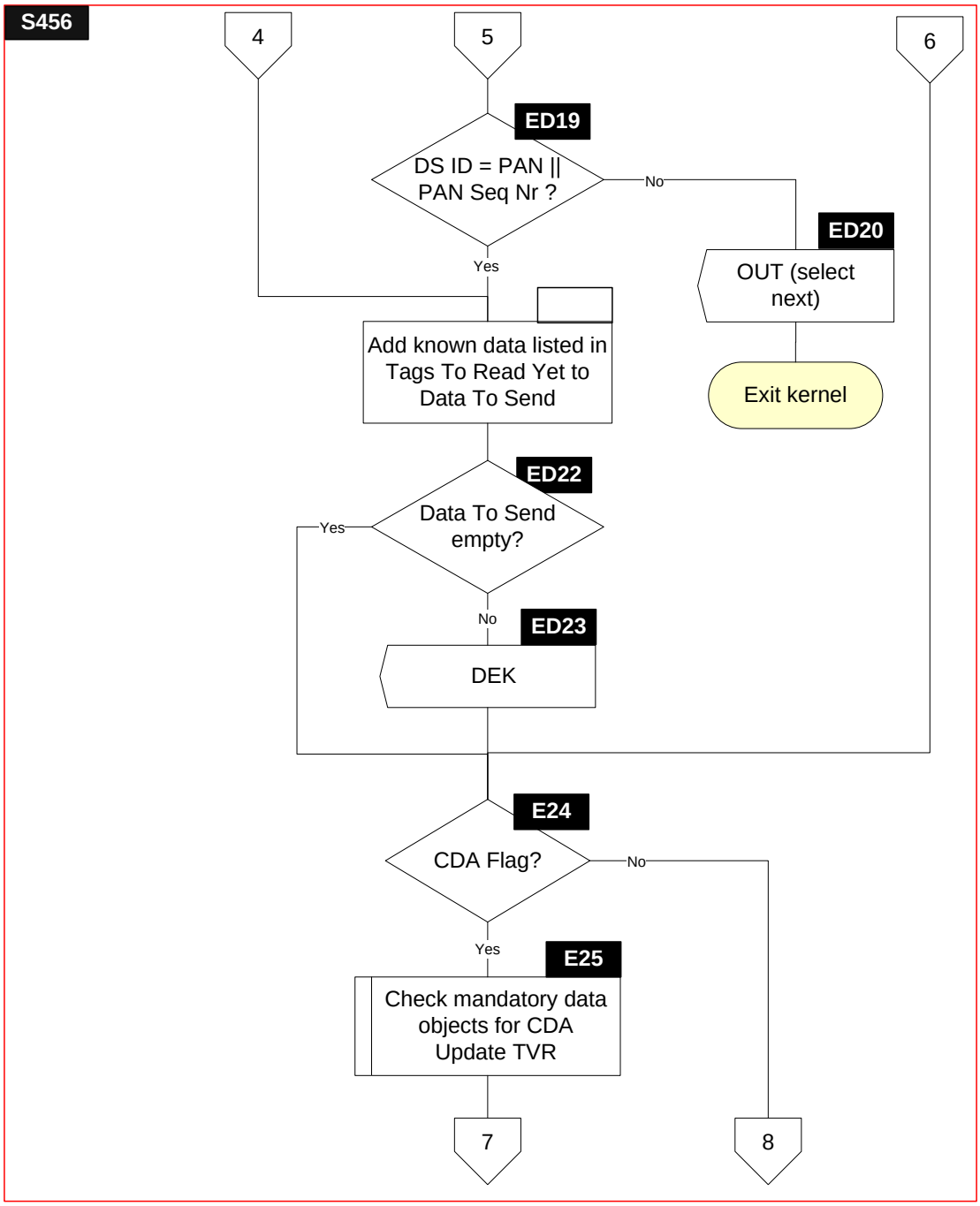
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 226



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 227

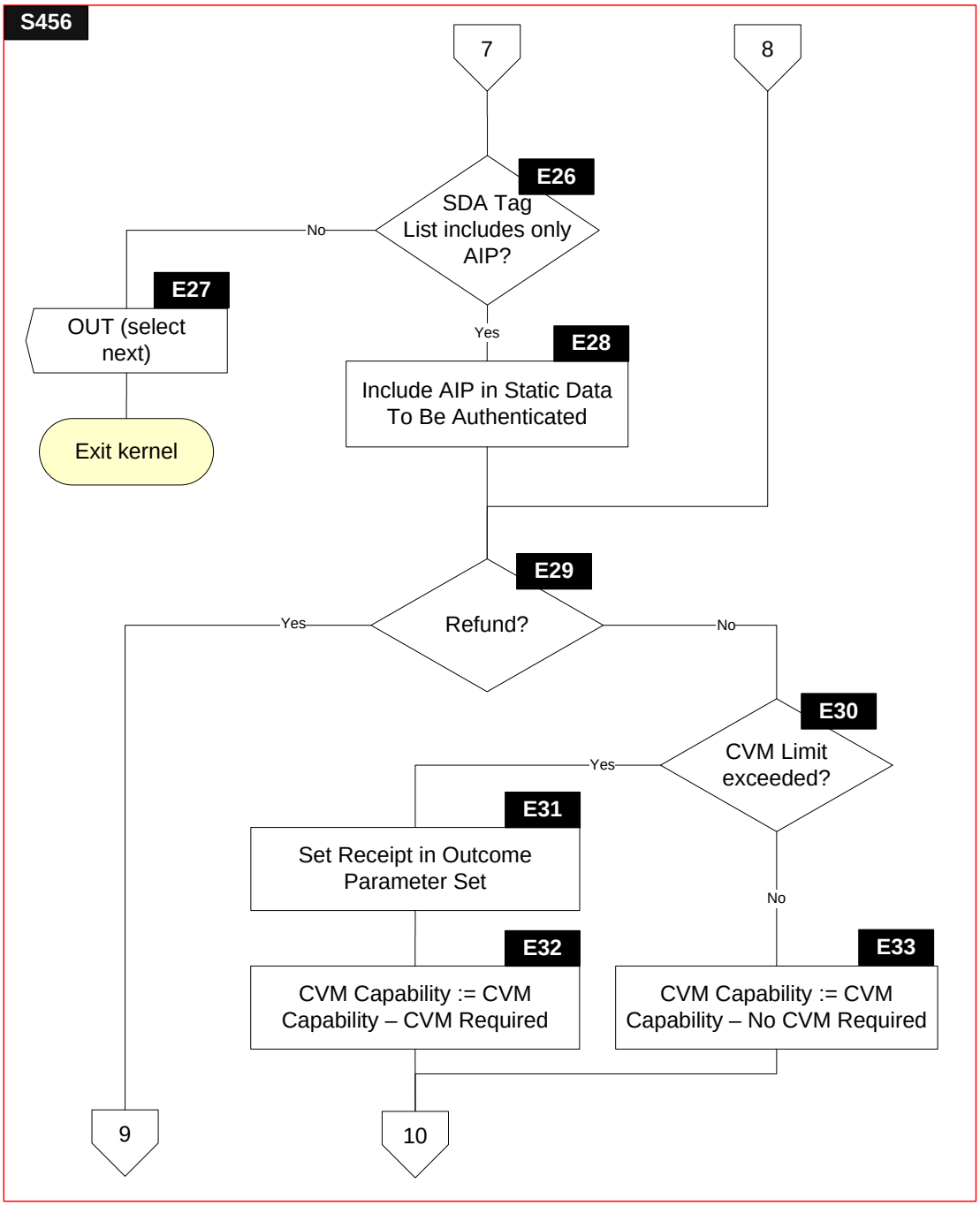


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 228

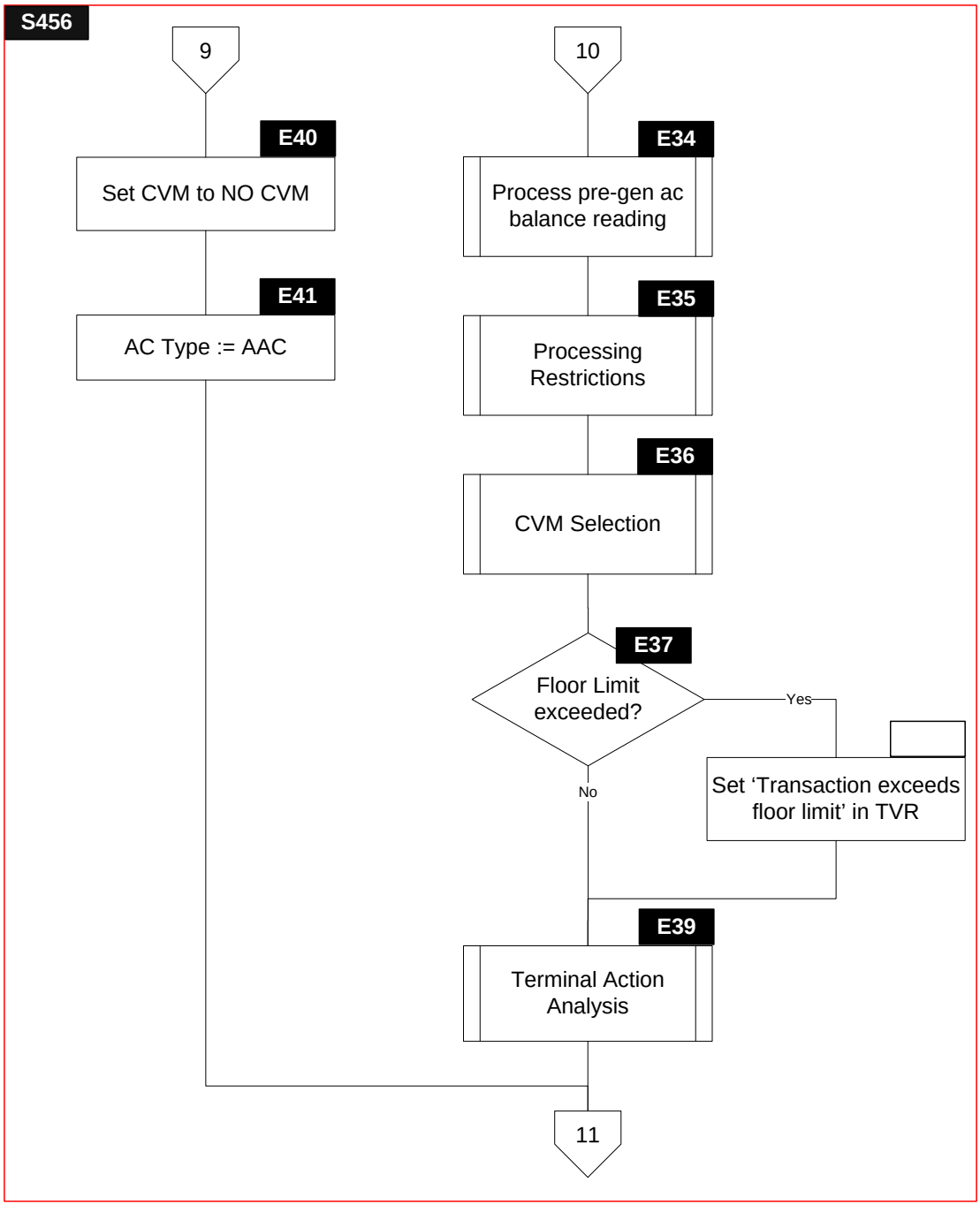


© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 229

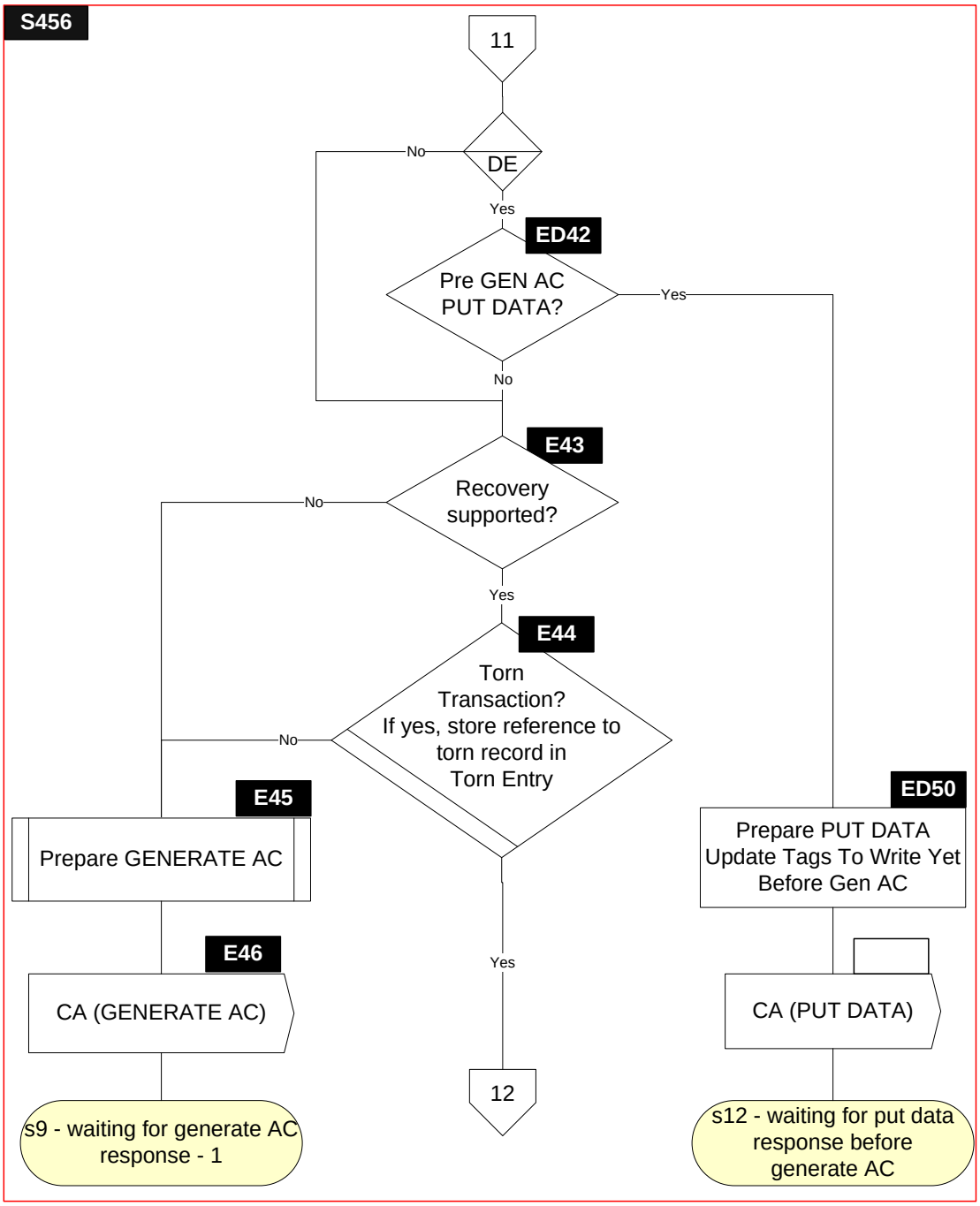


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 230



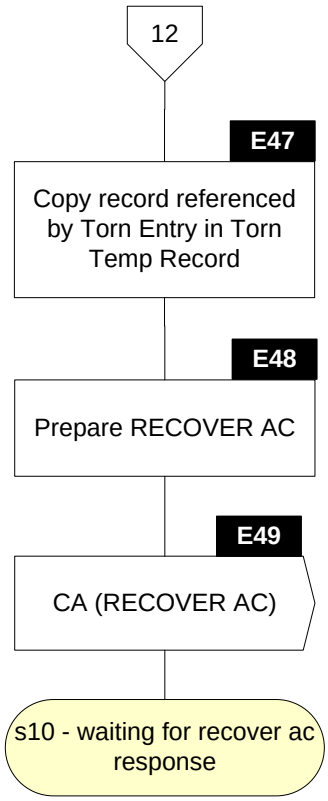
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 231



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 232

S456



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 233

6.10.3 Processing

S456.E1

IF ['Next Cmd' in *Next Cmd* = READ RECORD]
THEN
    GOTO S456.ED2 or `s4 – waiting for EMV read record response`
ELSE
    IF ['Next Cmd' in *Next Cmd* = GET DATA]
    THEN
        GOTO `s5 – waiting for get data response`
    ELSE
        GOTO S456.ED5 or S456.E12
    ENDIF
ENDIF

S456.ED2

FOR every T in *Tags To Read Yet*
{
    IF [IsNotEmpty(T)]
    THEN
        AddToList(GetTLV(T), *Data To Send*)
        RemoveFromList(T, *Tags To Read Yet*)
    ENDIF
}

S456.ED3

IF [IsNotEmptyList(*Data To Send*) AND IsEmptyList(*Tags To Read Yet*)]
THEN
    GOTO S456.ED4
ELSE
    GOTO `s4 – waiting for EMV read record response`
ENDIF

S456.ED4

Send DEK(*Data To Send, Data Needed*) signal
Initialize(*Data To Send*)
Initialize(*Data Needed*)

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 234

S456.ED5

IF [IsEmpty(TagOf(*Proceed To First Write Flag*))]
THEN
GOTO S456.ED6
ELSE
GOTO S456.ED11
ENDIF

S456.ED6

AddToList (TagOf (*Proceed To First Write Flag*), *Data Needed*)

S456.ED7

FOR every T in *Tags To Read Yet*
{
IF [IsNotEmpty(T)]
THEN
AddToList(GetTLV(T), *Data To Send*)
RemoveFromList(T, *Tags To Read Yet*)
ENDIF
}

S456.ED8

IF [IsNotEmptyList(*Data Needed*) OR
(IsNotEmptyList(*Data To Send*) AND IsEmptyList(*Tags To Read Yet*))]
THEN
GOTO S456.ED9
ELSE
GOTO S456.ED10
ENDIF

S456.ED9

Send DEK(*Data To Send*, *Data Needed*) signal
Initialize(*Data To Send*)
Initialize(*Data Needed*)

S456.ED10

Start Timer (*Time Out Value*)

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 235

S456.ED11

IF [IsPresent(TagOf(*Proceed To First Write Flag*)) AND (*Proceed To First Write Flag* = '00')]

THEN

GOTO S456.ED7

ELSE

GOTO S456.E12

ENDIF

S456.E12

IF [IsNotEmpty(TagOf(*Amount, Authorized (Numeric)*))]

THEN

GOTO S456.E14

ELSE

GOTO S456.E13

ENDIF

S456.E13

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := AMOUNT NOT PRESENT

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S456.E14

IF [*Amount, Authorized (Numeric) > Reader Contactless Transaction Limit*]

THEN

GOTO S456.E15

ELSE

GOTO S456.E16

ENDIF

S456.E15

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := MAX LIMIT EXCEEDED

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 236

S456.E16

Check if all mandatory data objects are present in the TLV Database

Table 6.2—Mandatory EMV Mode Data Objects

Data Object
Application Expiration Date
Application PAN
CDOL1

IF [IsNotEmpty(TagOf(*Application Expiration Date*)) AND
IsNotEmpty(TagOf(*Application PAN*)) AND
IsNotEmpty(TagOf(*CDOL1*))]

THEN

GOTO S456.ED18 or S456.E24

ELSE

GOTO S456.E17

ENDIF

S456.E17

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := CARD DATA MISSING

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S456.ED18

IF ['Read' in *IDS Status* is set]

THEN

GOTO S456.ED19

ELSE

GOTO S456.ED21

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

S456.ED19

Concatenate from left to right the *Application PAN* (without any 'F' padding) with the *Application PAN Sequence Number* (if the *Application PAN Sequence Number* is not present, then it is replaced by a '00' byte). If the result is less than 22 digits long, pad it to the right with hexadecimal zeros in order to obtain an 11-byte long number Y in numeric (n) format.

IF [*DS ID* = Y]
THEN
GOTO S456.ED21
ELSE
GOTO S456.ED20
ENDIF

S456.ED20

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := CARD DATA ERROR

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S456.ED21

FOR every T in *Tags To Read Yet*
{
IF [IsPresent(T)]
THEN
AddToList(GetTLV(T), *Data To Send*)
ELSE
Add an empty data object with tag T to *Data To Send* if the TLV Database does not include a data object with tag T:
AddToList(T || '00', *Data To Send*)
ENDIF
RemoveFromList(T, *Tags To Read Yet*)
}

S456.ED22

IF [IsEmptyList(*Data To Send*)]
THEN
GOTO S456.E24
ELSE
GOTO S456.ED23
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 238

S456.ED23

Send DEK(*Data To Send*) signal

Initialize(*Data To Send*)

S456.E24

IF ['CDA' in *ODA Status* is set]

THEN

GOTO S456.E25

ELSE

GOTO S456.E29

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 239

S456.E25

Check if all mandatory Card data objects for CDA are present in the TLV Database

Table 6.3—Mandatory Card CDA Data Objects

Data Object
CA Public Key Index (Card)
Issuer Public Key Certificate
Issuer Public Key Exponent
ICC Public Key Certificate
ICC Public Key Exponent
Static Data Authentication Tag List

IF [NOT (
IsNotEmpty(TagOf(*CA Public Key Index (Card)*)) AND
IsNotEmpty(TagOf(*Issuer Public Key Certificate*)) AND
IsNotEmpty(TagOf(*Issuer Public Key Exponent*)) AND
IsNotEmpty(TagOf(*ICC Public Key Certificate*)) AND
IsNotEmpty(TagOf(*ICC Public Key Exponent*)) AND
IsNotEmpty(TagOf(*Static Data Authentication Tag List*))
)]

THEN

SET 'ICC data missing' in *Terminal Verification Results*

SET 'CDA failed' in *Terminal Verification Results*

ENDIF

IF [The *CA Public Key Index (Card)* is not present in the CA Public Key Database]

THEN

SET 'CDA failed' in *Terminal Verification Results*

ENDIF

S456.E26

IF [IsNotEmpty(TagOf(*Static Data Authentication Tag List*)) AND (*Static Data Authentication Tag List* = '82')]

THEN

GOTO S456.E28

ELSE

GOTO S456.E27

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 240

S456.E27

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := CARD DATA ERROR

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S456.E28

IF [Enough space left in *Static Data To Be Authenticated* to append *Application Interchange Profile*]

THEN

Append *Application Interchange Profile* at the end of *Static Data To Be Authenticated*

ELSE

SET 'CDA failed' in *Terminal Verification Results*

ENDIF

S456.E29

IF [*Transaction Type* indicates refund]

THEN

GOTO S456.E40

ELSE

GOTO S456.E30

ENDIF

S456.E30

IF [*Amount, Authorized (Numeric) > Reader CVM Required Limit*]

THEN

GOTO S456.E31

ELSE

GOTO S456.E33

ENDIF

S456.E31

'Receipt' in *Outcome Parameter Set* := YES

S456.E32

Terminal Capabilities[2] := *CVM Capability – CVM Required*

S456.E33

Terminal Capabilities[2] := *CVM Capability – No CVM Required*

S456.E34

Process pre-generate AC balance reading as specified in section 7.1

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 241

S456.E35

Process Processing Restrictions as specified in section 7.7

S456.E36

Process CVM Selection as specified in section 7.5

S456.E37

IF [*Amount, Authorized (Numeric) > Reader Contactless Floor Limit*]
THEN
GOTO S456.E38
ELSE
GOTO S456.E39
ENDIF

S456.E38

SET 'Transaction exceeds floor limit' in *Terminal Verification Results*

S456.E39

Process Terminal Action Analysis as specified in section 7.8

S456.E40

'CVM' in *Outcome Parameter Set* := NO CVM
'CVM Performed' in *CVM Results* := '3F' (No CVM performed)
'CVM Condition' in *CVM Results* := '00'
'CVM Result' in *CVM Results* := '00' (unknown)

S456.E41

'AC type' in *AC Type* := AAC

S456.ED42

IF [IsNotEmptyList(*Tags To Write Before Gen AC*)]
THEN
GOTO S456.ED50
ELSE
GOTO S456.E43
ENDIF

S456.E43

IF [IsPresent(TagOf(*DRDOL*)) AND
Max Number of Torn Transaction Log Records ≠ 0]
THEN
GOTO S456.E44
ELSE
GOTO S456.E45
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 242

S456.E44

FOR every Record in Torn Transaction Log

{

IF [*Application PAN* in Record = *Application PAN*
AND
Application PAN Sequence Number in Record = *Application PAN Sequence Number*]

THEN

Store reference to Record in *Torn Entry* for later use

GOTO S456.E47

ENDIF

}

GOTO S456.E45

Note that the loop must be executed in such a way that the records are ordered by time and date with the most recent record first. This is because it is possible for there to be two records in the Torn Transaction Log for the same Card and in this case the most recent record must be found.

S456.E45

Prepare GENERATE AC command as specified in section 7.6

S456.E46

Send CA(GENERATE AC command) signal

S456.E47

Copy record referenced by *Torn Entry* into *Torn Temp Record*

S456.E48

DRDOL Related Data := *DRDOL Related Data* in *Torn Temp Record*

Prepare RECOVER AC command as specified in section 5.8

S456.E49

Send CA(RECOVER AC) signal

S456.ED50

TLV := GetAndRemoveFromList(*Tags To Write Yet Before Gen AC*)

Prepare PUT DATA command for TLV as specified in section 5.6

S456.ED51

Send CA(PUT DATA command) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 243

6.11 State 7 – Waiting for Mag-stripe Read Record Response

6.11.1 Local Variables

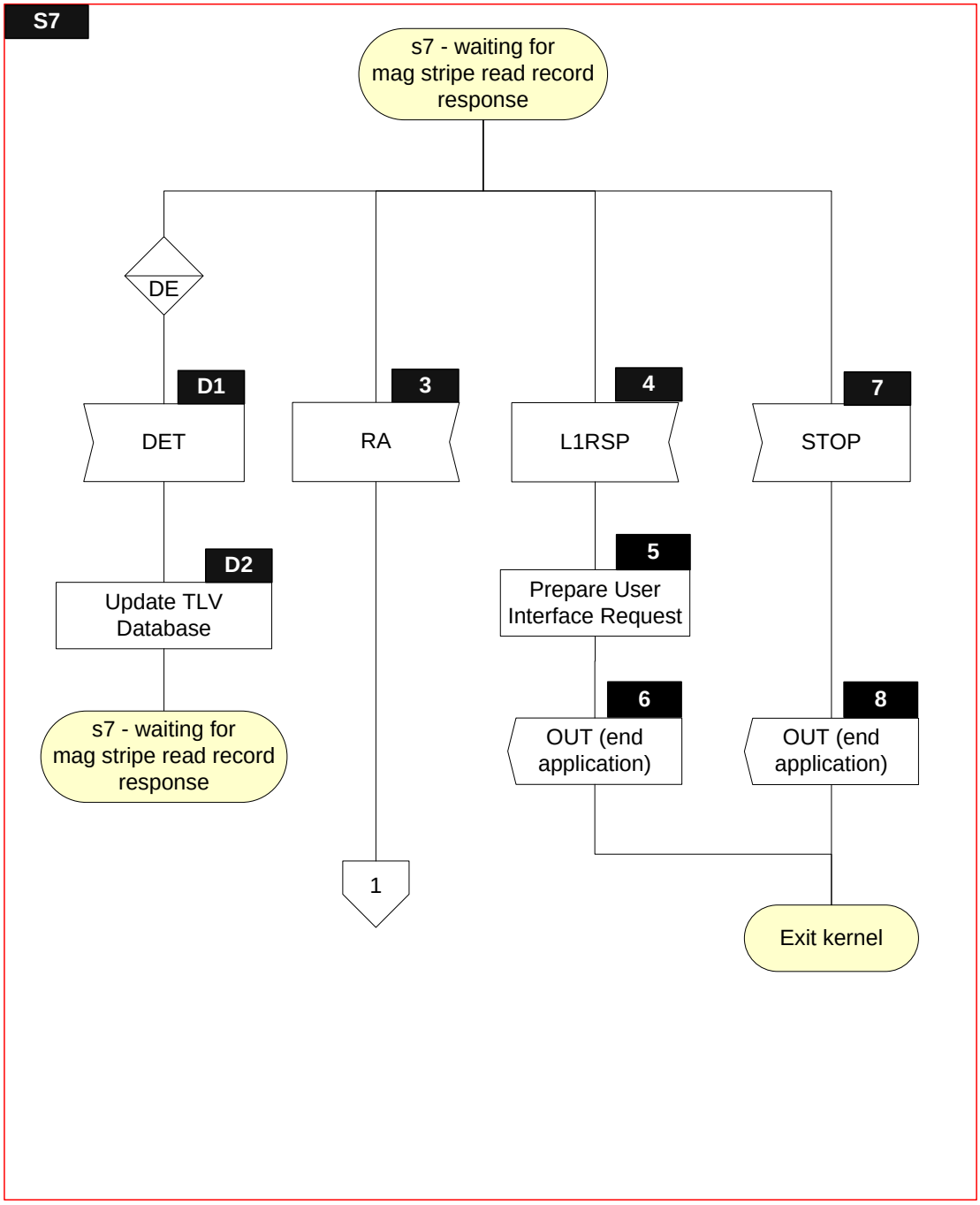
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Sync Data	var.	b	List of data objects returned with DET signal
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Record	var. up to 256	b	Response Message Data Field of the R-APDU of READ RECORD
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 253	b	Value of TLV encoded string

6.11.2 Flow Diagram

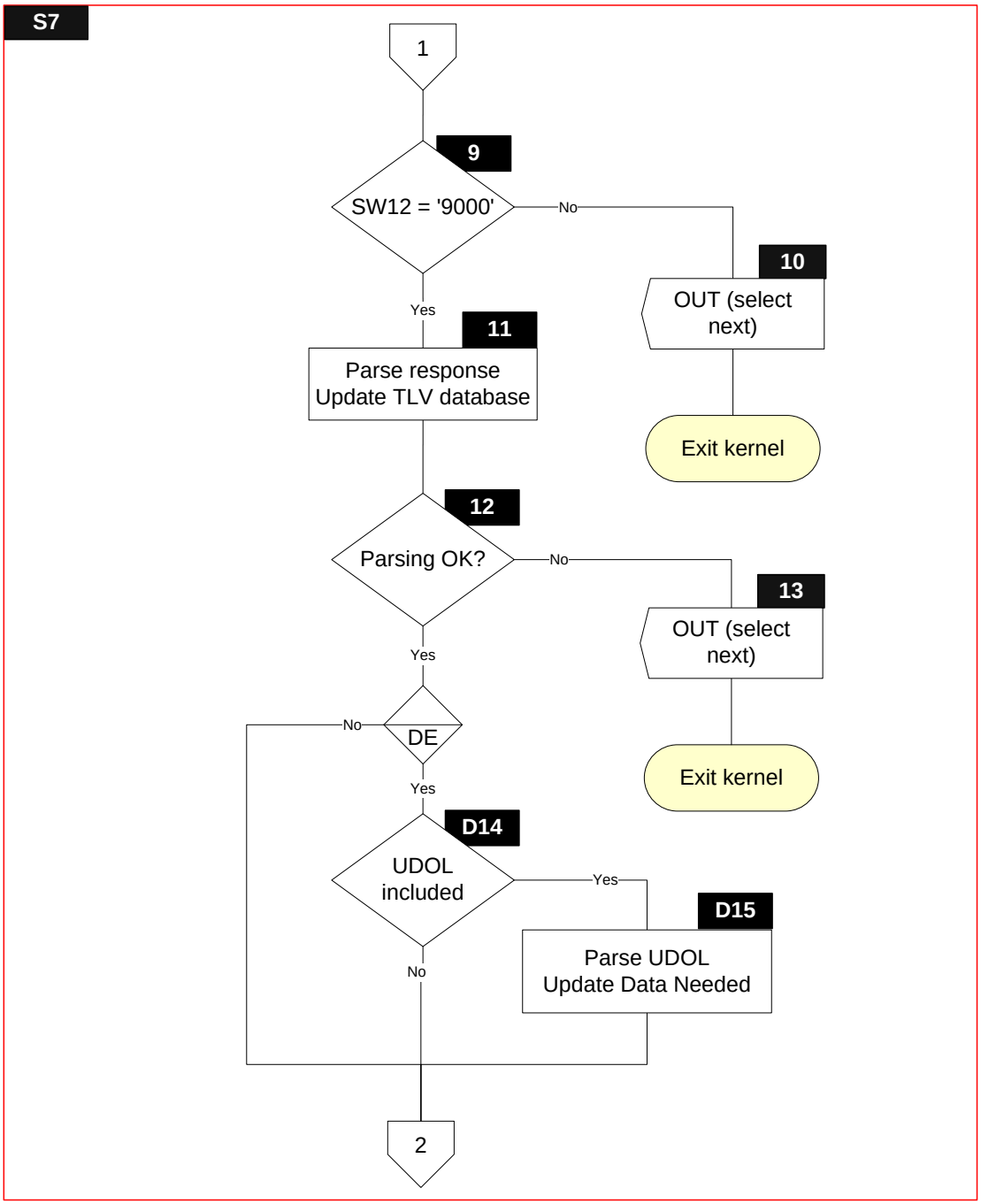
Figure 6.10 shows the flow diagram of `S7 – waiting for mag stripe read record response`. Symbols in this diagram are labeled S7.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 244

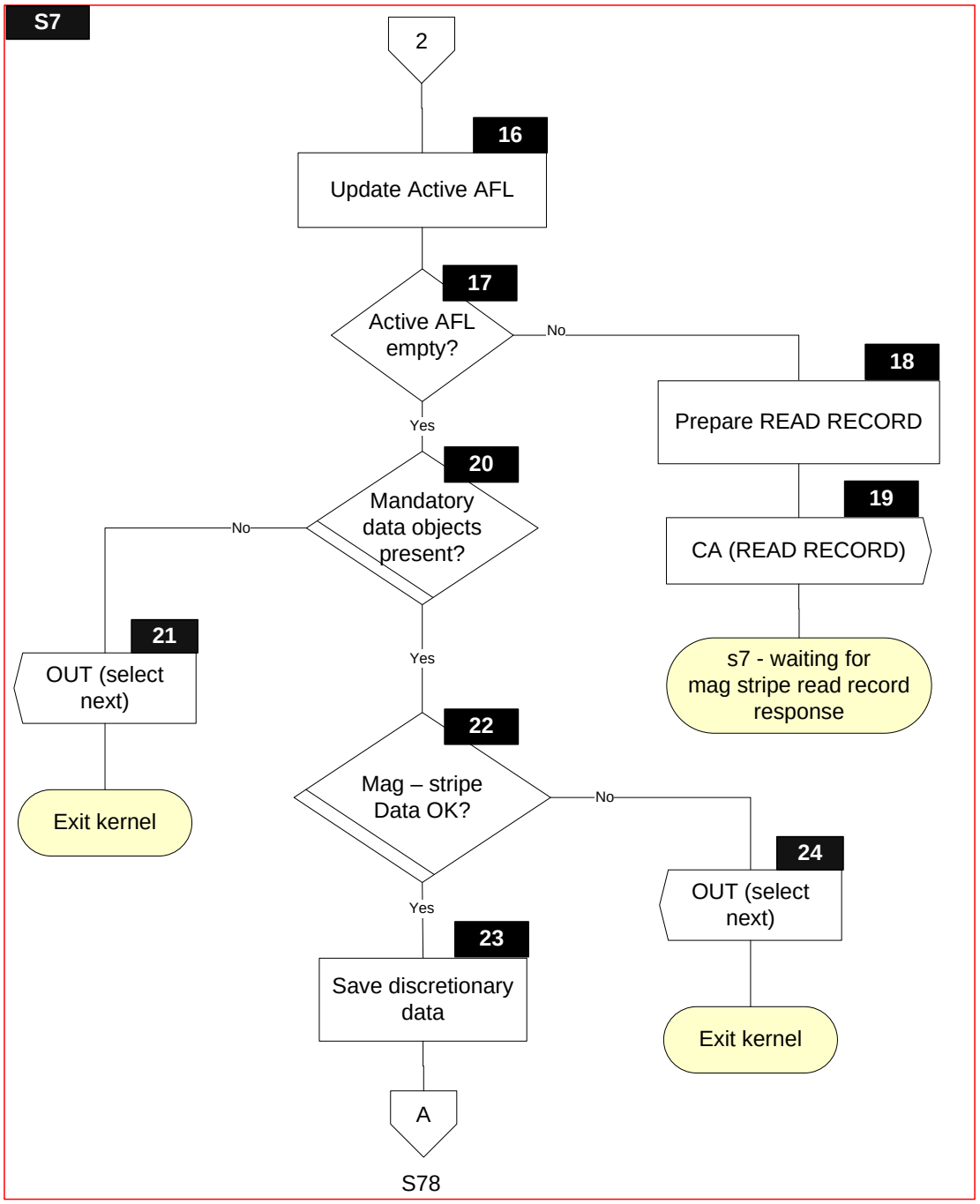
Figure 6.10—State 7 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 245



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 246



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 247

6.11.3 Processing

S7.D1

Receive DET signal with Sync Data

S7.D2

UpdateWithDetData(Sync Data)

S7.3

Receive RA signal with Record and SW12

S7.4

Receive L1RSP signal with Return Code

S7.5

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S7.6

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S7.7

Receive STOP signal

S7.8

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S7.9

IF [SW12 = '9000']

THEN

GOTO S7.11

ELSE

GOTO S7.10

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 248

S7.10

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := STATUS BYTES

'SW12' in *Error Indication* := SW12

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S7.11

IF [SFI of file of Record ≤ 10]

THEN

IF [(Length of Record > 0) AND (Record[1] = '70')]

THEN

Parsing Result := ParseAndStoreCardResponse(Record)

ELSE

Parsing Result := FALSE

ENDIF

ELSE

Processing of records in proprietary files is beyond the scope of this specification

ENDIF

S7.12

IF [Parsing Result]

THEN

GOTO S7.D14 or S7.16

ELSE

GOTO S7.13

ENDIF

S7.13

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := PARSING ERROR

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 249

S7.D14

IF [Record includes *UDOL*]
THEN
    GOTO S7.D15
ELSE
    GOTO S7.16
ENDIF

S7.D15

FOR every TL entry in the *UDOL*
{
    IF [IsEmpty(T)]
    THEN
        AddToList(T, *Data Needed*)
    ENDIF
}

S7.16

Remove first record from *Active AFL*

S7.17

IF [*Active AFL* is empty]
THEN
    GOTO S7.20
ELSE
    GOTO S7.18
ENDIF

S7.18

Prepare READ RECORD command for first record in *Active AFL* as specified in section 5.7

S7.19

Send CA(READ RECORD command) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 250

S7.20

Check if all mandatory data objects are present in the TLV Database

Table 6.4—Mandatory Mag-stripe Mode Data Objects

Data Object
Track 2 Data
PUNATC(Track2)
PCVC3(Track2)
NATC(Track2)

IF [IsNotEmpty(TagOf(*Track 2 Data*)) AND
IsNotEmpty(TagOf(*PUNATC(Track2)*)) AND
IsNotEmpty(TagOf(*PCVC3(Track2)*)) AND
IsNotEmpty(TagOf(*NATC(Track2)*))]

THEN

GOTO S7.22

ELSE

GOTO S7.21

ENDIF

S7.21

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := CARD DATA MISSING

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 251

S7.22

Verify correctness of the mag-stripe mode data objects as follows:

nUN := (Number of non-zero bits in *PUNATC(Track2)*) – *NATC(Track2)*

IF [(*nUN* < 0) OR (*nUN* > 8)]

THEN

GOTO S7.24

ENDIF

IF [IsNotEmpty(TagOf(*Track 1 Data*))]

THEN

IF [(IsNotPresent(TagOf(*NATC(Track1)*)) OR
IsEmpty(TagOf(*NATC(Track1)*)))
OR
(IsNotPresent(TagOf(*PCVC3(Track1)*)) OR
IsEmpty(TagOf(*PCVC3(Track1)*)))
OR
(IsNotPresent(TagOf(*PUNATC(Track1)*)) OR
IsEmpty(TagOf(*PUNATC(Track1)*)))
OR
(Number of non-zero bits in *PUNATC(Track1) – NATC(Track1) ≠ nUN)*]

THEN

GOTO S7.24

ELSE

GOTO S7.23

ENDIF

ELSE

GOTO S7.23

ENDIF

Note that the Kernel must not validate the individual data fields in *Track 1 Data* and *Track 2 Data*. Specifically:

- Validation of the values 2 and 6 in the first digit of the service code present in *Track 1 Data* or *Track 2 Data* to determine if a contact chip transaction is required must not be performed.
- Validation of the cardholder name, including the presence of the surname separator, must not be performed.

Any existing data validation carried out to support individual payment products is outside the scope of this specification.

However, if the Kernel is not able to localize a required data field in *Track 1 Data* or *Track 2 Data* due to one or more format errors, the Kernel must terminate the transaction as described in S7.24.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 252

S7.23

DD Card (Track2) := 'Discretionary Data' in *Track 2 Data*

IF [IsNotEmpty(TagOf(*Track 1 Data*))]

THEN

DD Card (Track1) := 'Discretionary Data' in *Track 1 Data*

ENDIF

S7.24

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := CARD DATA ERROR

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 253

6.12 State 8 – Waiting for Mag-stripe First Write Flag

6.12.1 Local Variables

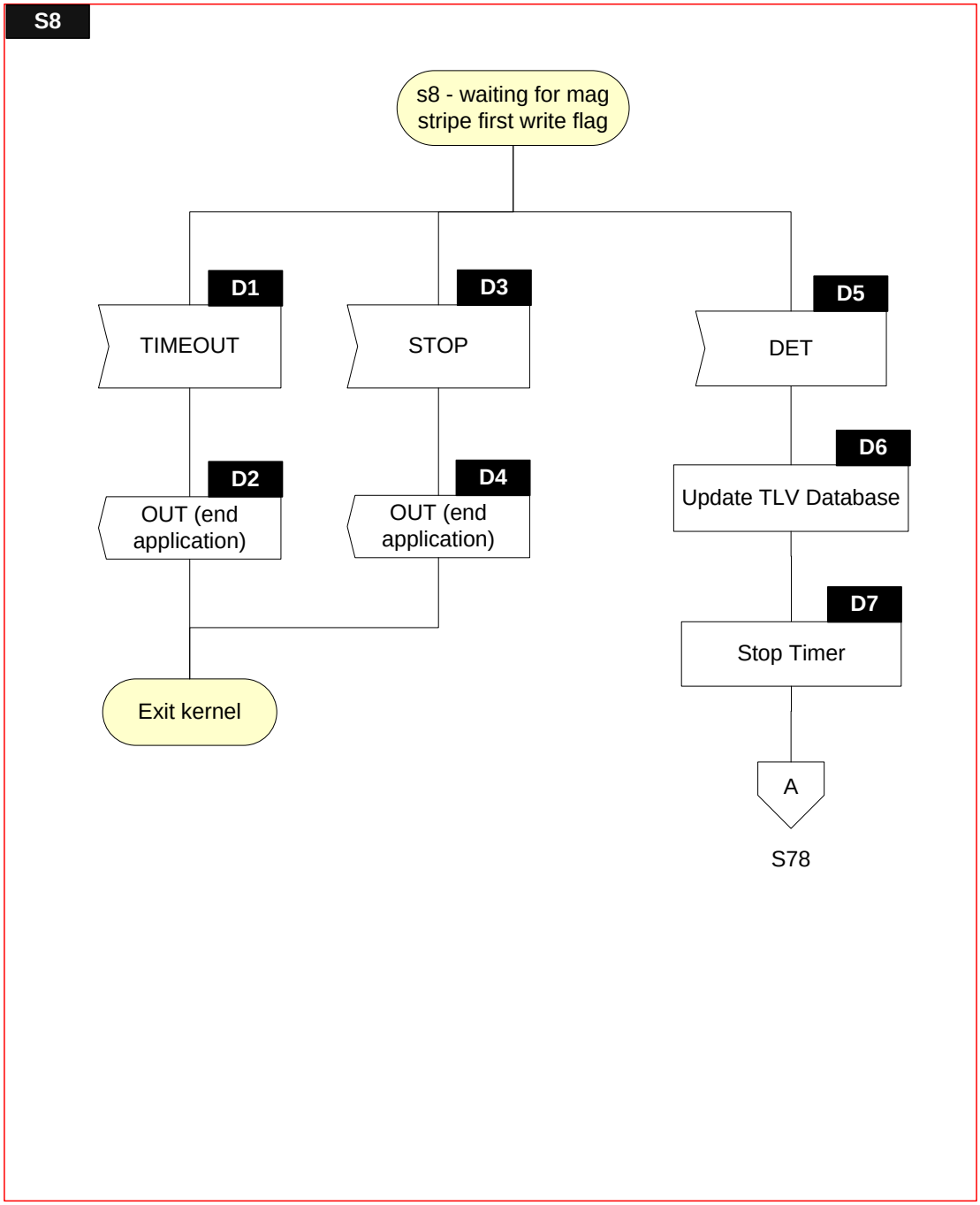
Name	Length	Format	Description
Sync Data	var.	b	List of data objects returned with DET signal
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 253	b	Value of TLV encoded string

6.12.2 Flow Diagram

Figure 6.11 shows the flow diagram of `s8 – waiting for mag stripe first write flag`. Symbols in this diagram are labeled S8.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 254

Figure 6.11—State 8 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 255

6.12.3 Processing

S8.D1

Receive TIMEOUT signal

S8.D2

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := TIME OUT

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S8.D3

Receive STOP signal

S8.D4

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S8.D5

Receive DET signal with Sync Data

S8.D6

UpdateWithDetData(Sync Data)

S8.D7

Stop Timer

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

6.13 States 7 and 8 – Common Processing

6.13.1 Local Variables

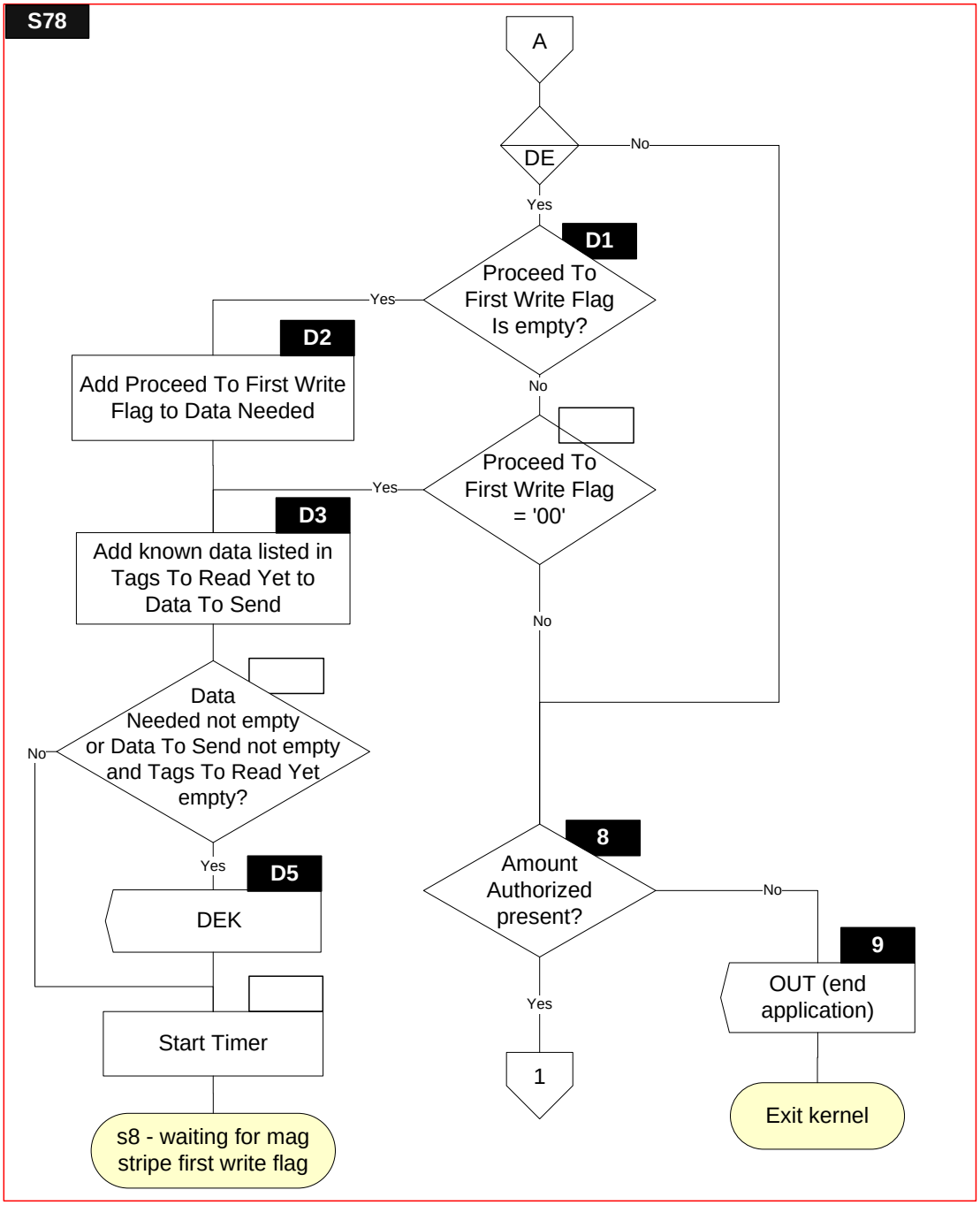
Local variables for common processing are defined in states 7 and 8.

6.13.2 Flow Diagram

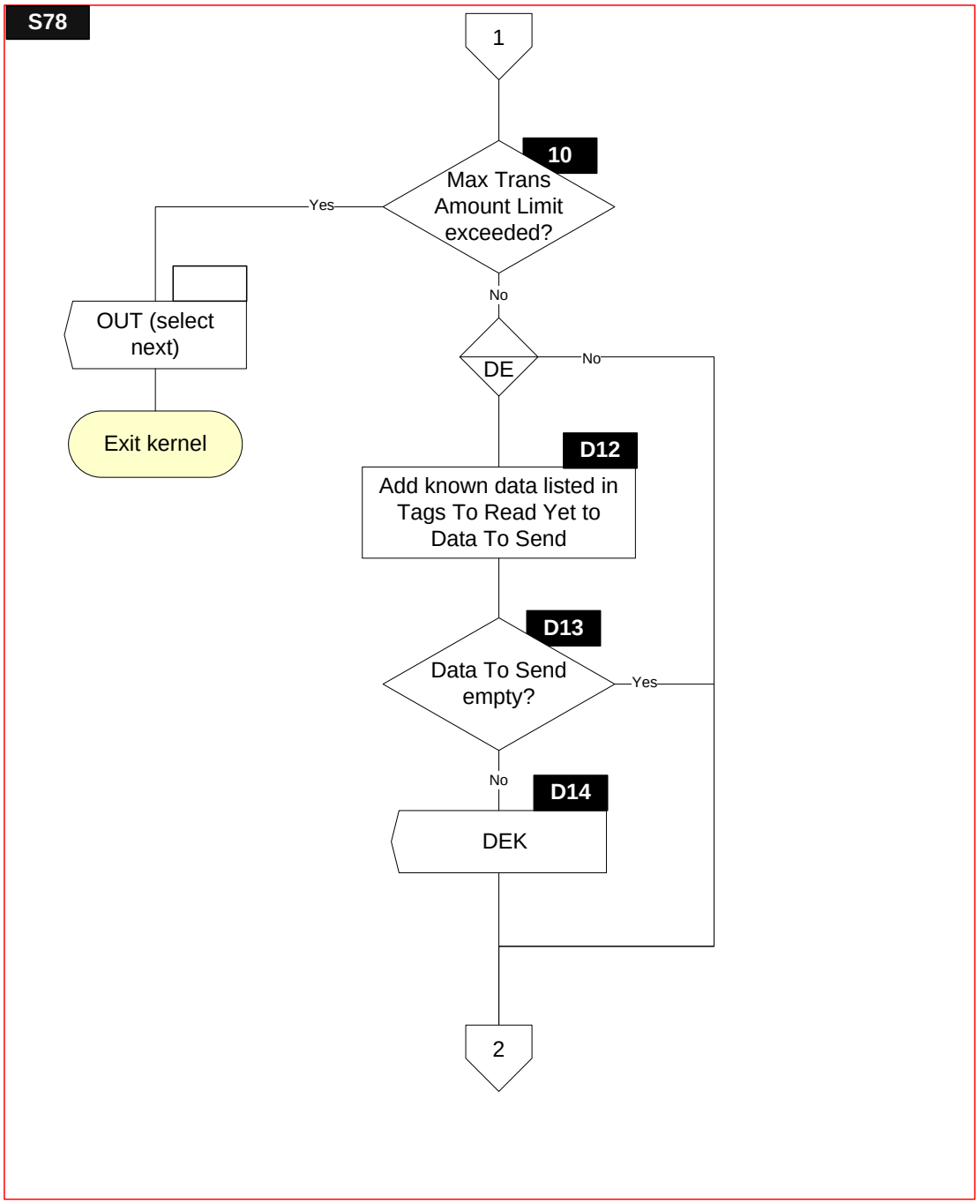
Figure 6.12 shows the flow diagram for common processing between states 7 and 8. Symbols in this diagram are labeled S78.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 257

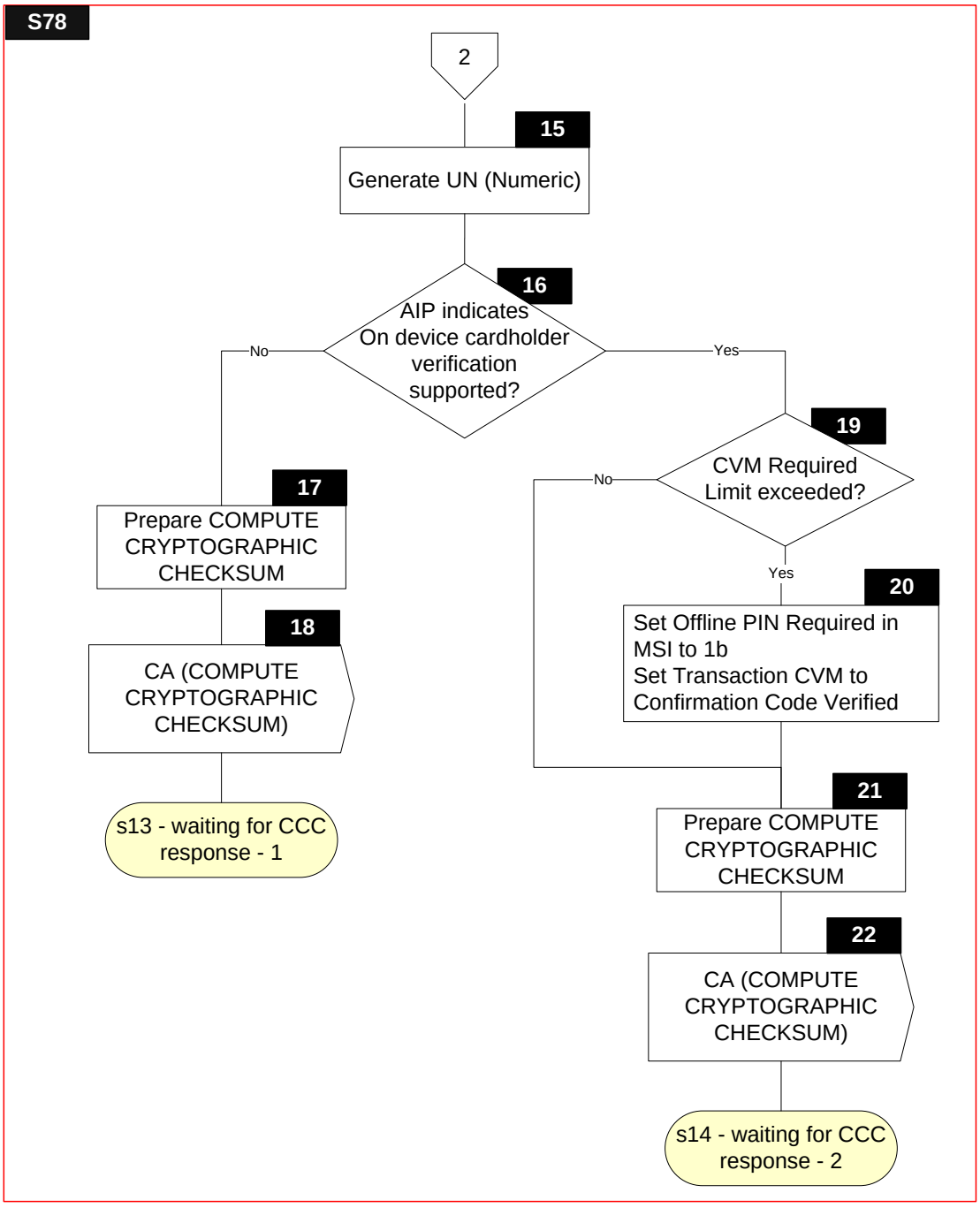
Figure 6.12—States 7 and 8 – Common Processing – Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 258



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 259



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 260

6.13.3 Processing

S78.D1

IF [IsEmpty(TagOf(*Proceed To First Write Flag*))]
THEN
GOTO S78.D2
ELSE
GOTO S78.D7
ENDIF

S78.D2

AddToList(TagOf(*Proceed To First Write Flag*), *Data Needed*)

S78.D3

FOR every T in *Tags To Read Yet*
{
IF [IsNotEmpty(T)]
THEN
AddToList(GetTLV(T), *Data To Send*)
RemoveFromList(T, *Tags To Read Yet*)
ENDIF
}

S78.D4

IF [IsNotEmptyList(*Data Needed*) OR
(IsNotEmptyList(*Data To Send*) AND IsEmptyList(*Tags To Read Yet*))]
THEN
GOTO S78.D5
ELSE
GOTO S78.D6
ENDIF

S78.D5

Send DEK(*Data To Send*, *Data Needed*) signal
Initialize *Data To Send*
Initialize *Data Needed*

S78.D6

Start Timer (*Time Out Value*)

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 261

S78.D7

IF [IsPresent(TagOf(*Proceed To First Write Flag*)) AND
(*Proceed To First Write Flag* = '00')]

THEN

GOTO S78.D3

ELSE

GOTO S78.8

ENDIF

S78.8

IF [IsNotEmpty(*Amount, Authorized (Numeric)*)]

THEN

GOTO S78.10

ELSE

GOTO S78.9

ENDIF

S78.9

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := AMOUNT NOT PRESENT

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S78.10

IF [*Amount, Authorized (Numeric) > Reader Contactless Transaction Limit*]

THEN

GOTO S78.11

ELSE

GOTO S78.D12

ENDIF

S78.11

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

'L2' in *Error Indication* := MAX LIMIT EXCEEDED

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 262

S78.D12

FOR every T in *Tags To Read Yet*

{

IF [IsPresent(T)]

THEN

AddToList(GetTLV(T), *Data To Send*)

ELSE

Add an empty data object with tag T to *Data To Send* if the TLV Database does not include a data object with tag T:

AddToList(T || '00', *Data To Send*)

ENDIF

RemoveFromList(T, *Tags To Read Yet*)

}

S78.D13

IF [IsEmptyList(*Data To Send*)]

THEN

GOTO S78.15

ELSE

GOTO S78.D14

ENDIF

S78.D14

Send DEK(*Data To Send*) signal

Initialize(*Data To Send*)

S78.15

Generate a 4 byte random value as described in section 8.1. Convert the random value to a 4 byte BCD encoded value and set the 8 – *nUN* most significant digits to zero.

Store this value in *Unpredictable Number (Numeric)*.

Note that it is possible to generate the value of the *Unpredictable Number (Numeric)* at other times in parallel with the processing of a CA signal without changing the external behavior of the Kernel. The *Unpredictable Number (Numeric)* could for example be generated after S3.81.

S78.16

IF ['On device cardholder verification is supported' in *Application Interchange Profile* is set]

THEN

GOTO S78.19

ELSE

GOTO S78.17

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 263

S78.17

Prepare COMPUTE CRYPTOGRAPHIC CHECKSUM command as specified in section 5.2

S78.18

Send CA(COMPUTE CRYPTOGRAPHIC CHECKSUM) signal

S78.19

IF [*Amount, Authorized (Numeric) > Reader CVM Required Limit*]

THEN

GOTO S78.20

ELSE

GOTO S78.21

ENDIF

S78.20

SET 'Offline PIN Required' in *Mobile Support Indicator*

'CVM' in *Outcome Parameter Set* := CONFIRMATION CODE VERIFIED

S78.21

Prepare COMPUTE CRYPTOGRAPHIC CHECKSUM command as specified in section 5.2

S78.22

Send CA(COMPUTE CRYPTOGRAPHIC CHECKSUM) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 264

6.14 State 9 – Waiting for Generate AC Response - 1

6.14.1 Local Variables

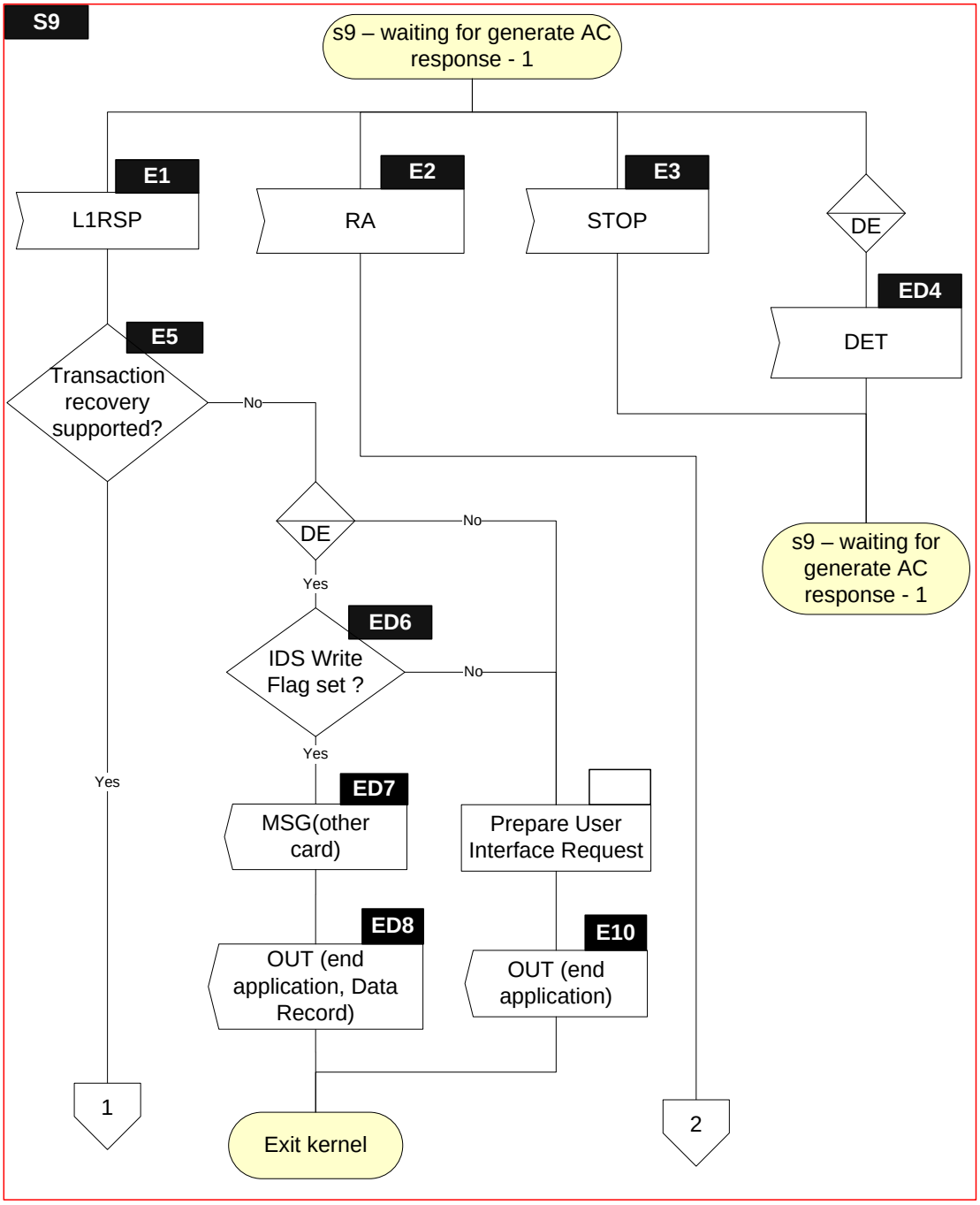
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of GENERATE AC
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 252	b	Value of TLV encoded string

6.14.2 Flow Diagram

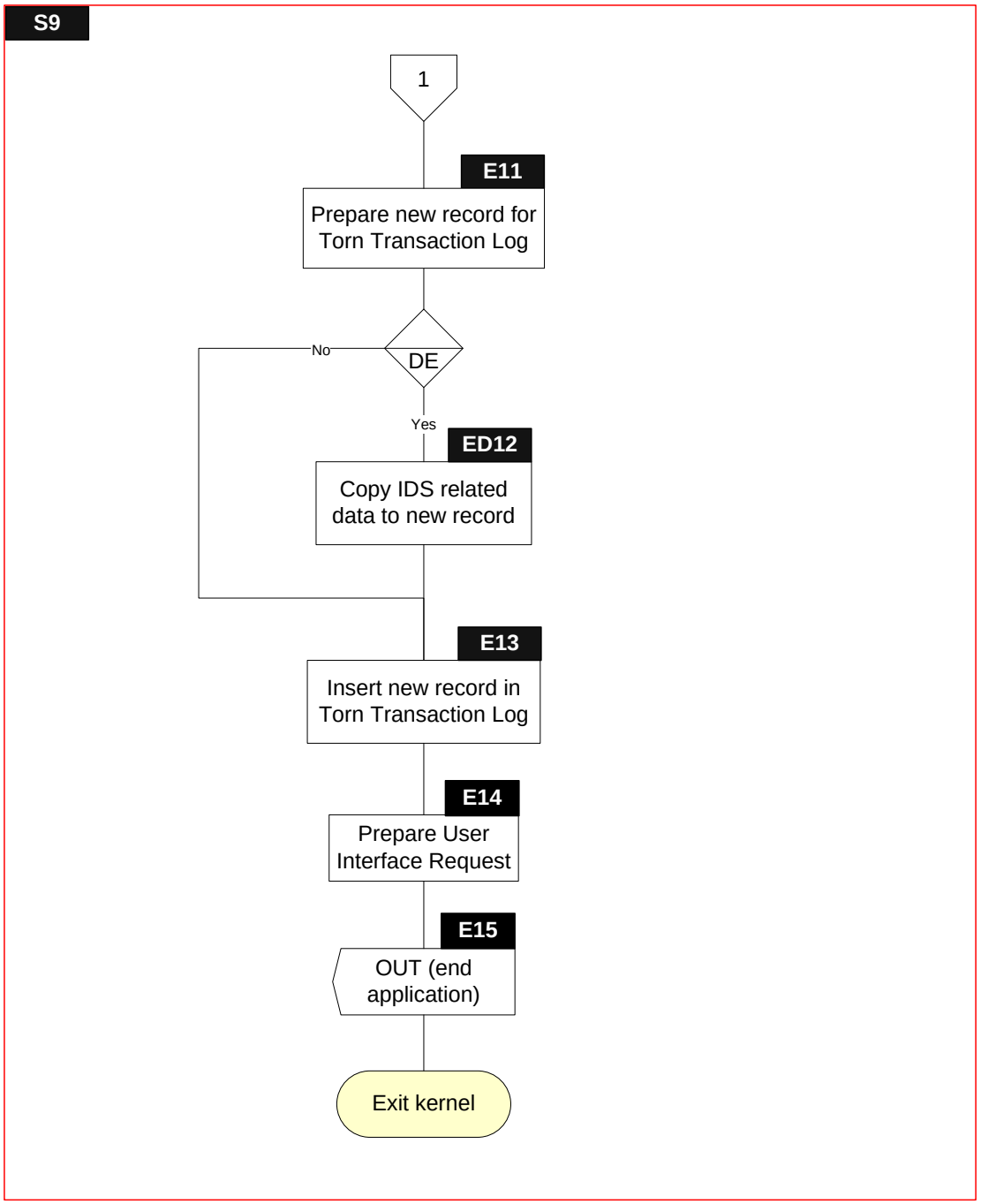
Figure 6.13 shows the flow diagram of `s9 – waiting for generate AC response - 1`. Symbols in this diagram are labeled S9.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 265

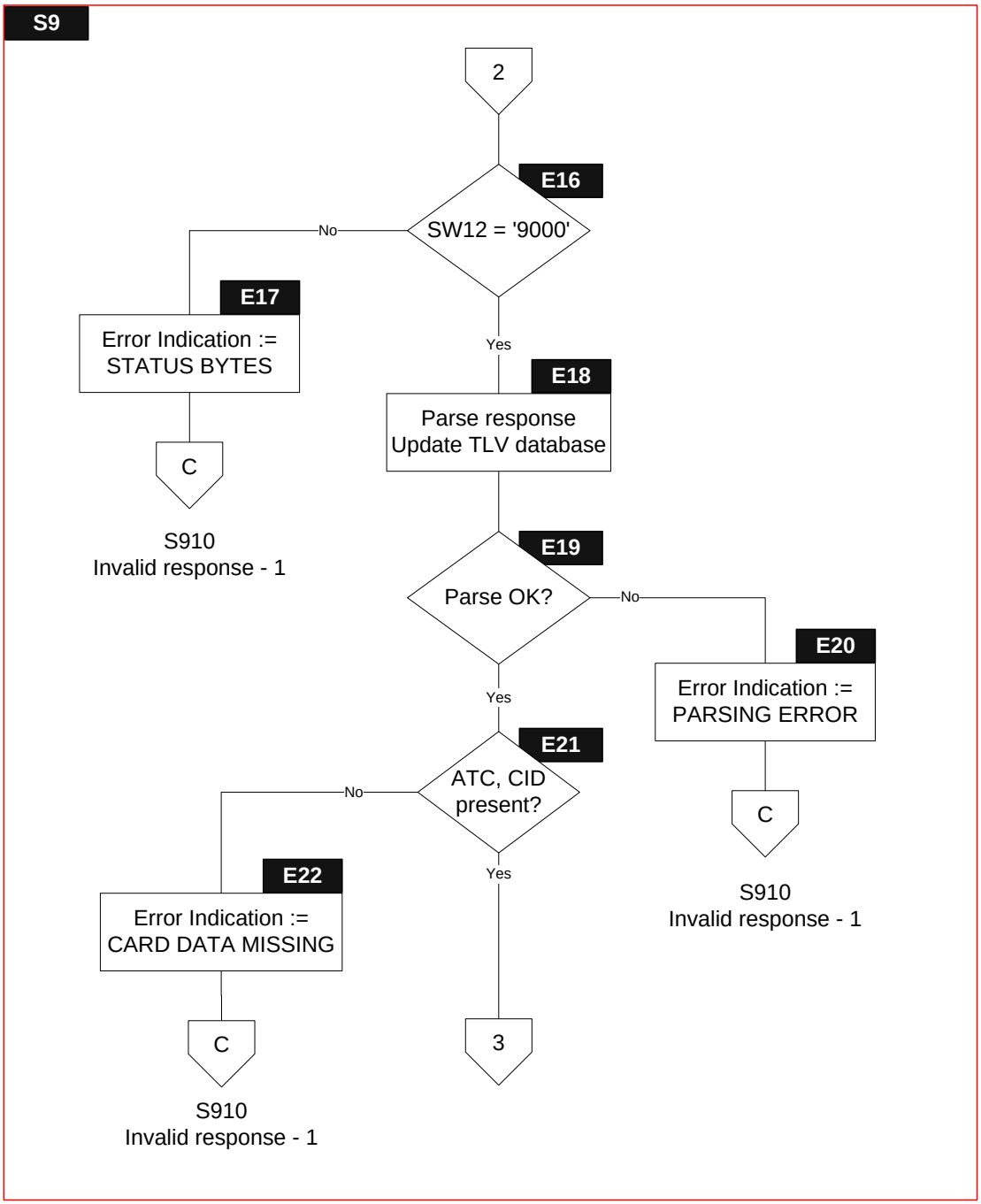
Figure 6.13—State 9 Flow Diagram



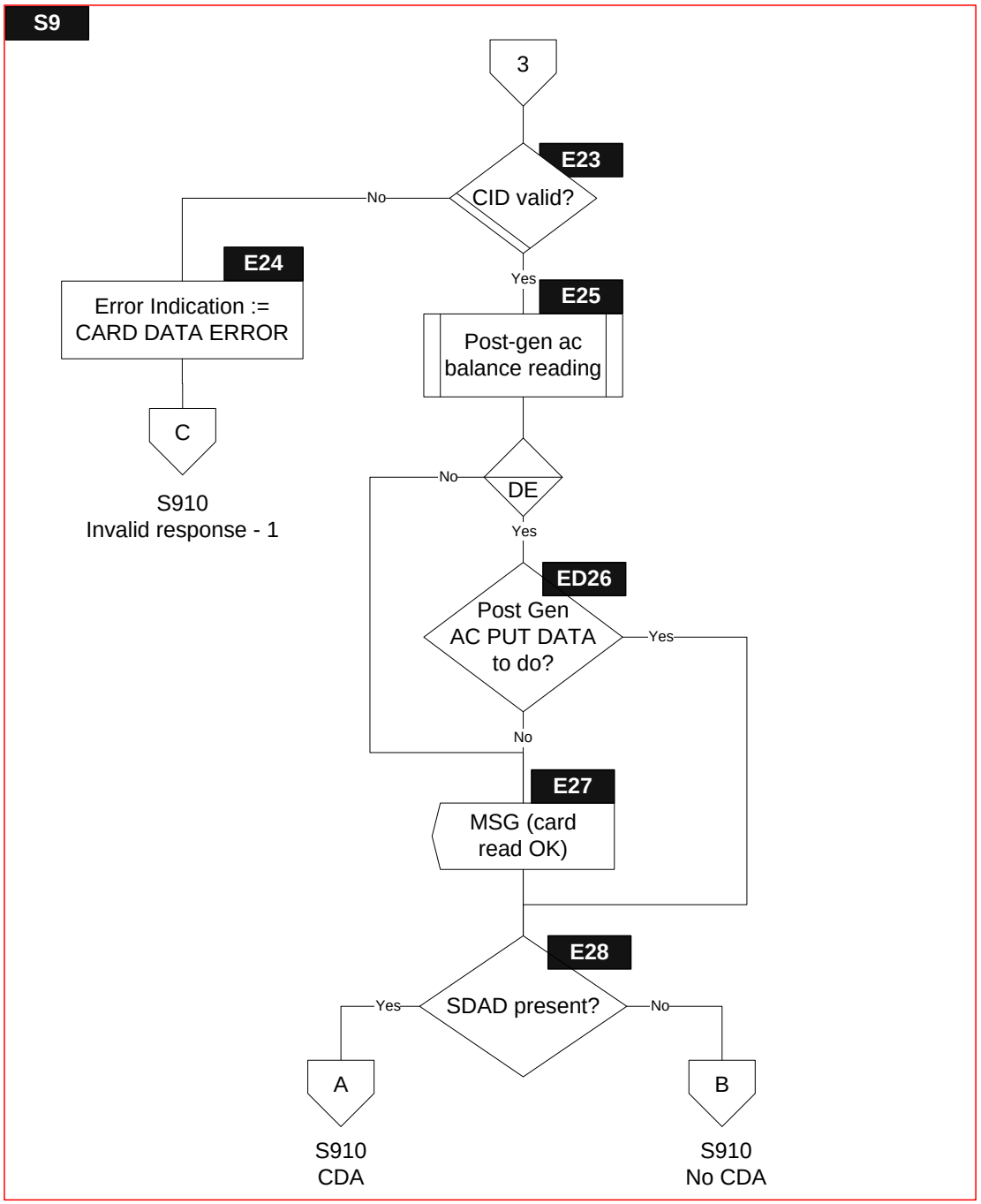
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 266



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 267



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 268



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 269

6.14.3 Processing

S9.E1

Receive L1RSP signal with Return Code

S9.E2

Receive RA signal with Response Message Data Field and SW12

S9.E3

Receive STOP signal

S9.ED4

Receive DET signal

S9.E5

IF [*Max Number of Torn Transaction Log Records* > 0
AND IsPresent(TagOf(*DRDOL*))]
THEN
GOTO S9.E11
ELSE
GOTO S9.ED6 or S9.E9
ENDIF

S9.ED6

IF ['Write' in *IDS Status* is set]
THEN
GOTO S9.ED7
ELSE
GOTO S9.E9
ENDIF

S9.ED7

'Message Identifier' in *User Interface Request Data* := ERROR - OTHER CARD
'Status' in *User Interface Request Data* := NOT READY
Send MSG(*User Interface Request Data*) signal

S9.ED8

'Status' in *Outcome Parameter Set* := END APPLICATION
'Msg On Error' in *Error Indication* := ERROR - OTHER CARD
'L1' in *Error Indication* := Return Code
SET 'Data Record Present' in *Outcome Parameter Set*
CreateEMVDataRecord ()
CreateEMVDiscretionaryData ()
Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 270

S9.E9

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S9.E10

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S9.E11

Use *DRDOL* to create *DRDOL Related Data* as a concatenated list of data objects without tags and lengths following the rules specified in section 4.1.4

Initialize(*Torn Temp Record*)

FOR every Data Object in Table 4.2 listed for both the EMV and EMV/DE implementations

{

IF [IsNotEmpty(TagOf(Data Object))]

THEN

AddToList(GetTLV(TagOf(Data Object)), *Torn Temp Record*)

ENDIF

}

S9.ED12

AddToList(GetTLV(TagOf(*IDS Status*)), *Torn Temp Record*)

IF ['Read' in *IDS Status* is set]

THEN

AddToList(GetTLV(TagOf(*DS Summary 1*)), *Torn Temp Record*)

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 271

S9.E13

IF [Number of records in Torn Transaction Log = *Max Number of Torn Transaction Log Records*]

THEN

Copy oldest record of Torn Transaction Log to *Torn Record*

Replace oldest record of Torn Transaction Log with *Torn Temp Record*

ELSE

Add *Torn Temp Record* to Torn Transaction Log

ENDIF

S9.E14

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S9.E15

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S9.E16

IF [SW12 = '9000']

THEN

GOTO S9.E18

ELSE

GOTO S9.E17

ENDIF

S9.E17

'L2' in *Error Indication* := STATUS BYTES

'SW12' in *Error Indication* := SW12

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 272

S9.E18

Parsing Result := FALSE

IF [(Length of Response Message Data Field > 0) AND
(Response Message Data Field[1] = '77')]

THEN

Parsing Result := ParseAndStoreCardResponse(Response Message Data Field)

ELSE

IF [(Length of Response Message Data Field > 0) AND
(Response Message Data Field[1] = '80')]

THEN

Retrieve *Cryptogram Information Data*, *Application Transaction Counter*, *Application Cryptogram* and *Issuer Application Data* from Response Message Data Field according to section 5.3.3

Store retrieved data objects in TLV Database

Set Parsing Result to TRUE if successful

ENDIF

ENDIF

S9.E19

IF [Parsing Result]

THEN

GOTO S9.E21

ELSE

GOTO S9.E20

ENDIF

S9.E20

'L2' in *Error Indication* := PARSING ERROR

S9.E21

IF [IsNotEmpty(TagOf(*Application Transaction Counter*))
AND IsNotEmpty(TagOf(*Cryptogram Information Data*))]

THEN

GOTO S9.E23

ELSE

GOTO S9.E22

ENDIF

S9.E22

'L2' in *Error Indication* := CARD DATA MISSING

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 273

S9.E23

IF [((*Cryptogram Information Data* AND 'C0' = '40')
AND
('AC type' in *Reference Control Parameter* = TC))
OR
((*Cryptogram Information Data* AND 'C0' = '80')
AND
(('AC type' in *Reference Control Parameter* = TC)
OR
('AC type' in *Reference Control Parameter* = ARQC)))
OR
(*Cryptogram Information Data* AND 'C0'= '00')]

THEN

GOTO S9.E25

ELSE

GOTO S9.E24

ENDIF

S9.E24

'L2' in *Error Indication* := CARD DATA ERROR

S9.E25

Perform Post-GenAC Balance Reading as specified in section 7.3

S9.ED26

IF [IsNotEmpty(*Tags To Write After Gen AC*)]

THEN

GOTO S9.E28

ELSE

GOTO S9.E27

ENDIF

S9.E27

'Message Identifier' in *User Interface Request Data* := CLEAR DISPLAY

'Status' in *User Interface Request Data* := CARD READ SUCCESSFULLY

'Hold Time' in *User Interface Request Data* := '000000'

Send MSG(*User Interface Request Data*) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 274

S9.E28

IF [IsNotEmpty(TagOf(*Signed Dynamic Application Data*))]
THEN
GOTO S910.E1
ELSE
GOTO S910.E30
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 275

6.15 State 10 – Waiting for Recover AC Response

6.15.1 Local Variables

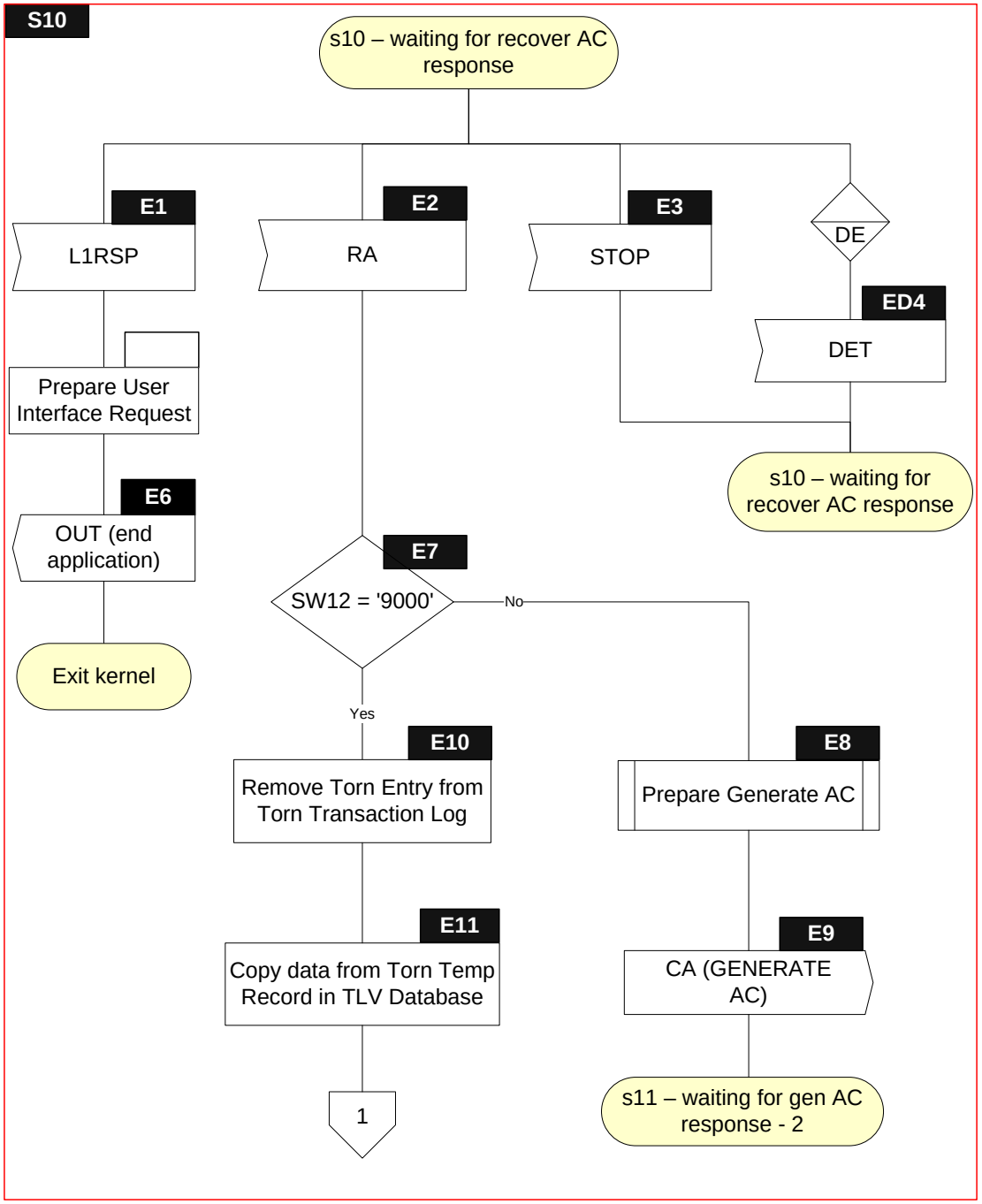
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of RECOVER AC
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 252	b	Value of TLV encoded string

6.15.2 Flow Diagram

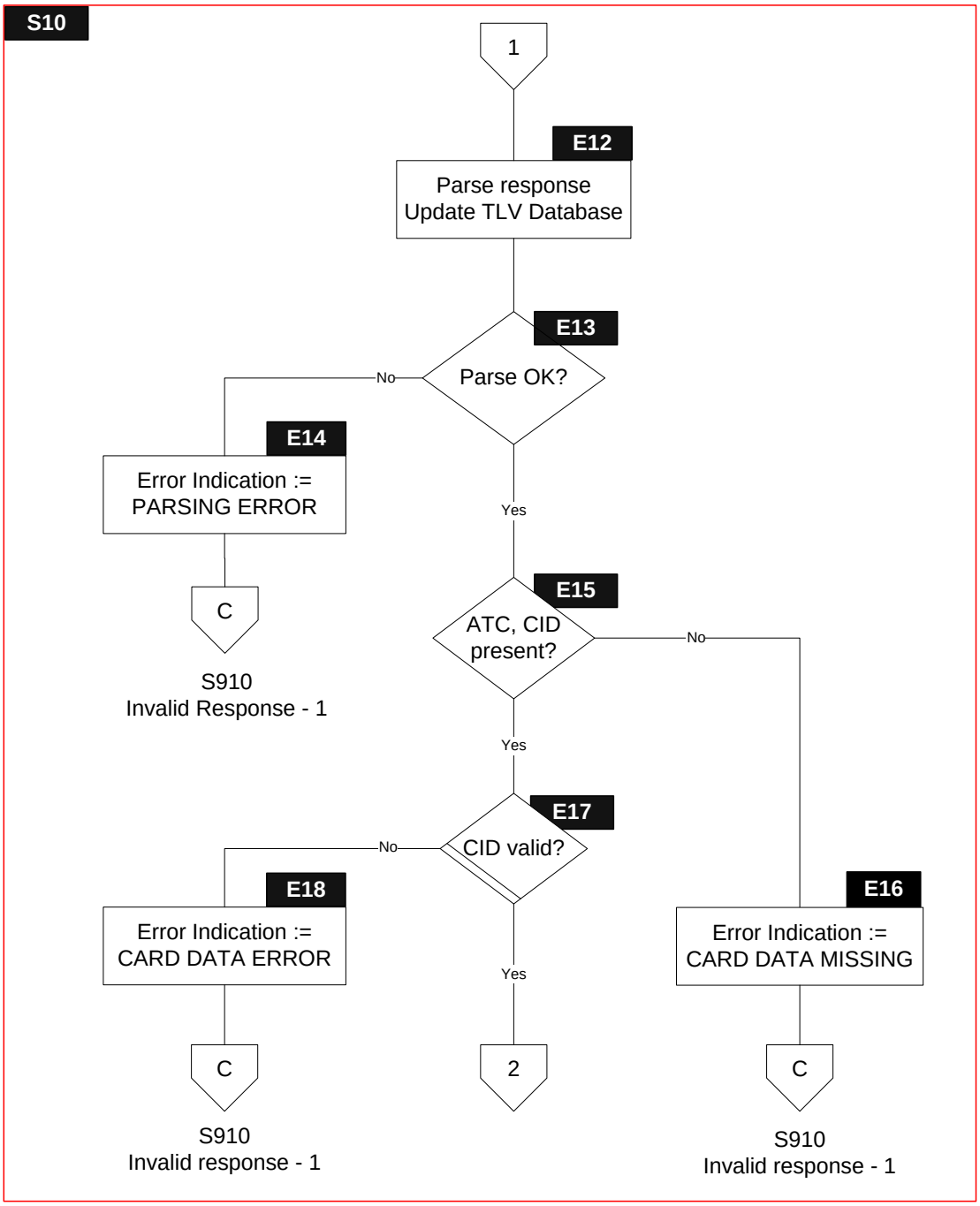
Figure 6.14 shows the flow diagram of `s10 – waiting for recover AC response`. Symbols in this diagram are labeled S10.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 276

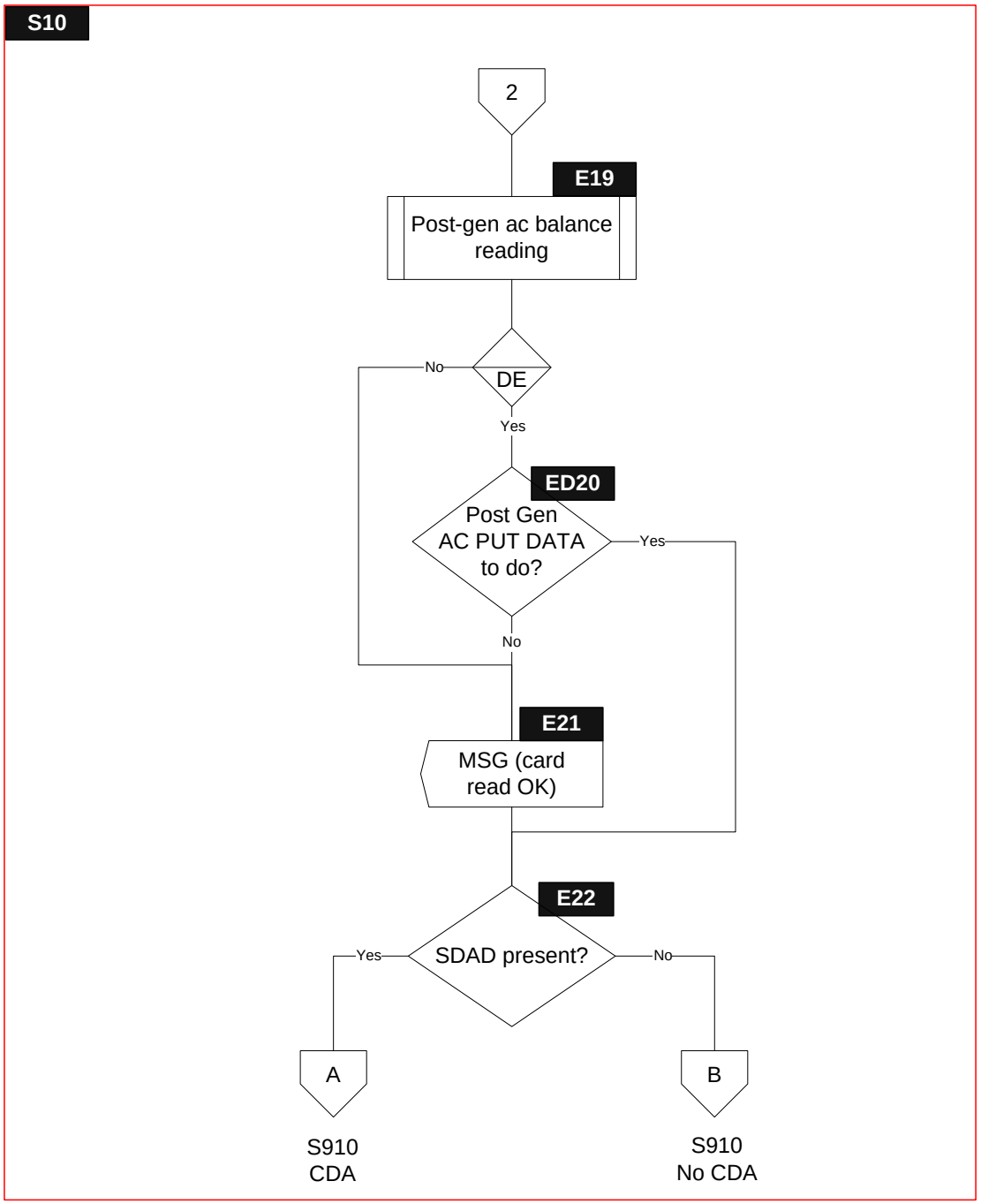
Figure 6.14—State 10 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 277



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 278



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 279

6.15.3 Processing

S10.E1

Receive L1RSP signal with Return Code

S10.E2

Receive RA signal with Response Message Data Field and SW12

S10.E3

Receive STOP signal

S10.ED4

Receive DET signal

S10.E5

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S10.E6

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S10.E7

IF [SW12 = '9000']

THEN

GOTO S10.E10

ELSE

GOTO S10.E8

ENDIF

S10.E8

Prepare GENERATE AC command as specified in section 7.6

S10.E9

Send the CA(GENERATE AC command) signal

S10.E10

Remove record referenced by *Torn Entry* from the Torn Transaction Log

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 280

S10.E11

FOR every primitive TLV in *Torn Temp Record*
{
    Store LV in the TLV Database for tag T
}
FOR every TL entry in *PDOL*
{
    Retrieve the corresponding value V from *PDOL Related Data*
    Store LV in the TLV Database for tag T
}
FOR every TL entry in *CDOL1*
{
    Retrieve the corresponding value V from *CDOL1 Related Data*
    Store LV in the TLV Database for tag T
}
FOR every TL entry in *DRDOL*
{
    Retrieve the corresponding value V from *DRDOL Related Data*
    Store LV in the TLV Database for tag T
}

S10.E12

Parsing Result := FALSE
IF [(Length of Response Message Data Field > 0) AND
(Response Message Data Field[1] = '77')]
THEN
    Parsing Result := ParseAndStoreCardResponse(Response Message Data Field)
ENDIF

S10.E13

IF [Parsing Result]
THEN
    GOTO S10.E15
ELSE
    GOTO S10.E14
ENDIF

S10.E14

'L2' in *Error Indication* := PARSING ERROR

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 281

S10.E15

IF [IsNotEmpty(TagOf(*Application Transaction Counter*))
AND IsNotEmpty(TagOf(*Cryptogram Information Data*))]

THEN

GOTO S10.E17

ELSE

GOTO S10.E16

ENDIF

S10.E16

'L2' in *Error Indication* := CARD DATA MISSING

S10.E17

IF [((*Cryptogram Information Data* AND 'C0' = '40')
AND
('AC type' in *Reference Control Parameter* = TC))
OR
((*Cryptogram Information Data* AND 'C0' = '80')
AND
(('AC type' in *Reference Control Parameter* = TC)
OR
('AC type' in *Reference Control Parameter* = ARQC)))
OR
(*Cryptogram Information Data* AND 'C0'= '00')]

THEN

GOTO S10.E19

ELSE

GOTO S10.E18

ENDIF

S10.E18

'L2' in *Error Indication* := CARD DATA ERROR

S10.E19

Perform Post-GenAC Balance Reading as specified in section 7.3

S10.ED20

IF [IsNotEmpty(*Tags To Write After Gen AC*)]

THEN

GOTO S10.E22

ELSE

GOTO S10.E21

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 282

S10.E21

'Message Identifier' in *User Interface Request Data* := CLEAR DISPLAY
'Status' in *User Interface Request Data* := CARD READ SUCCESSFULLY
'Hold Time' in *User Interface Request Data* := '000000'
Send MSG(*User Interface Request Data*) signal

S10.E22

IF [IsNotEmpty(TagOf(*Signed Dynamic Application Data*))]
THEN
    GOTO S910.E1
ELSE
    GOTO S910.E30
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 283

6.16 States 9 and 10 – Common Processing

6.16.1 Local Variables

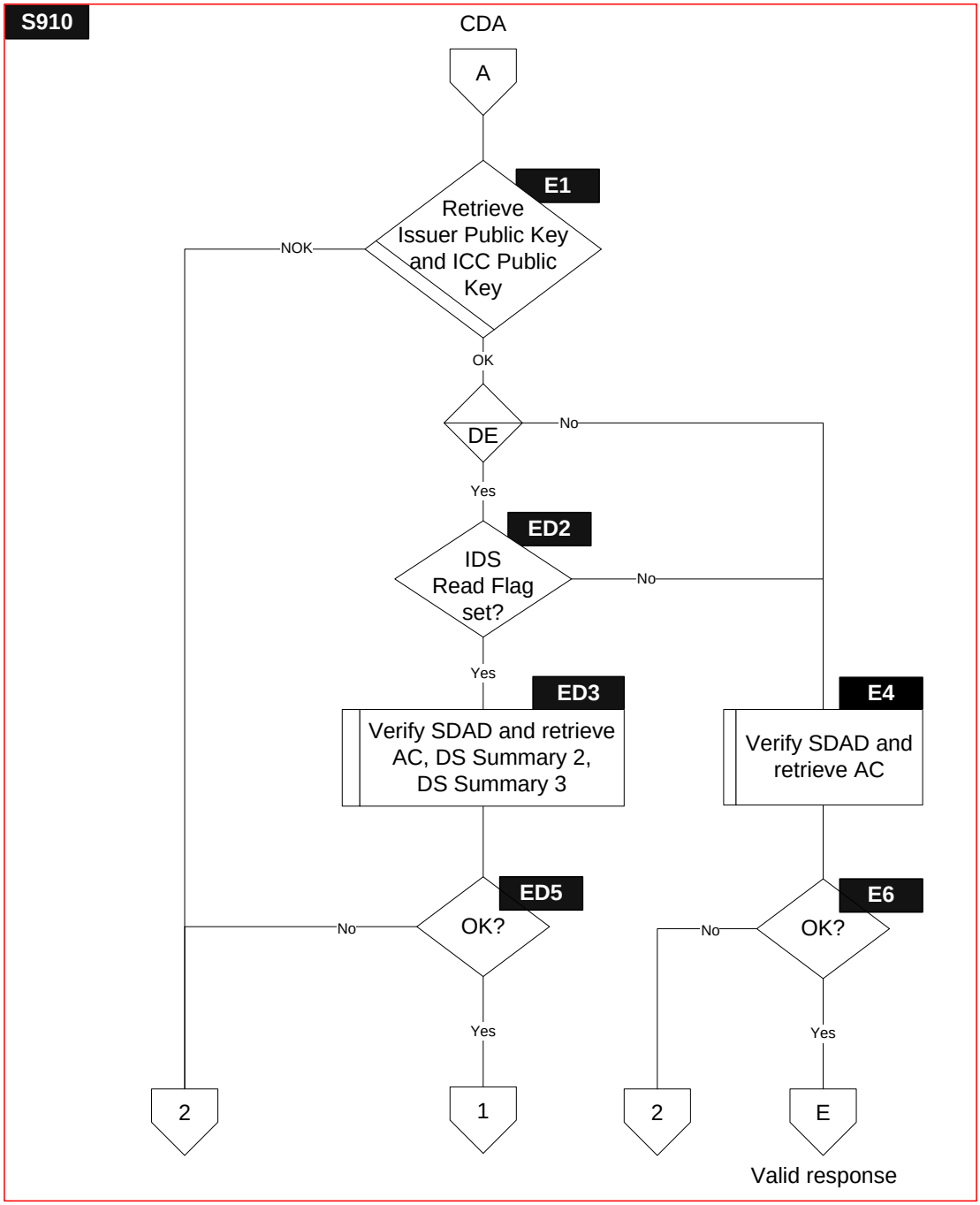
Local variables for common processing are defined in states 9 and 10.

6.16.2 Flow Diagram

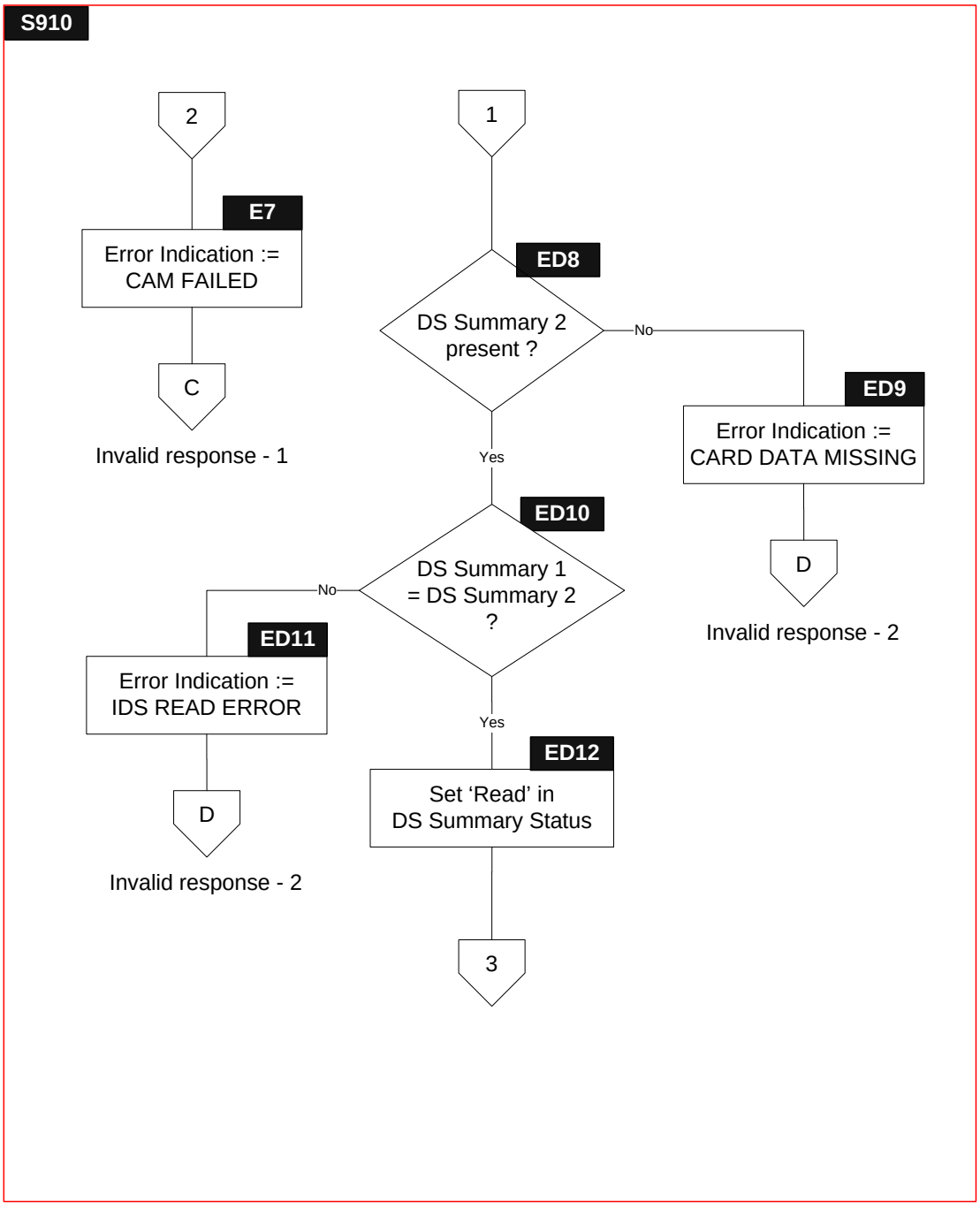
Figure 6.15 shows the flow diagram for common processing between states 9 and 10. Symbols in this diagram are labeled S910.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 284

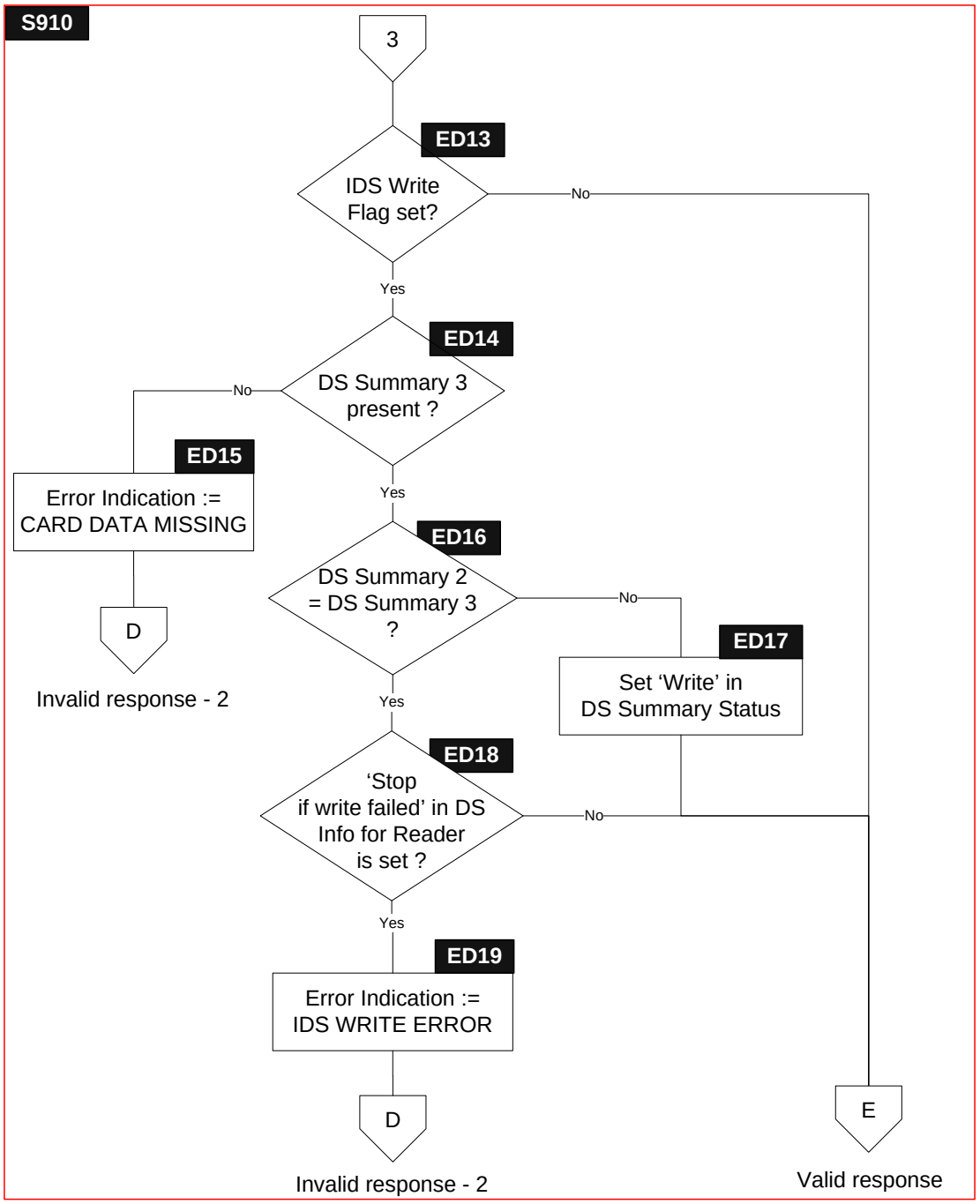
Figure 6.15—States 9 and 10 – Common Processing – Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 285

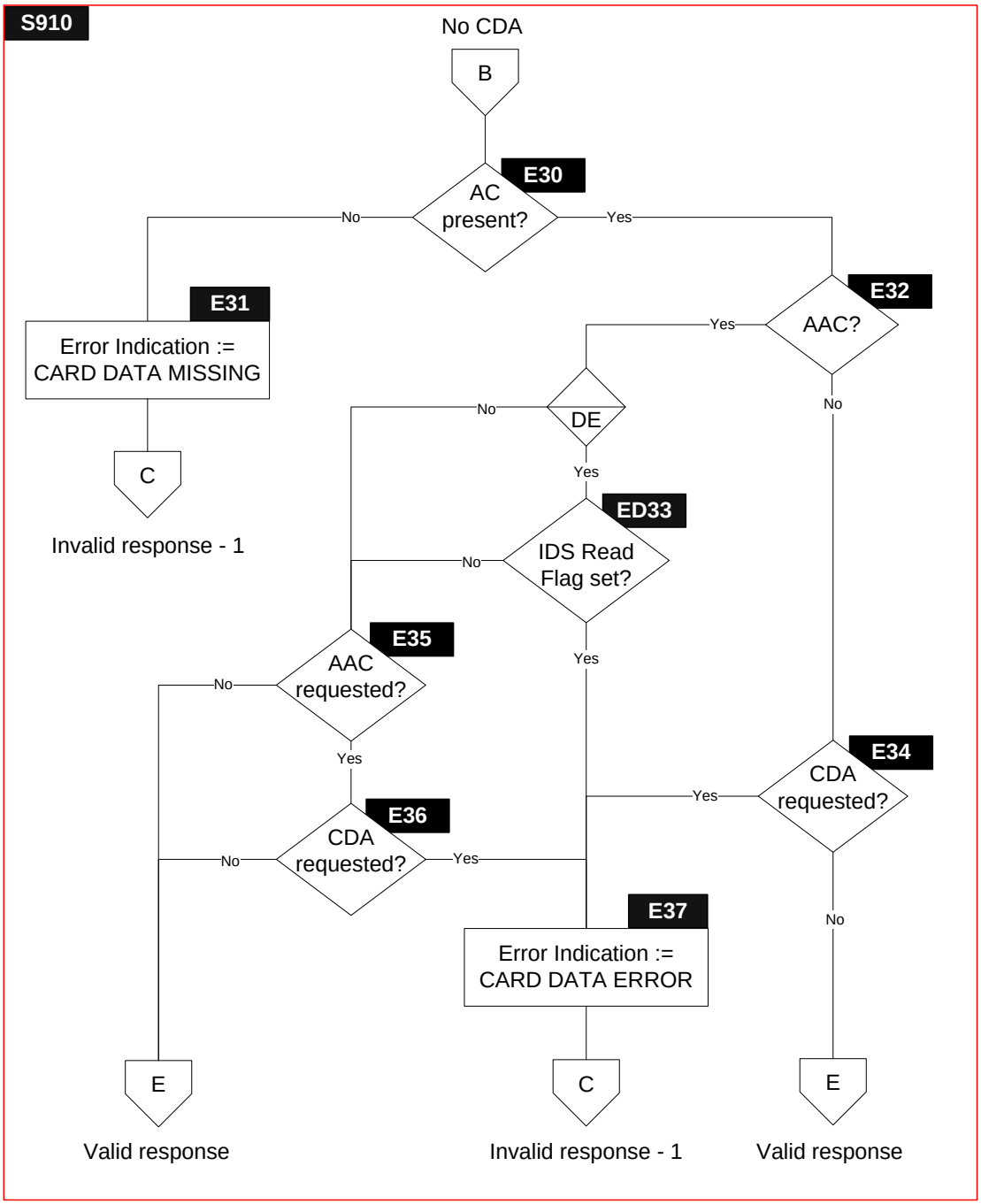


© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 286



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

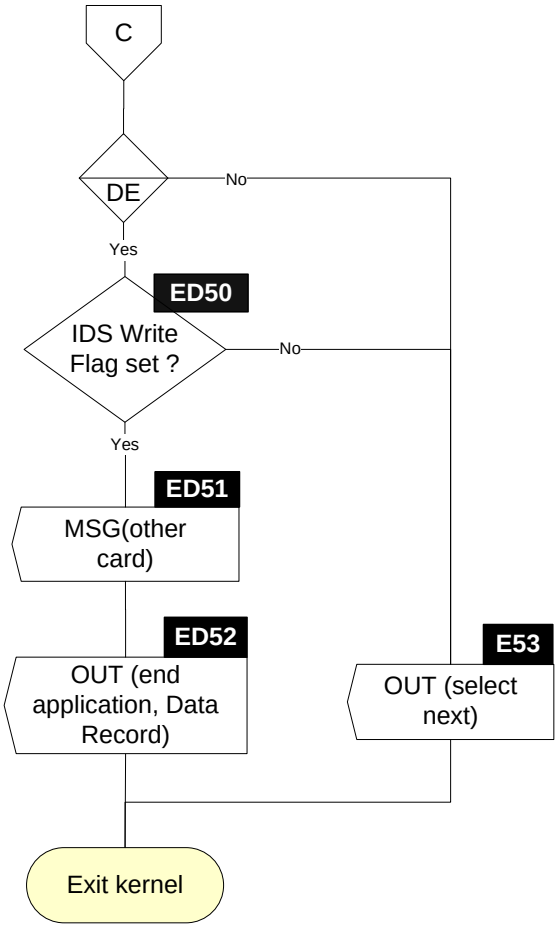
Mastercard, Ex1026, p. 287



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 288

S910

Invalid response - 1

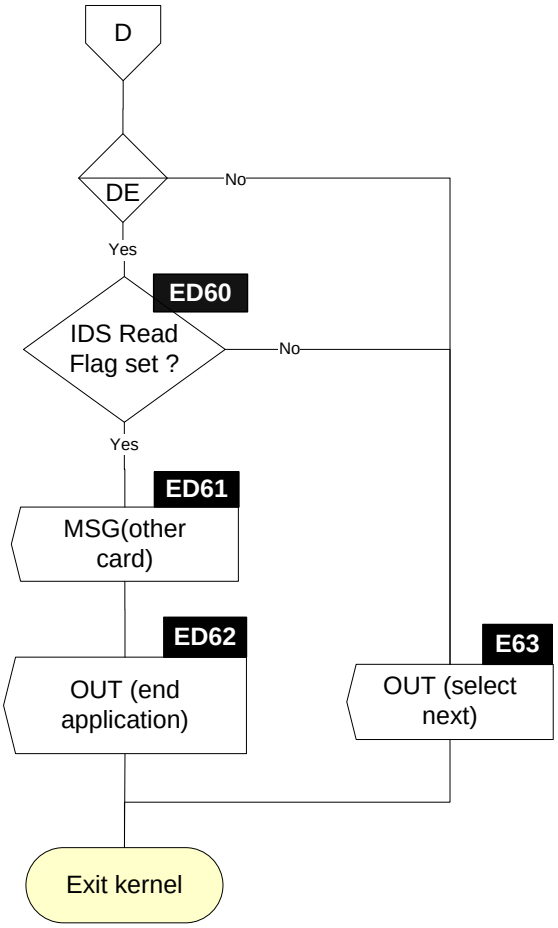


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

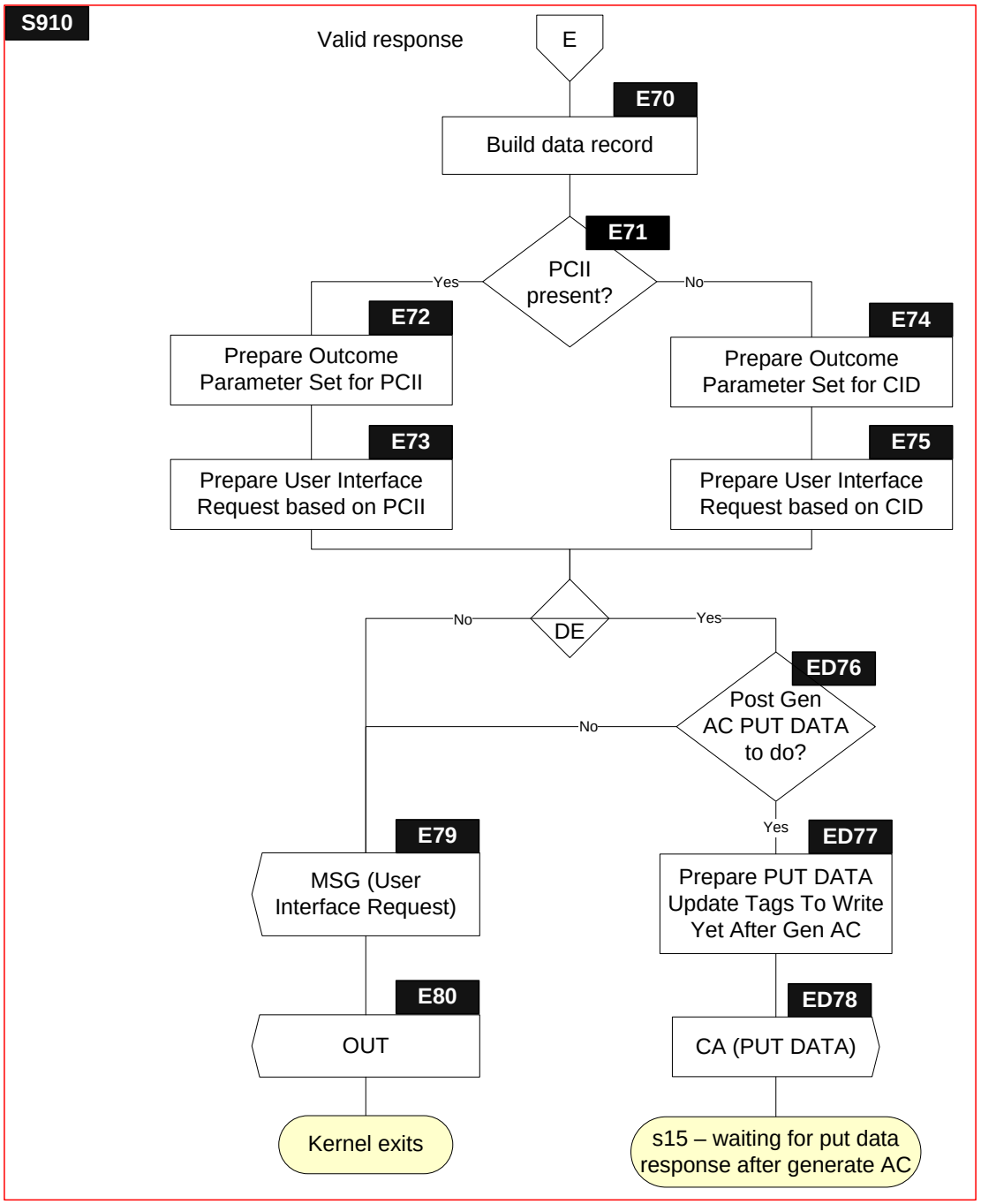
Mastercard, Ex1026, p. 289

S910

Invalid response - 2



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 290



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 291

6.16.3 Processing

CDA

S910.E1

Retrieve with the *CA Public Key Index (Card)* the Certification Authority Public Key Modulus and Exponent and associated key related information, and the corresponding algorithm to be used from the CA Public Key Database (see section 4.5.2).

Retrieve the Issuer Public Key and ICC Public Key as described in section 6.3 and 6.4 of [EMV Book 2].

Check if the concatenation of the *CA Public Key Index (Card)* and the Certificate Serial Number recovered from the *Issuer Public Key Certificate* appears in the CRL. If this is the case, then ICC Public Key retrieval is not successful.

IF [ICC Public Key retrieval was successful]

THEN

GOTO S910.ED2 or S910.E4

ELSE

GOTO S910.E7

ENDIF

S910.ED2

IF ['Read' in *IDS Status* is set]

THEN

GOTO S910.ED3

ELSE

GOTO S910.E4

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 292

S910.ED3

Verify *Signed Dynamic Application Data* as in section 6.6 of [EMV Book 2]. Retrieve from the ICC Dynamic Data (see Table 6.5) the *ICC Dynamic Number*, *Application Cryptogram*, *DS Summary 2* and *DS Summary 3* and store in the TLV Database.

If the ICC Dynamic Data does not include *DS Summary 3* (i.e. there are less than 16 bytes after Hash Result (if 'Data Storage Version Number' in *Application Capabilities Information* = VERSION 1) or less than 32 bytes (if 'Data Storage Version Number' in *Application Capabilities Information* = VERSION 2)), then do not store *DS Summary 3*. This is not a reason to fail CDA.

If the ICC Dynamic Data does also not include *DS Summary 2* (i.e. there are less than 8 bytes after Hash Result (if 'Data Storage Version Number' in *Application Capabilities Information* = VERSION 1) or less than 16 bytes (if 'Data Storage Version Number' in *Application Capabilities Information* = VERSION 2)), then do not store *DS Summary 2*. This is not a reason to fail CDA.

Table 6.5—ICC Dynamic Data (IDS)

Value	Length
Length of *ICC Dynamic Number*	1
ICC Dynamic Number	2-8
Cryptogram Information Data	1
Application Cryptogram	8
Hash Result	20
DS Summary 2	8 or 16
DS Summary 3	8 or 16

S910.E4

Verify *Signed Dynamic Application Data* as in section 6.6 of [EMV Book 2]. Retrieve from the ICC Dynamic Data (see Table 6.6) the *ICC Dynamic Number* and *Application Cryptogram* and store in the TLV Database.

Table 6.6—ICC Dynamic Data (No IDS)

Value	Length
Length of *ICC Dynamic Number*	1
ICC Dynamic Number	2-8
Cryptogram Information Data	1
Application Cryptogram	8
Hash Result	20

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 293

S910.ED5

IF [*Signed Dynamic Application Data* verification is OK]
THEN
GOTO S910.ED8
ELSE
GOTO S910.E7
ENDIF

S910.E6

IF [*Signed Dynamic Application Data* verification is OK]
THEN
GOTO S910.E70
ELSE
GOTO S910.E7
ENDIF

S910.E7

'L2' in *Error Indication* := CAM FAILED

S910.ED8

IF [IsPresent(TagOf(*DS Summary 2*))]
THEN
GOTO S910.ED10
ELSE
GOTO S910.ED9
ENDIF

S910.ED9

'L2' in *Error Indication* := CARD DATA MISSING

S910.ED10

IF [*DS Summary 1 = DS Summary 2*]
THEN
GOTO S910.ED12
ELSE
GOTO S910.ED11
ENDIF

S910.ED11

'L2' in *Error Indication* := IDS READ ERROR

S910.ED12

SET 'Successful Read' in *DS Summary Status*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 294

S910.ED13

IF ['Write' in *IDS Status* is set]
THEN
GOTO S910.ED14
ELSE
GOTO S910.E70
ENDIF

S910.ED14

IF [IsPresent(TagOf(*DS Summary 3*))]
THEN
GOTO S910.ED16
ELSE
GOTO S910.ED15
ENDIF

S910.ED15

'L2' in *Error Indication* := CARD DATA MISSING

S910.ED16

IF [*DS Summary 2 = DS Summary 3*]
THEN
GOTO S910.ED18
ELSE
GOTO S910.ED17
ENDIF

S910.ED17

SET 'Successful Write' in *DS Summary Status*

S910.ED18

IF ['Stop if write failed' in *DS ODS Info For Reader* is set]
THEN
GOTO S910.ED19
ELSE
GOTO S910.E70
ENDIF

S910.ED19

'L2' in *Error Indication* := IDS WRITE ERROR

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 295

No CDA

S910.E30

IF [IsNotEmpty(TagOf(*Application Cryptogram*))]
THEN
GOTO S910.E32
ELSE
GOTO S910.E31
ENDIF

S910.E31

'L2' in *Error Indication* := CARD DATA MISSING

S910.E32

IF [(*Cryptogram Information Data* AND 'C0') = '00']
THEN
GOTO S910.ED33or S910.E70
ELSE
GOTO S910.E34
ENDIF

S910.ED33

IF ['Read' in *IDS Status* is set]
THEN
GOTO S910.E37
ELSE
GOTO S910.E35
ENDIF

S910.E34

IF ['CDA signature requested' in *Reference Control Parameter* is set]
THEN
GOTO S910.E37
ELSE
GOTO S910.E70
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 296

S910.E35

IF ['AC type' in *Reference Control Parameter* = AAC]
THEN
GOTO S910.E36
ELSE
GOTO S910.E70
ENDIF

S910.E36

IF ['CDA signature requested' in *Reference Control Parameter* is set]
THEN
GOTO S910.E37
ELSE
GOTO S910.E70
ENDIF

S910.E37

'L2' in *Error Indication* := CARD DATA ERROR

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 297

Invalid Response – 1

S910.ED50

IF ['Write' in *IDS Status* is set]

THEN

GOTO S910.ED51

ELSE

GOTO S910.E53

ENDIF

S910.ED51

'Message Identifier' in *User Interface Request Data* := ERROR - OTHER CARD

'Status' in *User Interface Request Data* := NOT READY

Send MSG(*User Interface Request Data*) signal

S910.ED52

'Status' in *Outcome Parameter Set* := END APPLICATION

'Msg On Error' in *Error Indication* := ERROR - OTHER CARD

SET 'Data Record Present' in *Outcome Parameter Set*

CreateEMVDataRecord ()

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

S910.E53

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 298

Invalid Response – 2

S910.ED60

IF ['Read' in *IDS Status* is set]

THEN

GOTO S910.ED61

ELSE

GOTO S910.E63

ENDIF

S910.ED61

'Message Identifier' in *User Interface Request Data* := ERROR - OTHER CARD

'Status' in *User Interface Request Data* := NOT READY

Send MSG(*User Interface Request Data*) signal

S910.ED62

'Status' in *Outcome Parameter Set* := END APPLICATION

'Msg On Error' in *Error Indication* := ERROR - OTHER CARD

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S910.E63

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 299

Valid Response

S910.E70

SET 'Data Record Present' in *Outcome Parameter Set*

CreateEMVDataRecord ()

S910.E71

IF [IsNotEmpty(TagOf(*POS Cardholder Interaction Information*))]
THEN
    GOTO S910.E72
ELSE
    GOTO S910.E74
ENDIF

S910.E72

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

S910.E73

FOR every entry in the Phone Message Table (see section 4.5.4)
{
    IF [(PCII MASK[ID] AND *POS Cardholder Interaction Information*) = PCII VALUE[ID]]
    THEN
        'Hold Time' in *User Interface Request Data* := *Message Hold Time*
        'Message Identifier' in *User Interface Request Data* := MESSAGE[ID]
        'Status' in *User Interface Request Data* := STATUS[ID]
        EXIT loop
    ENDIF
}

S910.E74

IF [(*Cryptogram Information Data* AND 'C0') = '40']
THEN
    'Status' in *Outcome Parameter Set* := APPROVED
ELSE
    IF [(*Cryptogram Information Data* AND 'C0') = '80']
    THEN
        'Status' in *Outcome Parameter Set* := ONLINE REQUEST
    ELSE
        'Status' in *Outcome Parameter Set* := DECLINED
    ENDIF
ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 300

S910.E75

'Status' in *User Interface Request Data* := NOT READY
IF [(*Cryptogram Information Data* AND 'C0') = '40']
THEN
    'Hold Time' in *User Interface Request Data* := *Message Hold Time*
    IF [IsNotEmpty(*Balance Read After Gen AC*)]
    THEN
        'Value Qualifier' in *User Interface Request Data* := BALANCE
        'Value' in *User Interface Request Data* := *Balance Read After Gen AC*
        IF [IsNotEmpty(TagOf(*Application Currency Code*))]
        THEN
            'Currency Code' in *User Interface Request Data* := *Application Currency Code*
        ENDIF
    ENDIF
    IF ['CVM' in *Outcome Parameter Set* = OBTAIN SIGNATURE]
    THEN
        'Message Identifier' in *User Interface Request Data* := APPROVED - SIGN
    ELSE
        'Message Identifier' in *User Interface Request Data* := APPROVED
    ENDIF
ELSE
    IF [(*Cryptogram Information Data* AND 'C0') = '80']
    THEN
        'Hold Time' in *User Interface Request Data* := '000000'
        'Message Identifier' in *User Interface Request Data* := AUTHORISING – PLEASE WAIT
    ELSE
        'Hold Time' in *User Interface Request Data* := *Message Hold Time*
        'Message Identifier' in *User Interface Request Data* := DECLINED
    ENDIF
ENDIF

S910.ED76

IF [IsNotEmptyList(*Tags To Write After Gen AC*)]
THEN
    GOTO S910.ED77
ELSE
    GOTO S910.E79
ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 301

S910.ED77

TLV = GetAndRemoveFromList(*Tags To Write Yet After Gen AC*)

Prepare the PUT DATA command with TLV as defined in section 5.6

S910.ED78

Send CA(PUT DATA command) signal

S910.E79

Send MSG(*User Interface Request Data*) signal

S910.E80

CreateEMVDiscretionaryData ()

IF [IsNotEmpty(TagOf(*POS Cardholder Interaction Information*))]

THEN

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

ELSE

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 302

6.17 State 11 – Waiting for Generate AC Response - 2

6.17.1 Local Variables

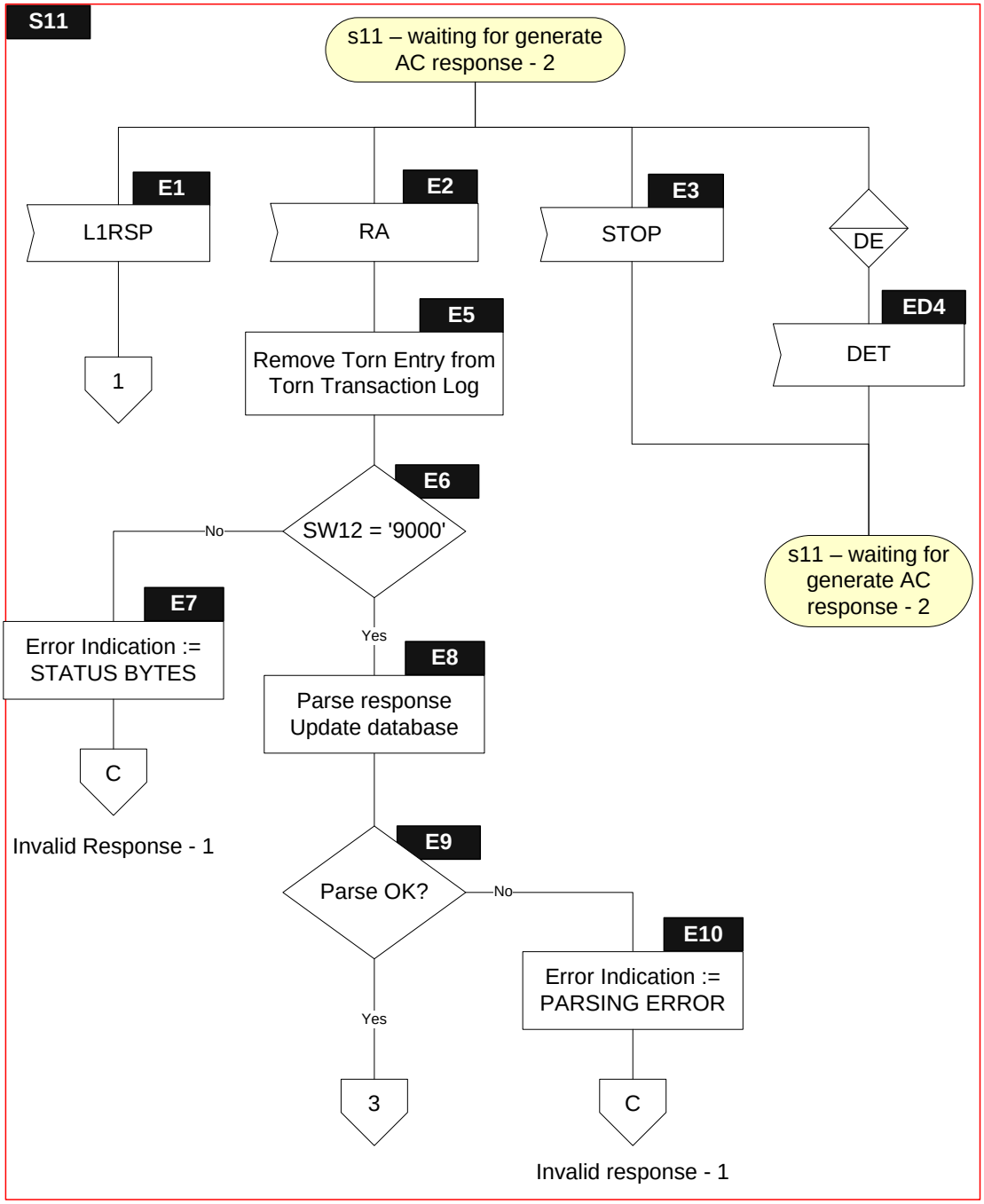
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of GENERATE AC
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 252	b	Value of TLV encoded string

6.17.2 Flow Diagram

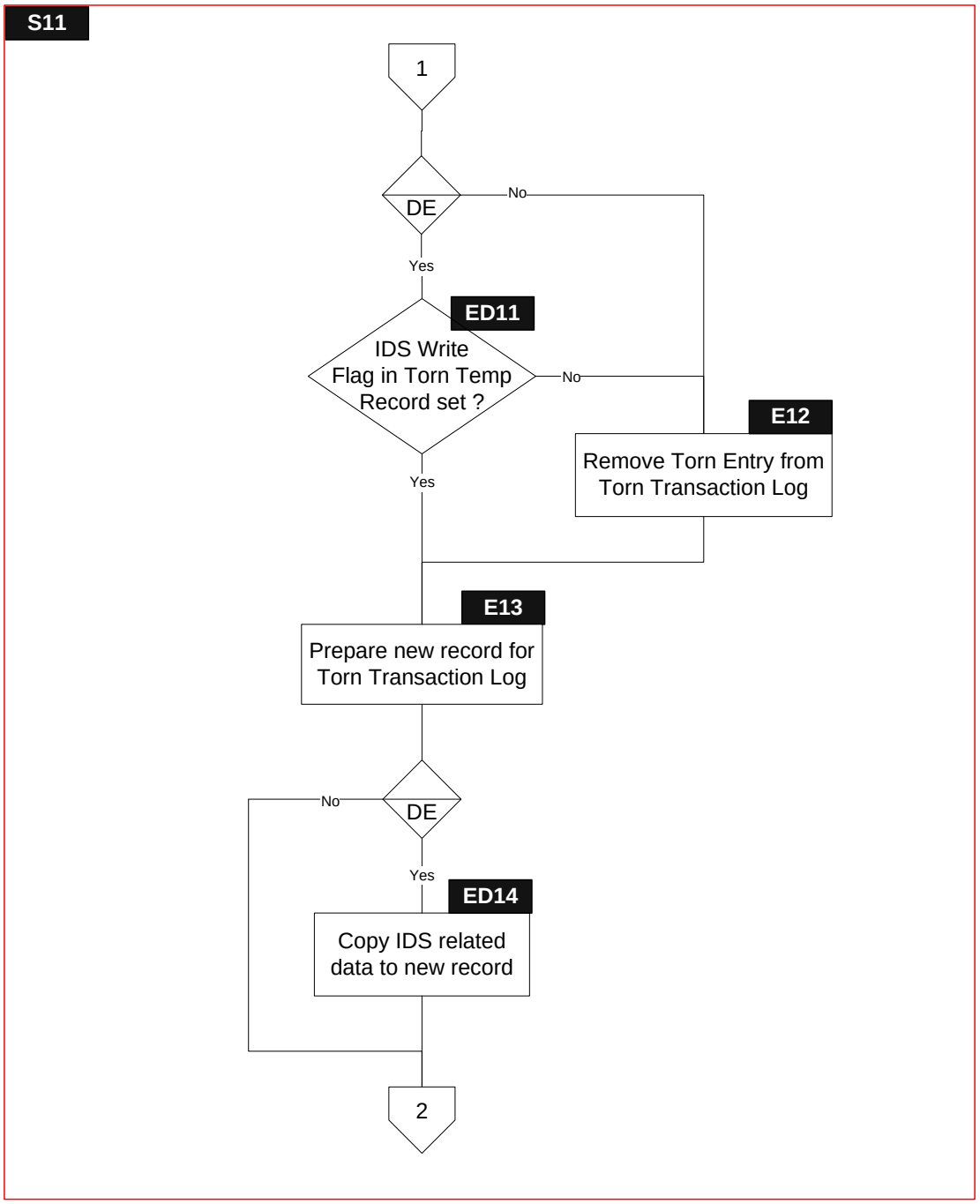
Figure 6.16 shows the flow diagram of `s11 – waiting for generate AC response -` 2. Symbols in this diagram are labeled S11.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 303

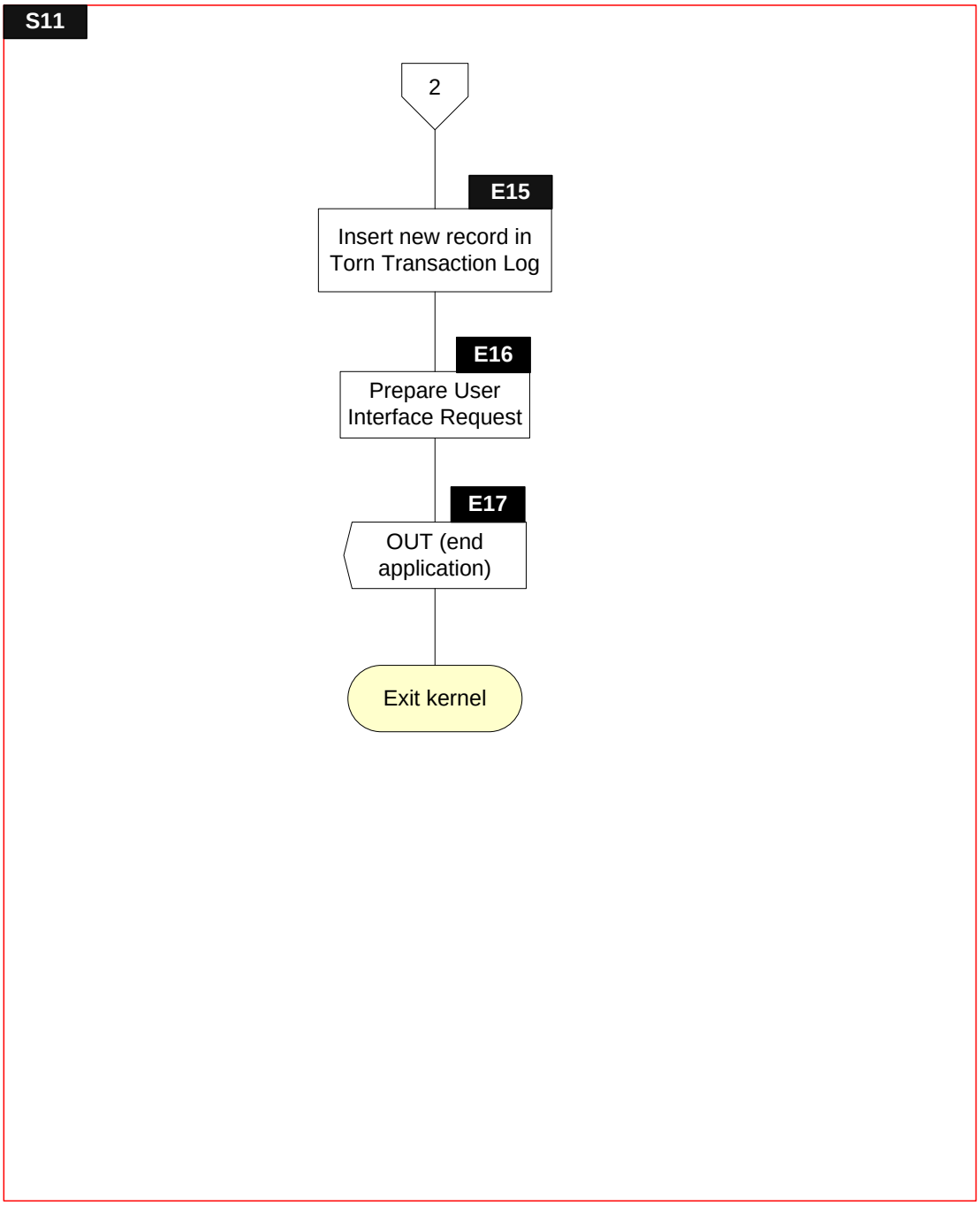
Figure 6.16—State 11 Flow Diagram



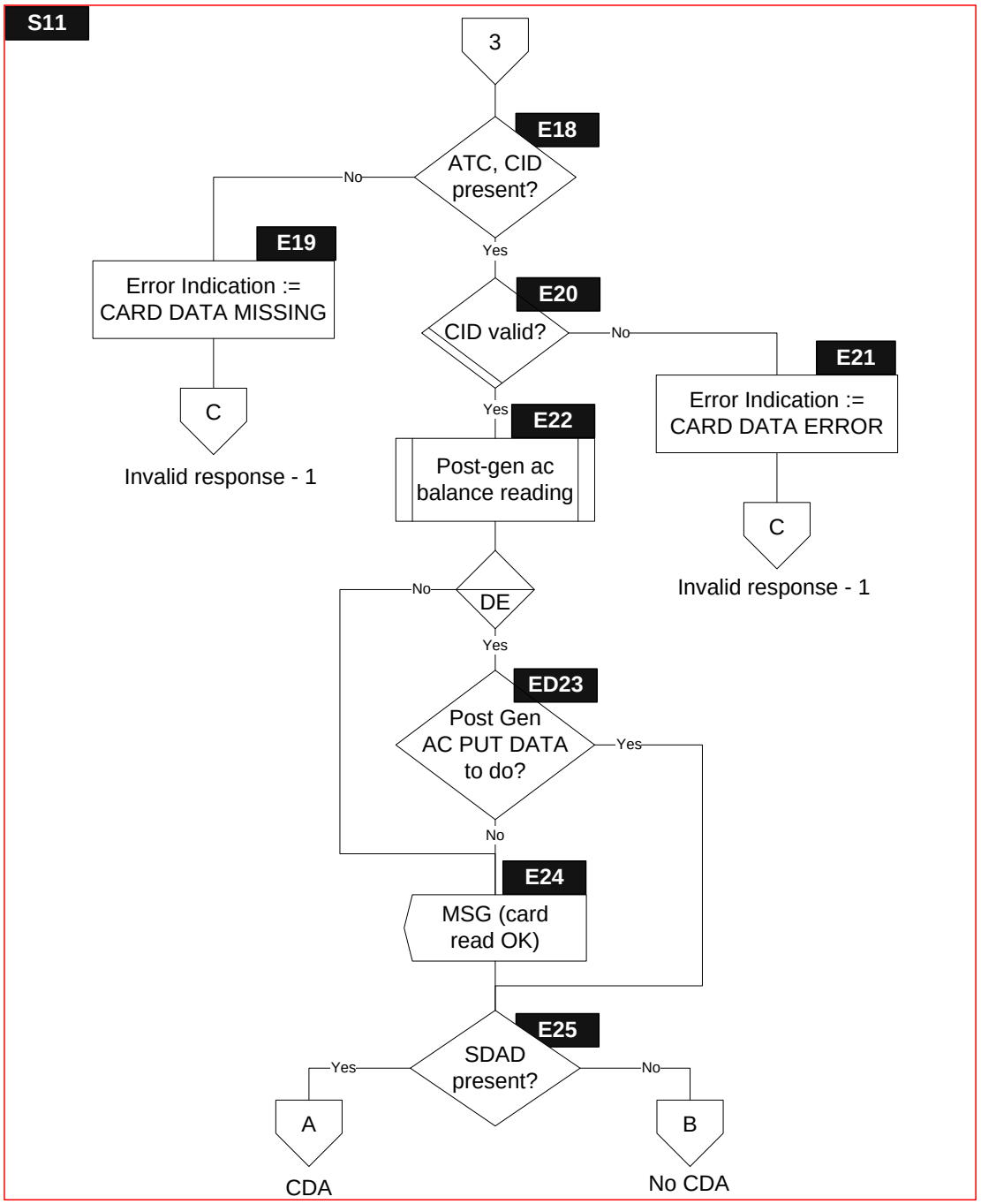
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 304



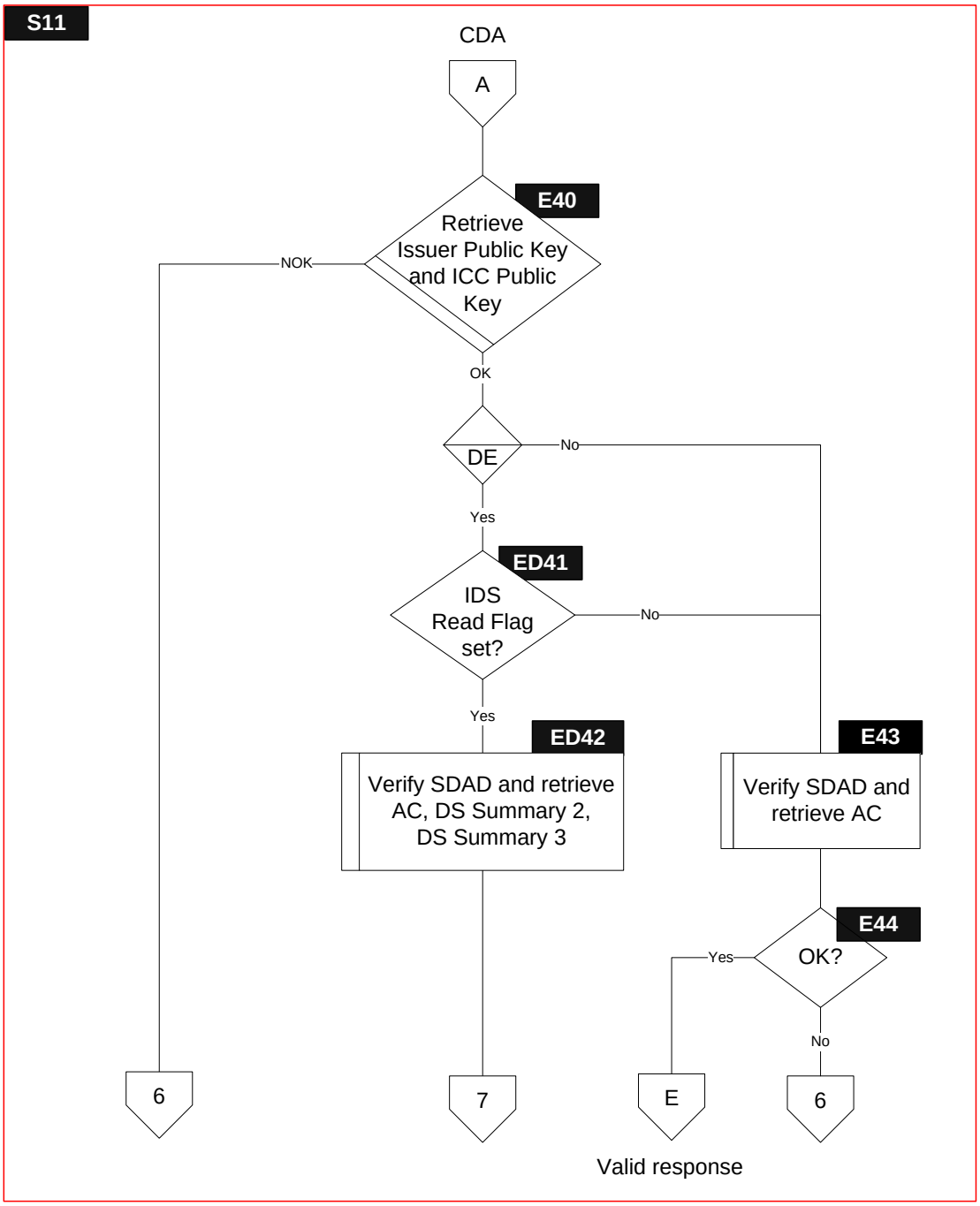
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 305



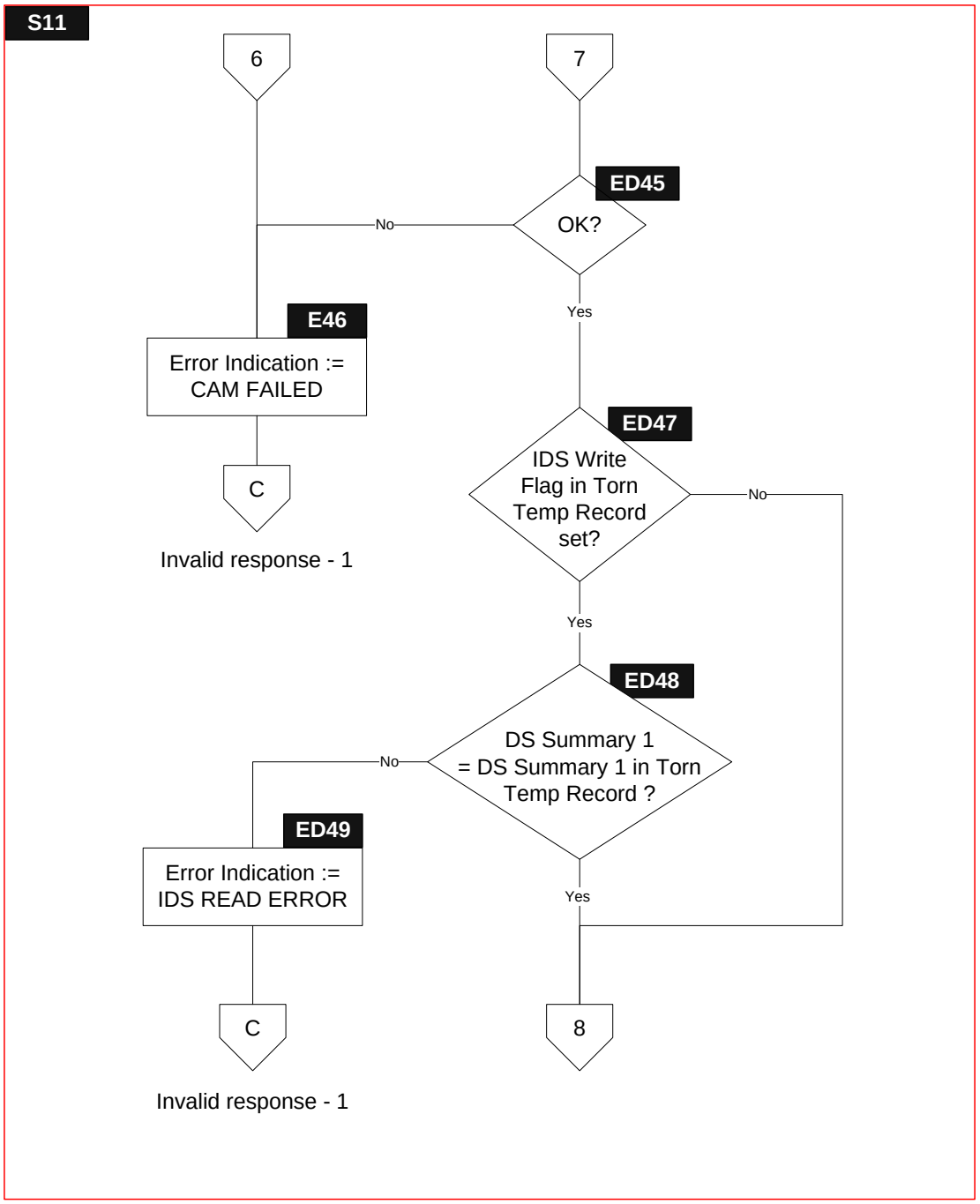
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 306



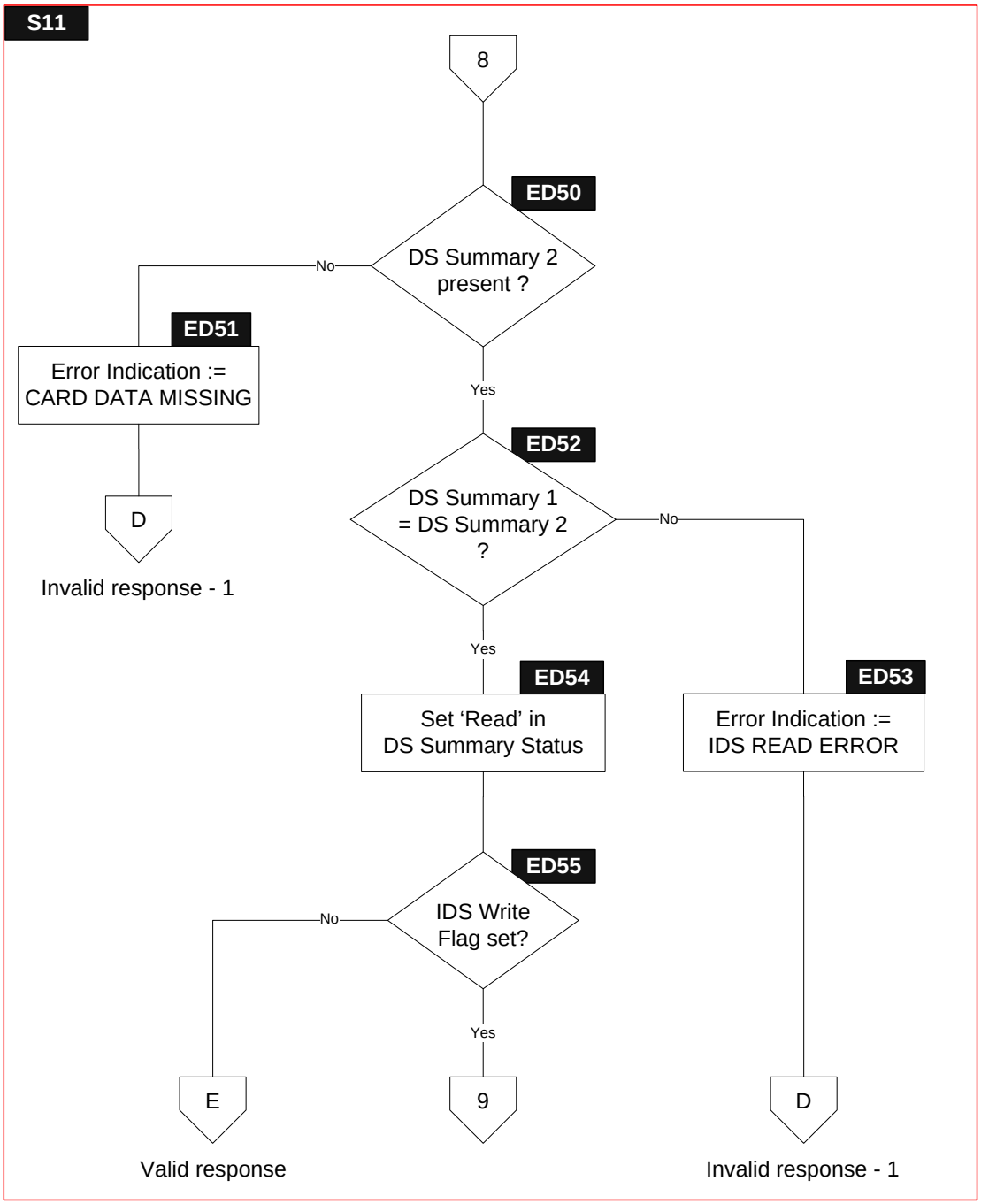
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 307



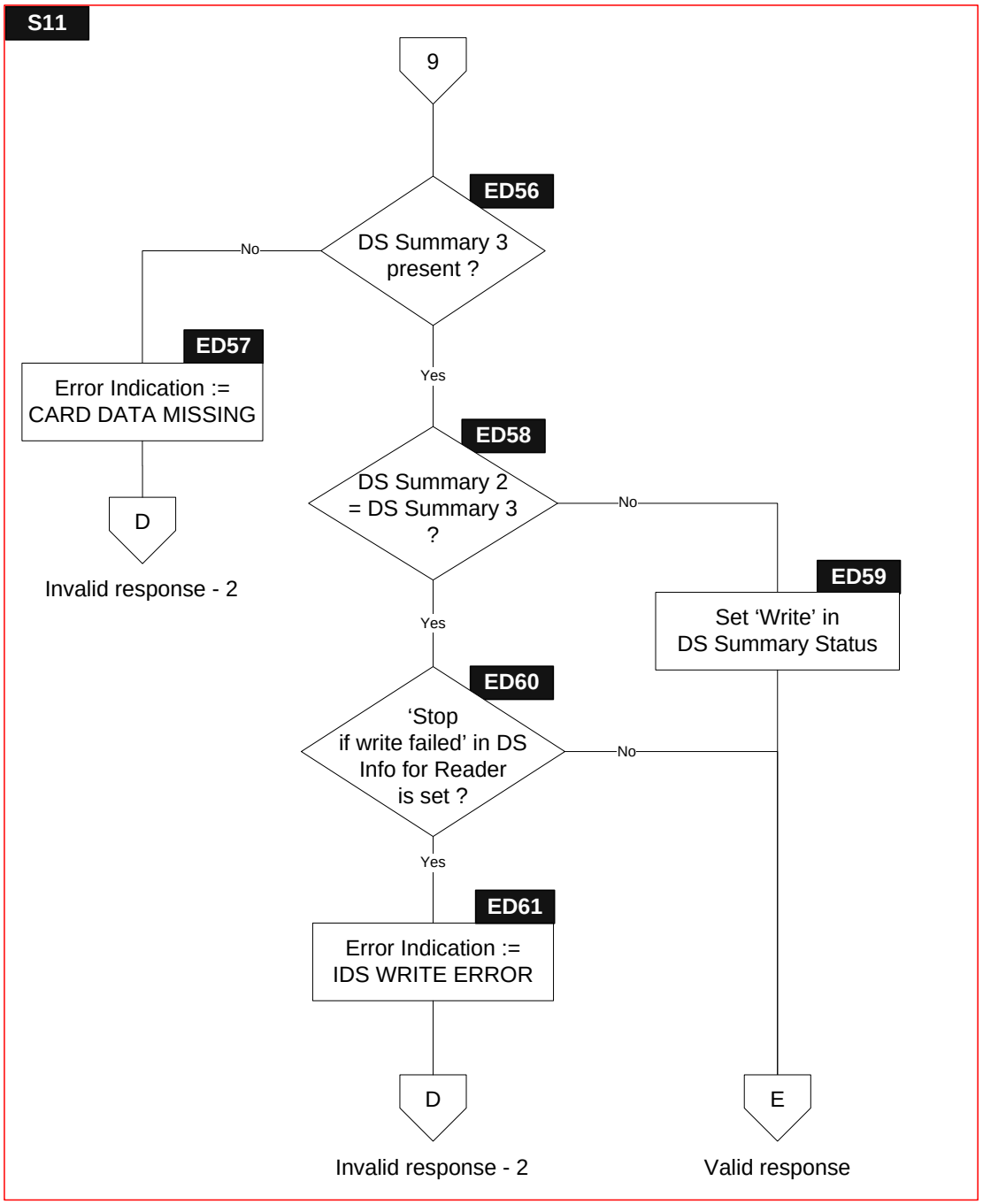
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 308



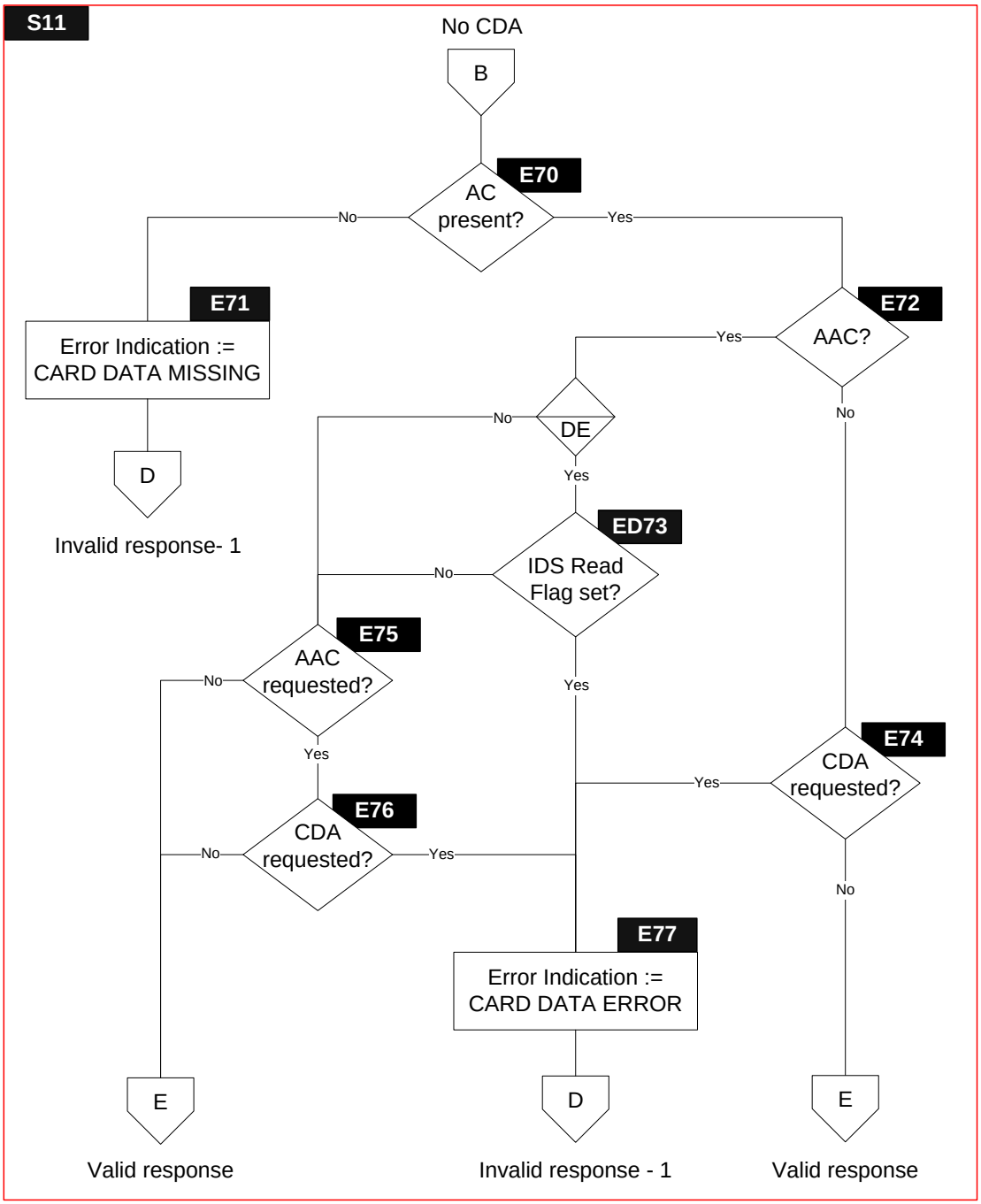
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 309



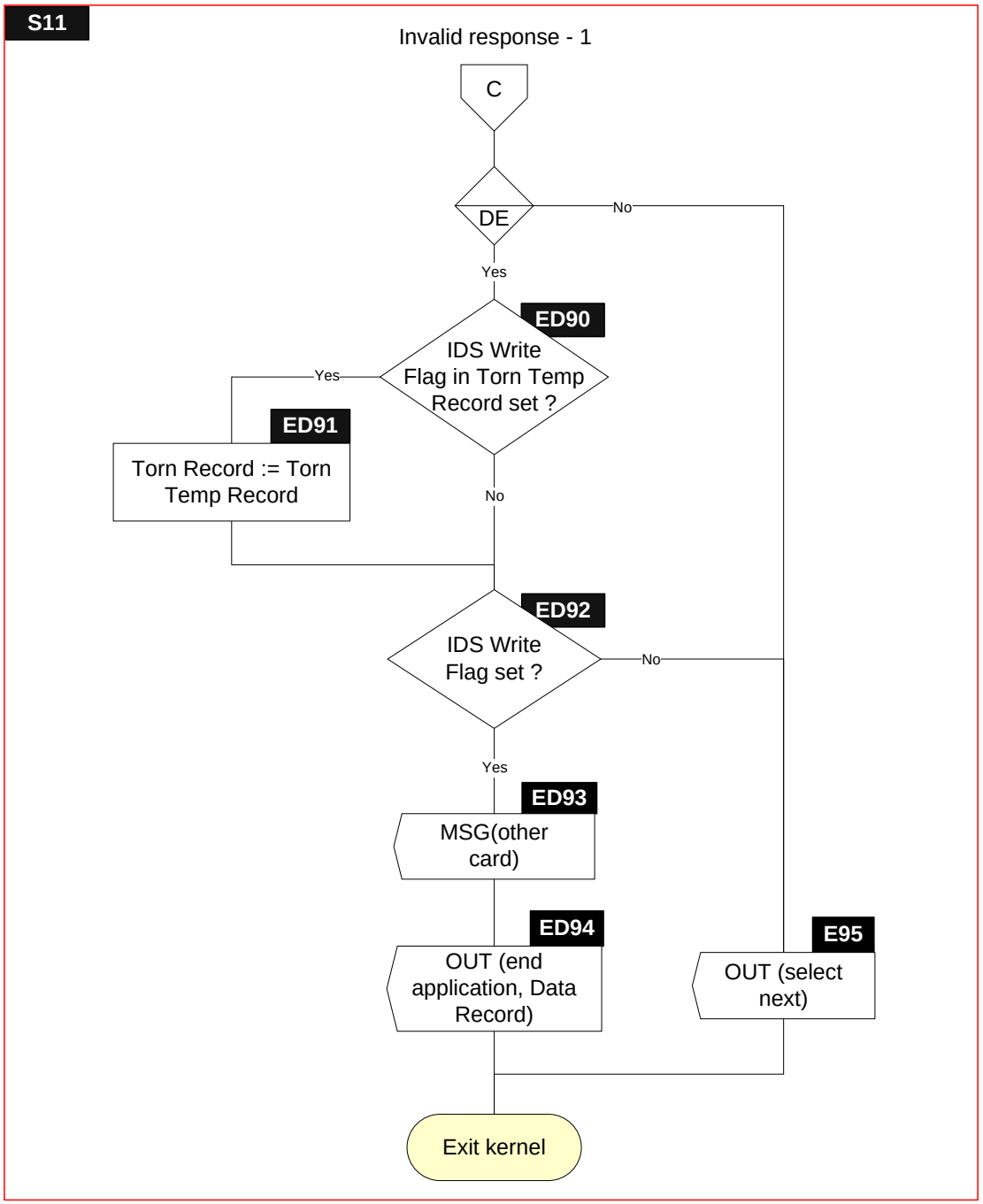
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 310



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 311



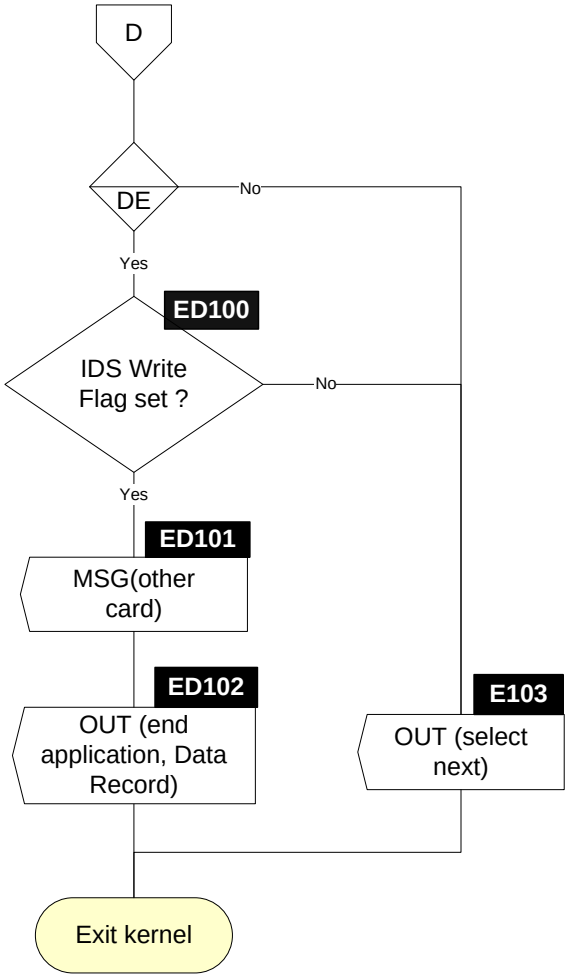
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 312



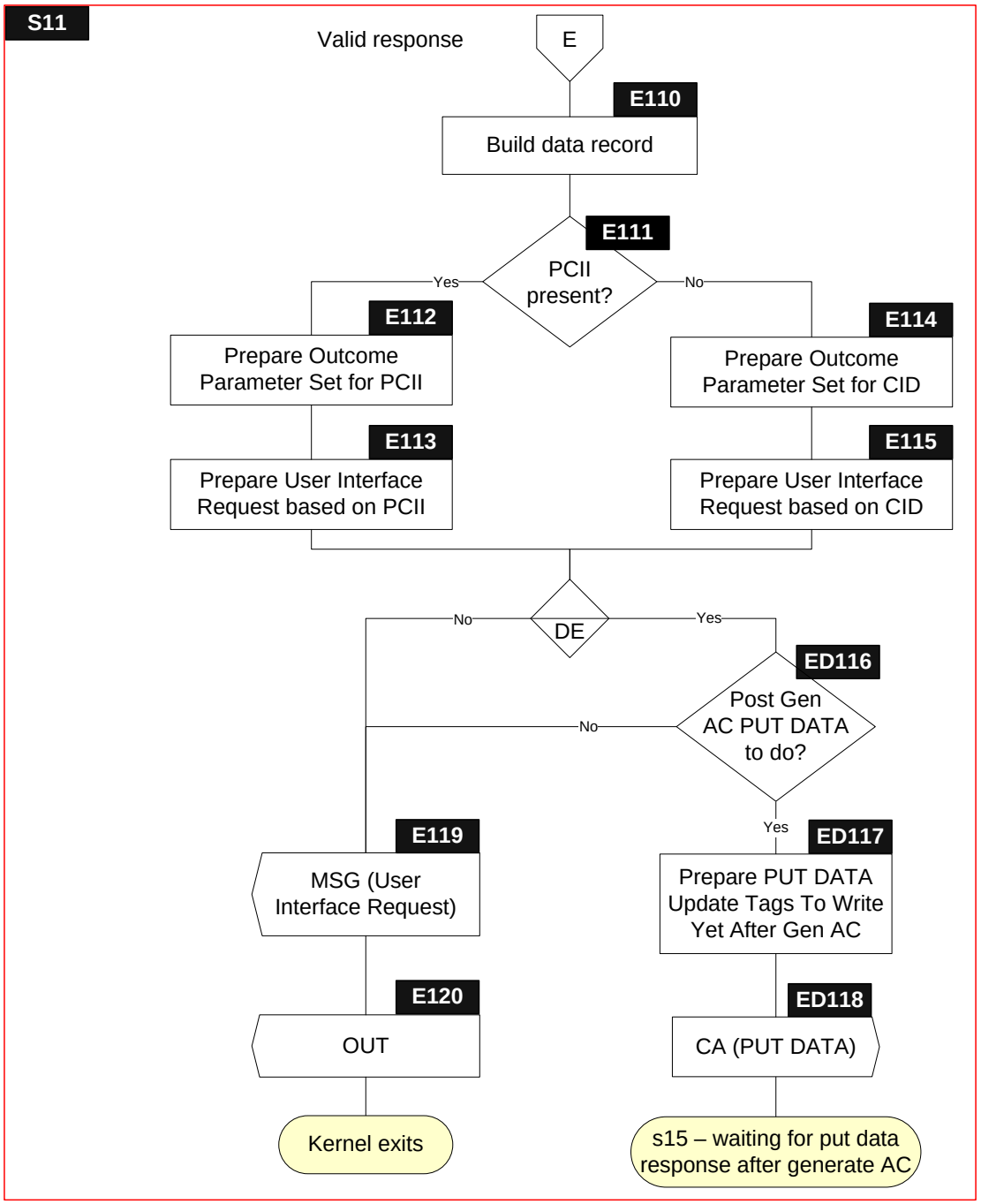
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 313

S11

Invalid response - 2



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 314



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 315

6.17.3 Processing

S11.E1

Receive L1RSP signal with Return Code

S11.E2

Receive RA signal with Response Message Data Field and SW12

S11.E3

Receive STOP signal

S11.ED4

Receive DET signal

S11.E5

Remove record referenced by *Torn Entry* from the Torn Transaction Log

S11.E6

IF [SW12 = '9000']
THEN
    GOTO S11.E8
ELSE
    GOTO S11.E7
ENDIF

S11.E7

'L2' in *Error Indication* := STATUS BYTES
'SW12' in *Error Indication* := SW12

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 316

S11.E8

Parsing Result := FALSE

IF [(Length of Response Message Data Field > 0) AND
(Response Message Data Field[1] = '77')]

THEN

Parsing Result := ParseAndStoreCardResponse(Response Message Data Field)

ELSE

IF [(Length of Response Message Data Field > 0) AND
(Response Message Data Field[1] = '80')]

THEN

Retrieve *Cryptogram Information Data*, *Application Transaction Counter*, *Application Cryptogram* and *Issuer Application Data* from Response Message Data Field according to section 5.3.3 and store in TLV Database

Set Parsing Result to TRUE if successful

ENDIF

ENDIF

S11.E9

IF [Parsing Result]

THEN

GOTO S11.E18

ELSE

GOTO S11.E10

ENDIF

S11.E10

'L2' in *Error Indication* := PARSING ERROR

S11.ED11

IF ['Write' in *IDS Status* in *Torn Temp Record* is set]

THEN

GOTO S11.E13

ELSE

GOTO S11.E15

ENDIF

S11.E12

Remove record referenced by *Torn Entry* from the Torn Transaction Log

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 317

S11.E13

Use *DRDOL* to create *DRDOL Related Data* as a concatenated list of data objects without tags and lengths following the rules specified in section 4.1.4

Initialize(*Torn Temp Record*)

FOR every Data Object in Table 4.2 listed for both the EMV and EMV/DE implementations

{

IF [IsNotEmpty(TagOf(Data Object))]

THEN

AddToList(GetTLV(TagOf(Data Object)), *Torn Temp Record*)

ENDIF

}

S11.ED14

AddToList(GetTLV(TagOf(*IDS Status*)), *Torn Temp Record*)

IF ['Read' in *IDS Status* is set]

THEN

AddToList(GetTLV(TagOf(*DS Summary 1*)), *Torn Temp Record*)

ENDIF

S11.E15

IF [Number of records in Torn Transaction Log = *Max Number of Torn Transaction Log Records*]

THEN

Copy oldest record of Torn Transaction Log in *Torn Record*

Replace oldest record of Torn Transaction Log with *Torn Temp Record*

ELSE

Add *Torn Temp Record* to Torn Transaction Log

ENDIF

S11.E16

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 318

S11.E17

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication* := TRY AGAIN

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S11.E18

IF [IsNotEmpty(TagOf(*Application Transaction Counter*))
AND IsNotEmpty(TagOf(*Cryptogram Information Data*))]

THEN

GOTO S11.E20

ELSE

GOTO S11.E19

ENDIF

S11.E19

'L2' in *Error Indication* := CARD DATA MISSING

S11.E20

IF [((*Cryptogram Information Data* AND 'C0' = '40')
AND
('AC type' in *Reference Control Parameter* = TC))

OR

((*Cryptogram Information Data* AND 'C0' = '80')
AND
(('AC type' in *Reference Control Parameter* = TC)
OR
('AC type' in *Reference Control Parameter* = ARQC)))

OR

(*Cryptogram Information Data* AND 'C0'= '00')]

THEN

GOTO S11.E22

ELSE

GOTO S11.E21

ENDIF

S11.E21

'L2' in *Error Indication* := CARD DATA ERROR

S11.E22

Perform Post-GenAC Balance Reading as specified in section 7.3

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 319

S11.ED23

IF [IsNotEmpty(*Tags To Write After Gen AC*)]
THEN
GOTO S11.E25
ELSE
GOTO S11.E24
ENDIF

S11.E24

'Message Identifier' in *User Interface Request Data* := CLEAR DISPLAY
'Status' in *User Interface Request Data* := CARD READ SUCCESSFULLY
'Hold Time' in *User Interface Request Data* := '000000'
Send MSG(*User Interface Request Data*) signal

S11.E25

IF [IsNotEmpty(TagOf(*Signed Dynamic Application Data*))]
THEN
GOTO S11.E40
ELSE
GOTO S11.E70
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 320

CDA

S11.E40

Retrieve With the *CA Public Key Index (Card)* the Certification Authority Public Key Modulus and Exponent and associated key related information, and the corresponding algorithm to be used from the CA Public Key Database (see section 4.5.2).

Retrieve the Issuer Public Key and ICC Public Key as described in section 6.3 and 6.4 of [EMV Book 2].

Check if the concatenation of the *CA Public Key Index (Card)* and the Certificate Serial Number recovered from the *Issuer Public Key Certificate* appears in the CRL. If this is the case, then ICC Public Key retrieval is not successful.

IF [ICC Public Key retrieval was successful]

THEN

GOTO S11.ED41 or S11.E43

ELSE

GOTO S11.E46

ENDIF

S11.ED41

IF ['Read' in *IDS Status* is set]

THEN

GOTO S11.ED42

ELSE

GOTO S11.E43

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 321

S11.ED42

Verify *Signed Dynamic Application Data* as in section 6.6 of [EMV Book 2]. Retrieve from the ICC Dynamic Data (see Table 6.7) the *ICC Dynamic Number*, *Application Cryptogram*, *DS Summary 2* and *DS Summary 3* and store in the TLV Database.

If the ICC Dynamic Data does not include *DS Summary 3* (i.e. there are less than 16 bytes after Hash Result (if 'Data Storage Version Number' in *Application Capabilities Information* = VERSION 1) or less than 32 bytes (if 'Data Storage Version Number' in *Application Capabilities Information* = VERSION 2)), then do not store *DS Summary 3*. This is not a reason to fail CDA.

If the ICC Dynamic Data does also not include *DS Summary 2* (i.e. there are less than 8 bytes after Hash Result (if 'Data Storage Version Number' in *Application Capabilities Information* = VERSION 1) or less than 16 bytes (if 'Data Storage Version Number' in *Application Capabilities Information* = VERSION 2)), then do not store *DS Summary 2*. This is not a reason to fail CDA.

Table 6.7—ICC Dynamic Data (IDS)

Value	Length
Length of *ICC Dynamic Number*	1
ICC Dynamic Number	2-8
Cryptogram Information Data	1
Application Cryptogram	8
Hash Result	20
DS Summary 2	8 or 16
DS Summary 3	8 or 16

S11.E43

Verify *Signed Dynamic Application Data* as in section 6.6 of [EMV Book 2]. Retrieve from the ICC Dynamic Data (see Table 6.8) the *ICC Dynamic Number* and *Application Cryptogram* and store in the TLV Database.

Table 6.8—ICC Dynamic Data (No IDS)

Value	Length
Length of *ICC Dynamic Number*	1
ICC Dynamic Number	2-8
Cryptogram Information Data	1
Application Cryptogram	8
Hash Result	20

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 322

S11.E44

IF [*Signed Dynamic Application Data* verification is OK]
THEN
GOTO S11.E110
ELSE
GOTO S11.E46
ENDIF

S11.ED45

IF [*Signed Dynamic Application Data* verification is OK]
THEN
GOTO S11.ED47
ELSE
GOTO S11.E46
ENDIF

S11.E46

'L2' in *Error Indication* := CAM FAILED

S11.ED47

IF ['Write' in *IDS Status* in *Torn Temp Record* is set]
THEN
GOTO S11.ED48
ELSE
GOTO S11.ED50
ENDIF

S11.ED48

IF [*DS Summary 1 = DS Summary 1* in *Torn Temp Record*]
THEN
GOTO S11.ED50
ELSE
GOTO S11.ED49
ENDIF

S11.ED49

'L2' in *Error Indication* := IDS READ ERROR

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 323

S11.ED50

IF [IsPresent(TagOf(*DS Summary 2*))]
THEN
GOTO S11.ED52
ELSE
GOTO S11.ED51
ENDIF

S11.ED51

'L2' in *Error Indication* := CARD DATA MISSING

S11.ED52

IF [*DS Summary 1 = DS Summary 2*]
THEN
GOTO S11.ED54
ELSE
GOTO S11.ED53
ENDIF

S11.ED53

'L2' in *Error Indication* := IDS READ ERROR

S11.ED54

SET 'Successful Read' in *DS Summary Status*

S11.ED55

IF ['Write' in *IDS Status* is set]
THEN
GOTO S11.ED56
ELSE
GOTO S11.E110
ENDIF

S11.ED56

IF [IsPresent(TagOf(*DS Summary 3*))]
THEN
GOTO S11.ED58
ELSE
GOTO S11.ED57
ENDIF

S11.ED57

'L2' in *Error Indication* := CARD DATA MISSING

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 324

S11.ED58

IF [*DS Summary 2 = DS Summary 3*]

THEN

GOTO S11.ED60

ELSE

GOTO S11.ED59

ENDIF

S11.ED59

SET 'Successful Write' in *DS Summary Status*

S11.ED60

IF ['Stop if write failed' in *DS ODS Info For Reader* is set]

THEN

GOTO S11.ED61

ELSE

GOTO S11.E110

ENDIF

S11.ED61

'L2' in *Error Indication* := IDS WRITE ERROR

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 325

No CDA

S11.E70

IF [IsNotEmpty(TagOf(*Application Cryptogram*))]
THEN
GOTO S11.E72
ELSE
GOTO S11.E71
ENDIF

S11.E71

'L2' in *Error Indication* := CARD DATA MISSING

S11.E72

IF [(*Cryptogram Information Data* AND 'C0') = '00']
THEN
GOTO S11.ED73 or S11.E110
ELSE
GOTO S11.E74
ENDIF

S11.ED73

IF ['Read' in *IDS Status* is set]
THEN
GOTO S11.E77
ELSE
GOTO S11.E75
ENDIF

S11.E74

IF ['CDA signature requested' in *Reference Control Parameter* is set]
THEN
GOTO S11.E77
ELSE
GOTO S11.E110
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 326

S11.E75

IF ['AC type' in *Reference Control Parameter* = AAC]
THEN
GOTO S11.E76
ELSE
GOTO S11.E110
ENDIF

S11.E76

IF ['CDA signature requested' in *Reference Control Parameter* is set]
THEN
GOTO S11.E77
ELSE
GOTO S11.E110
ENDIF

S11.E77

'L2' in *Error Indication* := CARD DATA ERROR

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 327

Invalid Response – 1

S11.ED90

IF ['Write' in *IDS Status* in *Torn Temp Record* is set]
THEN
GOTO S11.ED91
ELSE
GOTO S11.ED92
ENDIF

S11.ED91

Torn Record := *Torn Temp Record*

S11.ED92

IF ['Write' in *IDS Status* is set]
THEN
GOTO S11.ED93
ELSE
GOTO S11.E95
ENDIF

S11.ED93

'Message Identifier' in *User Interface Request Data* := ERROR - OTHER CARD
'Status' in *User Interface Request Data* := NOT READY
Send MSG(*User Interface Request Data*) signal

S11.ED94

'Status' in *Outcome Parameter Set* := END APPLICATION
'Msg On Error' in *Error Indication* := ERROR - OTHER CARD
SET 'Data Record Present' in *Outcome Parameter Set*
CreateEMVDataRecord ()
CreateEMVDiscretionaryData ()
Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

S11.E95

'Status' in *Outcome Parameter Set* := SELECT NEXT
'Start' in *Outcome Parameter Set* := C
CreateEMVDiscretionaryData ()
Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 328

Invalid Response – 2

S11.ED100

IF ['Write' in *IDS Status* is set]

THEN

GOTO S11.ED101

ELSE

GOTO S11.E103

ENDIF

S11.ED101

'Message Identifier' in *User Interface Request Data* := ERROR - OTHER CARD

'Status' in *User Interface Request Data* := NOT READY

Send MSG(*User Interface Request Data*) signal

S11.ED102

'Status' in *Outcome Parameter Set* := END APPLICATION

'Msg On Error' in *Error Indication* := ERROR - OTHER CARD

SET 'Data Record Present' in *Outcome Parameter Set*

CreateEMVDataRecord ()

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

S11.E103

'Status' in *Outcome Parameter Set* := SELECT NEXT

'Start' in *Outcome Parameter Set* := C

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 329

Valid Response

S11.E110

SET 'Data Record Present' in *Outcome Parameter Set*

CreateEMVDataRecord ()

S11.E111

```
IF      [IsNotEmpty(TagOf(POS Cardholder Interaction Information))]
THEN
        GOTO S11.E112
ELSE
        GOTO S11.E114
ENDIF
```

S11.E112

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

S11.E113

```
FOR every entry in the Phone Message Table (see section 4.5.4)
{
        IF      [(PCII MASK[ID] AND POS Cardholder Interaction Information) =
                  PCII VALUE[ID]]
        THEN
                'Message Identifier' in User Interface Request Data := MESSAGE[ID]
                'Status' in User Interface Request Data := STATUS[ID]
                'Hold Time' in User Interface Request Data := Message Hold Time
                EXIT loop
        ENDIF
}
```

S11.E114

```
IF      [(Cryptogram Information Data AND 'C0') = '40']
THEN
        'Status' in Outcome Parameter Set := APPROVED
ELSE
        IF      [(Cryptogram Information Data AND 'C0') = '80']
        THEN
                'Status' in Outcome Parameter Set := ONLINE REQUEST
        ELSE
                'Status' in Outcome Parameter Set := DECLINED
        ENDIF
ENDIF
```


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 330

S11.E115

'Status' in *User Interface Request Data* := CARD READ SUCCESSFULLY
IF [(*Cryptogram Information Data* AND 'C0') = '40']
THEN
'Hold Time' in *User Interface Request Data* := *Message Hold Time*
IF [IsNotEmpty(*Balance Read After Gen AC*)]
THEN
'Value Qualifier' in *User Interface Request Data* := BALANCE
'Value' in *User Interface Request Data* := *Balance Read After Gen AC*
IF [IsNotEmpty(TagOf(*Application Currency Code*))]
THEN
'Currency Code' in *User Interface Request Data* := *Application Currency Code*
ENDIF
ENDIF
IF ['CVM' in *Outcome Parameter Set* = OBTAIN SIGNATURE]
THEN
'Message Identifier' in *User Interface Request Data* := APPROVED - SIGN
ELSE
'Message Identifier' in *User Interface Request Data* := APPROVED
ENDIF
ELSE
IF [(*Cryptogram Information Data* AND 'C0') = '80']
THEN
'Hold Time' in *User Interface Request Data* :='000000'
'Message Identifier' in *User Interface Request Data* := AUTHORISING – PLEASE WAIT
ELSE
'Hold Time' in *User Interface Request Data* := *Message Hold Time*
'Message Identifier' in *User Interface Request Data* := DECLINED
ENDIF
ENDIF

S11.ED116

IF [IsNotEmptyList(*Tags To Write After Gen AC*)]
THEN
GOTO S11.ED117
ELSE
GOTO S11.E119
ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 331

S11.ED117

TLV = GetAndRemoveFromList(*Tags To Write Yet After Gen AC*)

Prepare the PUT DATA command with TLV as defined in section 5.6

S11.ED118

Send CA(PUT DATA command) signal

S11.E119

Send MSG(*User Interface Request Data*) signal

S11.E120

CreateEMVDiscretionaryData ()

IF [IsNotEmpty(TagOf(*POS Cardholder Interaction Information*))]

THEN

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

ELSE

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 332

6.18 State 12 – Waiting for Put Data Response Before Generate AC

6.18.1 Local Variables

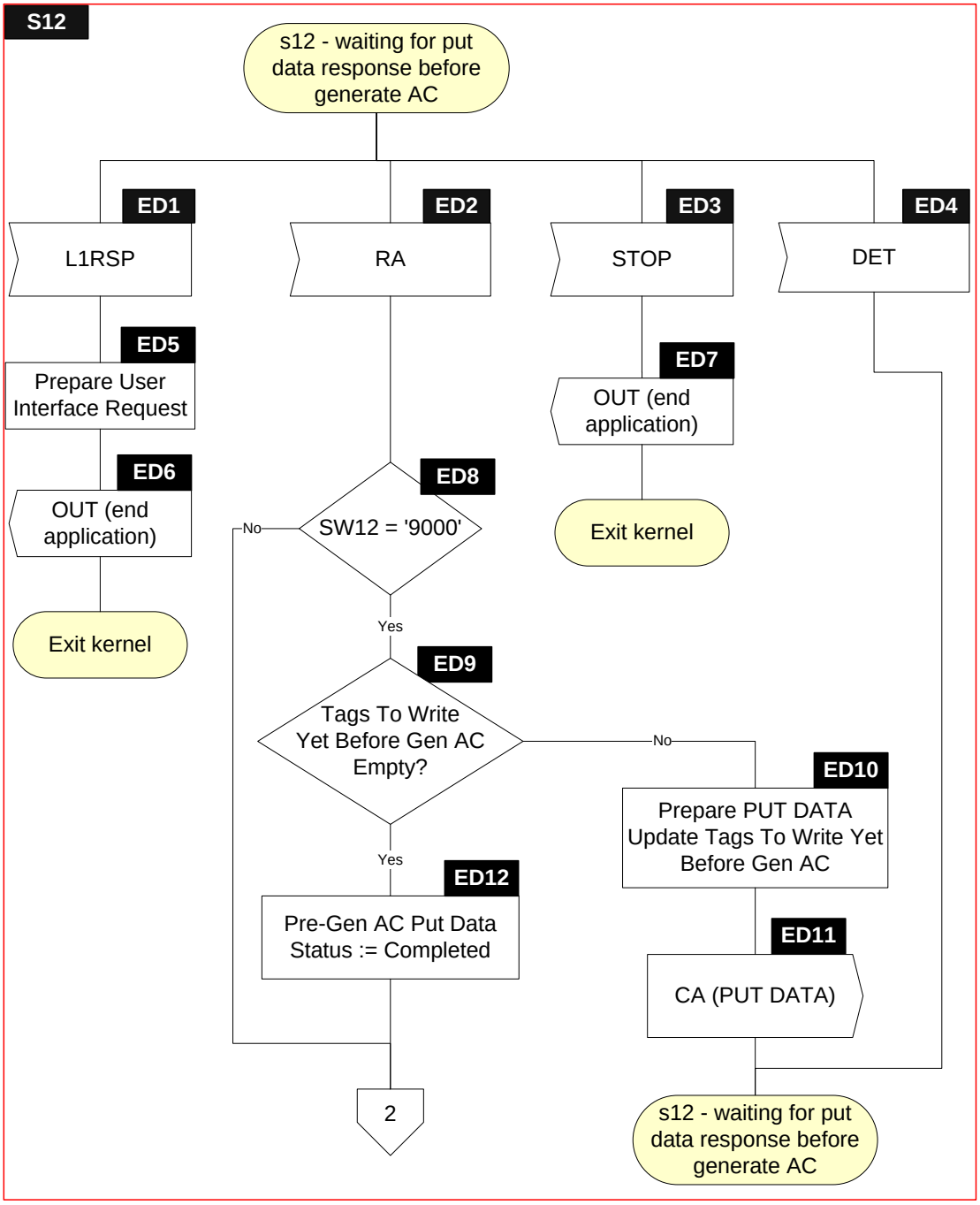
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
SW12	2	b	Status bytes
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 252	b	Value of TLV encoded string

6.18.2 Flow Diagram

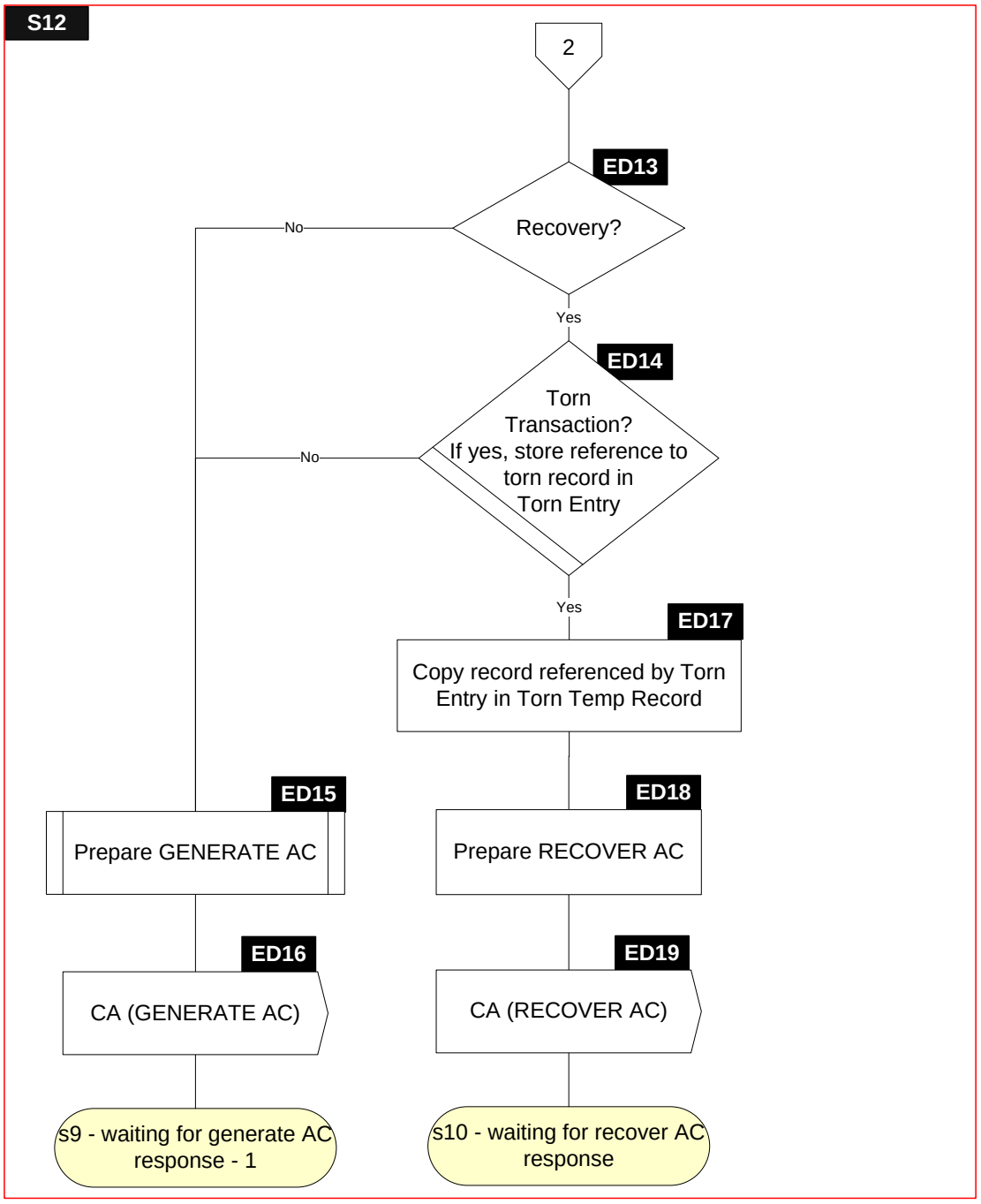
Figure 6.17 shows the flow diagram of `s12 – waiting for put data response before generate AC`. Symbols in this diagram are labeled S12.X.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 333

Figure 6.17—State 12 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 334



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 335

6.18.3 Processing

S12.ED1

Receive L1RSP signal with Return Code

S12.ED2

Receive RA signal with SW12

S12.ED3

Receive STOP signal

S12.ED4

Receive DET signal

S12.ED5

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S12.ED6

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S12.ED7

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S12.ED8

IF [SW12 = '9000']

THEN

GOTO S12.ED9

ELSE

GOTO S12.ED13

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 336

S12.ED9

IF [IsEmptyList(*Tags To Write Yet Before Gen AC*)]
THEN
GOTO S12.ED12
ELSE
GOTO S12.ED10
ENDIF

S12.ED10

TLV := GetAndRemoveFromList(*Tags To Write Yet Before Gen AC*)
Prepare PUT DATA command for TLV as specified in section 5.6

S12.ED11

Send CA(PUT DATA command) signal

S12.ED12

SET 'Completed' in *Pre-Gen AC Put Data Status*

S12.ED13

IF [IsPresent(TagOf(*DRDOL*)) AND
Max Number of Torn Transaction Log Records ≠ 0]
THEN
GOTO S12.ED14
ELSE
GOTO S12.ED15
ENDIF

S12.ED14

FOR every Record in Torn Transaction Log
{
IF [*Application PAN* in Record = *Application PAN*
AND
Application PAN Sequence Number in Record = *Application PAN Sequence Number*]
THEN
Store reference to Record in *Torn Entry* for later use
GOTO S12.ED17
ENDIF
}
GOTO S12.ED15

S12.ED15

Prepare GENERATE AC command as specified in section 7.6

S12.ED16

Send CA(GENERATE AC) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 337

S12.ED17

Copy record referenced by *Torn Entry* into *Torn Temp Record*

S12.ED18

DRDOL Related Data := *DRDOL Related Data* in *Torn Temp Record*

Prepare RECOVER AC command as specified in section 5.8

S12.ED19

Send CA(RECOVER AC) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 338

6.19 State 13 – Waiting for CCC Response – 1

6.19.1 Local Variables

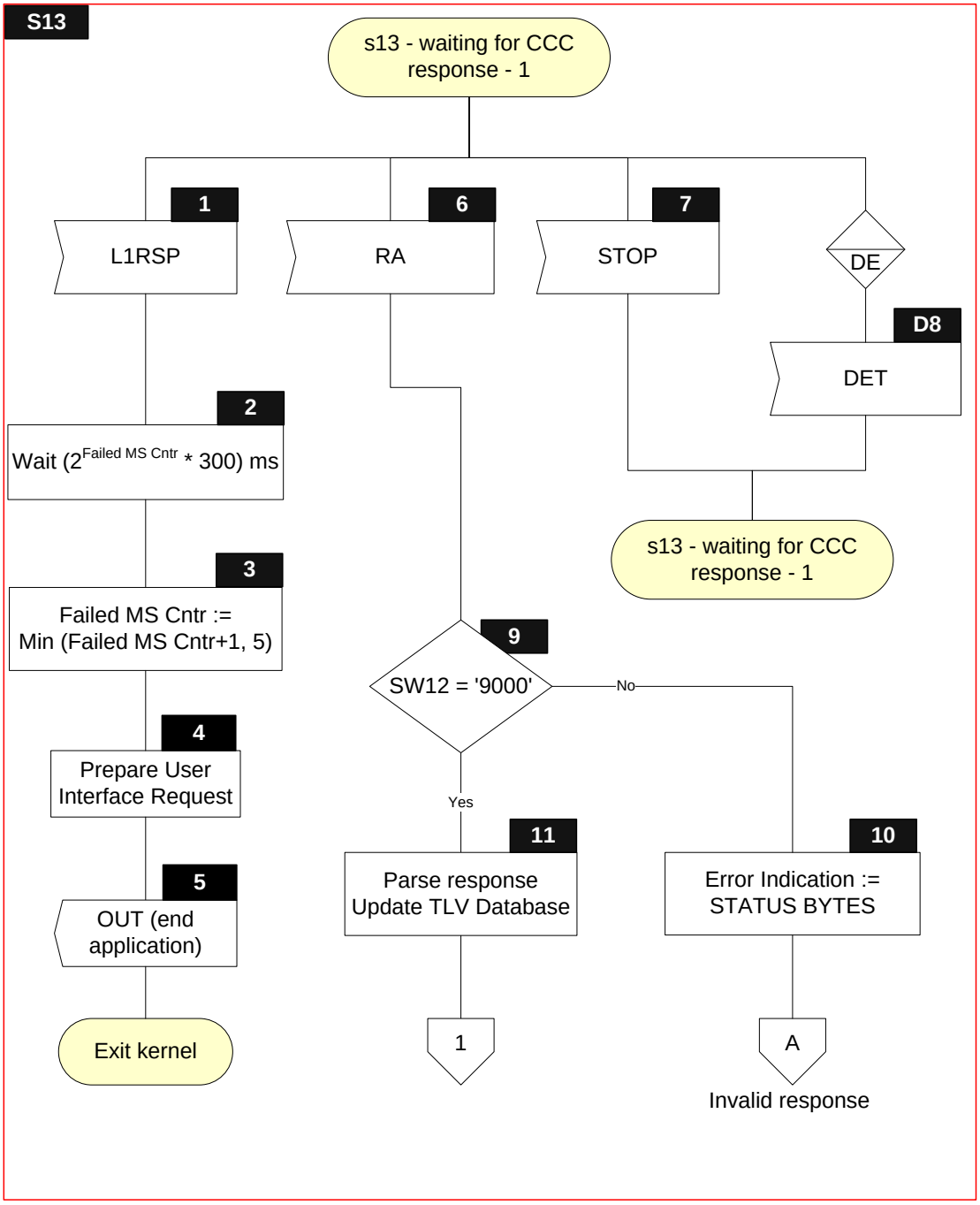
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of COMPUTE CRYPTOGRAPHIC CHECKSUM
q	1	n	Number of CVC3 digits to be copied in the 'Discretionary Data' in *Track 1 Data* and *Track 2 Data*
t	1	n	Number of ATC digits to be copied in the 'Discretionary Data' in *Track 1 Data* and *Track 2 Data*

6.19.2 Flow Diagram

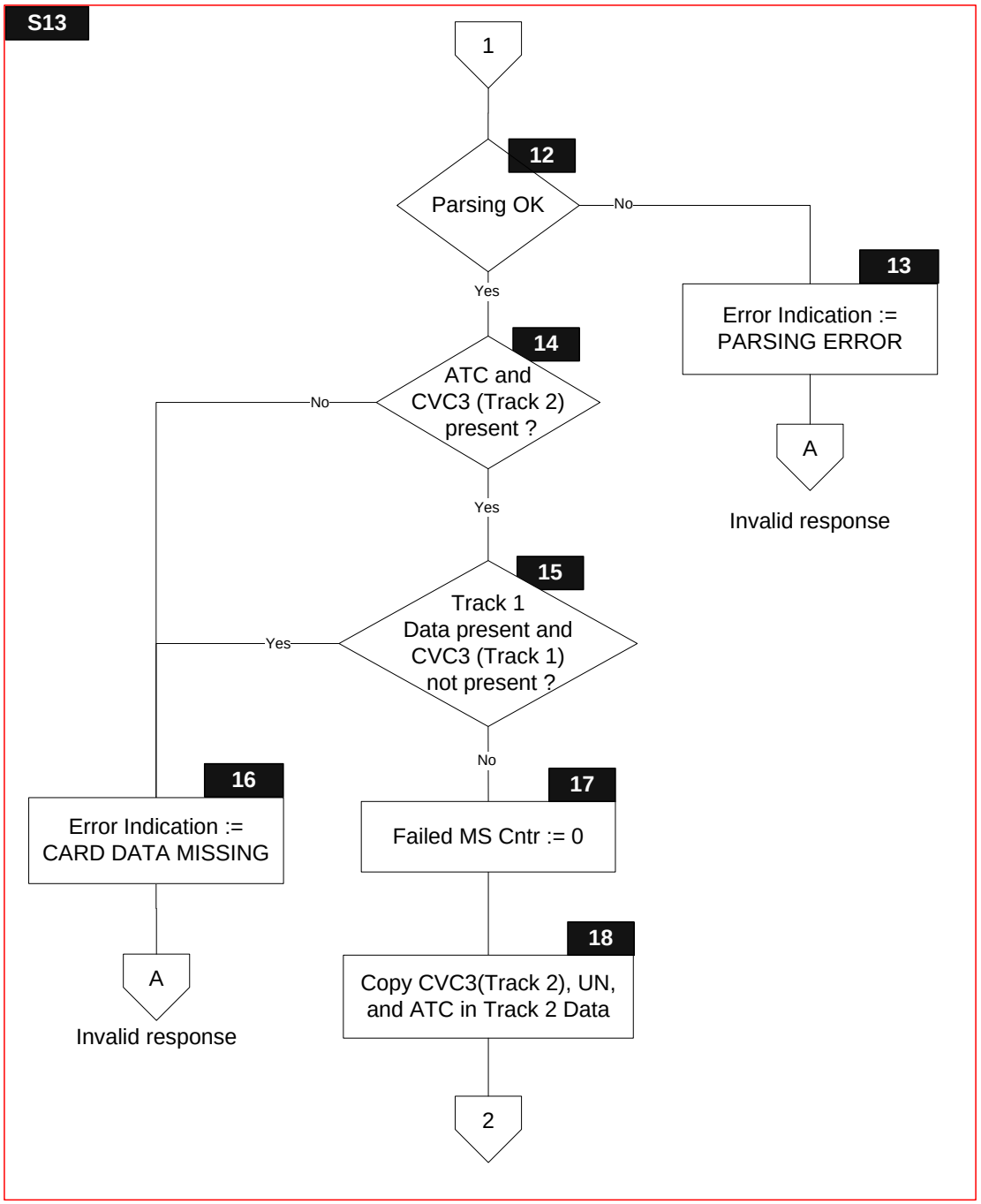
Figure 6.18 shows the flow diagram of `s13 – waiting for CCC response - 1`. Symbols in this diagram are labeled S13.X.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 339

Figure 6.18—State 13 Flow Diagram

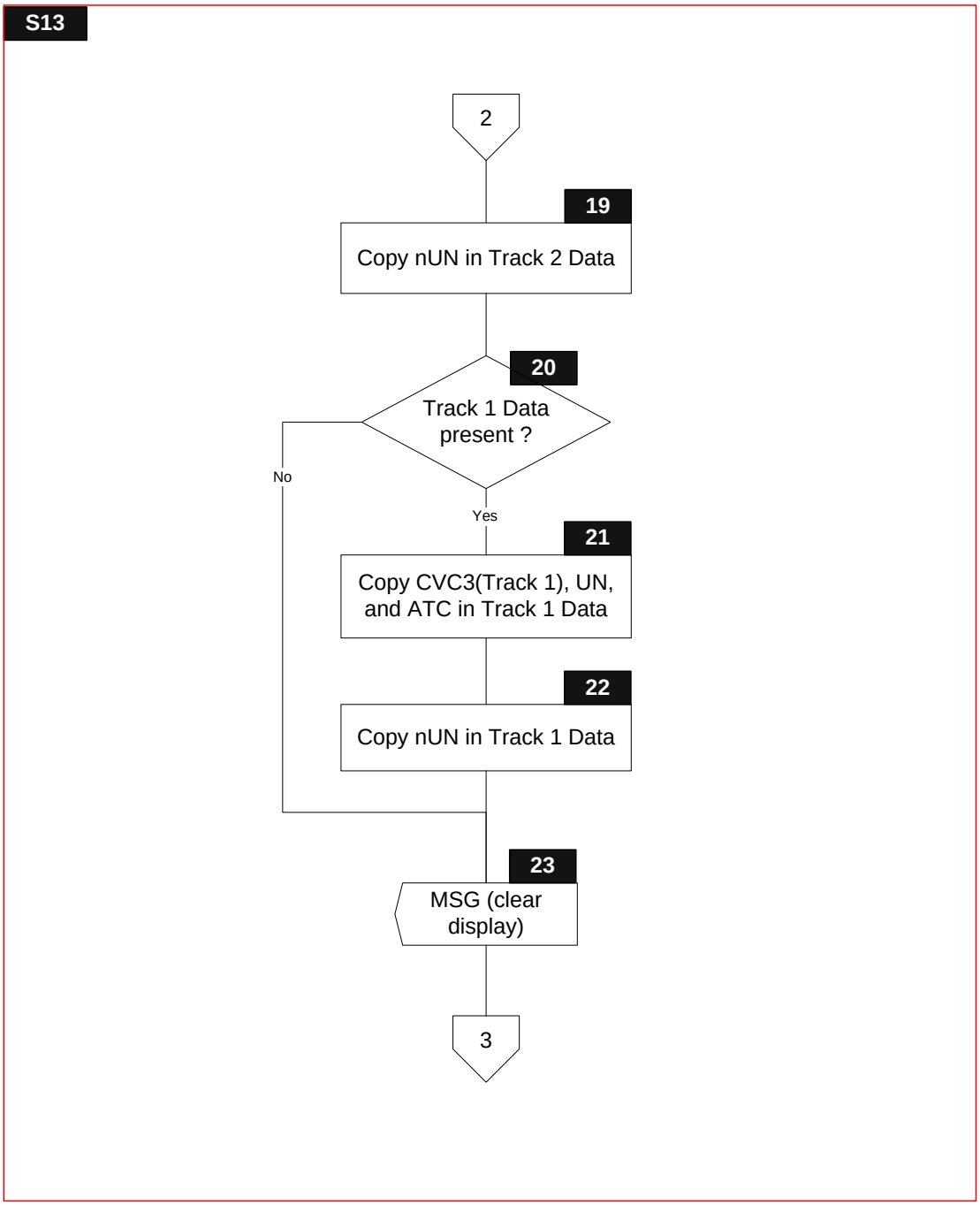


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 340

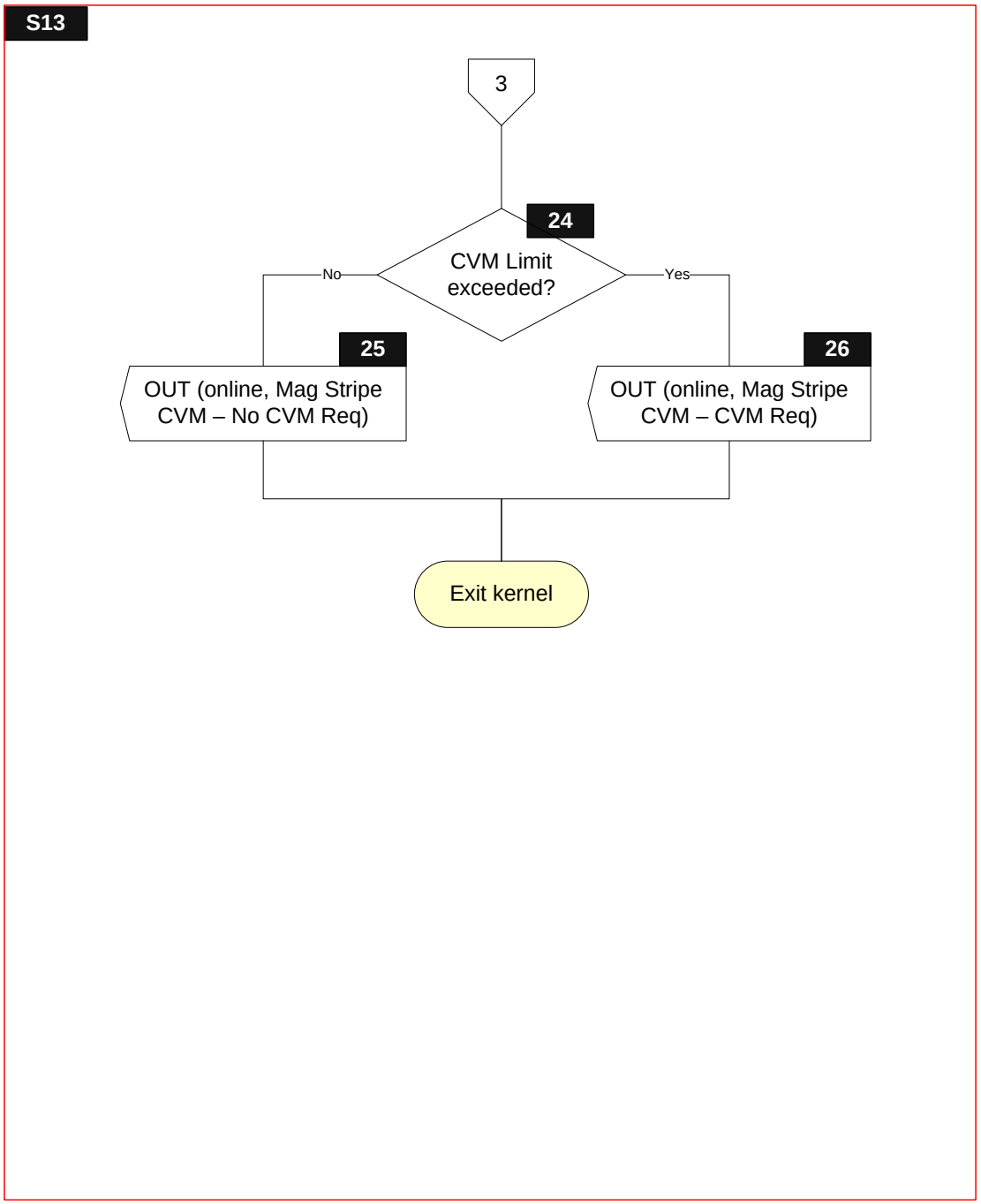


© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 341

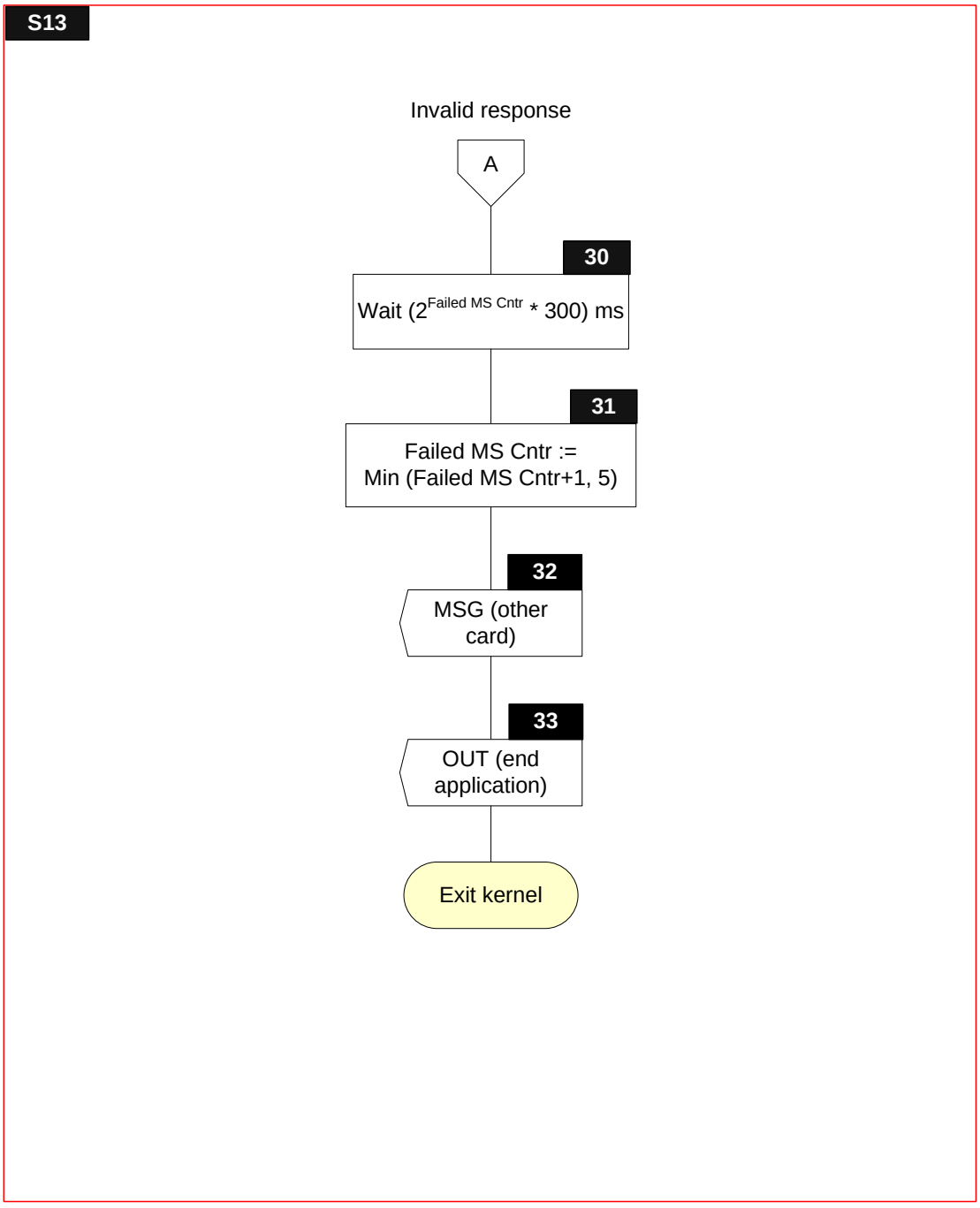


© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 342



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 343



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 344

6.19.3 Processing

S13.1

Receive L1RSP signal with Return Code

S13.2

Wait for ($2^{Failed\ MS\ Cntr} * 300$) ms

Note that *Failed MS Cntr* is stored in the scratch pad provided to the Kernel at instantiation. Dependent on the implementation, it may be that *Failed MS Cntr* does not exist the first time the Kernel is executed. In this case, *Failed MS Cntr* must be created and initialized to zero.

S13.3

Failed MS Cntr := min(*Failed MS Cntr* + 1, 5)

S13.4

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S13.5

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S13.6

Receive RA signal with Response Message Data Field and SW12

S13.7

Receive STOP signal

S13.D8

Receive DET signal

S13.9

IF [SW12 = '9000']

THEN

GOTO S13.11

ELSE

GOTO S13.10

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 345

S13.10

'L2' in *Error Indication* := STATUS BYTES

'SW12' in *Error Indication* := SW12

S13.11

IF [(Length of Response Message Data Field > 0) AND
(Response Message Data Field[1] = '77')]

THEN

Parsing Result := ParseAndStoreCardResponse(Response Message Data Field)

ELSE

Parsing Result := FALSE

ENDIF

S13.12

IF [Parsing Result]

THEN

GOTO S13.14

ELSE

GOTO S13.13

ENDIF

S13.13

'L2' in *Error Indication* := PARSING ERROR

S13.14

IF [IsNotEmpty(TagOf(*Application Transaction Counter*)) AND
IsNotEmpty(TagOf(*CVC3 (Track2)*))]

THEN

GOTO S13.15

ELSE

GOTO S13.16

ENDIF

S13.15

IF [IsNotEmpty(TagOf(*Track 1 Data*)) AND
(IsNotPresent(TagOf(*CVC3 (Track1)*)) OR
IsEmpty(TagOf(*CVC3 (Track1)*)))]

THEN

GOTO S13.16

ELSE

GOTO S13.17

ENDIF

S13.16

'L2' in *Error Indication* := CARD DATA MISSING

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 346

S13.17

Failed MS Cntr := 0

S13.18

q := Number of non-zero bits in *PCVC3(Track2)*

t := *NATC(Track2)*

Convert the binary encoded *CVC3 (Track2)* to the BCD encoding of the corresponding number expressed in base 10. Copy the q least significant digits of the BCD encoded *CVC3 (Track2)* in the eligible positions of the 'Discretionary Data' in *Track 2 Data*. The eligible positions are indicated by the q non-zero bits in *PCVC3(Track2)*.

Replace the *nUN* least significant eligible positions of the 'Discretionary Data' in *Track 2 Data* by the *nUN* least significant digits of *Unpredictable Number (Numeric)*. The eligible positions in the 'Discretionary Data' in *Track 2 Data* are indicated by the *nUN* least significant non-zero bits in *PUNATC(Track2)*.

If $t \neq 0$, convert the *Application Transaction Counter* to the BCD encoding of the corresponding number expressed in base 10. Replace the t most significant eligible positions of the 'Discretionary Data' in *Track 2 Data* by the t least significant digits of the BCD encoded *Application Transaction Counter*. The eligible positions in the 'Discretionary Data' in *Track 2 Data* are indicated by the t most significant non-zero bits in *PUNATC(Track2)*.

S13.19

Copy *nUN* into the least significant digit of the 'Discretionary Data' in *Track 2 Data*

S13.20

IF [IsNotEmpty(TagOf(*Track 1 Data*))]
THEN
    GOTO S13.21
ELSE
    GOTO S13.23
ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 347

S13.21

q := Number of non-zero bits in *PCVC3(Track1)*

t := *NATC(Track1)*

Convert the binary encoded *CVC3 (Track1)* to the BCD encoding of the corresponding number expressed in base 10. Convert the q least significant digits of the BCD encoded *CVC3 (Track1)* into ASCII format and copy the q ASCII encoded *CVC3 (Track1)* characters into the eligible positions of the 'Discretionary Data' in *Track 1 Data*. The eligible positions are indicated by the q non-zero bits in *PCVC3(Track1).*

Convert the BCD encoded *Unpredictable Number (Numeric)* into ASCII format and replace the *nUN* least significant eligible positions of the 'Discretionary Data' in *Track 1 Data* by the *nUN* least significant characters of the ASCII encoded *Unpredictable Number (Numeric)*. The eligible positions in the 'Discretionary Data' in *Track 1 Data* are indicated by the *nUN* least significant non-zero bits in *PUNATC(Track1).*

If t ≠ 0, convert the *Application Transaction Counter* to the BCD encoding of the corresponding number expressed in base 10. Convert the t least significant digits of the BCD encoded *Application Transaction Counter* into ASCII format. Replace the t most significant eligible positions of the 'Discretionary Data' in *Track 1 Data* by the t ASCII encoded *Application Transaction Counter* characters. The eligible positions in the 'Discretionary Data' in *Track 1 Data* are indicated by the t most significant non-zero bits in *PUNATC(Track1).*

S13.22

Convert *nUN* into the ASCII format

Copy the ASCII encoded *nUN* character into the least significant position of the 'Discretionary Data' in *Track 1 Data*

S13.23

'Message Identifier' in *User Interface Request Data* := CLEAR DISPLAY

'Status' in *User Interface Request Data* := CARD READ SUCCESSFULLY

'Hold Time' in *User Interface Request Data* := '000000'

Send MSG(*User Interface Request Data*) signal

S13.24

IF [*Amount, Authorized (Numeric) > Reader CVM Required Limit*]

THEN

GOTO S13.26

ELSE

GOTO S13.25

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 348

S13.25

'Status' in *Outcome Parameter Set* := ONLINE REQUEST

'CVM' in *Outcome Parameter Set* := 'CVM' in *Mag-stripe CVM Capability – No CVM Required*

IF ['CVM' in *Mag-stripe CVM Capability – No CVM Required* = OBTAIN SIGNATURE]

THEN

'Receipt' in *Outcome Parameter Set* := YES

ENDIF

SET 'Data Record Present' in *Outcome Parameter Set*

CreateMSDataRecord ()

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

S13.26

'Status' in *Outcome Parameter Set* := ONLINE REQUEST

'CVM' in *Outcome Parameter Set* := 'CVM' in *Mag-stripe CVM Capability – CVM Required*

'Receipt' in *Outcome Parameter Set* := YES

SET 'Data Record Present' in *Outcome Parameter Set*

CreateMSDataRecord ()

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 349

Invalid Response

S13.30

Wait for ($2^{Failed\ MS\ Cntr} * 300$) ms

Note that *Failed MS Cntr* is stored in the scratch pad provided to the Kernel at instantiation. Dependent on the implementation, it may be that *Failed MS Cntr* does not exist the first time the Kernel is executed. In this case, *Failed MS Cntr* must be created and initialized to zero.

S13.31

Failed MS Cntr := min(*Failed MS Cntr* + 1, 5)

S13.32

'Message Identifier' in *User Interface Request Data* := ERROR - OTHER CARD

'Status' in *User Interface Request Data* := NOT READY

Send MSG(*User Interface Request Data*) signal

S13.33

'Status' in *Outcome Parameter Set* := END APPLICATION

'Msg On Error' in *Error Indication*:= 'Message Identifier' in *User Interface Request Data*

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 350

6.20 State 14 – Waiting for CCC Response - 2

6.20.1 Local Variables

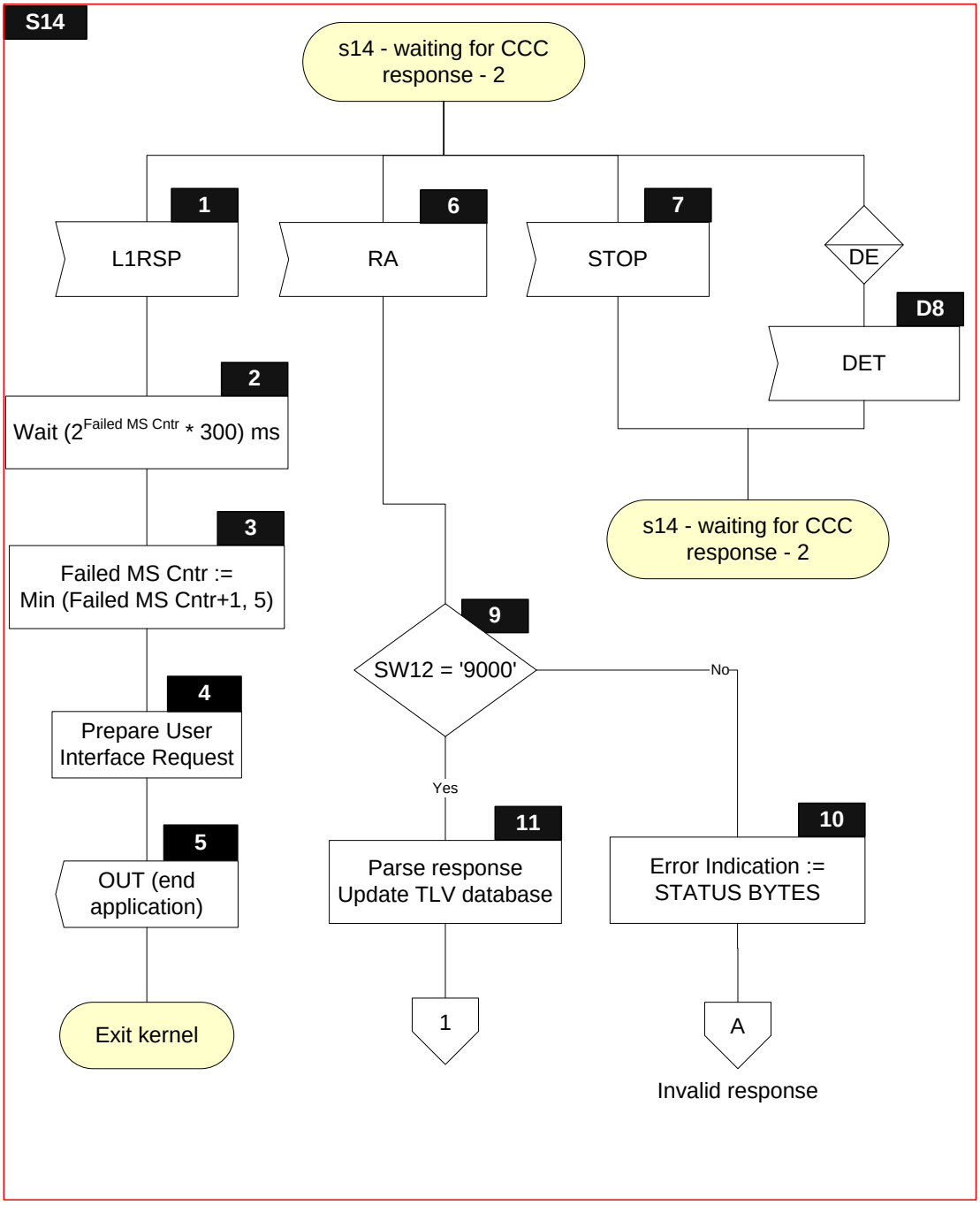
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
Parsing Result	1	b	Boolean used to store result of parsing a TLV string
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of COMPUTE CRYPTOGRAPHIC CHECKSUM
nUN'	1	n	nUN' is used to store the value to be copied in the last digit of the 'Discretionary Data' in *Track 1 Data* and *Track 2 Data*
q	1	n	Number of CVC3 digits to be copied in the 'Discretionary Data' in *Track 1 Data* and *Track 2 Data*
t	1	n	Number of ATC digits to be copied in the 'Discretionary Data' in *Track 1 Data* and *Track 2 Data*

6.20.2 Flow Diagram

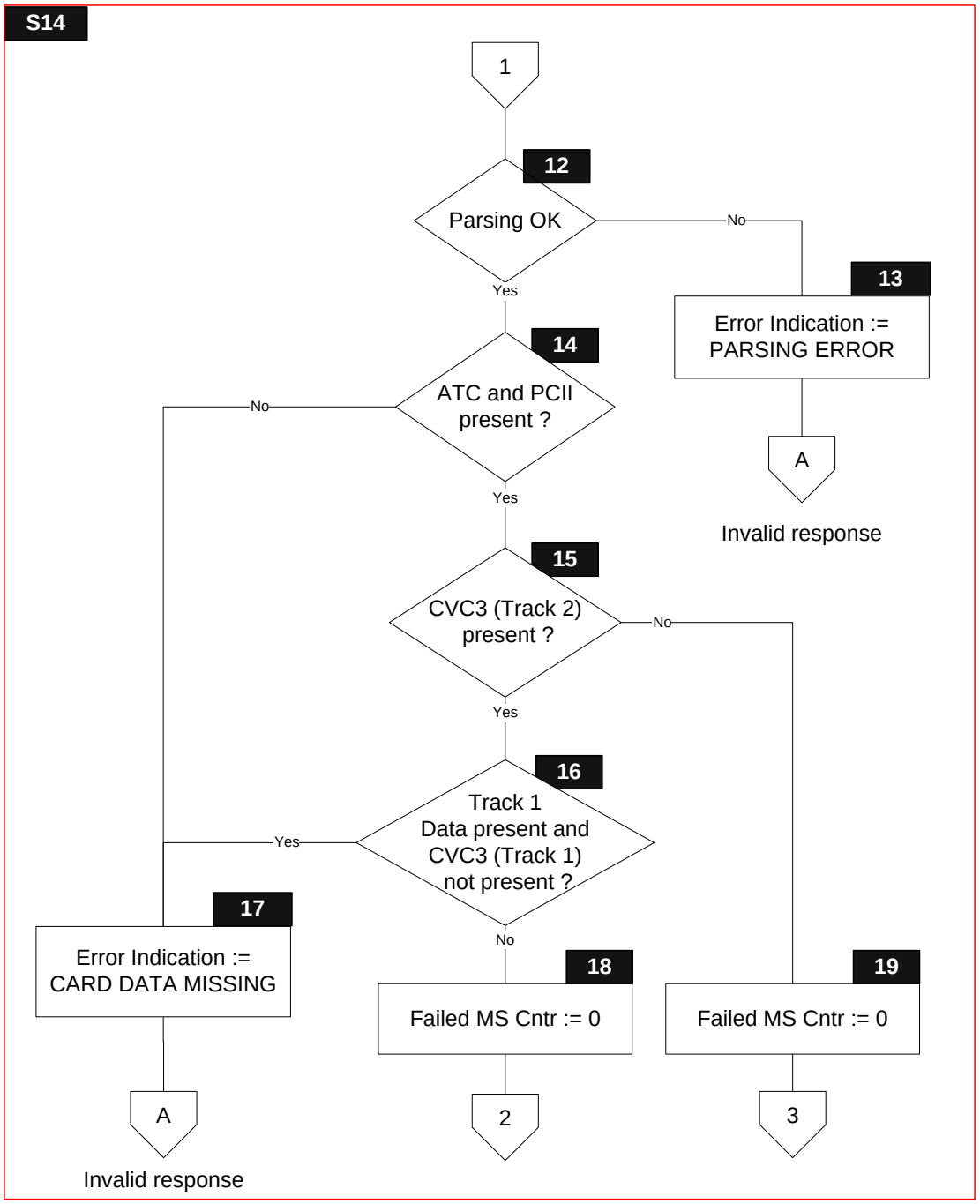
Figure 6.19 shows the flow diagram of `s14 – waiting for CCC response - 2`. Symbols in this diagram are labeled S14.X.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 351

Figure 6.19—State 14 Flow Diagram

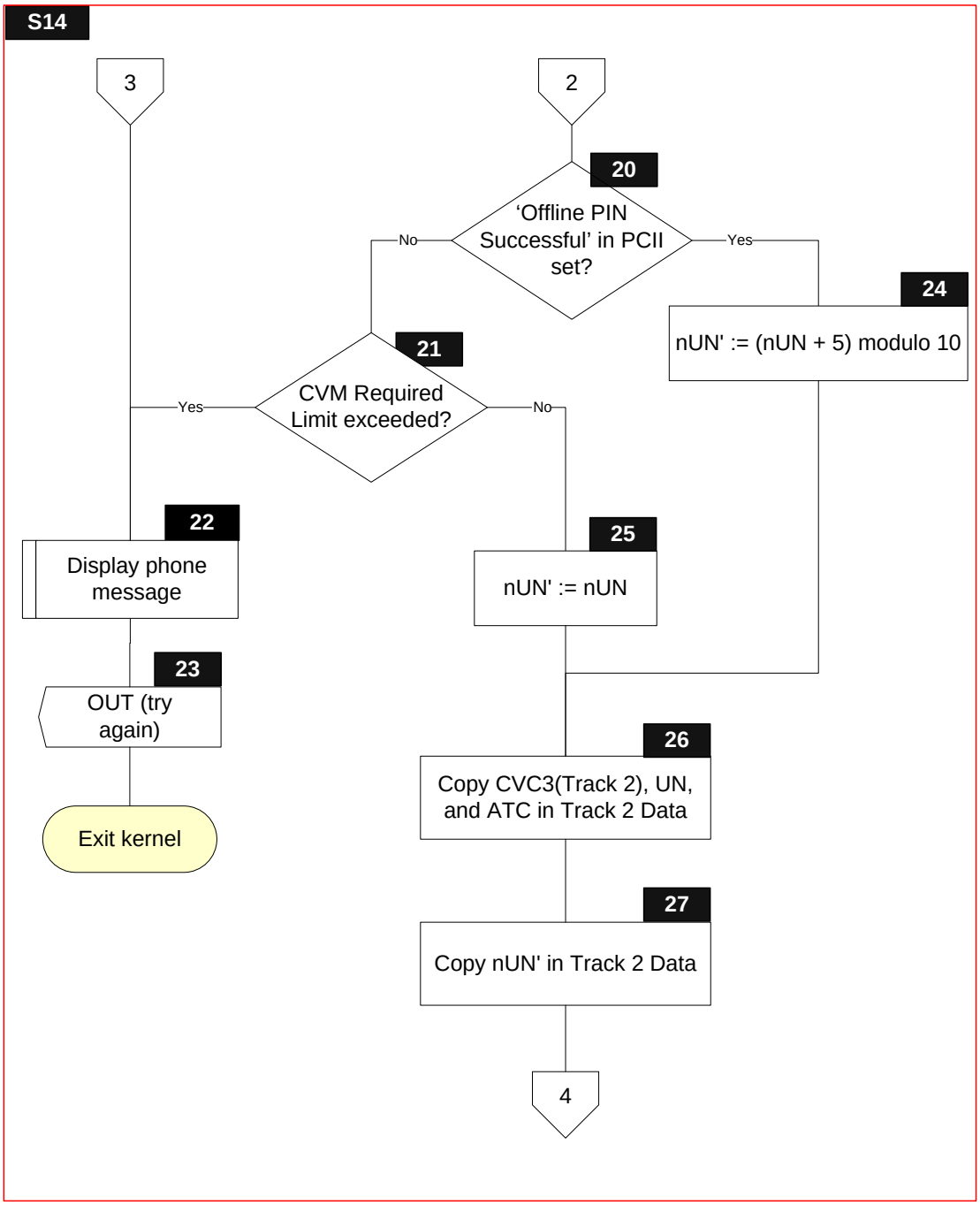


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 352

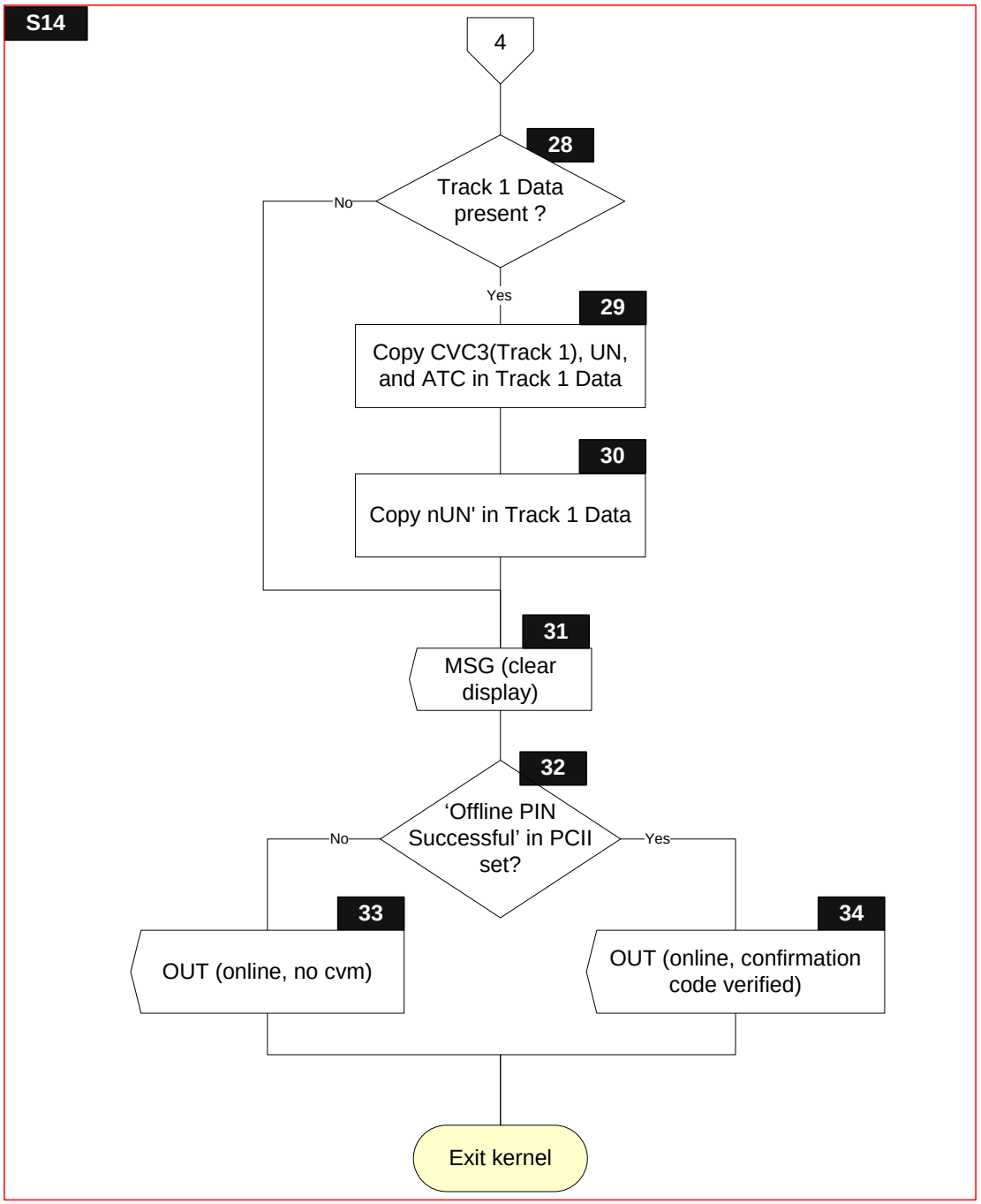


© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

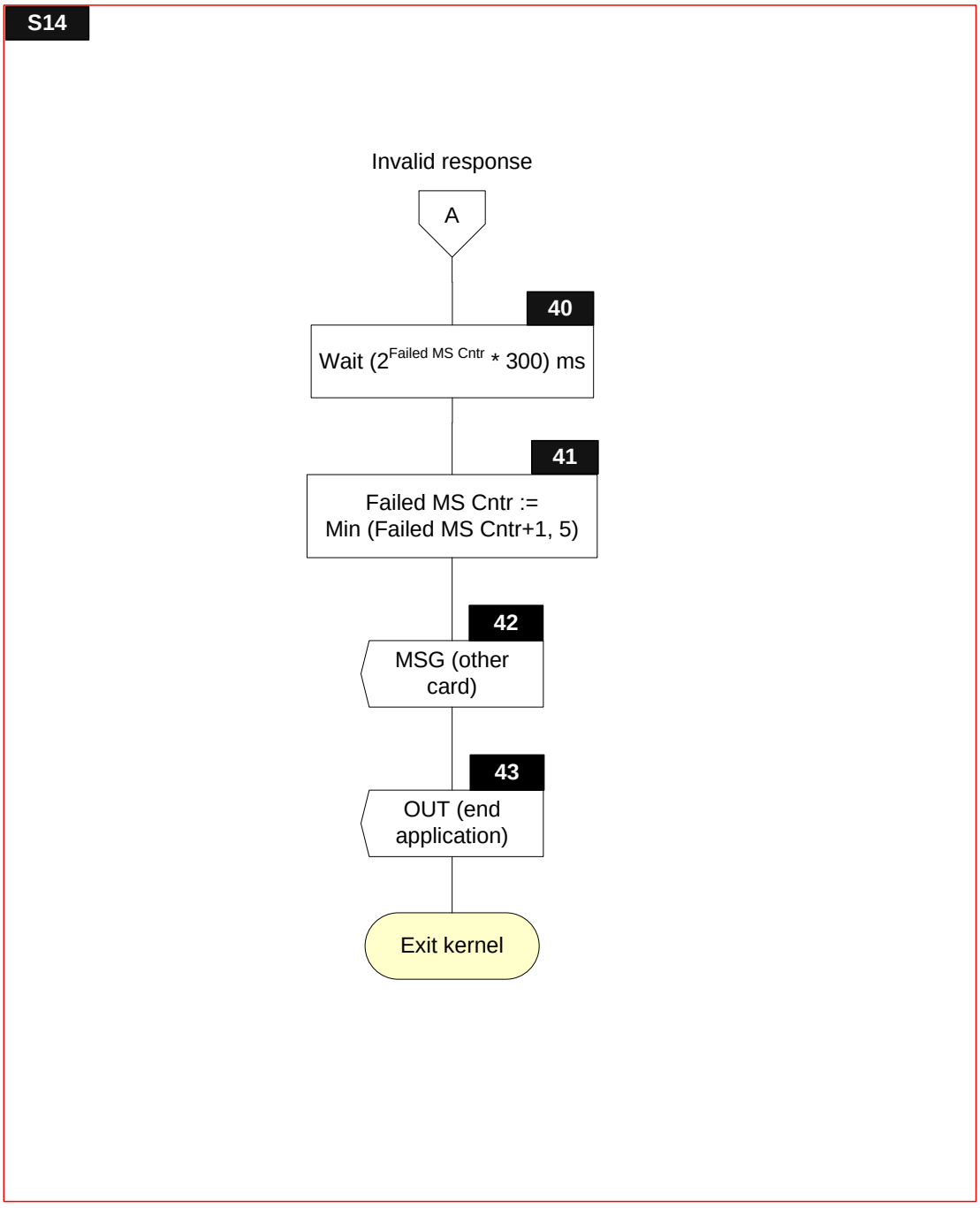
Mastercard, Ex1026, p. 353



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 354



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 355



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 356

6.20.3 Processing

S14.1

Receive L1RSP signal with Return Code

S14.2

Wait for ($2^{Failed\ MS\ Cntr} * 300$) ms

Note that *Failed MS Cntr* is stored in the scratch pad provided to the Kernel at instantiation. Dependent on the implementation, it may be that *Failed MS Cntr* does not exist the first time the Kernel is executed. In this case, *Failed MS Cntr* must be created and initialized to zero.

S14.3

Failed MS Cntr := min(*Failed MS Cntr* + 1, 5)

S14.4

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S14.5

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)) , GetTLV(TagOf(*User Interface Request Data*))) signal

S14.6

Receive RA signal with Response Message Data Field and SW12

S14.7

Receive STOP signal

S14.D8

Receive DET signal

S14.9

IF [SW12 = '9000']

THEN

GOTO S14.11

ELSE

GOTO S14.10

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 357

S14.10

'L2' in *Error Indication* := STATUS BYTES

'SW12' in *Error Indication* := SW12

S14.11

IF [(Length of Response Message Data Field > 0) AND
(Response Message Data Field[1] = '77')]

THEN

Parsing Result := ParseAndStoreCardResponse(Response Message Data Field)

ELSE

Parsing Result := FALSE

ENDIF

S14.12

IF [Parsing Result]

THEN

GOTO S14.14

ELSE

GOTO S14.13

ENDIF

S14.13

'L2' in *Error Indication* := PARSING ERROR

S14.14

IF [IsNotEmpty(TagOf(*Application Transaction Counter*)) AND
IsNotEmpty(TagOf(*POS Cardholder Interaction Information*))]

THEN

GOTO S14.15

ELSE

GOTO S14.17

ENDIF

S14.15

IF [IsNotEmpty(TagOf(*CVC3 (Track2)*))]

THEN

GOTO S14.16

ELSE

GOTO S14.19

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 358

S14.16

IF [IsNotEmpty(TagOf(*Track 1 Data*)) AND
(IsNotPresent(TagOf(*CVC3 (Track1)*)) OR
IsEmpty(TagOf(*CVC3 (Track1)*)))]

THEN

GOTO S14.17

ELSE

GOTO S14.18

ENDIF

S14.17

'L2' in *Error Indication* := CARD DATA MISSING

S14.18

Failed MS Cntr := 0

S14.19

Failed MS Cntr := 0

S14.20

IF ['Offline PIN verification successful' in *POS Cardholder Interaction Information* is set]

THEN

GOTO S14.24

ELSE

GOTO S14.21

ENDIF

S14.21

IF [*Amount, Authorized (Numeric) > Reader CVM Required Limit*]

THEN

GOTO S14.22

ELSE

GOTO S14.25

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 359

S14.22

FOR every entry in the Phone Message Table (See section 4.5.4)
{
IF [(PCII MASK[ID] AND *POS Cardholder Interaction Information*) = PCII VALUE[ID]]
THEN
'Message Identifier' in *User Interface Request Data* := MESSAGE[ID]
'Status' in *User Interface Request Data* := STATUS[ID]
Send MSG(*User Interface Request Data*) signal
EXIT loop
ENDIF
}

S14.23

'Hold Time' in *User Interface Request Data* := '000000'
'Status' in *User Interface Request Data* := READY TO READ
SET 'UI Request on Restart Present' in *Outcome Parameter Set*
'Status' in *Outcome Parameter Set* := END APPLICATION
'Start' in *Outcome Parameter Set* := B
SET 'Data Record Present' in *Outcome Parameter Set*
CreateMSDataRecord ()
CreateMSDiscretionaryData ()
Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*)) , GetTLV(TagOf(*User Interface Request Data*))) signal

S14.24

nUN' := (*nUN* + 5) modulo 10

S14.25

nUN' := *nUN*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 360

S14.26

q := Number of non-zero bits in *PCVC3(Track2)*

t := *NATC(Track2)*

Convert the binary encoded *CVC3 (Track2)* to the BCD encoding of the corresponding number expressed in base 10. Copy the q least significant digits of the BCD encoded *CVC3 (Track2)* in the eligible positions of the 'Discretionary Data' in *Track 2 Data*. The eligible positions are indicated by the q non-zero bits in *PCVC3(Track2)*.

Replace the *nUN* least significant eligible positions of the 'Discretionary Data' in *Track 2 Data* by the *nUN* least significant digits of *Unpredictable Number (Numeric)*. The eligible positions in the 'Discretionary Data' in *Track 2 Data* are indicated by the *nUN* least significant non-zero bits in *PUNATC(Track2)*.

If $t \neq 0$, convert the *Application Transaction Counter* to the BCD encoding of the corresponding number expressed in base 10. Replace the t most significant eligible positions of the 'Discretionary Data' in *Track 2 Data* by the t least significant digits of the BCD encoded *Application Transaction Counter*. The eligible positions in the 'Discretionary Data' in *Track 2 Data* are indicated by the t most significant non-zero bits in *PUNATC(Track2)*.

S14.27

Copy nUN' into the least significant digit of the 'Discretionary Data' in *Track 2 Data*

S14.28

IF [IsNotEmpty(TagOf(*Track 1 Data*))]

THEN

GOTO S14.29

ELSE

GOTO S14.31

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 361

S14.29

q := Number of non-zero bits in *PCVC3(Track1)*

t := *NATC(Track1)*

Convert the binary encoded *CVC3 (Track1)* to the BCD encoding of the corresponding number expressed in base 10. Convert the q least significant digits of the BCD encoded *CVC3 (Track1)* into the ASCII format and copy the q ASCII encoded *CVC3 (Track1)* characters into the eligible positions of the 'Discretionary Data' in *Track 1 Data*. The eligible positions are indicated by the q non-zero bits in *PCVC3(Track1).*

Convert the BCD encoded *Unpredictable Number (Numeric)* into the ASCII format and replace the *nUN* least significant eligible positions of the 'Discretionary Data' in *Track 1 Data* by the *nUN* least significant characters of the ASCII encoded *Unpredictable Number (Numeric)*. The eligible positions in the 'Discretionary Data' in *Track 1 Data* are indicated by the *nUN* least significant non-zero bits in *PUNATC(Track1).*

If t ≠ 0, convert the *Application Transaction Counter* to the BCD encoding of the corresponding number expressed in base 10. Convert the t least significant digits of the *Application Transaction Counter* into the ASCII format. Replace the t most significant eligible positions of the 'Discretionary Data' in *Track 1 Data* by the t ASCII encoded *Application Transaction Counter* characters. The eligible positions in the 'Discretionary Data' in *Track 1 Data* are indicated by the t most significant non-zero bits in *PUNATC(Track1).*

S14.30

Convert nUN' into the ASCII format

Copy the ASCII encoded nUN' character into the least significant position of the 'Discretionary Data' in *Track 1 Data*

S14.31

'Message Identifier' in *User Interface Request Data* := CLEAR DISPLAY

'Status' in *User Interface Request Data* := CARD READ SUCCESSFULLY

'Hold Time' in *User Interface Request Data* := '000000'

Send MSG(*User Interface Request Data*) signal

S14.32

IF ['Offline PIN verification successful' in *POS Cardholder Interaction Information* is set]

THEN

GOTO S14.34

ELSE

GOTO S14.33

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 362

S14.33

'Status' in *Outcome Parameter Set* := ONLINE REQUEST

'CVM' in *Outcome Parameter Set* := NO CVM

SET 'Data Record Present' in *Outcome Parameter Set*

CreateMSDataRecord ()

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

S14.34

'Status' in *Outcome Parameter Set* := ONLINE REQUEST

'CVM' in *Outcome Parameter Set* := CONFIRMATION CODE VERIFIED

IF [*Amount, Authorized (Numeric) > Reader CVM Required Limit*]

THEN

'Receipt' in *Outcome Parameter Set* := YES

ENDIF

SET 'Data Record Present' in *Outcome Parameter Set*

CreateMSDataRecord ()

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 363

Invalid Response

S14.40

Wait for ($2^{Failed\ MS\ Cntr} * 300$) ms

Note that *Failed MS Cntr* is stored in the scratch pad provided to the Kernel at instantiation. Dependent on the implementation, it may be that *Failed MS Cntr* does not exist the first time the Kernel is executed. In this case, *Failed MS Cntr* must be created and initialized to zero.

S14.41

Failed MS Cntr := min(*Failed MS Cntr* + 1, 5)

S14.42

'Message Identifier' in *User Interface Request Data* := ERROR - OTHER CARD

'Status' in *User Interface Request Data* := NOT READY

Send MSG(*User Interface Request Data*) signal

S14.43

'Status' in *Outcome Parameter Set* := END APPLICATION

'Msg On Error' in *Error Indication*:= 'Message Identifier' in *User Interface Request Data*

CreateMSDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 364

6.21 State 15 – Waiting for Put Data Response After Generate AC

6.21.1 Local Variables

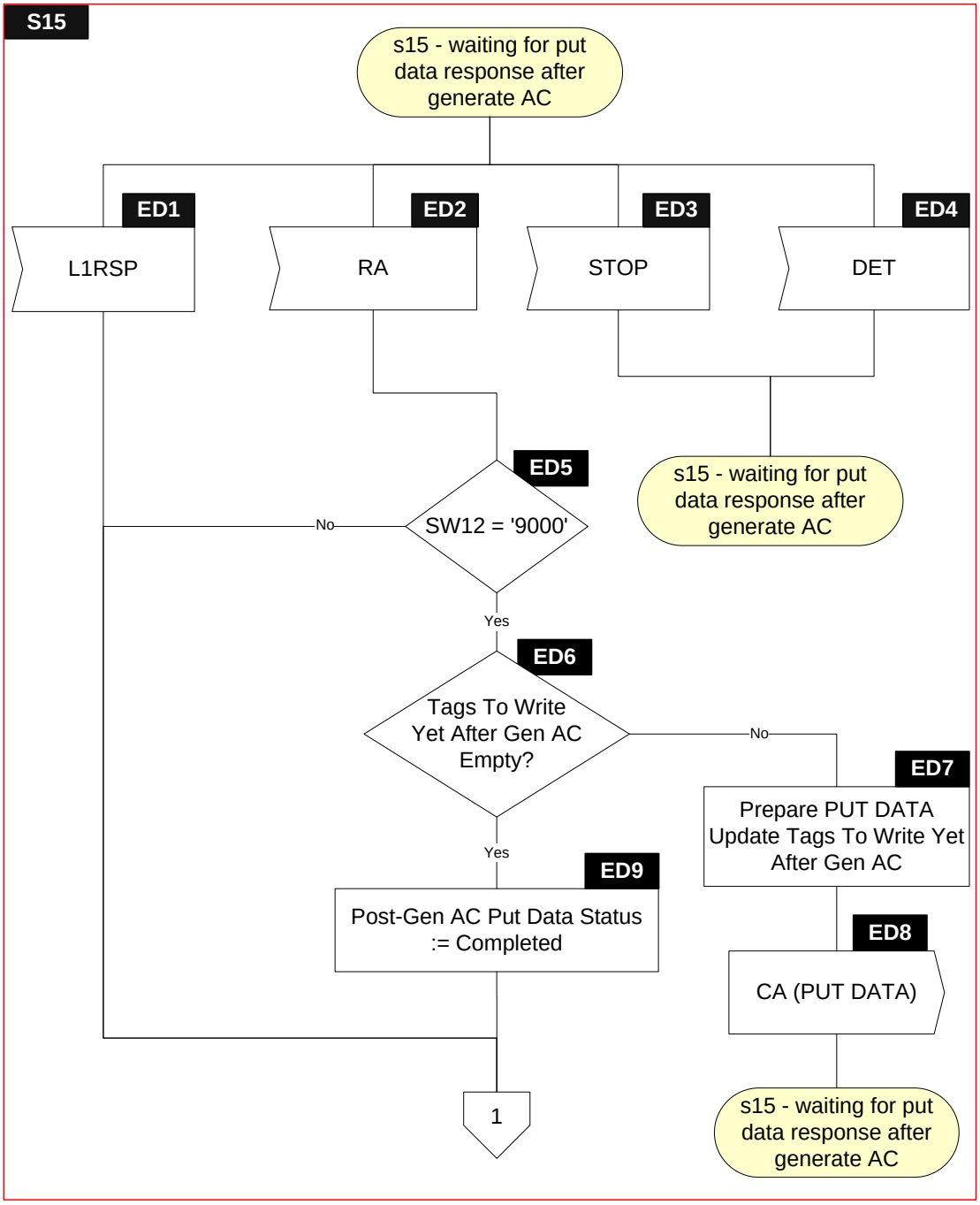
Name	Length	Format	Description
SW12	2	b	Status bytes
T	var.	b	Tag of TLV encoded string
L	var.	b	Length of TLV encoded string
V	var. up to 252	b	Value of TLV encoded string

6.21.2 Flow Diagram

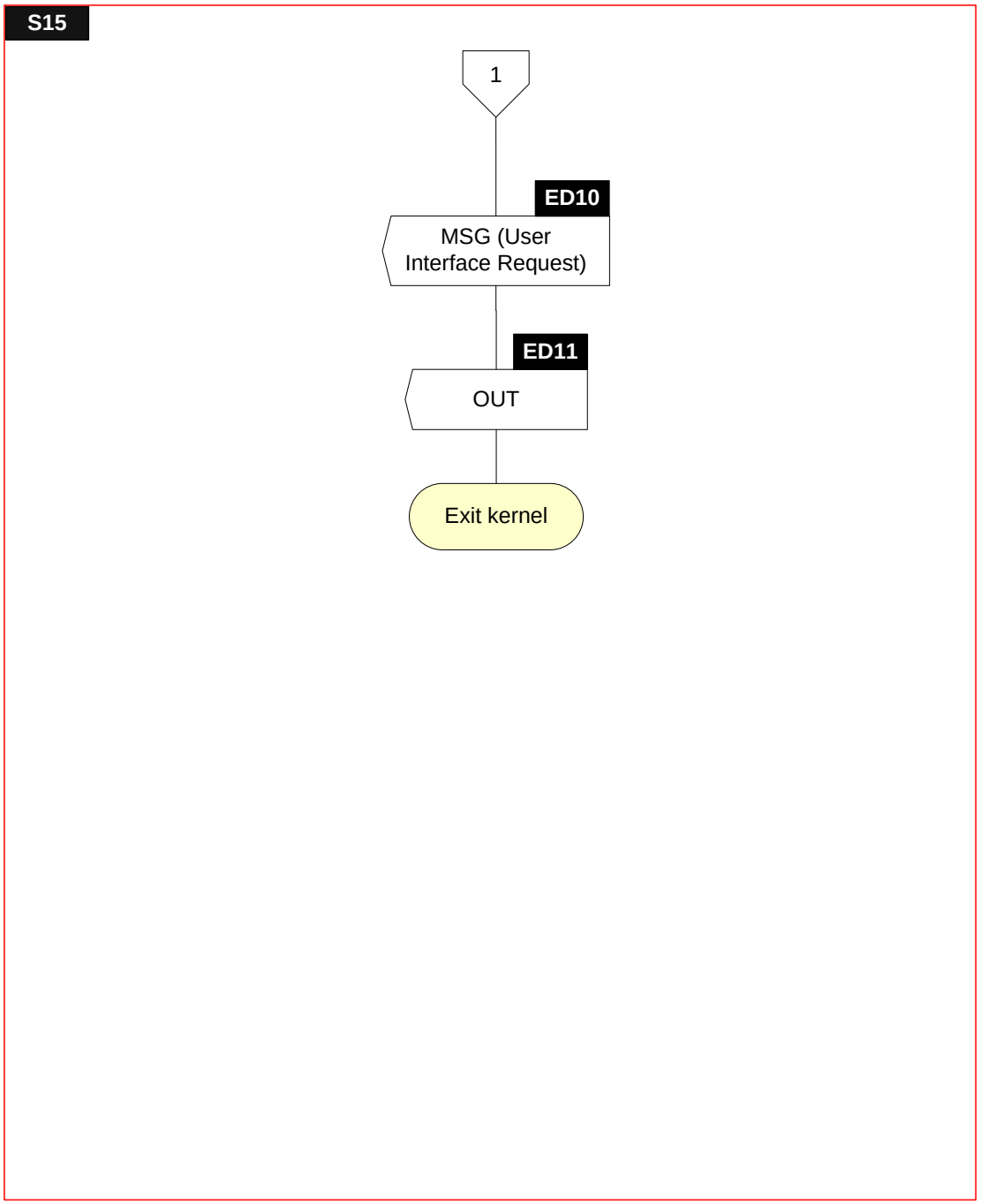
Figure 6.13 shows the flow diagram of `s15 – waiting for put data response after generate AC`. Symbols in this diagram are labeled S15.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 365

Figure 6.20—State 15 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 366



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 367

6.21.3 Processing

S15.ED1

Receive L1RSP signal

S15.ED2

Receive RA signal with SW12

S15.ED3

Receive STOP signal

S15.ED4

Receive DET signal

S15.ED5

IF [SW12 = '9000']
THEN
GOTO S15.ED6
ELSE
GOTO S15.ED10
ENDIF

S15.ED6

IF [IsEmptyList(*Tags To Write Yet After Gen AC*)]
THEN
GOTO S15.ED9
ELSE
GOTO S15.ED7
ENDIF

S15.ED7

TLV := GetAndRemoveFromList(*Tags To Write Yet After Gen AC*)
Prepare PUT DATA command for TLV as specified in section 5.6

S15.ED8

Send CA(PUT DATA command) signal

S15.ED9

SET 'Completed' in *Post-Gen AC Put Data Status*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 368

S15.ED10

'Status' in *User Interface Request Data* := CARD READ SUCCESSFULLY

Send MSG(*User Interface Request Data*) signal

S15.ED11

CreateEMVDiscretionaryData ()

IF [IsNotEmpty(TagOf(*POS Cardholder Interaction Information*))]

THEN

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

ELSE

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Data Record*)), GetTLV(TagOf(*Discretionary Data*))) signal

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 369

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 370

7 Procedures

7.1 Procedure – Pre-gen AC Balance Reading

7.1.1 Local Variables

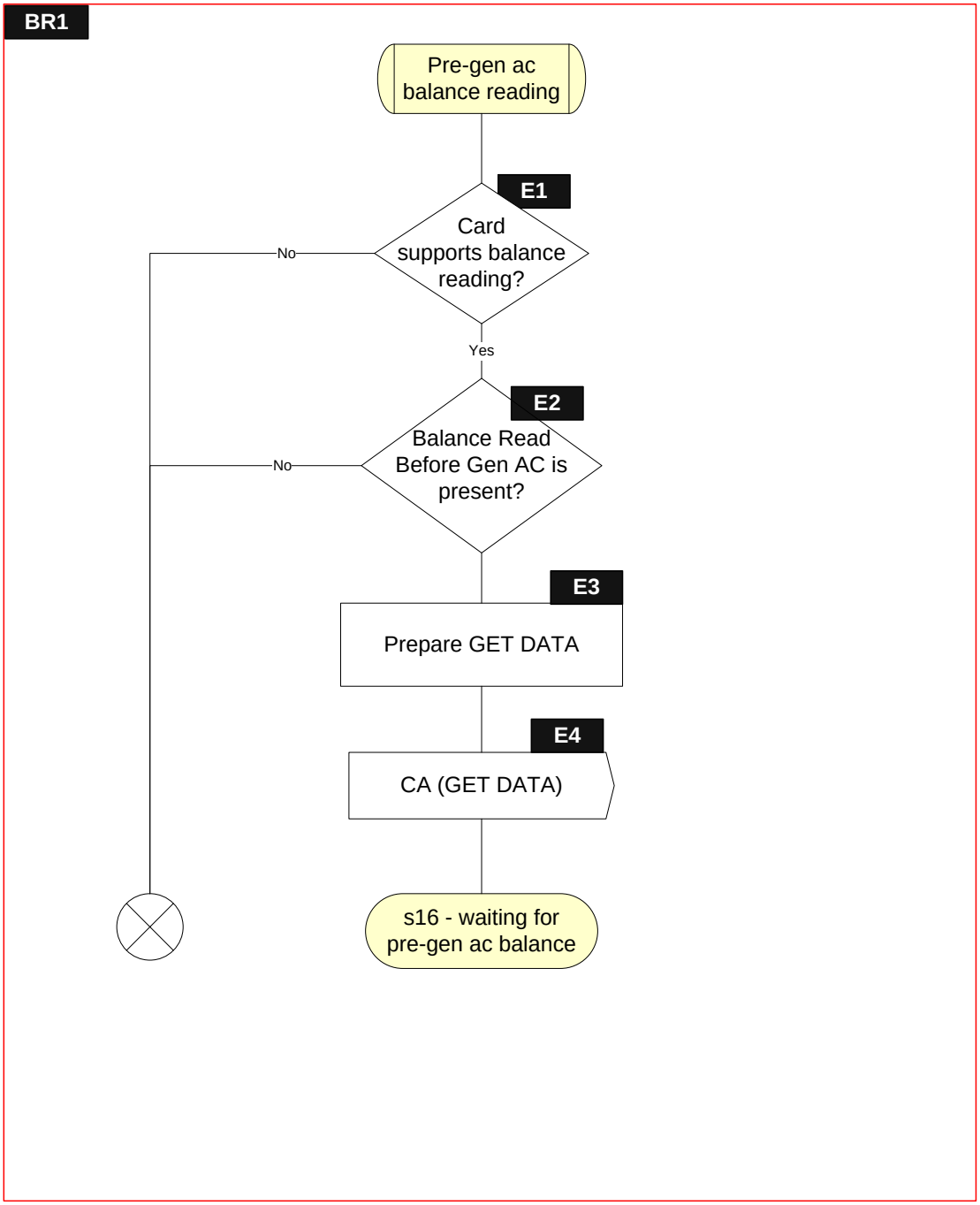
None

7.1.2 Flow Diagram

Figure 7.1 shows the flow diagram of the Pre-gen AC Balance Reading procedure. Symbols in this diagram are labeled BR1.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 371

Figure 7.1—Pre-gen AC Balance Reading Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 372

7.1.3 Processing

BR1.E1

IF [IsNotEmpty(TagOf(*Application Capabilities Information*)) AND 'Support for balance reading' in *Application Capabilities Information* is set]

THEN

GOTO BR1.E2

ELSE

EXIT BR1

ENDIF

BR1.E2

IF [IsPresent(TagOf(*Balance Read Before Gen AC*))]

THEN

GOTO BR1.E3

ELSE

EXIT BR1

ENDIF

BR1.E3

Prepare GET DATA command for '9F50' (*Offline Accumulator Balance*) as specified in section 5.4

BR1.E4

Send CA(GET DATA) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

7.2 State 16 – Waiting for Pre-gen AC Balance

7.2.1 Local Variables

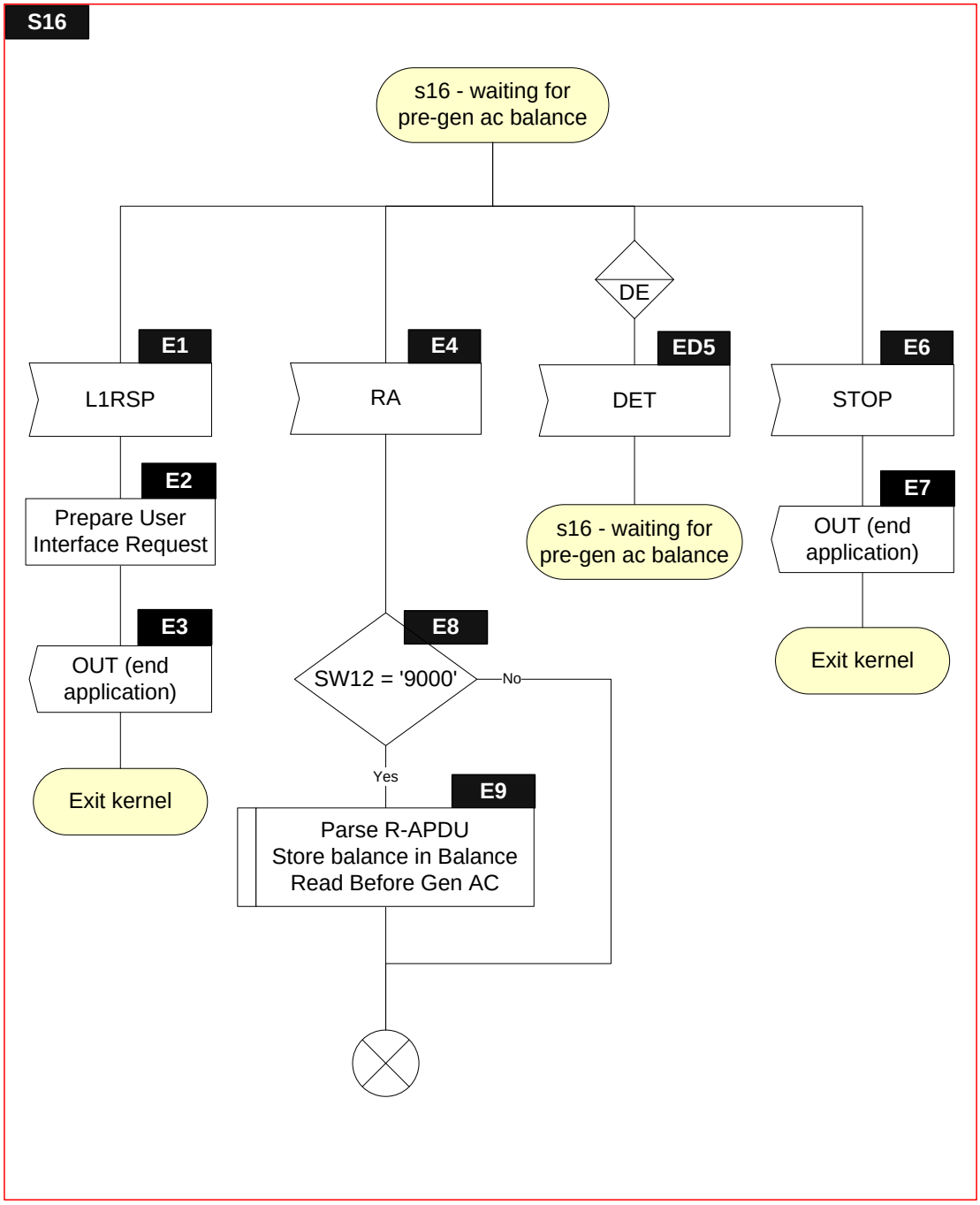
Name	Length	Format	Description
Return Code	1	b	Value returned with L1RSP signal (TIME OUT ERROR, PROTOCOL ERROR, TRANSMISSION ERROR)
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of GET DATA

7.2.2 Flow Diagram

Figure 7.2 shows the flow diagram of `s16 – waiting for pre-gen AC balance`. Symbols in this diagram are labeled S16.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 374

Figure 7.2—State 16 Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 375

7.2.3 Processing

S16.E1

Receive L1RSP signal with Return Code

S16.E2

'Message Identifier' in *User Interface Request Data* := TRY AGAIN

'Status' in *User Interface Request Data* := READY TO READ

'Hold Time' in *User Interface Request Data* := '000000'

S16.E3

'Status' in *Outcome Parameter Set* := END APPLICATION

'Start' in *Outcome Parameter Set* := B

SET 'UI Request on Restart Present' in *Outcome Parameter Set*

'L1' in *Error Indication* := Return Code

'Msg On Error' in *Error Indication*:= TRY AGAIN

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*)), GetTLV(TagOf(*User Interface Request Data*))) signal

S16.E4

Receive RA signal with Response Message Data Field and SW12

S16.ED5

Receive DET signal

S16.E6

Receive STOP signal

S16.E7

'Status' in *Outcome Parameter Set* := END APPLICATION

'L3' in *Error Indication* := STOP

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

S16.E8

IF [SW12 = '9000']

THEN

GOTO S16.E9

ELSE

EXIT BR1

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

S16.E9

Table 7.1—Response Message Data Field

'9F50'	'06'	Offline balance

IF [(Length of Response Message Data Field = 9) AND
(Response Message Data Field[1:2] = '9F50') AND
(Response Message Data Field[3] = '06')]

THEN

Balance Read Before Gen AC := Response Message Data Field[4:9]

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 377

7.3 Procedure – Post-gen AC Balance Reading

7.3.1 Local Variables

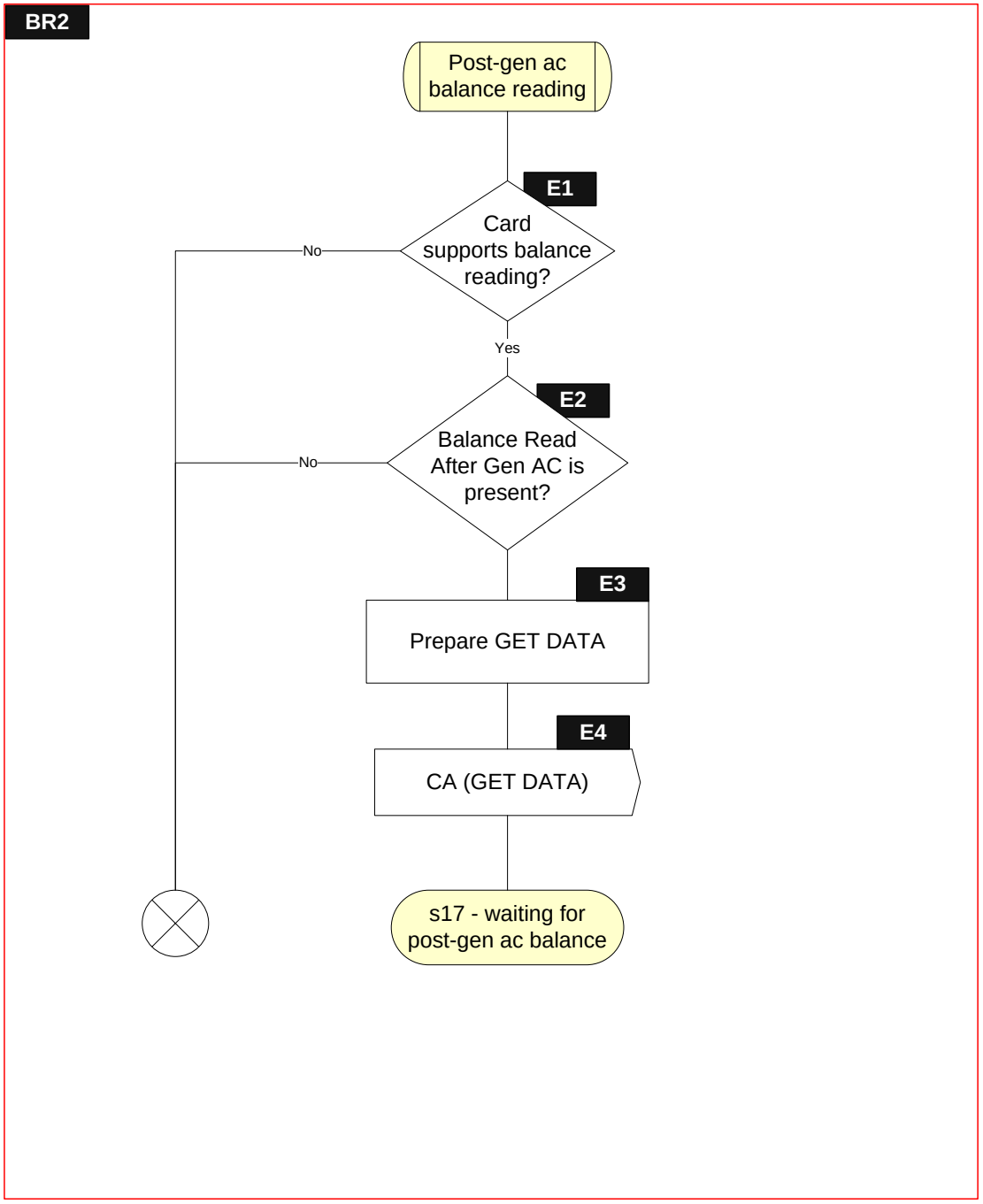
None

7.3.2 Flow Diagram

Figure 7.3 shows the flow diagram of the Post-gen AC Balance Reading procedure. Symbols in this diagram are labeled BR2.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 378

Figure 7.3—Post-gen AC Balance Reading Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 379

7.3.3 Processing

BR2.E1

IF [IsNotEmpty(TagOf(*Application Capabilities Information*)) AND 'Support for balance reading' in *Application Capabilities Information* is set]

THEN

GOTO BR2.E2

ELSE

EXIT BR2

ENDIF

BR2.E2

IF [IsPresent(TagOf(*Balance Read After Gen AC*))]

THEN

GOTO BR2.E3

ELSE

EXIT BR2

ENDIF

BR2.E3

Prepare GET DATA command for '9F50' (*Offline Accumulator Balance*) as specified in section 5.4

BR2.E4

Send CA(GET DATA) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 380

7.4 State 17 – Waiting for Post-gen AC Balance

7.4.1 Local Variables

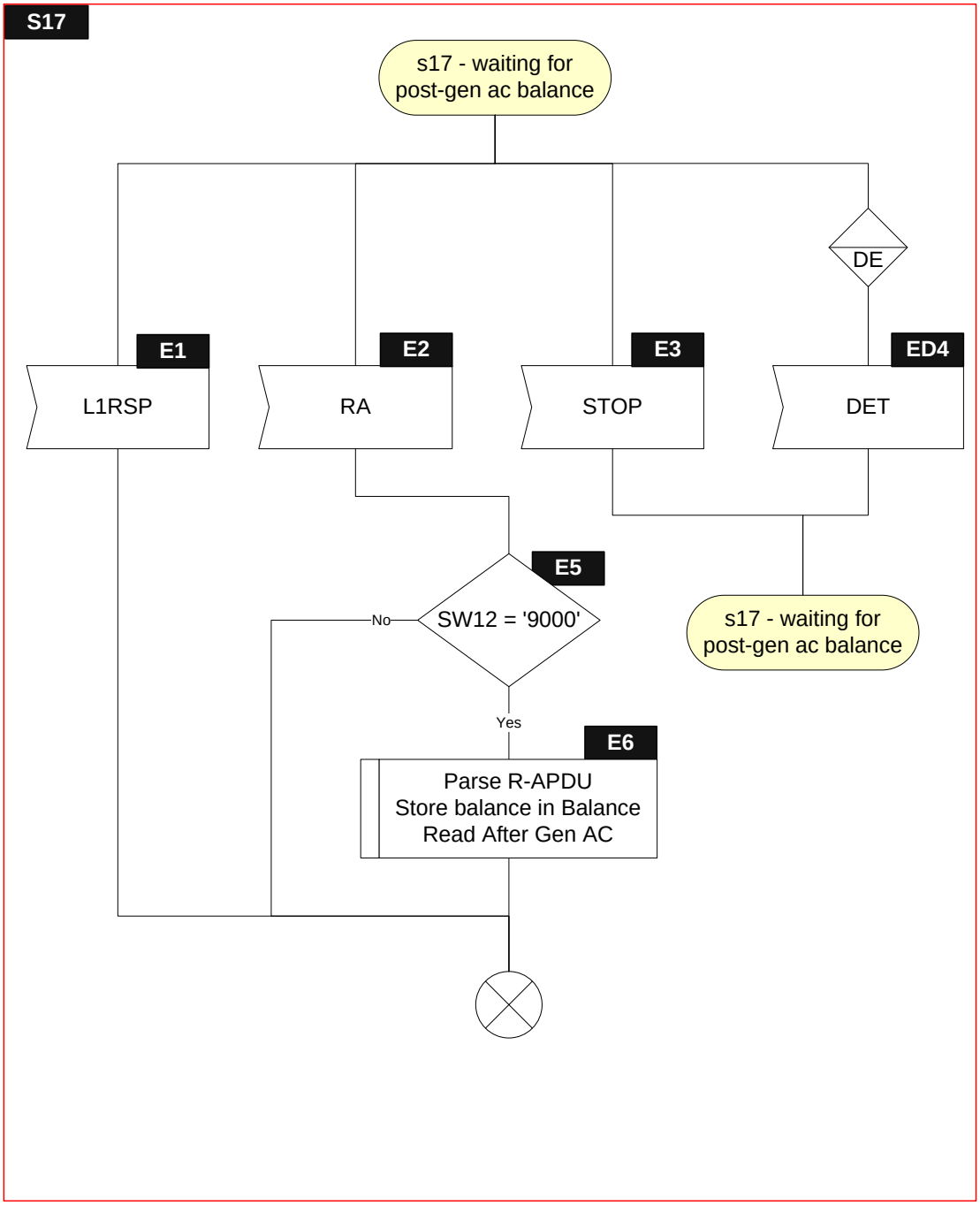
Name	Length	Format	Description
SW12	2	b	Status bytes
Response Message Data Field	var. up to 256	b	TLV encoded string included in R-APDU of GET DATA

7.4.2 Flow Diagram

Figure 7.4 shows the flow diagram of `s17 – waiting for post-gen AC balance`. Symbols in this diagram are labeled S17.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 381

Figure 7.4—State 17 Flow Diagram



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 382

7.4.3 Processing

S17.E1

Receive L1RSP signal

S17.E2

Receive RA signal with Response Message Data Field and SW12

S17.E3

Receive STOP signal

S17.ED4

Receive DET signal

S17.E5

IF [SW12 = '9000']
THEN
GOTO S17.E6
ELSE
EXIT BR2
ENDIF

S17.E6

Table 7.2—Response Message Data Field

'9F50'	'06'	Offline balance

IF [(Length of Response Message Data Field = 9) AND
(Response Message Data Field[1:2] = '9F50') AND
(Response Message Data Field[3] = '06')]
THEN
Balance Read After Gen AC := Response Message Data Field[4:9]
ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 383

7.5 Procedure – CVM Selection

7.5.1 Local Variables

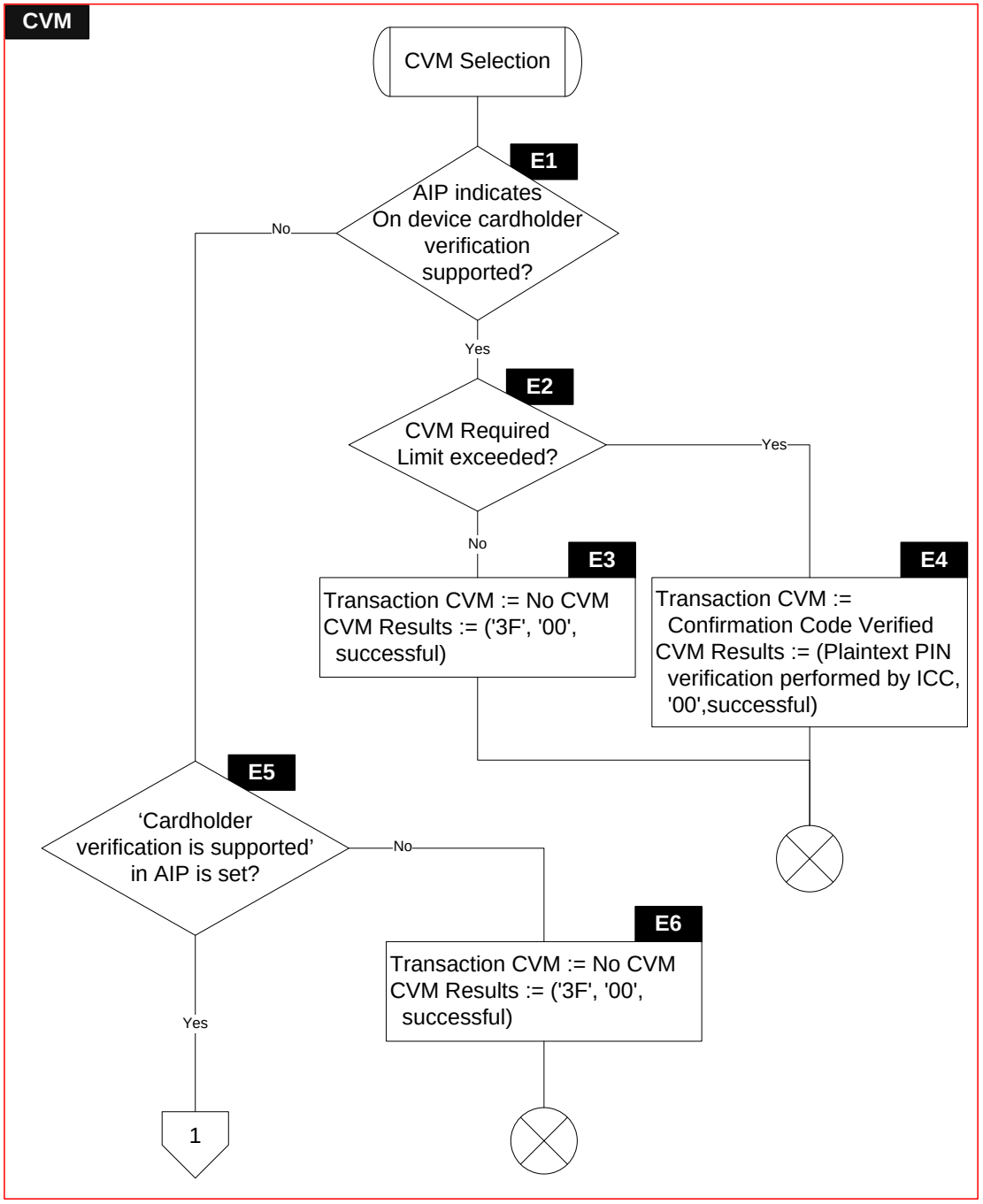
Name	Length	Format	Description
CVR	2	b	Cardholder Verification Rule
CVM Condition Code	1	b	Second byte of a CVR
CVM Code	1	b	First byte of a CVR

7.5.2 Flow Diagram

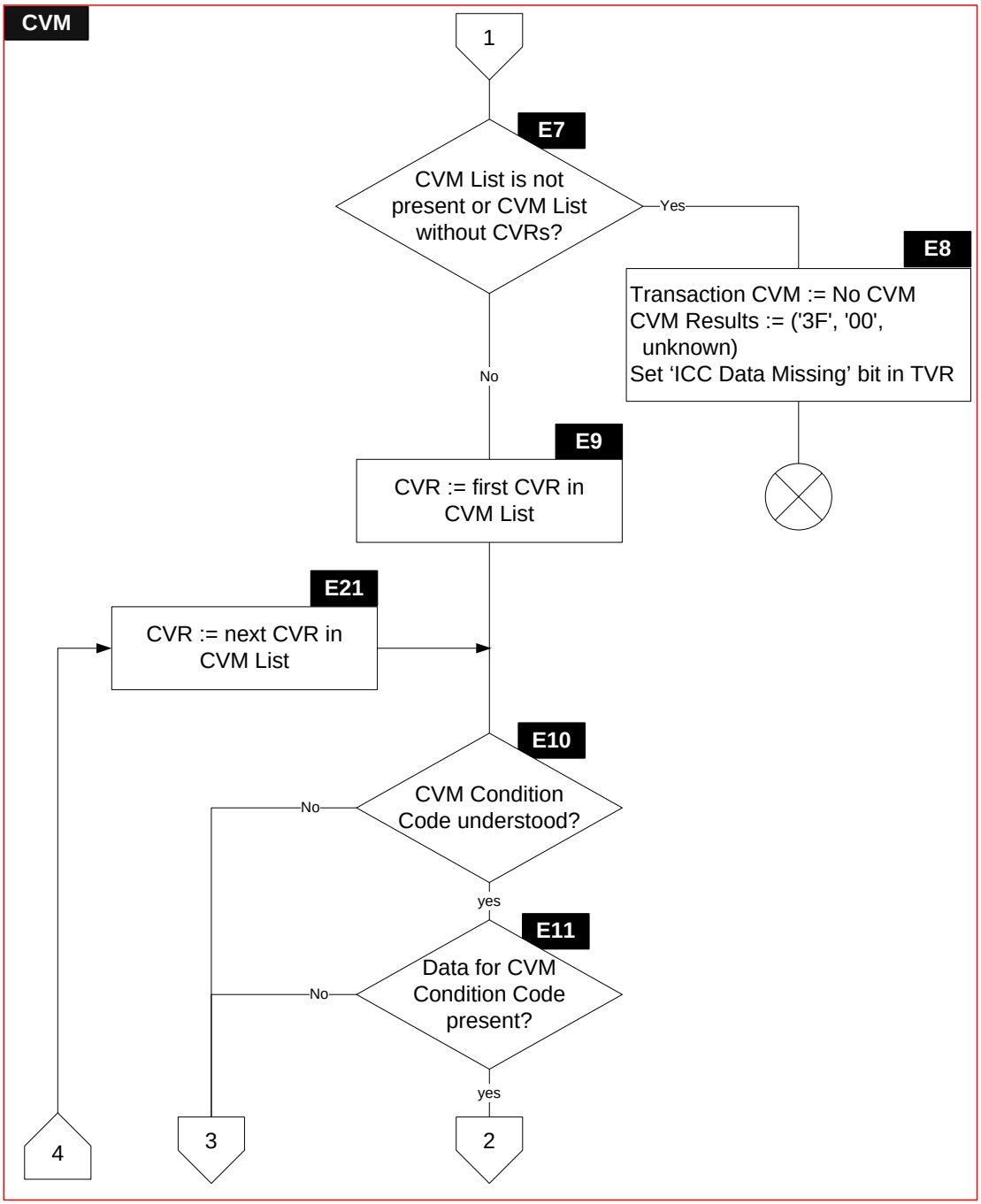
Figure 7.5 shows the flow diagram of the CVM Selection procedure. Symbols in this diagram are labeled CVM.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 384

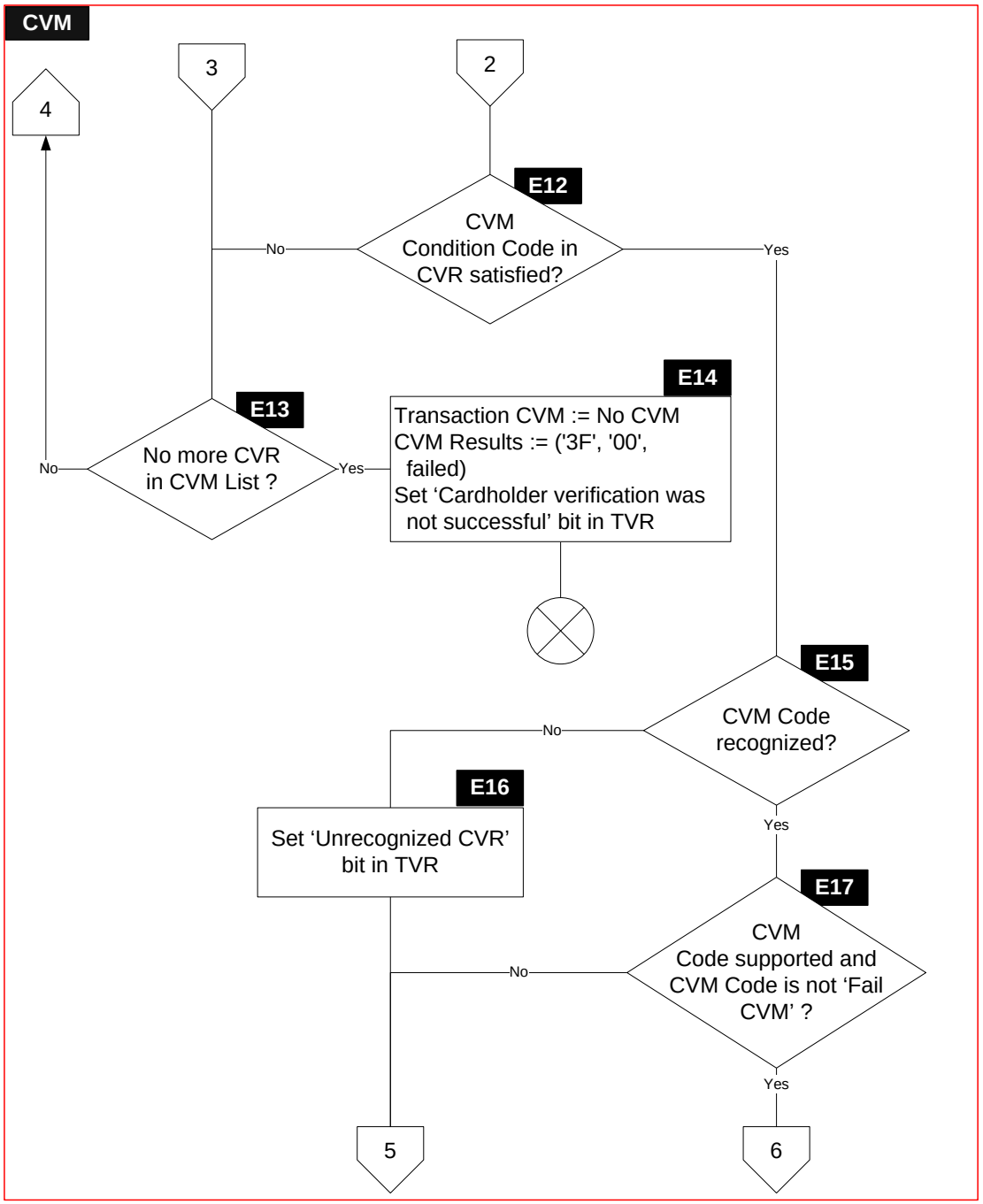
Figure 7.5—CVM Selection Flow Diagram



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 385

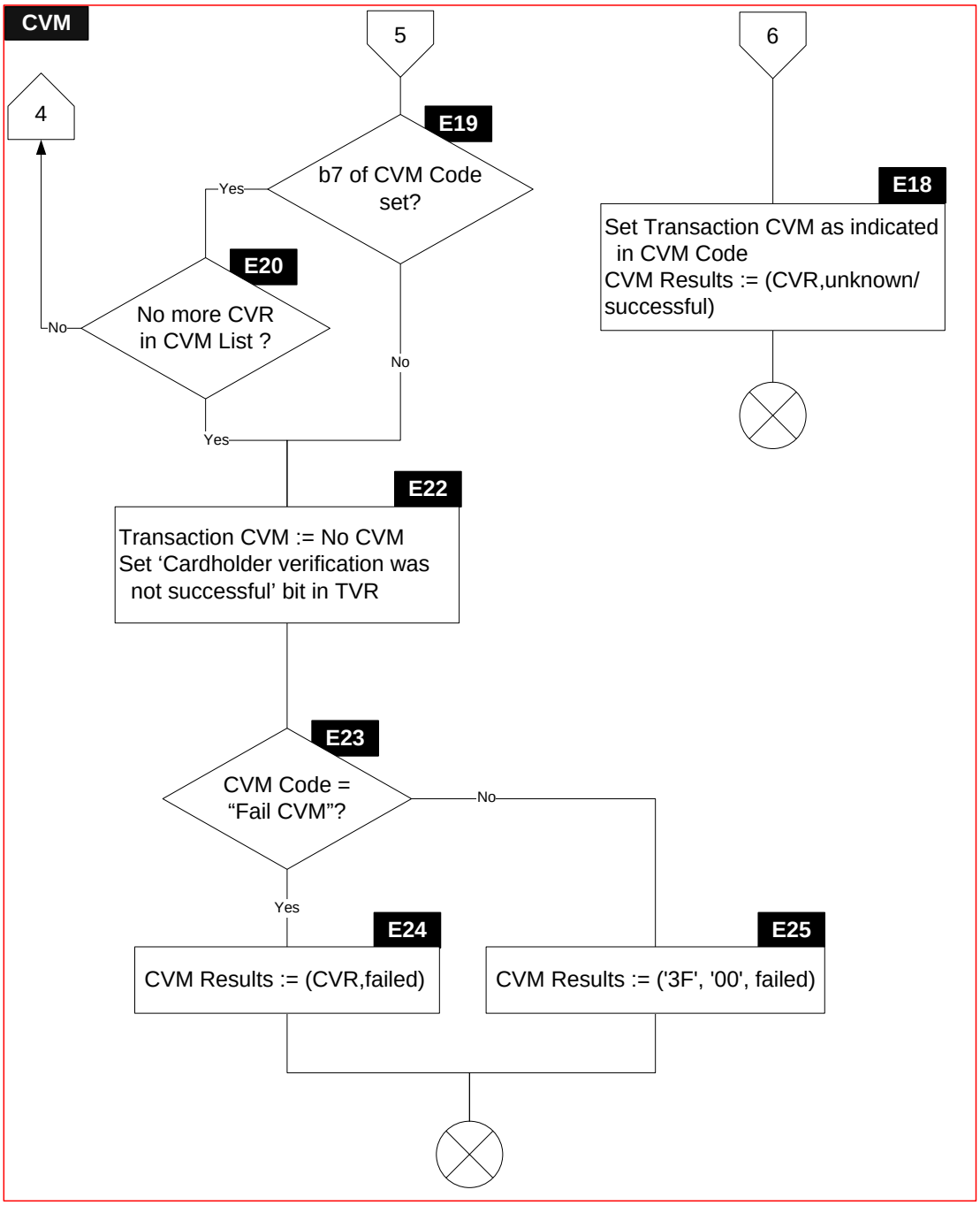


© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 386



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 387



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 388

7.5.3 Processing

CVM.E1

IF ['On device cardholder verification is supported' in *Application Interchange Profile* is set]

THEN

GOTO CVM.E2

ELSE

GOTO CVM.E5

ENDIF

CVM.E2

IF [*Amount, Authorized (Numeric)* > *Reader CVM Required Limit*]

THEN

GOTO CVM.E4

ELSE

GOTO CVM.E3

ENDIF

CVM.E3

'CVM' in *Outcome Parameter Set* := NO CVM

'CVM Performed' in *CVM Results* := '3F' (No CVM performed)

'CVM Condition' in *CVM Results* := '00'

'CVM Result' in *CVM Results* := '02' (successful)

CVM.E4

'CVM' in *Outcome Parameter Set* := CONFIRMATION CODE VERIFIED

'CVM Performed' in *CVM Results* := '01' (Plaintext PIN verification performed by ICC)

'CVM Condition' in *CVM Results* := '00'

'CVM Result' in *CVM Results* := '02' (successful)

CVM.E5

IF ['Cardholder verification is supported' in *Application Interchange Profile* is set]

THEN

GOTO CVM.E7

ELSE

GOTO CVM.E6

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 389

CVM.E6

'CVM' in *Outcome Parameter Set* := NO CVM

'CVM Performed' in *CVM Results* := '3F' (No CVM performed)

'CVM Condition' in *CVM Results* := '00'

'CVM Result' in *CVM Results* := '02' (successful)

CVM.E7

IF [IsNotPresent(TagOf(*CVM List*)) OR (GetLength(TagOf(*CVM List*)) = 8)]

THEN

GOTO CVM.E8

ELSE

GOTO CVM.E9

ENDIF

CVM.E8

'CVM' in *Outcome Parameter Set* := NO CVM

'CVM Performed' in *CVM Results* := '3F' (No CVM performed)

'CVM Condition' in *CVM Results* := '00'

'CVM Result' in *CVM Results* := '00' (unknown)

SET 'ICC data missing' in *Terminal Verification Results*

CVM.E9

CVR := first CV Rule in *CVM List*

CVM Code := CVR[1]

CVM Condition Code := CVR[2]

CVM.E10

IF [CVM Condition Code is understood (i.e. the CVM Condition Code is included in Table 40 of Annex C.3 of [EMV Book 3])]

THEN

GOTO CVM.E11

ELSE

GOTO CVM.E13

ENDIF

Note that the Kernel may also understand proprietary CVM condition codes not defined in Annex C.3 of [EMV Book 3].

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 390

CVM.E11

IF [Data required by the conditions expressed by the CVM Condition Code are present in the TLV Database]

THEN

GOTO CVM.E12

ELSE

GOTO CVM.E13

ENDIF

CVM.E12

IF [Conditions expressed by the CVM Condition Code are satisfied]

THEN

GOTO CVM.E15

ELSE

GOTO CVM.E13

ENDIF

CVM.E13

IF [CVR is last CV Rule in CVM List]

THEN

GOTO CVM.E14

ELSE

GOTO CVM.E21

ENDIF

CVM.E14

'CVM' in *Outcome Parameter Set* := NO CVM

'CVM Performed' in *CVM Results* := '3F' (No CVM performed)

'CVM Condition' in *CVM Results* := '00'

'CVM Result' in *CVM Results* := '01' (failed)

SET 'Cardholder verification was not successful' in *Terminal Verification Results*

CVM.E15

IF [CVM Code is recognized (i.e. the CVM Code is included in Table 39 of Annex C.3 of [EMV Book 3])]

THEN

GOTO CVM.E17

ELSE

GOTO CVM.E16

ENDIF

Note that the Kernel may also recognize proprietary CVM codes not defined in Annex C.3 of [EMV Book 3].

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 391

CVM.E16

SET 'Unrecognised CVM' in *Terminal Verification Results*

CVM.E17

Verify if the CVM Code is supported:

- For CVM Codes defined in Annex C.3 of [EMV Book 3], support must be indicated in *Terminal Capabilities*.
- For CVM Codes not defined in Annex C.3 of [EMV Book 3], support may be known explicitly.
- For combination CVMs, both CVM codes must be supported.
- Fail CVM processing ('00' or '40') must always be supported.

IF [CVM Code is supported AND ((CVM Code AND '3F') ≠ '00')]

THEN

GOTO CVM.E18

ELSE

GOTO CVM.E19

ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 392

CVM.E18

IF [(CVM Code AND '3F')= '02']
THEN
'CVM' in *Outcome Parameter Set* := ONLINE PIN
'CVM Result' in *CVM Results* := '00' (unknown)
SET 'Online PIN entered' in *Terminal Verification Results*
ELSE
IF [(CVM Code AND '3F') = '1E']
THEN
'CVM' in *Outcome Parameter Set* := OBTAIN SIGNATURE
'CVM Result' in *CVM Results* := '00' (unknown)
'Receipt' in *Outcome Parameter Set* := YES
ELSE
IF [(CVM Code AND '3F') = '1F']
THEN
'CVM' in *Outcome Parameter Set* := NO CVM
'CVM Result' in *CVM Results* := '02' (successful)
ELSE
Set 'CVM' in *Outcome Parameter Set* to proprietary value
'CVM Result' in *CVM Results* := '00' or '02'
ENDIF
ENDIF
ENDIF

'CVM Performed' in *CVM Results* := CVM Code
'CVM Condition' in *CVM Results* := CVM Condition Code

CVM.E19

IF [CVM Code[7] is set (i.e. apply succeeding CV Rule if this CVM is unsuccessful)]
THEN
GOTO CVM.E20
ELSE
GOTO CVM.E22
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 393

CVM.E20

IF [CVR is last CV Rule in CVM List]
THEN
GOTO CVM.E22
ELSE
GOTO CVM.E21
ENDIF

CVM.E21

CVR := next CV Rule in *CVM List*
CVM Code := CVR[1]
CVM Condition Code := CVR[2]

CVM.E22

'CVM' in *Outcome Parameter Set* := NO CVM
SET 'Cardholder verification was not successful' in *Terminal Verification Results*

CVM.E23

IF [(CVM Code AND '3F') = '00']
THEN
GOTO CVM.E24
ELSE
GOTO CVM.E25
ENDIF

CVM.E24

'CVM Performed' in *CVM Results* := CVM Code
'CVM Condition' in *CVM Results* := 'CVM Condition Code
'CVM Result' in *CVM Results* := '01' (failed)

CVM.E25

'CVM Performed' in *CVM Results* := '3F'
'CVM Condition' in *CVM Results* := '00'
'CVM Result' in *CVM Results* := '01' (failed)

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 394

7.6 Procedure – Prepare Generate AC Command

7.6.1 Local Variables

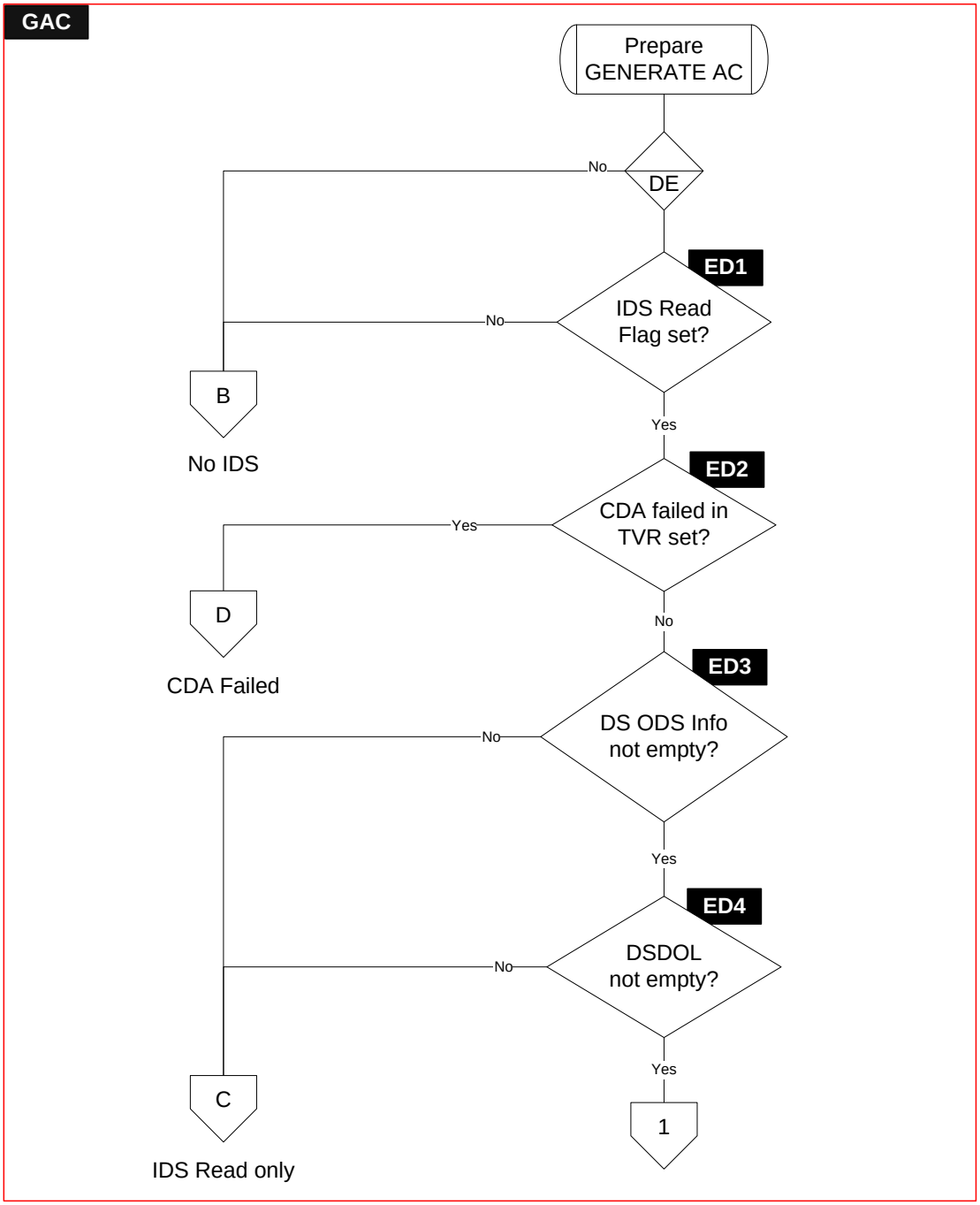
None

7.6.2 Flow Diagram

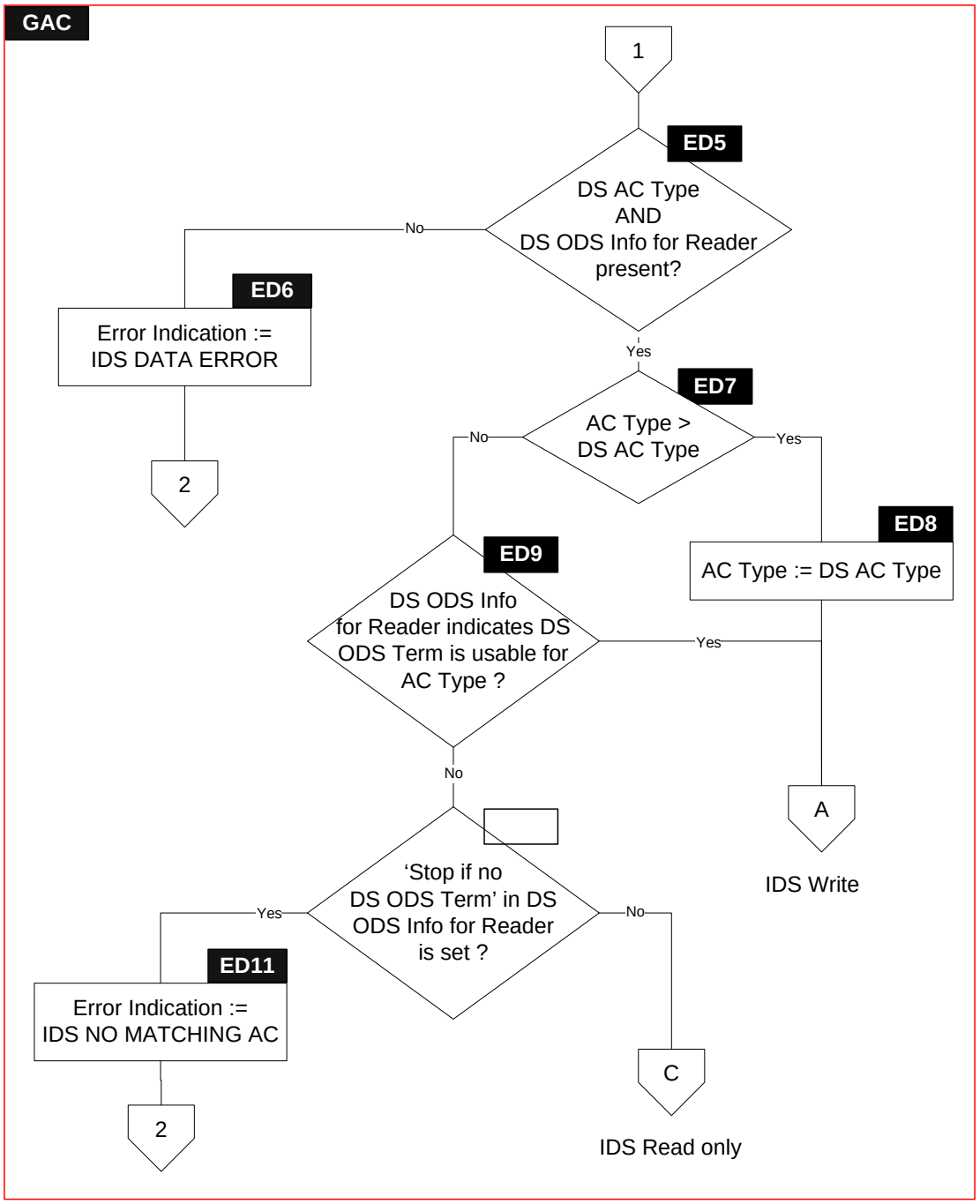
Figure 7.6 shows the flow diagram of the Prepare Generate AC Command procedure. Symbols in this diagram are labeled GAC.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 395

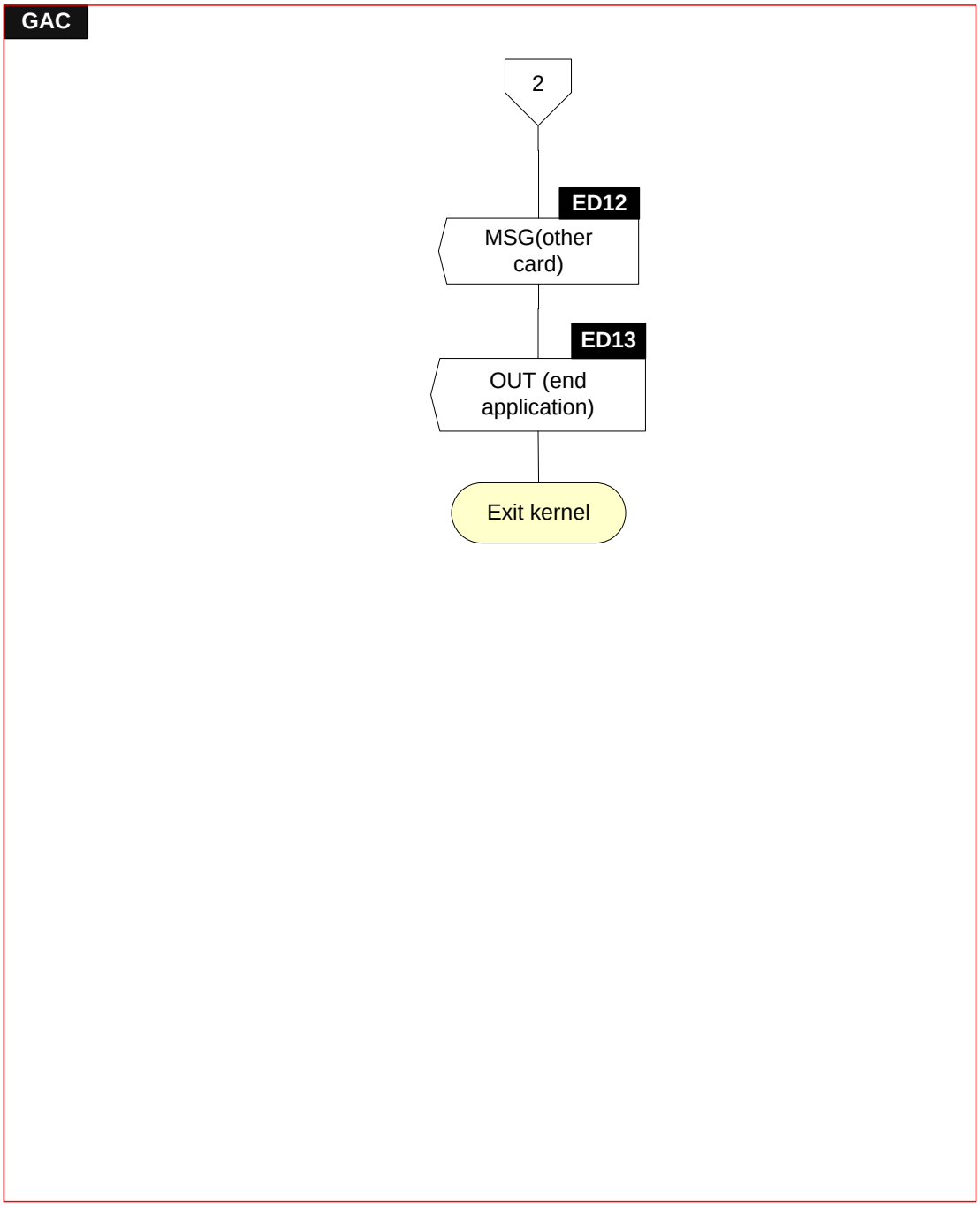
Figure 7.6—Prepare Generate AC Command Flow Diagram



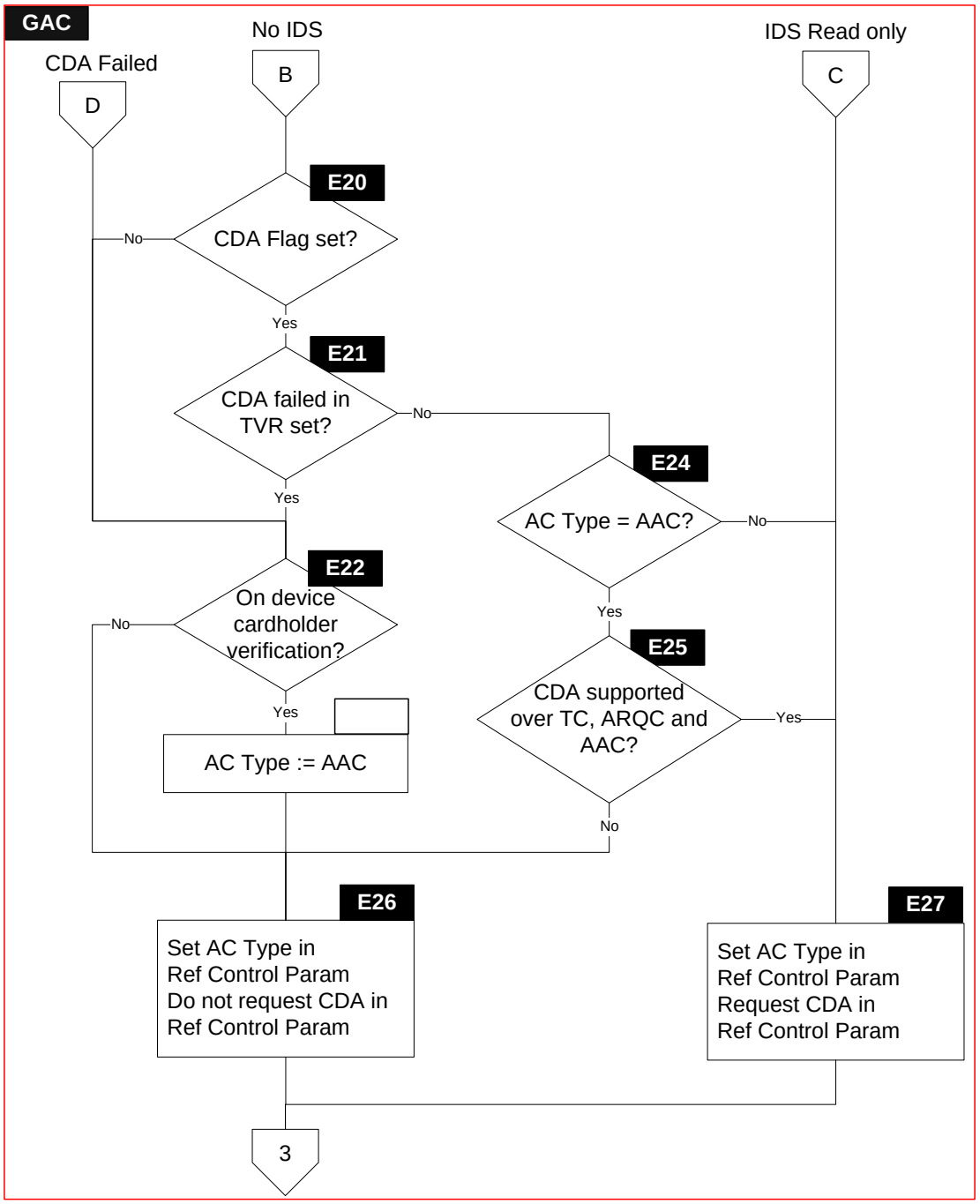
© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 396



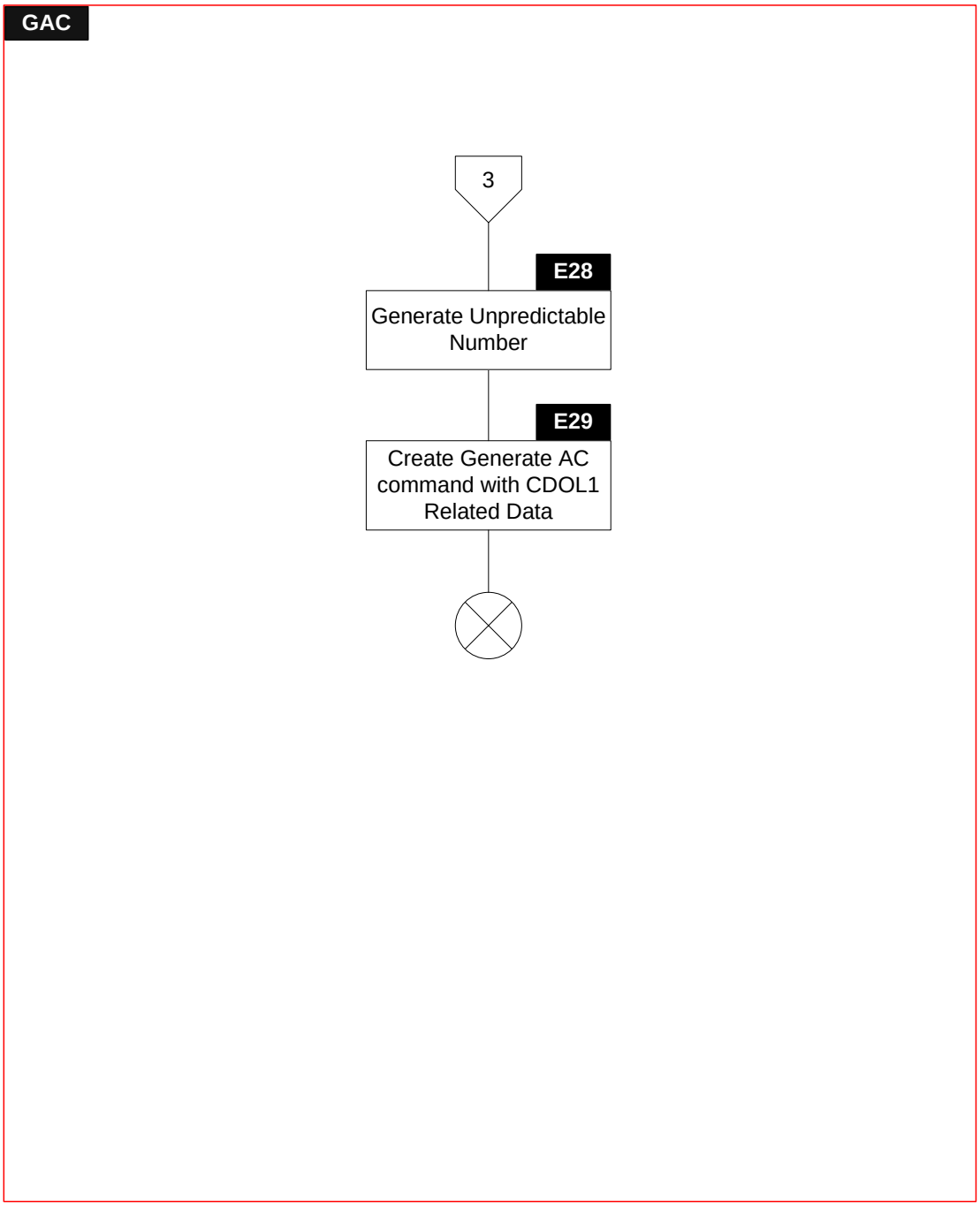
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 397



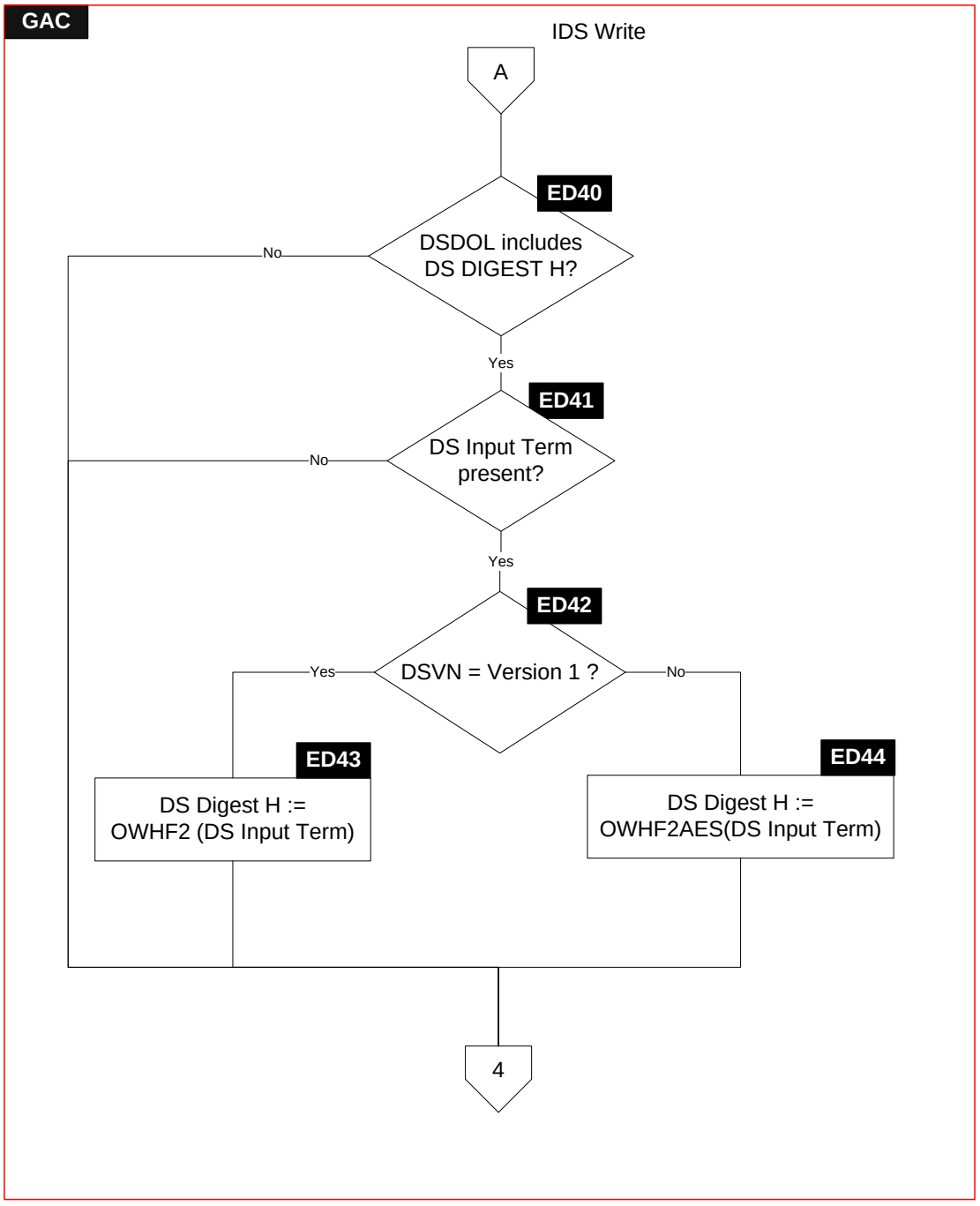
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 398



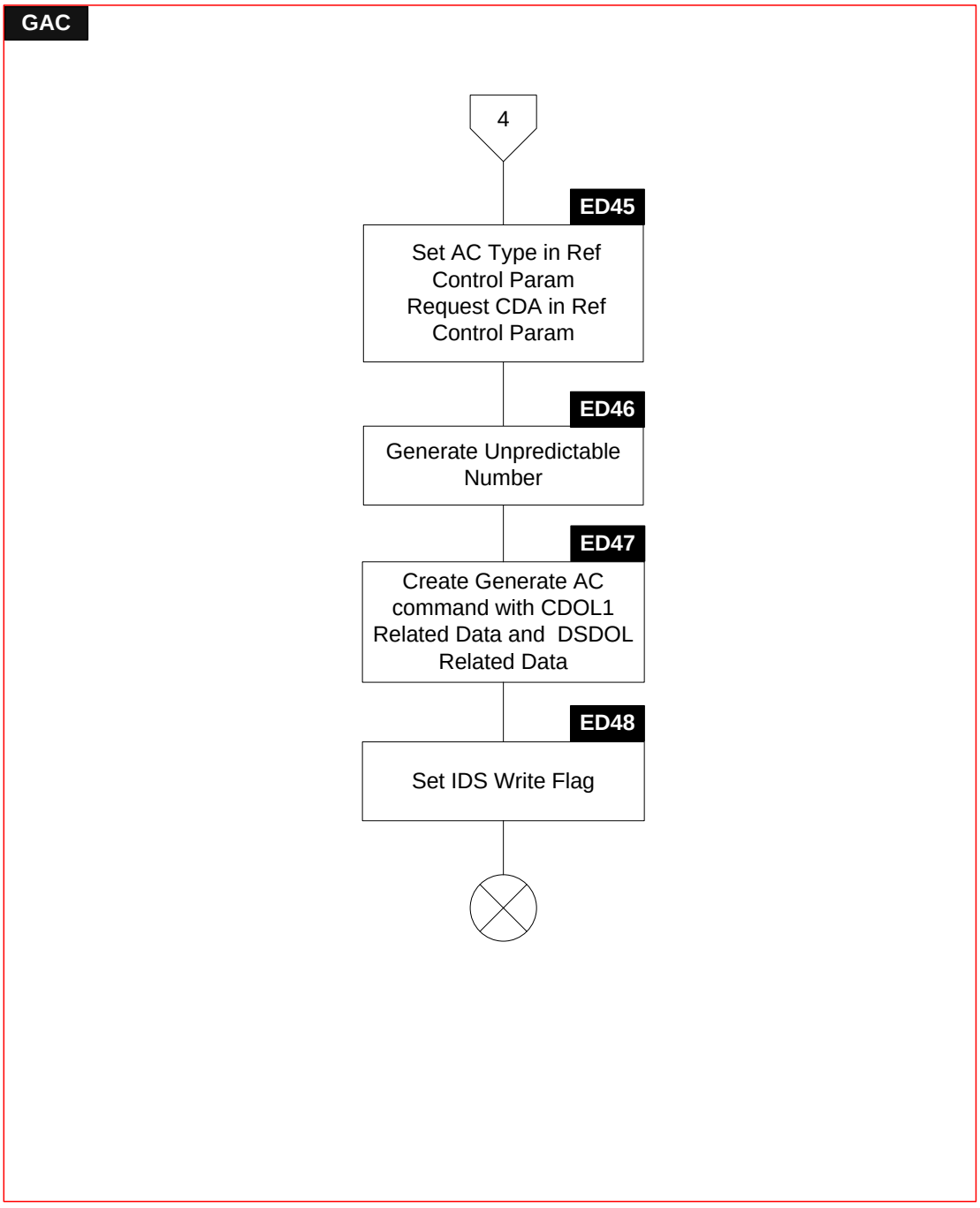
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 399



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 400



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 401



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 402

7.6.3 Processing

GAC.ED1

IF ['Read' in *IDS Status* is set]
THEN
GOTO GAC.ED2
ELSE
GOTO GAC.E20
ENDIF

GAC.ED2

IF ['CDA failed' in *Terminal Verification Results* is set]
THEN
GOTO GAC.E22
ELSE
GOTO GAC.ED3
ENDIF

GAC.ED3

IF [IsNotEmpty(*DS ODS Info*)]
THEN
GOTO GAC.ED4
ELSE
GOTO GAC.E27
ENDIF

GAC.ED4

IF [IsNotEmpty(TagOf(*DSDOL*))]
THEN
GOTO GAC.ED5
ELSE
GOTO GAC.E27
ENDIF

GAC.ED5

IF [IsNotEmpty(TagOf(*DS AC Type*)) AND
IsNotEmpty(TagOf(*DS ODS Info For Reader*))]
THEN
GOTO GAC.ED7
ELSE
GOTO GAC.ED6
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 403

GAC.ED6

'L2' in *Error Indication* := IDS DATA ERROR

GAC.ED7

IF [('AC type' in *DS AC Type* = AAC)
OR
('AC type' in *AC Type* = 'AC type' in *DS AC Type*)
OR
(('AC type' in *DS AC Type* = ARQC) AND ('AC type' in *AC Type* = TC))]
THEN
GOTO GAC.ED8
ELSE
GOTO GAC.ED9
ENDIF

GAC.ED8

'AC type' in *AC Type* := 'AC type' in *DS AC Type*

GAC.ED9

IF [(('AC type' in *AC Type* = AAC) AND 'Usable for AAC' in *DS ODS Info For Reader* is set)
OR
(('AC type' in *AC Type* = ARQC) AND 'Usable for ARQC' in *DS ODS Info For Reader* is set)
OR
(('AC type' in *AC Type* = TC) AND 'Usable for TC' in *DS ODS Info For Reader* is set)]
THEN
GOTO GAC.ED40
ELSE
GOTO GAC.ED10
ENDIF

GAC.ED10

IF ['Stop if no DS ODS Term' in *DS ODS Info For Reader* is set]
THEN
GOTO GAC.ED11
ELSE
GOTO GAC.E27
ENDIF

GAC.ED11

'L2' in *Error Indication* := IDS NO MATCHING AC

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 404

GAC.ED12

'Message Identifier' in *User Interface Request Data* := ERROR - OTHER CARD

'Status' in *User Interface Request Data* := NOT READY

Send MSG(*User Interface Request Data*) signal

GAC.ED13

'Status' in *Outcome Parameter Set* := END APPLICATION

'Msg On Error' in *Error Indication* := ERROR - OTHER CARD

CreateEMVDiscretionaryData ()

Send OUT(GetTLV(TagOf(*Outcome Parameter Set*)), GetTLV(TagOf(*Discretionary Data*))) signal

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 405

No IDS

GAC.E20

IF ['CDA' in *ODA Status* is set]
THEN
GOTO GAC.E21
ELSE
GOTO GAC.E22
ENDIF

GAC.E21

IF ['CDA failed' in *Terminal Verification Results* is set]
THEN
GOTO GAC.E22
ELSE
GOTO GAC.E24
ENDIF

GAC.E22

IF ['On device cardholder verification is supported' in *Application Interchange Profile* is set]
THEN
GOTO GAC.E23
ELSE
GOTO GAC.E26
ENDIF

GAC.E23

'AC type' in *AC Type* := AAC

GAC.E24

IF ['AC type' in *AC Type* = AAC]
THEN
GOTO GAC.E25
ELSE
GOTO GAC.E27
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 406

GAC.E25

IF [IsNotEmpty(TagOf(*Application Capabilities Information*)) AND
'CDA Indicator' in *Application Capabilities Information* = CDA SUPPORTED OVER TC, ARQC AND AAC]

THEN

GOTO GAC.E27

ELSE

GOTO GAC.E26

ENDIF

GAC.E26

Reference Control Parameter : = '00'

'AC type' in *Reference Control Parameter* := 'AC type' in *AC Type*

GAC.E27

Reference Control Parameter : = '00'

'AC type' in *Reference Control Parameter* := 'AC type' in *AC Type*

SET 'CDA signature requested' in *Reference Control Parameter*

GAC.E28

Generate *Unpredictable Number* as specified in section 8.1

GAC.E29

Prepare GENERATE AC command as specified in section 5.3.2. Use *CDOL1* to create *CDOL1 Related Data* as a concatenated list of data objects without tags or lengths following the rules specified in section 4.1.4.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 407

IDS Write

GAC.ED40

IF [*DSDOL* includes TagOf(*DS Digest H*)]
THEN
GOTO GAC.ED41
ELSE
GOTO GAC.ED45
ENDIF

GAC.ED41

IF [IsPresent(TagOf(*DS Input (Term)*))]
THEN
GOTO GAC.ED42
ELSE
GOTO GAC.ED45
ENDIF

GAC.ED42

IF ['Data Storage Version Number' in *Application Capabilities Information* = VERSION 1]
THEN
GOTO GAC.ED43
ELSE
GOTO GAC.ED44
ENDIF

GAC.ED43

DS Digest H := OWHF2(*DS Input (Term)*)
Refer to section 8.2 for the description of OWHF2

GAC.ED44

DS Digest H := OWHF2AES(*DS Input (Term)*)
Refer to section 8.3 for the description of OWHF2AES

GAC.ED45

Reference Control Parameter : = '00'
'AC type' in *Reference Control Parameter* := 'AC type' in *AC Type*
SET 'CDA signature requested' in *Reference Control Parameter*

GAC.ED46

Generate *Unpredictable Number* as specified in section 8.1

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 408

GAC.ED47

Prepare GENERATE AC command as specified in section 5.3.2. Use *CDOL1* and *DSDOL* to create *CDOL1 Related Data* and *DSDOL* related data as concatenated lists of data objects without tags or lengths following the rules specified in section 4.1.4.

GAC.ED48

SET 'Write' in *IDS Status*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 409

7.7 Procedure – Processing Restrictions

7.7.1 Local Variables

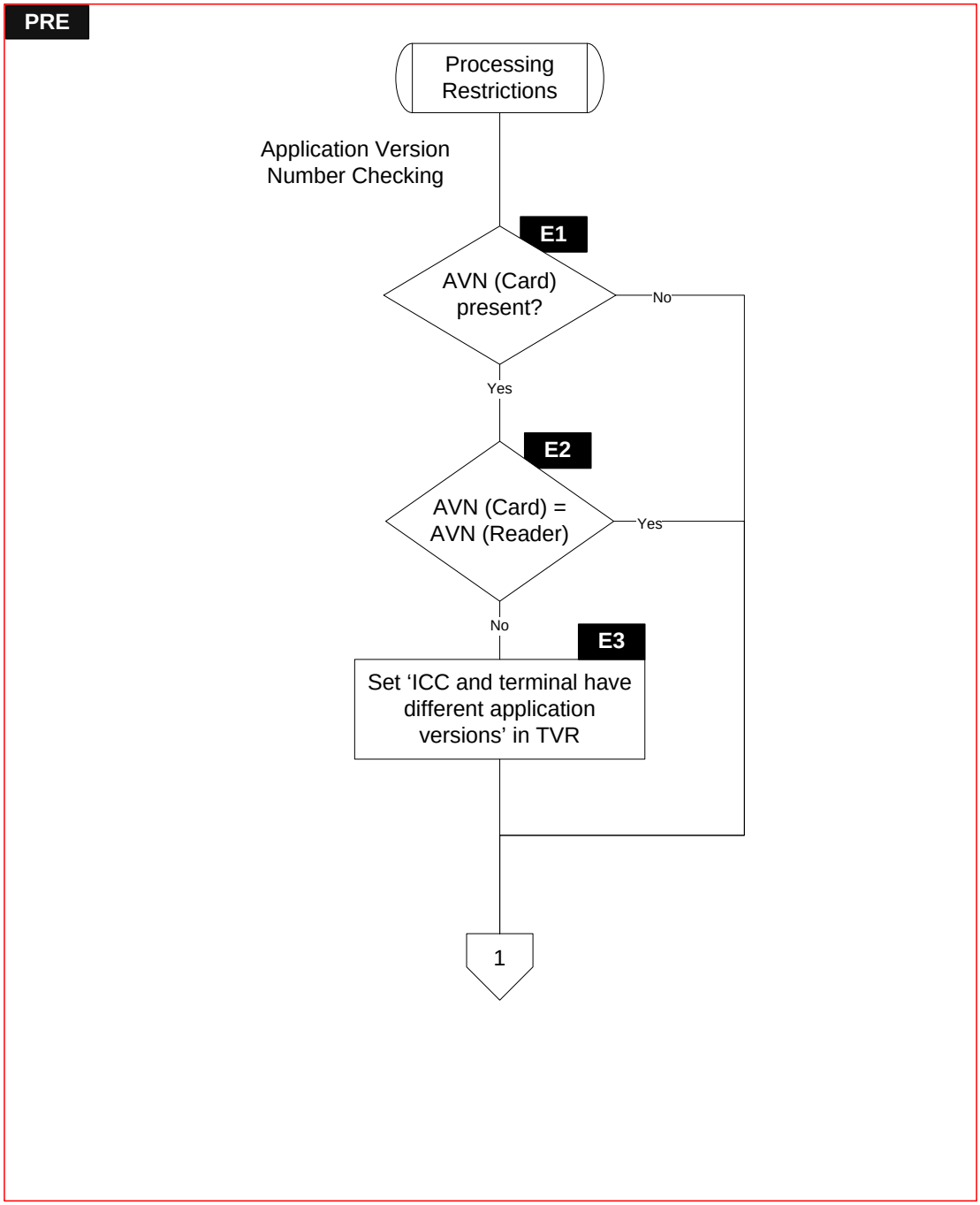
None

7.7.2 Flow Diagram

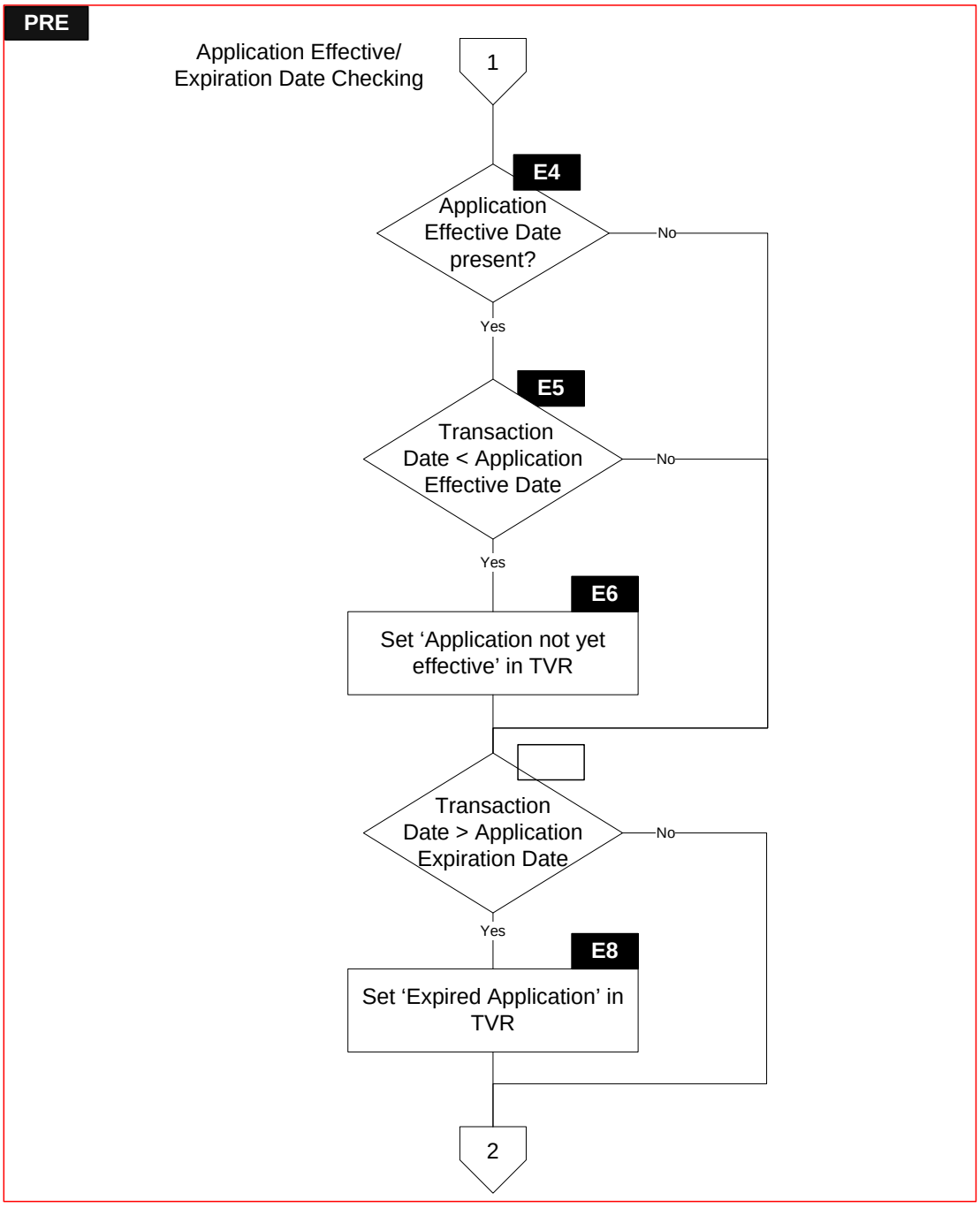
Figure 7.7 shows the flow diagram of the Processing Restrictions procedure. Symbols in this diagram are labeled PRE.X.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

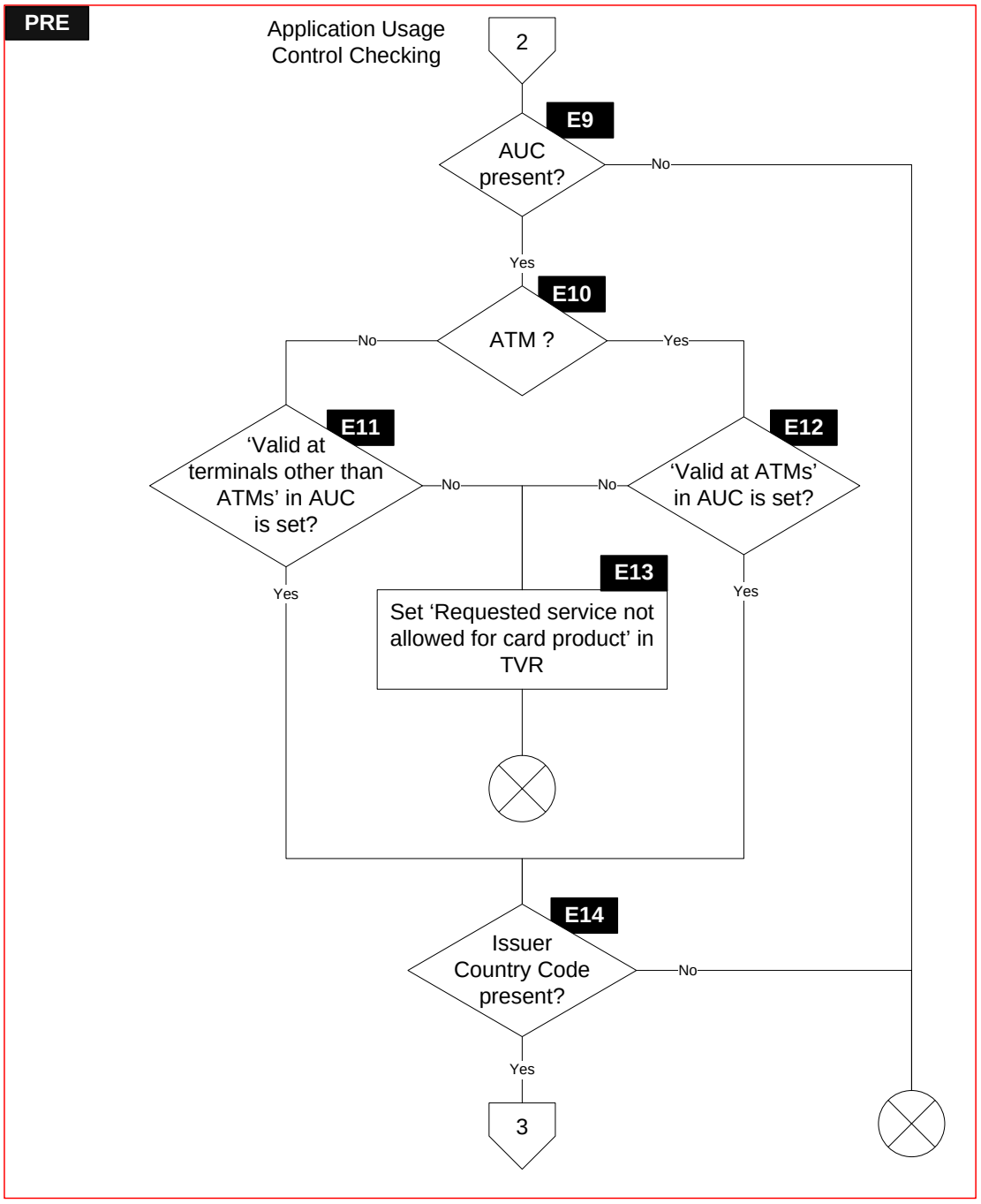
Figure 7.7—Processing Restrictions Flow Diagram



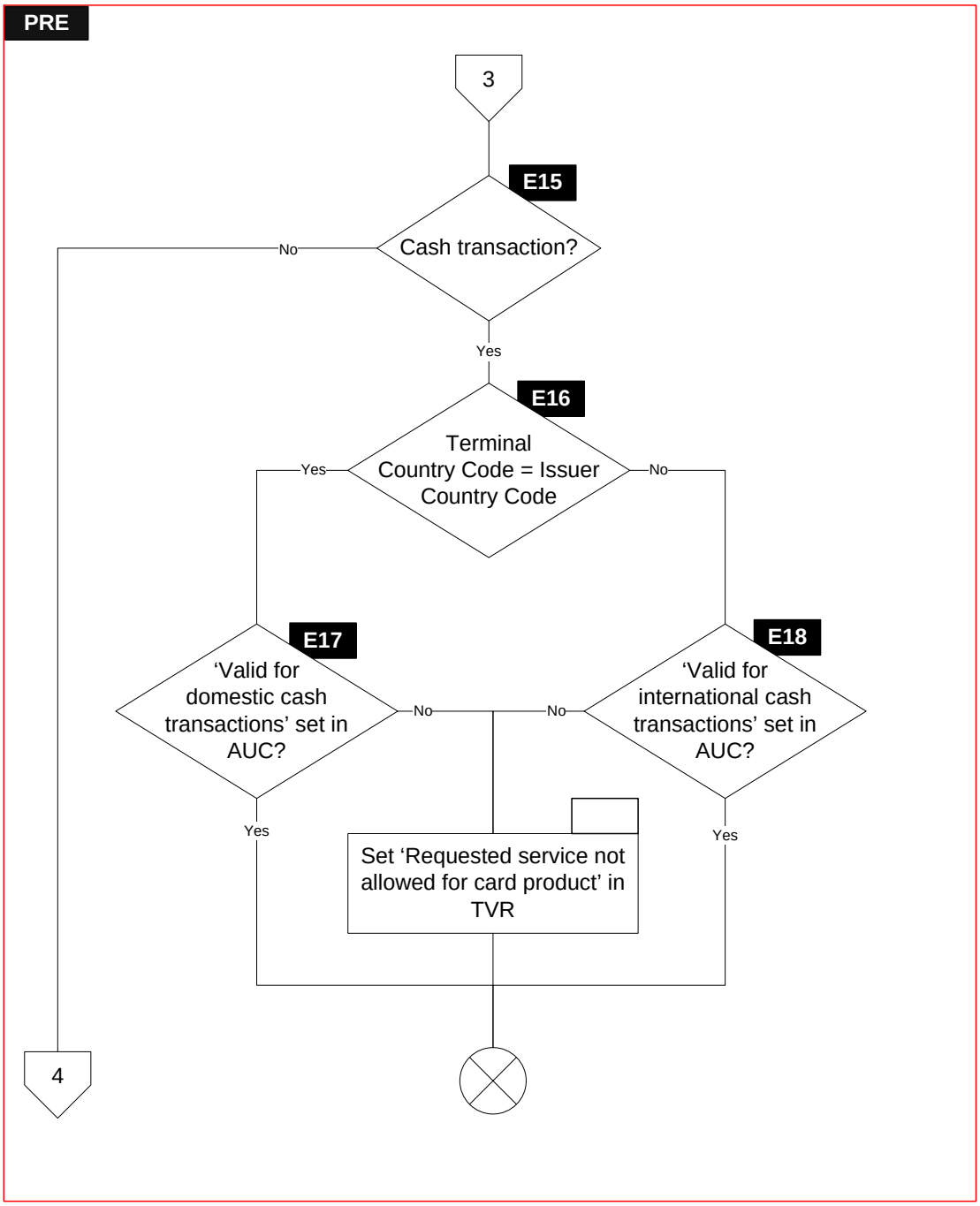
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 411



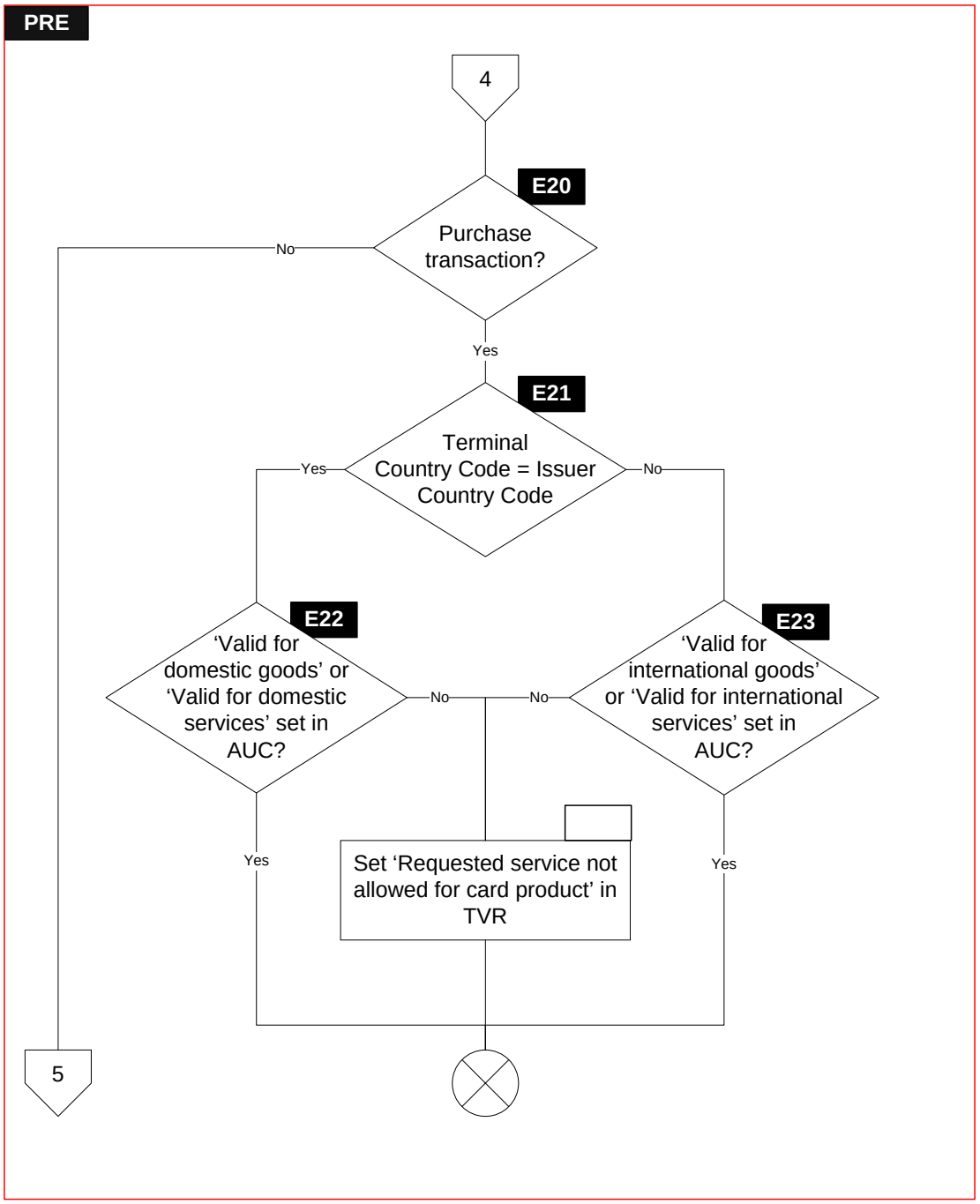
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 412



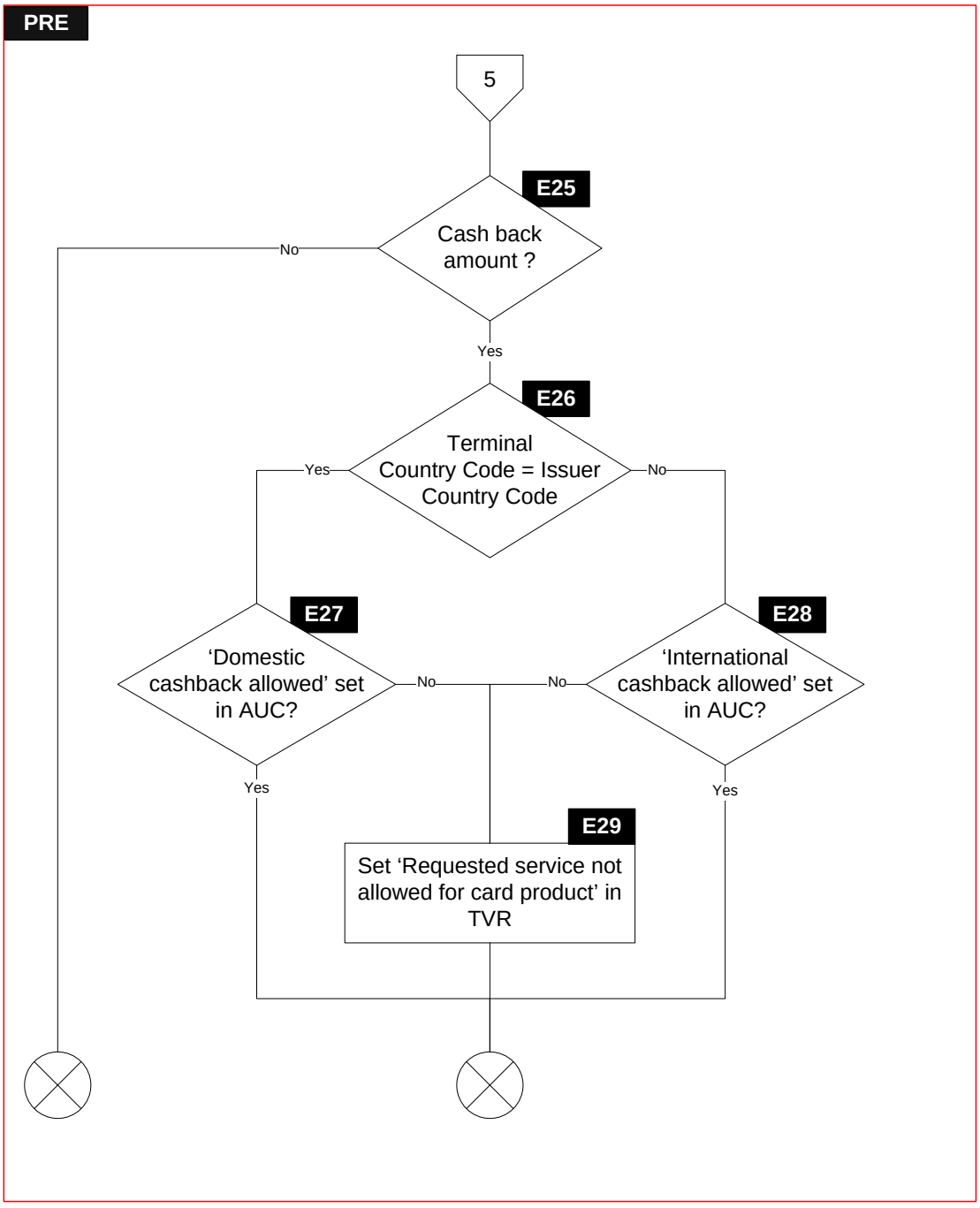
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 414



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 415



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 416

7.7.3 Processing

Application Version Number Checking

PRE.E1

IF [IsNotEmpty(TagOf(*Application Version Number (Card)*))]

THEN

GOTO PRE.E2

ELSE

GOTO PRE.E4

ENDIF

PRE.E2

IF [*Application Version Number (Card) = Application Version Number (Reader)*]

THEN

GOTO PRE.E4

ELSE

GOTO PRE.E3

ENDIF

PRE.E3

SET 'ICC and terminal have different application versions' in *Terminal Verification Results*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 417

Application Effective/Expiration Date Checking

PRE.E4

IF [IsNotEmpty(TagOf(*Application Effective Date*))]
THEN
GOTO PRE.E5
ELSE
GOTO PRE.E7
ENDIF

PRE.E5

IF [*Transaction Date* is before *Application Effective Date*]
THEN
GOTO PRE.E6
ELSE
GOTO PRE.E7
ENDIF

PRE.E6

SET 'Application not yet effective' in *Terminal Verification Results*

PRE.E7

IF [*Transaction Date* is after *Application Expiration Date*]
THEN
GOTO PRE.E8
ELSE
GOTO PRE.E9
ENDIF

PRE.E8

SET 'Expired application' in *Terminal Verification Results*

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 418

Application Usage Control Checking

PRE.E9

IF [IsNotEmpty(TagOf(*Application Usage Control*))]

THEN

GOTO PRE.E10

ELSE

EXIT Processing Restrictions

ENDIF

PRE.E10

IF [((*Terminal Type* = '14') OR (*Terminal Type* = '15') OR (*Terminal Type* = '16')) AND 'Cash' in *Additional Terminal Capabilities* is set]

THEN

GOTO PRE.E12

ELSE

GOTO PRE.E11

ENDIF

PRE.E11

IF ['Valid at terminals other than ATMs' in *Application Usage Control* is set]

THEN

GOTO PRE.E14

ELSE

GOTO PRE.E13

ENDIF

PRE.E12

IF ['Valid at ATMs' in *Application Usage Control* is set]

THEN

GOTO PRE.E14

ELSE

GOTO PRE.E13

ENDIF

PRE.E13

SET 'Requested service not allowed for card product' in *Terminal Verification Results*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 419

PRE.E14

IF [IsNotEmpty(TagOf(*Issuer Country Code*))]
THEN
GOTO PRE.E15
ELSE
EXIT Processing Restrictions
ENDIF

PRE.E15

IF [*Transaction Type* indicates cash transaction]
THEN
GOTO PRE.E16
ELSE
GOTO PRE.E20
ENDIF

PRE.E16

IF [*Terminal Country Code = Issuer Country Code*]
THEN
GOTO PRE.E17
ELSE
GOTO PRE.E18
ENDIF

PRE.E17

IF ['Valid for domestic cash transactions' in *Application Usage Control* is set]
THEN
EXIT Processing Restrictions
ELSE
GOTO PRE.E19
ENDIF

PRE.E18

IF ['Valid for international cash transactions' in *Application Usage Control* is set]
THEN
EXIT Processing Restrictions
ELSE
GOTO PRE.E19
ENDIF

PRE.E19

SET 'Requested service not allowed for card product' in *Terminal Verification Results*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 420

PRE.E20

IF [*Transaction Type* indicates purchase transaction]

THEN

GOTO PRE.E21

ELSE

GOTO PRE.E25

ENDIF

PRE.E21

IF [*Terminal Country Code = Issuer Country Code*]

THEN

GOTO PRE.E22

ELSE

GOTO PRE.E23

ENDIF

PRE.E22

IF ['Valid for domestic goods' in *Application Usage Control* is set OR 'Valid for domestic services' in *Application Usage Control* is set]

THEN

EXIT Processing Restrictions

ELSE

GOTO PRE.E24

ENDIF

PRE.E23

IF ['Valid for international goods' in *Application Usage Control* is set OR 'Valid for international services' in *Application Usage Control* is set]

THEN

EXIT Processing Restrictions

ELSE

GOTO PRE.E24

ENDIF

PRE.E24

SET 'Requested service not allowed for card product' in *Terminal Verification Results*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 421

PRE.E25

IF [IsPresent(TagOf(*Amount, Other (Numeric)*)) AND
(Amount, Other (Numeric) ≠ '000000000000')]

THEN

GOTO PRE.E26

ELSE

EXIT Processing Restrictions

ENDIF

PRE.E26

IF [*Terminal Country Code = Issuer Country Code*]

THEN

GOTO PRE.E27

ELSE

GOTO PRE.E28

ENDIF

PRE.E27

IF ['Domestic cashback allowed' in *Application Usage Control* is set]

THEN

EXIT Processing Restrictions

ELSE

GOTO PRE.E29

ENDIF

PRE.E28

IF ['International cashback allowed' in *Application Usage Control* is set]

THEN

EXIT Processing Restrictions

ELSE

GOTO PRE.E29

ENDIF

PRE.E29

SET 'Requested service not allowed for card product' in *Terminal Verification Results*

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 422

7.8 Procedure – Terminal Action Analysis

7.8.1 Local Variables

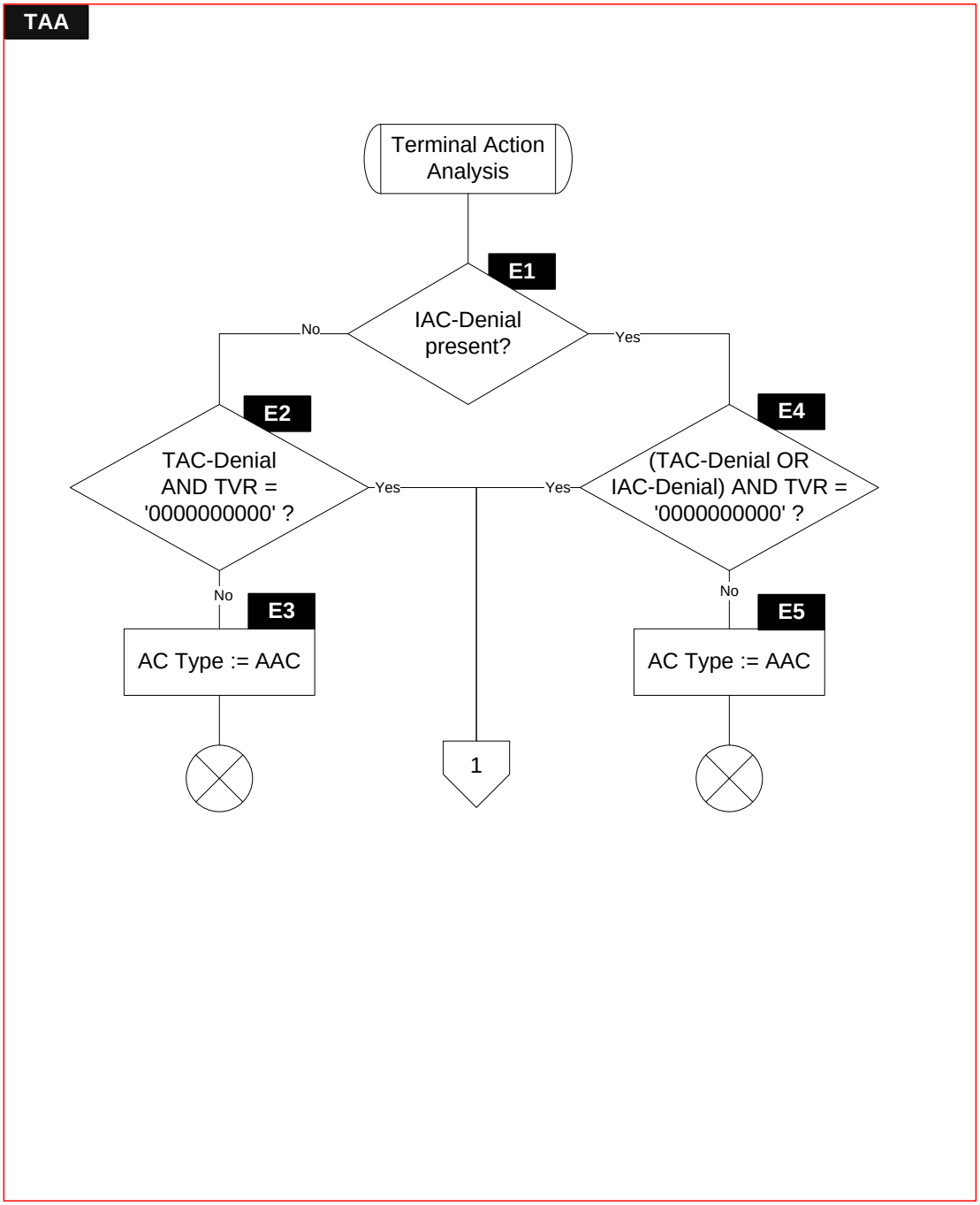
None

7.8.2 Flow Diagram

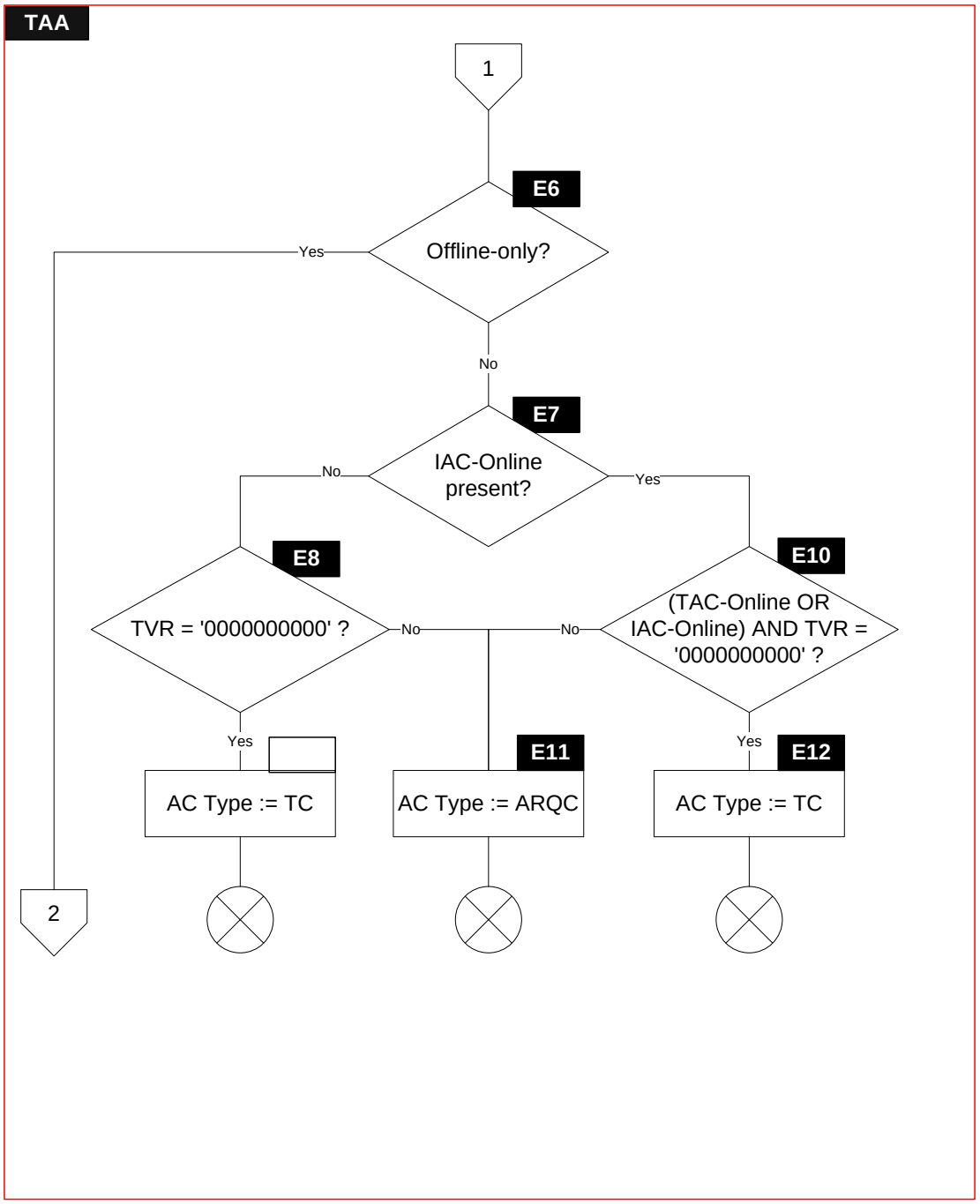
Figure 7.8 shows the flow diagram of the Terminal Action Analysis procedure. Symbols in this diagram are labeled TAA.X.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 423

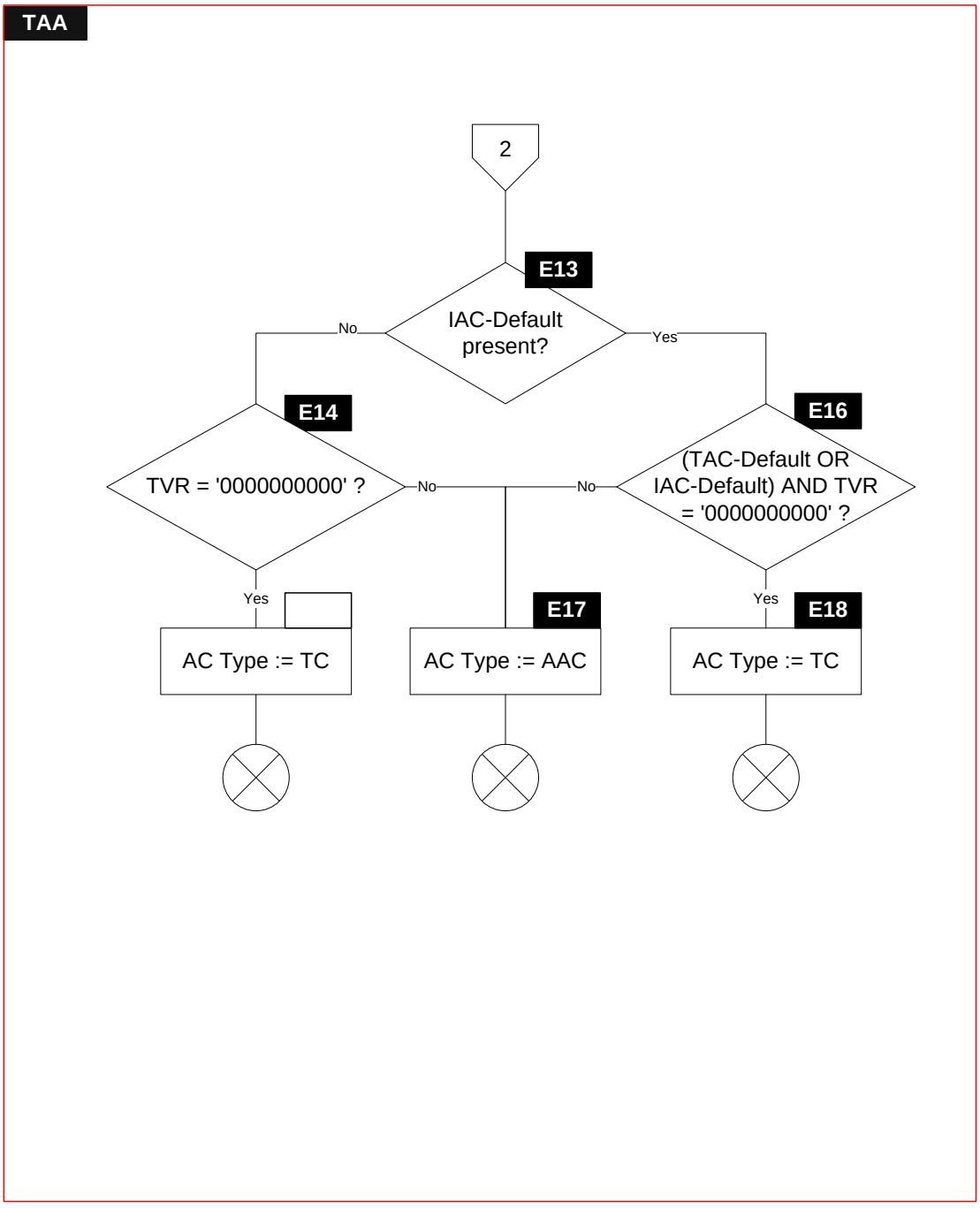
Figure 7.8—Terminal Action Analysis Flow Diagram



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 424



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 425



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 426

7.8.3 Processing

TAA.E1

IF [IsNotEmpty(TagOf(*Issuer Action Code – Denial*))]

THEN

GOTO TAA.E4

ELSE

GOTO TAA.E2

ENDIF

TAA.E2

IF [(*Terminal Action Code – Denial* AND *Terminal Verification Results*) = '0000000000']

THEN

GOTO TAA.E6

ELSE

GOTO TAA.E3

ENDIF

TAA.E3

'AC type' in *AC Type* := AAC

TAA.E4

IF [((*Terminal Action Code – Denial* OR *Issuer Action Code – Denial*) AND *Terminal Verification Results*) = '0000000000']

THEN

GOTO TAA.E6

ELSE

GOTO TAA.E5

ENDIF

TAA.E5

'AC type' in *AC Type* := AAC

TAA.E6

IF [(*Terminal Type* = '23') OR (*Terminal Type* = '26') OR (*Terminal Type* = '36') OR (*Terminal Type* = '13') OR (*Terminal Type* = '16')]

THEN

GOTO TAA.E13

ELSE

GOTO TAA.E7

ENDIF

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 427

TAA.E7

IF [IsNotEmpty(TagOf(*Issuer Action Code – Online*))]
THEN
GOTO TAA.E10
ELSE
GOTO TAA.E8
ENDIF

TAA.E8

IF [*Terminal Verification Results* = '0000000000']
THEN
GOTO TAA.E9
ELSE
GOTO TAA.E11
ENDIF

TAA.E9

'AC type' in *AC Type* := TC

TAA.E10

IF [((*Terminal Action Code – Online* OR *Issuer Action Code – Online*) AND *Terminal Verification Results*) = '0000000000']
THEN
GOTO TAA.E12
ELSE
GOTO TAA.E11
ENDIF

TAA.E11

'AC type' in *AC Type* := ARQC

TAA.E12

'AC type' in *AC Type* := TC

TAA.E13

IF [IsNotEmpty(TagOf(*Issuer Action Code – Default*))]
THEN
GOTO TAA.E16
ELSE
GOTO TAA.E14
ENDIF

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 428

TAA.E14

IF [*Terminal Verification Results* = '0000000000']

THEN

GOTO TAA.E15

ELSE

GOTO TAA.E17

ENDIF

TAA.E15

'AC type' in *AC Type* := TC

TAA.E16

IF [((*Terminal Action Code – Default* OR *Issuer Action Code – Default*) AND *Terminal Verification Results*) = '0000000000']

THEN

GOTO TAA.E18

ELSE

GOTO TAA.E17

ENDIF

TAA.E17

'AC type' in *AC Type* := AAC

TAA.E18

'AC type' in *AC Type* := TC

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 429

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 430

8 Security Algorithms

8.1 Unpredictable Number Generation

Random numbers needed by the Kernel (for example for the *Unpredictable Number* and *Unpredictable Number (Numeric)*) should be generated in a hardware Random Number Generator. Any hardware random number generator must be tested in operation according to [NIST SP 800-22A]. A software random number generator must be seeded from an unpredictable source of data. A software whitening process may be applied to a hardware Number Generator if required. Regardless of the method used, there must be no observable correlation from one set of random data to a preceding set of random data and the Terminal must raise a suitable alarm in the event of a random number generator failure. A software Number Generator may be temporarily used until a hardware Number Generator is reinstated.

All values of random number (for example when used as the 4 byte *Unpredictable Number*) must be equally likely to occur, and the value of the random numbers must be unpredictable from the perspective of an attacker (even given knowledge of previous values). This may be achieved using a Random Number Generator compliant with [ISO 18031:2005] and tested using [NIST SP800-22A].

As generation of random data can be a slow process and transaction performance is important, an implementation may consider generating random data before it is needed, for example in a frequently refreshed buffer of random data. If random data is generated ahead of its use it must not be possible to observe this externally and thus to predict all or part of a number that may be used for a specific transaction.

The random number generator must not be susceptible to external perturbation that might reduce its quality, for example EM fields, glitch or other attacks. It must also not be possible for an attacker to deliberately cause fallback from the hardware RNG to a software one.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 431

8.2 OWHF2

OWHF2 is the DES-based variant of the one-way function for computing the digest. OWHF2 computes an 8-byte output R based on an 8-byte input PD.

Let PL be the index of the byte from *DS ID* containing the least significant digit of the *Application PAN Sequence Number* (i.e. GetLength(TagOf(*Application PAN*)) + 1)

Compute two 6-byte values DSPKL and DSPKR as follows:

DSPKL[i] := ((*DS ID* [i] div 16) x 10 + (*DS ID* [i] mod 16)) x 2, for i = 1, 2, . . . , 6

DSPKR[i] := ((*DS ID* [PL - 6 + i] div 16) x 10 + (*DS ID* [PL - 6 + i] mod 16)) x 2, for i = 1, 2, . . . , 6

Compute an 8 byte value OID as follows:

IF [IsNotEmpty(TagOf(*DS Slot Management Control*)) AND
'Permanent slot type' in *DS Slot Management Control* is set AND
'Volatile slot type' in *DS ODS Info* is set]

THEN

OID := '0000000000000000'

ELSE

OID := *DS Requested Operator ID*

ENDIF

Generate two DES keys KL and KR as follows:

KL[i] := DSPKL[i], for i = 1, 2, . . . , 6

KL[i] := OID [i - 2], for i = 7, . . . , 8

KR[i] := DSPKR[i], for i = 1, 2, . . . , 6

KR[i] := OID[i], for i = 7, . . . , 8

Compute R as follows:

R := DES(KL)[DES-1(KR)[DES(KL)[OID $\oplus$ PD]]] $\oplus$ PD

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 432

8.3 OWHF2AES

OWHF2AES is the AES-based variant of the one-way function for computing the digest. OWHF2AES computes an 8-byte output R based on an 8-byte input C.

Compute an 8 byte value OID as follows:

IF [IsNotEmpty(TagOf(*DS Slot Management Control*)) AND
'Permanent slot type' in *DS Slot Management Control* is set AND
'Volatile slot type' in *DS ODS Info* is set]

THEN

OID := '0000000000000000'

ELSE

OID := *DS Requested Operator ID*

ENDIF

Compute R as follows:

Create a 16-byte message M by concatenating the following data:

M := C || OID

Create a 16-byte AES key K by concatenating the following data:

K := *DS ID* || OID[5..8] || '3F'

Compute a 16-byte value T as follows:

T := AES(K)[M] $\oplus$ M

Compute R as the 8 rightmost byes from T

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 433

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 434

Annex A Data Dictionary

This section contains the data dictionary of the Kernel. It lists all the data objects known to the Kernel other than local working variables.

A.1 Data Objects by Name

A.1.1 Account Type

Tag:	'5F57'
Length:	1
Format:	n 2
Update:	K
Implementations:	EMV, EMV/DE
Description:	Indicates the type of account selected on the Terminal, coded as specified in Annex G of [EMV Book 3].

A.1.2 Acquirer Identifier

Tag:	'9F01'
Length:	6
Format:	n 6-11
Update:	K
Implementations:	EMV, EMV/DE
Description:	Uniquely identifies the acquirer within each payment system.

A.1.3 Active AFL

Tag:	—
Length:	var. up to 252
Format:	b
Update:	K
Implementations:	Always
Description:	Contains the AFL indicating the (remaining) terminal file records to be read from the Card. The *Active AFL* is updated after each successful READ RECORD.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 435

A.1.4 Active Tag

Tag: —
Length: var. up to 2
Format: b
Update: K
Implementations: EMV, EMV/DE
Description: Contains the tag requested by the GET DATA command.

A.1.5 AC Type

Tag: —
Length: 1
Format: b
Update: K
Implementations: EMV, EMV/DE
Description: Contains the AC type to be requested from the Card with the GENERATE AC command. This is the outcome of Terminal Action Analysis.

AC Type			
Byte 1	b8-7	AC type	
			00: AAC
			01: TC
			10: ARQC
			11: RFU
	b6-1	RFU	

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 436

A.1.6 Additional Terminal Capabilities

Tag: '9F40'

Length: 5

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Indicates the data input and output capabilities of the Terminal and Reader.

The *Additional Terminal Capabilities* is coded according to Annex A.3 of [EMV Book 4].

Additional Terminal Capabilities		
Byte 1	b8	Cash
	b7	Goods
	b6	Services
	b5	Cashback
	b4	Inquiry
	b3	Transfer
	b2	Payment
	b1	Administrative
Byte 2	b8	Cash Deposit
	b7-1	RFU
Byte 3	b8	Numeric keys
	b7	Alphabetical and special characters keys
	b6	Command keys
	b5	Function keys
	b4-1	RFU

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 437

Additional Terminal Capabilities		
Byte 4	b8	Print, attendant
	b7	Print, cardholder
	b6	Display, attendant
	b5	Display, cardholder
	b4-3	RFU
	b2	Code table 10
	b1	Code table 9
Byte 5	b8	Code table 8
	b7	Code table 7
	b6	Code table 6
	b5	Code table 5
	b4	Code table 4
	b3	Code table 3
	b2	Code table 2
	b1	Code table 1

A.1.7 Amount, Authorized (Numeric)

Tag: '9F02'

Length: 6

Format: n 12

Update: K/ACT/DET

Implementations: Always

Description: Authorized amount of the transaction (excluding adjustments).

This amount is expressed with implicit decimal point corresponding to the minor unit of currency as defined by [ISO 4217] (for example the six bytes '00 00 00 00 01 23' represent USD 1.23 when the currency code is '840').

If the initial transaction amount, which is provided when the Kernel is started, needs to be replaced with a revised transaction amount, the Terminal must provide it before the chokepoint.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 438

A.1.8 Amount, Other (Numeric)

Tag: '9F03'

Length: 6

Format: n 12

Update: K/ACT/DET

Implementations: EMV, EMV/DE

Description: Secondary amount associated with the transaction representing a cash back amount.

This amount is expressed with implicit decimal point corresponding to the minor unit of currency as defined by [ISO 4217] (for example the 6 bytes '00 00 00 00 01 23' represent GBP 1.23 when the currency code is '826').

A.1.9 Application Capabilities Information

Tag: '9F5D'

Length: 3

Format: b

Update: K/RA

Implementations: EMV, EMV/DE

Description: Lists a number of card features beyond regular payment.

Application Capabilities Information			
Byte 1	b8-5	ACI Version number	
			0000: VERSION 0
			Other values: RFU
	b4-1	Data Storage Version Number	
			0000: DATA STORAGE NOT SUPPORTED
			0001: VERSION 1
			0010: VERSION 2
			Other values: RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 439

Application Capabilities Information			
Byte 2	b8-3	RFU	
	b2	Support for balance reading	
	b1	CDA Indicator	
			0: CDA SUPPORTED AS IN EMV
			1: CDA SUPPORTED OVER TC, ARQC AND AAC
Byte 3	b8-1	SDS Scheme Indicator	
			00000000: Undefined SDS configuration
			00000001: All 10 tags 32 bytes
			00000010: All 10 tags 48 bytes
			00000011: All 10 tags 64 bytes
			00000100: All 10 tags 96 bytes
			00000101: All 10 tags 128 bytes
			00000110: All 10 tags 160 bytes
			00000111: All 10 tags 192 bytes
			00001000: All SDS tags 32 bytes except '9F78' which is 64 bytes
			Other values: RFU

A.1.10 Application Cryptogram

Tag: '9F26'
Length: 8
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: Cryptogram returned by the Card in response to the GENERATE AC or RECOVER AC command.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 440

A.1.11 Application Currency Code

Tag:	'9F42'
Length:	2
Format:	n 3
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Indicates the currency in which the account is managed in accordance with [ISO 4217].

A.1.12 Application Currency Exponent

Tag:	'9F44'
Length:	1
Format:	n 1
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Indicates the implied position of the decimal point from the right of the amount represented in accordance with [ISO 4217].

A.1.13 Application Effective Date

Tag:	'5F25'
Length:	3
Format:	n 6 (YYMMDD)
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Date from which the application may be used. The date is expressed in the YYMMDD format. For MasterCard branded applications if the value of YY ranges from '00' to '49' the date reads 20YYMMDD. If the value of YY ranges from '50' to '99', the date reads 19YYMMDD.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 441

A.1.14 Application Expiration Date

Tag:	'5F24'
Length:	3
Format:	n 6 (YYMMDD)
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Date after which application expires. The date is expressed in the YYMMDD format. For MasterCard applications, if the value of YY ranges from '00' to '49' the date reads 20YYMMDD. If the value of YY ranges from '50' to '99' the date reads 19YYMMDD.

A.1.15 Application File Locator

Tag:	'94'
Length:	var. up to 252
Format:	b
Update:	K/RA
Implementations:	Always
Description:	Indicates the location (SFI range of records) of the Application Elementary Files associated with a particular AID, and read by the Kernel during a transaction. The *Application File Locator* is a list of entries of 4 bytes each. Each entry codes an SFI and a range of records as follows: • The five most significant bits of the first byte indicate the SFI. • The second byte indicates the first (or only) record number to be read for that SFI. • The third byte indicates the last record number to be read for that SFI. When the third byte is greater than the second byte, all the records ranging from the record number in the second byte to and including the record number in the third byte must be read for that SFI. When the third byte is equal to the second byte, only the record number coded in the second byte must be read for that SFI. • The fourth byte indicates the number of records involved in offline data authentication starting with the record number coded in the second byte. The fourth byte may range from zero to the value of the third byte less the value of the second byte plus 1.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 442

A.1.16 Application Interchange Profile

Tag: '82'

Length: 2

Format: b

Update: K/RA

Implementations: Always

Description: Indicates the capabilities of the Card to support specific functions in the application.

The *Application Interchange Profile* is returned in the response message of the GET PROCESSING OPTIONS command. It is coded as specified in Annex C.1 of [EMV Book 3]. This specification extends the definition by allocating:

- RFU bit b2 of byte 1 to indicate that on-device cardholder verification is supported
- RFU bit b8 in byte 2 to indicate support for EMV mode transactions

Application Interchange Profile		
Byte 1	b8	RFU
	b7	SDA Supported
	b6	DDA supported
	b5	Cardholder verification is supported
	b4	Terminal risk management is to be performed
	b3	Issuer Authentication is supported
	b2	On device cardholder verification is supported
	b1	CDA supported
Byte 2	b8	EMV mode is supported
	b7-1	RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 443

A.1.17 Application Label

Tag:	'50'
Length:	var. up to 16
Format:	ans
Update:	K/RA
Implementations:	Always
Description:	Name associated with the AID, in accordance with [ISO/IEC 7816-5].

A.1.18 Application Preferred Name

Tag:	'9F12'
Length:	var. up to 16
Format:	ans
Update:	K/RA
Implementations:	Always
Description:	Preferred name associated with the AID.

A.1.19 Application PAN

Tag:	'5A'
Length:	var. up to 10
Format:	cn var. up to 19
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Valid cardholder account number.

A.1.20 Application PAN Sequence Number

Tag:	'5F34'
Length:	1
Format:	n 2
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Identifies and differentiates cards with the same *Application PAN*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 444

A.1.21 Application Priority Indicator

Tag:	'87'
Length:	1
Format:	b
Update:	K/RA
Implementations:	Always
Description:	Indicates the priority of a given application or group of applications in a directory.

A.1.22 Application Transaction Counter

Tag:	'9F36'
Length:	2
Format:	b
Update:	K/RA
Implementations:	Always
Description:	Counter maintained by the application in the Card (incrementing the *Application Transaction Counter* is managed by the Card).

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 445

A.1.23 Application Usage Control

Tag: '9F07'

Length: 2

Format: b

Update: K/RA

Implementations: EMV, EMV/DE

Description: Indicates the issuer's specified restrictions on the geographic use and services allowed for the application.

The *Application Usage Control* is coded as specified in Annex C.2 of [EMV Book 3].

Application Usage Control		
Byte 1	b8	Valid for domestic cash transactions
	b7	Valid for international cash transactions
	b6	Valid for domestic goods
	b5	Valid for international goods
	b4	Valid for domestic services
	b3	Valid for international services
	b2	Valid at ATMs
	b1	Valid at terminals other than ATMs
Byte 2	b8	Domestic cashback allowed
	b7	International cashback allowed
	b6-1	RFU

A.1.24 Application Version Number (Card)

Tag: '9F08'

Length: 2

Format: b

Update: K/RA

Implementations: EMV, EMV/DE

Description: Version number assigned by the payment system for the application in the Card.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 446

A.1.25 Application Version Number (Reader)

Tag:	'9F09'
Length:	2
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Version number assigned by the payment system for the Kernel application.

A.1.26 Balance Read Before Gen AC

Tag:	'DF8104'
Length:	6
Format:	n 12
Update:	K/ACT/DET
Implementations:	EMV, EMV/DE
Description:	The presence of *Balance Read Before Gen AC* in the TLV Database is an indication to the Kernel to read the offline balance from the Card before the GENERATE AC command. The Kernel stores the offline balance read from the Card in *Balance Read Before Gen AC*.

A.1.27 Balance Read After Gen AC

Tag:	'DF8105'
Length:	6
Format:	n 12
Update:	K/ACT/DET
Implementations:	EMV, EMV/DE
Description:	The presence of *Balance Read After Gen AC* in the TLV Database is an indication to the Kernel to read the offline balance from the Card after the GENERATE AC command. The Kernel stores the offline balance read from the Card in *Balance Read After Gen AC*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 447

A.1.28 CA Public Key Index (Card)

Tag: '8F'
Length: 1
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: Identifies the CA public key in conjunction with the RID.

A.1.29 Card Data Input Capability

Tag: 'DF8117'
Length: 1
Format: b
Update: K
Implementations: EMV, EMV/DE
Description: Indicates the card data input capability of the Terminal and Reader. The *Card Data Input Capability* is coded according to Annex A.2 of [EMV Book 4].

Card Data Input Capability		
Byte 1	b8	Manual key entry
	b7	Magnetic stripe
	b6	IC with contacts
	b5-1	RFU

A.1.30 CDOL1

Tag: '8C'
Length: var. up to 252
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: A data object in the Card that provides the Kernel with a list of data objects that must be passed to the Card in the data field of the GENERATE AC command.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 448

A.1.31 CDOL1 Related Data

Tag:	'DF8107'
Length:	var.
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Command data field of the GENERATE AC command, coded according to *CDOL1*.

A.1.32 Cryptogram Information Data

Tag:	'9F27'
Length:	1
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Indicates the type of cryptogram and the actions to be performed by the Kernel. The *Cryptogram Information Data* is coded according to Table 14 of [EMV Book 3].

A.1.33 CVC3 (Track1)

Tag:	'9F60'
Length:	2
Format:	b
Update:	K/RA
Implementations:	Always
Description:	The *CVC3 (Track1)* is a 2-byte cryptogram returned by the Card in the response to the COMPUTE CRYPTOGRAPHIC CHECKSUM command.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 449

A.1.34 CVC3 (Track2)

Tag: '9F61'

Length: 2

Format: b

Update: K/RA

Implementations: Always

Description: The *CVC3 (Track2)* is a 2-byte cryptogram returned by the Card in the response to the COMPUTE CRYPTOGRAPHIC CHECKSUM command.

A.1.35 CVM Capability – CVM Required

Tag: 'DF8118'

Length: 1

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Indicates the CVM capability of the Terminal and Reader when the transaction amount is greater than the *Reader CVM Required Limit*.

The *CVM Capability – CVM Required* is coded according to Annex A.2 of [EMV Book 4].

CVM Capability – CVM Required		
Byte 1	b8	Plaintext PIN for ICC verification
	b7	Enciphered PIN for online verification
	b6	Signature (paper)
	b5	Enciphered PIN for offline verification
	b4	No CVM required
	b3-1	RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 450

A.1.36 CVM Capability – No CVM Required

Tag: 'DF8119'

Length: 1

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Indicates the CVM capability of the Terminal and Reader when the transaction amount is less than or equal to the *Reader CVM Required Limit*.

The *CVM Capability – No CVM Required* is coded according to Annex A.2 of [EMV Book 4].

CVM Capability – No CVM Required		
Byte 1	b8	Plaintext PIN for ICC verification
	b7	Enciphered PIN for online verification
	b6	Signature (paper)
	b5	Enciphered PIN for offline verification
	b4	No CVM required
	b3-1	RFU

A.1.37 CVM List

Tag: '8E'

Length: var. up to 252

Format: b

Update: K/RA

Implementations: EMV, EMV/DE

Description: Identifies the methods of verification of the cardholder supported by the application.

The *CVM List* is coded as specified in section 10.5 of [EMV Book 3].

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 451

A.1.38 CVM Results

Tag: '9F34'

Length: 3

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Indicates the results of the last CVM performed.

The *CVM Results* are coded as specified in Annex A.4 of [EMV Book 4].

CVM Results		
Byte 1	b8-1	CVM Performed
Byte 2	b8-1	CVM Condition
Byte 3	b8-1	CVM Result

A.1.39 Data Needed

Tag: 'DF8106'

Length: var.

Format: b

Update: K

Implementations: DE, EMV/DE

Description: List of tags included in the DEK signal to request information from the Terminal.

A.1.40 Data Record

Tag: 'FF8105'

Length: var.

Format: b

Update: K

Implementations: Always

Description: The *Data Record* is a list of TLV encoded data objects returned with the *Outcome Parameter Set* on the completion of transaction processing.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 452

A.1.41 Data To Send

Tag:	'FF8104'
Length:	var.
Format:	b
Update:	K
Implementations:	DE, EMV/DE
Description:	List of data objects that contains the accumulated data sent by the Kernel to the Terminal in a DEK signal. These data may correspond to Terminal reading requests, obtained from the Card by means of GET DATA or READ RECORD commands, or may correspond to data that the Kernel posts to the Terminal as part of its own processing.

A.1.42 DD Card (Track1)

Tag:	'DF812A'
Length:	var. up to 56
Format:	ans
Update:	K
Implementations:	Always
Description:	If *Track 1 Data* is present, then *DD Card (Track1)* contains a copy of the discretionary data field of *Track 1 Data* as returned by the Card in the file read using the READ RECORD command during a mag-stripe mode transaction (i.e. without *Unpredictable Number (Numeric)*, *Application Transaction Counter*, *CVC3 (Track1)* and *nUN* included).

A.1.43 DD Card (Track2)

Tag:	'DF812B'
Length:	var. up to 8 bytes
Format:	cn
Update:	K
Implementations:	Always
Description:	*DD Card (Track2)* contains a copy of the discretionary data field of *Track 2 Data* as returned by the Card in the file read using the READ RECORD command during a mag-stripe mode transaction (i.e. without *Unpredictable Number (Numeric)*, *Application Transaction Counter*, *CVC3 (Track2)* and *nUN* included).

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 453

A.1.44 Default UDOL

Tag:	'DF811A'
Length:	3
Format:	b
Update:	K
Implementations:	Always
Description:	The *Default UDOL* is the *UDOL* to be used for constructing the value field of the COMPUTE CRYPTOGRAPHIC CHECKSUM command if the *UDOL* in the Card is not present. The *Default UDOL* must contain as its only entry the tag and length of the *Unpredictable Number (Numeric)* and has the value: '9F6A04'.

A.1.45 DF Name

Tag:	'84'
Length:	5-16
Format:	b
Update:	K/RA
Implementations:	Always
Description:	Identifies the name of the DF, as described in [ISO 7816-4].

A.1.46 Discretionary Data

Tag:	'FF8106'
Length:	var.
Format:	b
Update:	K
Implementations:	Always
Description:	The *Discretionary Data* is a list of Kernel-specific data objects sent to the Terminal as a separate field in the OUT signal.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 454

A.1.47 DRDOL

Tag: '9F51'
Length: var.
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: A data object in the Card that provides the Kernel with a list of data objects that must be passed to the Card in the data field of the RECOVER AC command.

A.1.48 DRDOL Related Data

Tag: 'DF8113'
Length: var.
Format: b
Update: K
Implementations: EMV, EMV/DE
Description: Command data field of the RECOVER AC command, coded according to *DRDOL*.

A.1.49 DS AC Type

Tag: 'DF8108'
Length: 1
Format: b
Update: K/ACT/DET
Implementations: EMV/DE
Description: Contains the AC type indicated by the Terminal for which IDS data must be stored in the Card.

DS AC Type			
Byte 1	b8-7	AC type	
			00: AAC
			01: TC
			10: ARQC
			11: RFU
	b6-1	RFU	

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 455

A.1.50 DS Digest H

Tag:	'DF61'
Length:	8
Format:	b
Update:	K
Implementations:	EMV/DE
Description:	Contains the result of OWHF2(*DS Input (Term)*) or OWHF2AES(*DS Input (Term)*), if *DS Input (Term)* is provided by the Terminal. This data object is to be supplied to the Card with the GENERATE AC command, as per *DSDOL* formatting.

A.1.51 DSDOL

Tag:	'9F5B'
Length:	var.
Format:	b
Update:	K/RA
Implementations:	EMV/DE
Description:	A data object in the Card that provides the Kernel with a list of data objects that must be passed to the Card in the data field of the GENERATE AC command after the *CDOL1 Related Data*. An example of value for *DSDOL* is 'DF6008DF6108DF6201DF6381A0', representing TL*DS Input (Card)* || TL*DS Digest H* || TL*DS ODS Info* || TL*DS ODS Term*. The Kernel must not presume that this is a given though, as the sequence and presence of data objects can vary. The presence of TL *DS ODS Info* is mandated and the processing of the last TL entry in *DSDOL* is different from normal TL processing as described in section 4.1.4.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 456

A.1.52 DS ID

Tag:	'9F5E'
Length:	11
Format:	n
Update:	K/RA
Implementations:	EMV/DE
Description:	Data Storage Identifier constructed as follows: *Application PAN* (without any 'F' padding) || *Application PAN Sequence Number* (+ zero padding)

A.1.53 DS Input (Card)

Tag:	'DF60'
Length:	8
Format:	b
Update:	K/ACT/DET
Implementations:	EMV/DE
Description:	Contains Terminal provided data if permanent data storage in the Card was applicable (*DS Slot Management Control*[8]=1b), remains applicable, or becomes applicable (*DS ODS Info*[8]=1b). Otherwise this data item is a filler to be supplied by the Kernel. The data is forwarded to the Card with the GENERATE AC command, as per *DSDOL* formatting.

A.1.54 DS Input (Term)

Tag:	'DF8109'
Length:	8
Format:	b
Update:	K/ACT/DET
Implementations:	EMV/DE
Description:	Contains Terminal provided data if permanent data storage in the Card was (*DS Slot Management Control*[8]=1b), stays or becomes (*DS ODS Info*[8]=1b) applicable. *DS Input (Term)* is used by the Kernel as input to calculate *DS Digest H*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 457

A.1.55 DS ODS Card

Tag: '9F54'
Length: var. up to 160
Format: b
Update: K/RA
Implementations: EMV/DE
Description: Contains the Card stored operator proprietary data obtained in the response to the GET PROCESSING OPTIONS command.

A.1.56 DS ODS Info

Tag: 'DF62'
Length: 1
Format: b
Update: K/ACT/DET
Implementations: EMV/DE
Description: Contains Terminal provided data to be forwarded to the Card with the GENERATE AC command, as per *DSDOL* formatting.

DS ODS Info		
Byte 1	b8	Permanent slot type
	b7	Volatile slot type
	b6	Low volatility
	b5	RFU
	b4	Decline payment transaction in case of data storage error
	b3-1	RFU

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 458

A.1.57 DS ODS Info For Reader

Tag: 'DF810A'

Length: 1

Format: b

Update: K/ACT/DET

Implementations: EMV/DE

Description: Contains instructions from the Terminal on how to proceed with the transaction if:

- The AC requested by the Terminal does not match the AC proposed by the Kernel
- The update of the slot data has failed

DS ODS Info For Reader		
Byte 1	b8	Usable for TC
	b7	Usable for ARQC
	b6	Usable for AAC
	b5-4	RFU
	b3	Stop if no DS ODS Term
	b2	Stop if write failed
	b1	RFU

A.1.58 DS ODS Term

Tag: 'DF63'

Length: var. up to 160

Format: b

Update: K/ACT/DET

Implementations: EMV/DE

Description: Contains Terminal provided data to be forwarded to the Card with the GENERATE AC command, as per *DSDOL* formatting.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 459

A.1.59 DS Requested Operator ID

Tag: '9F5C'

Length: 8

Format: b

Update: K/ACT/DET

Implementations: EMV/DE

Description: Contains the Terminal determined operator identifier for data storage. It is sent to the Card in the GET PROCESSING OPTIONS command.

A.1.60 DS Slot Availability

Tag: '9F5F'

Length: 1

Format: b

Update: K/RA

Implementations: EMV/DE

Description: Contains the Card indication, obtained in the response to the GET PROCESSING OPTIONS command, about the slot type(s) available for data storage.

DS Slot Availability		
Byte 1	b8	Permanent slot type
	b7	Volatile slot type
	b6-1	RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 460

A.1.61 DS Slot Management Control

Tag: '9F6F'

Length: 1

Format: b

Update: K/RA

Implementations: EMV/DE

Description: Contains the Card indication, obtained in the response to the GET PROCESSING OPTIONS command, about the status of the slot containing data associated to the *DS Requested Operator ID*.

DS Slot Management Control		
Byte 1	b8	Permanent slot type
	b7	Volatile slot type
	b6	Low volatility
	b5	Locked slot
	b4-2	RFU
	b1	Deactivated slot

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 461

A.1.62 DS Summary 1

Tag:	'9F7D'
Length:	8 or 16
Format:	b
Update:	K/RA
Implementations:	EMV/DE
Description:	Contains the Card indication, obtained in the response to the GET PROCESSING OPTIONS command, about either the stored summary associated with *DS ODS Card* if present, or about a default zero-filled summary if *DS ODS Card* is not present and *DS Unpredictable Number* is present.

A.1.63 DS Summary 2

Tag:	—
Length:	8 or 16
Format:	b
Update:	K/RA
Implementations:	EMV/DE
Description:	This data allows the Kernel to check the consistency between *DS Summary 1* and *DS Summary 2*, and so to ensure that *DS ODS Card* is provided by a genuine Card. It is located in the ICC Dynamic Data recovered from the *Signed Dynamic Application Data*.

A.1.64 DS Summary 3

Tag:	—
Length:	8 or 16
Format:	b
Update:	K/RA
Implementations:	EMV/DE
Description:	This data allows the Kernel to check whether the Card has seen the same transaction data as were sent by the Terminal/Kernel. It is located in the ICC Dynamic Data recovered from the *Signed Dynamic Application Data*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 462

A.1.65 DS Summary Status

Tag: 'DF810B'

Length: 1

Format: b

Update: K

Implementations: EMV/DE

Description: Information reported by the Kernel to the Terminal about:

- The consistency between *DS Summary 1* and *DS Summary 2* (successful read)
- The difference between *DS Summary 2* and *DS Summary 3* (successful write)

This data object is part of the *Discretionary Data*.

DS Summary Status		
Byte 1	b8	Successful Read
	b7	Successful Write
	b6-1	RFU

A.1.66 DS Unpredictable Number

Tag: '9F7F'

Length: 4

Format: b

Update: K/RA

Implementations: EMV/DE

Description: Contains the Card challenge (random), obtained in the response to the GET PROCESSING OPTIONS command, to be used by the Terminal in the summary calculation when providing *DS ODS Term*.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 463

A.1.67 DSVN Term

Tag: 'DF810D'

Length: var.

Format: b

Update: K

Implementations: EMV/DE

Description: Data storage support by the Kernel depends on the presence of this data object. If it is absent, data storage is not supported.

Its value is '02' for this version of data storage functionality.

This variable length data item has an initial byte that defines the maximum version number supported by the Terminal and a variable number of subsequent bytes that define how the Terminal supports earlier versions of the specification. As this is the first version, no legacy support is described and no additional bytes are present.

A.1.68 Error Indication

Tag: 'DF8115'

Length: 6

Format: b

Update: K

Implementations: Always

Description: Contains information regarding the nature of the error that has been encountered during the transaction processing.

This data object is part of the *Discretionary Data*.

Data Field	Length	Format
L1	1	b (see below)
L2	1	b (see below)
L3	1	b (see below)
SW12	2	b
Msg On Error	1	b (see Message Identifier as defined in A.1.176)

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 464

L1			
Byte 1	b8-1	L1	
			00000000: OK
			00000001: TIME OUT ERROR
			00000010: TRANSMISSION ERROR
			00000011: PROTOCOL ERROR
			Other values: RFU

L2			
Byte 1	b8-1	L2	
			00000000: OK
			00000001: CARD DATA MISSING
			00000010: CAM FAILED
			00000011: STATUS BYTES
			00000100: PARSING ERROR
			00000101: MAX LIMIT EXCEEDED
			00000110: CARD DATA ERROR
			00000111: MAGSTRIPE NOT SUPPORTED
			00001000: NO PPSE
			00001001: PPSE FAULT
			00001010: EMPTY CANDIDATE LIST
			00001011: IDS READ ERROR
			00001100: IDS WRITE ERROR
			00001101: IDS DATA ERROR
			00001110: IDS NO MATCHING AC
			Other values: RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 465

L3			
Byte 1	b8-1	L3	
			00000000: OK
			00000001: TIME OUT
			00000010: STOP
			00000011: AMOUNT NOT PRESENT
			Other values: RFU

A.1.69 Failed MS Cntr

Tag: —
Length: 1
Format: b
Update: K
Implementations: Always
Description: Counts the number of failed consecutive mag-stripe mode transactions. The *Failed MS Cntr* is stored in the scratch pad provided to the Kernel at instantiation.

A.1.70 File Control Information Issuer Discretionary Data

Tag: 'BF0C'
Length: var. up to 222
Format: b
Update: K/RA
Implementations: Always
Description: Issuer discretionary part of the *File Control Information Proprietary Template*.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 466

A.1.71 File Control Information Proprietary Template

Tag:	'A5'
Length:	var.
Format:	b
Update:	K/RA
Implementations:	Always
Description:	Identifies the data object proprietary to this specification in the *File Control Information Template*, in accordance with [ISO 7816-4].

A.1.72 File Control Information Template

Tag:	'6F'
Length:	var. up to 252
Format:	b
Update:	K/RA
Implementations:	Always
Description:	Identifies the *File Control Information Template*, in accordance with [ISO 7816-4].

A.1.73 ICC Dynamic Number

Tag:	'9F4C'
Length:	var. 2-8
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Time-variant number generated by the Card, to be captured by the Kernel.

A.1.74 ICC Public Key Certificate

Tag:	'9F46'
Length:	NI
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	ICC public key certified by the issuer.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 467

A.1.75 ICC Public Key Exponent

Tag: '9F47'
Length: 1 or 3
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: Exponent used for the verification of the *Signed Dynamic Application Data*.

A.1.76 ICC Public Key Remainder

Tag: '9F48'
Length: NIC-NI + 42
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: Remaining digits of the modulus of the ICC public key.

A.1.77 IDS Status

Tag: 'DF8128'
Length: 1
Format: b
Update: K
Implementations: EMV/DE
Description: Indicates if the transaction performs an IDS read and/or write.

IDS Status		
Byte 1	b8	Read
	b7	Write
	b6-1	RFU

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 468

A.1.78 Interface Device Serial Number

Tag:	'9F1E'
Length:	8
Format:	an
Update:	K
Implementations:	EMV, EMV/DE
Description:	Unique and permanent serial number assigned to the IFD by the manufacturer.

A.1.79 Issuer Action Code – Default

Tag:	'9F0D'
Length:	5
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Specifies the issuer's conditions that cause a transaction to be rejected on an offline only Terminal.

A.1.80 Issuer Action Code – Denial

Tag:	'9F0E'
Length:	5
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Specifies the issuer's conditions that cause the denial of a transaction without any attempt to go online.

A.1.81 Issuer Action Code – Online

Tag:	'9F0F'
Length:	5
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Specifies the issuer's conditions that cause a transaction to be transmitted online on an online capable Terminal.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 469

A.1.82 Issuer Application Data

Tag:	'9F10'
Length:	var. up to 32
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Contains proprietary application data for transmission to the issuer in an online transaction.

A.1.83 Issuer Code Table Index

Tag:	'9F11'
Length:	1
Format:	n 2
Update:	K/RA
Implementations:	Always
Description:	Indicates the code table, in accordance with [ISO/IEC 8859], for displaying the *Application Preferred Name*. The *Issuer Code Table Index* is coded as specified in Annex C.4 of [EMV Book 3].

A.1.84 Issuer Country Code

Tag:	'5F28'
Length:	2
Format:	n 3
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Indicates the country of the issuer, in accordance with [ISO 3166-1].

A.1.85 Issuer Public Key Certificate

Tag:	'90'
Length:	NCA
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Issuer public key certified by a certification authority.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 470

A.1.86 Issuer Public Key Exponent

Tag: '9F32'
Length: 1 or 3
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: Exponent used for the recovery and verification of the *ICC Public Key Certificate*.

A.1.87 Issuer Public Key Remainder

Tag: '92'
Length: NI - NCA + 36
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: Remaining digits of the modulus of the Issuer public key.

A.1.88 Kernel Configuration

Tag: 'DF811B'
Length: 1
Format: b
Update: K
Implementations: EMV, EMV/DE
Description: Indicates the Kernel configuration options.

Kernel Configuration		
Byte 1	b8	Only EMV mode transactions supported
	b7	Only mag-stripe mode transactions supported
	b6-1	RFU

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 471

A.1.89 Kernel ID

Tag:	'DF810C'
Length:	1
Format:	b
Update:	K
Implementations:	Always
Description:	Contains a value that uniquely identifies each Kernel. There is one occurrence of this data object for each Kernel in the Reader.

A.1.90 Language Preference

Tag:	'5F2D'
Length:	2-8
Format:	an
Update:	K/RA
Implementations:	Always
Description:	1-4 languages stored in order of preference, each represented by two alphabetical characters, in accordance with [ISO 639-1].

A.1.91 Log Entry

Tag:	'9F4D'
Length:	2
Format:	b
Update:	K/RA
Implementations:	EMV, EMV/DE
Description:	Provides the SFI of the Transaction Log file and its number of records.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 472

A.1.92 Mag-stripe Application Version Number (Reader)

Tag: '9F6D'
Length: 2
Format: b
Update: K
Implementations: Always
Description: Version number assigned by the payment system for the specific mag-stripe mode functionality of the Kernel.

A.1.93 Mag-stripe CVM Capability – CVM Required

Tag: 'DF811E'
Length: 1
Format: b
Update: K
Implementations: Always
Description: Indicates the CVM capability of the Terminal/Reader in the case of a mag-stripe mode transaction when the *Amount, Authorized (Numeric)* is greater than the *Reader CVM Required Limit*.

Mag-stripe CVM Capability – CVM Required			
Byte 1	b8-5	CVM	
			0000: NO CVM
			0001: OBTAIN SIGNATURE
			0010: ONLINE PIN
			1111: N/A
			Other values: RFU
	b4-1	RFU	

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 473

A.1.94 Mag-stripe CVM Capability – No CVM Required

Tag: 'DF812C'

Length: 1

Format: b

Update: K

Implementations: Always

Description: Indicates the CVM capability of the Terminal/Reader in the case of a mag-stripe mode transaction when the *Amount, Authorized (Numeric)* is less than or equal to the *Reader CVM Required Limit*.

Mag-stripe CVM Capability – No CVM Required			
Byte 1	b8-5	CVM	
			0000: NO CVM
			0001: OBTAIN SIGNATURE
			0010: ONLINE PIN
			1111: N/A
			Other values: RFU
	b4-1	RFU	

A.1.95 Max Lifetime of Torn Transaction Log Record

Tag: 'DF811C'

Length: 2

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Maximum time, in seconds, that a record can remain in the Torn Transaction Log.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 474

A.1.96 Max Number of Torn Transaction Log Records

Tag:	'DF811D'
Length:	1
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Indicates the maximum number of records that can be stored in the Torn Transaction Log.

A.1.97 Merchant Category Code

Tag:	'9F15'
Length:	2
Format:	n 4
Update:	K
Implementations:	EMV, EMV/DE
Description:	Classifies the type of business being done by the merchant, represented in accordance with [ISO 8583:1993] for Card Acceptor Business Code.

A.1.98 Merchant Custom Data

Tag:	'9F7C'
Length:	20
Format:	b
Update:	K/ACT/DET
Implementations:	EMV, EMV/DE
Description:	Proprietary merchant data that may be requested by the Card.

A.1.99 Merchant Identifier

Tag:	'9F16'
Length:	15
Format:	ans 15
Update:	K
Implementations:	EMV, EMV/DE
Description:	When concatenated with the *Acquirer Identifier*, uniquely identifies a given merchant.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 475

A.1.100 Merchant Name and Location

Tag: '9F4E'
Length: var.
Format: ans
Update: K
Implementations: EMV, EMV/DE
Description: Indicates the name and location of the merchant.

A.1.101 Message Hold Time

Tag: 'DF812D'
Length: 3
Format: n 6
Update: K
Implementations: Always
Description: Indicates the default delay for the processing of the next MSG signal. The *Message Hold Time* is an integer in units of 100ms.

A.1.102 Mobile Support Indicator

Tag: '9F55'
Length: 1
Format: b
Update: K
Implementations: Always
Description: The *Mobile Support Indicator* informs the Card that the Kernel supports extensions for mobile and requires on-device cardholder verification.

Mobile Support Indicator		
Byte 1	b8-3	RFU
	b2	Offline PIN Required
	b1	Mobile supported

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 476

A.1.103 NATC(Track1)

Tag:	'9F64'
Length:	1
Format:	b
Update:	K/RA
Implementations:	Always
Description:	The value of *NATC(Track1)* represents the number of digits of the *Application Transaction Counter* to be included in the discretionary data field of *Track 1 Data*.

A.1.104 NATC(Track2)

Tag:	'9F67'
Length:	1
Format:	b
Update:	K/RA
Implementations:	Always
Description:	The value of *NATC(Track2)* represents the number of digits of the *Application Transaction Counter* to be included in the discretionary data field of *Track 2 Data*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 477

A.1.105 Next Cmd

Tag: —

Length: 1

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: An internal working variable used to indicate the C-APDU that is currently being processed by the Card.

Next Cmd			
Byte 1	b8-7	Next Cmd	
			00: READ RECORD
			01: GET DATA
			10: NONE
			11: RFU
	b6-1	RFU	

A.1.106 nUN

Tag: —

Length: 1

Format: b

Update: K

Implementations: Always

Description: Number of non-zero bits in *PUNATC(Track2) – NATC(Track2)*

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

A.1.107 ODA Status

Tag: —

Length: 1

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Indicates if CDA is to be performed for the transaction in progress.

ODA Status		
Byte 1	b8	CDA
	b7-1	RFU

A.1.108 Offline Accumulator Balance

Tag: '9F50'

Length: 6

Format: n 12

Update: K/RA

Implementations: EMV, EMV/DE

Description: Represents the amount of offline spending available in the Card.

The *Offline Accumulator Balance* is retrievable by the GET DATA command, if allowed by the Card configuration.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 479

A.1.109 Outcome Parameter Set

Tag: 'DF8129'

Length: 8

Format: b

Update: K

Implementations: Always

Description: This data object is used to indicate to the Terminal the outcome of the transaction processing by the Kernel. Its value is an accumulation of results about applicable parts of the transaction.

Outcome Parameter Set			
Byte 1	b8-5	Status	
			0001: APPROVED
			0010: DECLINED
			0011: ONLINE REQUEST
			0100: END APPLICATION
			0101: SELECT NEXT
			0111: TRY AGAIN
			1111: N/A
			Other values: RFU
	b4-1	RFU	
Byte 2	b8-5	Start	
			0000: A
			0001: B
			0010: C
			0011: D
			1111: N/A
			Other values: RFU
	b4-1	RFU	

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 480

Outcome Parameter Set			
Byte 3	b8-5	Online Response Data	
			1111: N/A
			Other values: RFU
	b4-1	RFU	
Byte 4	b8-5	CVM	
			0000: NO CVM
			0001: OBTAIN SIGNATURE
			0010: ONLINE PIN
			0011: CONFIRMATION CODE VERIFIED
			1111: N/A
			Other values: RFU
	b4-1	RFU	
Byte 5	b8	UI Request on Outcome Present	
	b7	UI Request on Restart Present	
	b6	Data Record Present	
	b5	Discretionary Data Present	
	b4	Receipt	
			0: N/A
			1: YES
	b3-1	RFU	
Byte 6	b8-5	Alternate Interface Preference	
			1111: N/A
			Other values: RFU
	b4-1	RFU	
Byte 7	b8-1	Field Off Request	
Byte 8	b8-1	Removal Timeout	

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 481

A.1.110 PCVC3(Track1)

Tag:	'9F62'
Length:	6
Format:	b
Update:	K/RA
Implementations:	Always
Description:	*PCVC3(Track1)* indicates to the Kernel the positions in the discretionary data field of the *Track 1 Data* where the *CVC3 (Track1)* digits must be copied.

A.1.111 PCVC3(Track2)

Tag:	'9F65'
Length:	2
Format:	b
Update:	K/RA
Implementations:	Always
Description:	*PCVC3(Track2)* indicates to the Kernel the positions in the discretionary data field of the *Track 2 Data* where the *CVC3 (Track2)* digits must be copied.

A.1.112 PDOL

Tag:	'9F38'
Length:	var.
Format:	b
Update:	K/RA
Implementations:	Always
Description:	A data object in the Card that provides the Kernel with a list of data objects that must be passed to the Card in the GET PROCESSING OPTIONS command.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 482

A.1.113 PDOL Related Data

Tag: 'DF8111'
Length: var.
Format: b
Update: K
Implementations: Always
Description: Command data field of the GET PROCESSING OPTIONS command, coded according to *PDOL*.

A.1.114 POS Cardholder Interaction Information

Tag: 'DF4B'
Length: 3
Format: b
Update: K/RA
Implementations: Always
Description: The *POS Cardholder Interaction Information* informs the Kernel about the indicators set in the mobile phone that may influence the action flow of the merchant and cardholder.

POS Cardholder Interaction Information		
Byte 1	b8-1	Version Number
Byte 2	b8-6	RFU
	b5	Offline PIN verification successful
	b4	Context is conflicting
	b3	Offline change PIN required
	b2	ACK required
	b1	PIN required
Byte 3	b8-1	RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 483

A.1.115 Post-Gen AC Put Data Status

Tag: 'DF810E'

Length: 1

Format: b

Update: K

Implementations: EMV/DE

Description: Information reported by the Kernel to the Terminal, about the processing of PUT DATA commands after processing the GENERATE AC command.

Possible values are 'completed' or 'not completed'. In the latter case, this status is not specific about which of the PUT DATA commands failed, or about how many of these commands have failed or succeeded.

This data object is part of the *Discretionary Data* provided by the Kernel to the Terminal.

Post-Gen AC Put Data Status		
Byte 1	b8	Completed
	b7-1	RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

A.1.116 Pre-Gen AC Put Data Status

Tag: 'DF810F'

Length: 1

Format: b

Update: K

Implementations: EMV/DE

Description: Information reported by the Kernel to the Terminal, about the processing of PUT DATA commands before sending the GENERATE AC command.

Possible values are 'completed' or 'not completed'. In the latter case, this status is not specific about which of the PUT DATA commands failed, or about how many of these commands have failed or succeeded.

This data object is part of the *Discretionary Data* provided by the Kernel to the Terminal.

Pre-Gen AC Put Data Status		
Byte 1	b8	Completed
	b7-1	RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 485

A.1.117 Proceed To First Write Flag

Tag:	'DF8110'
Length:	1
Format:	b
Update:	K/ACT/DET
Implementations:	DE, EMV/DE
Description:	Indicates that the Terminal will send no more requests to read data other than as indicated in *Tags To Read*. This data item indicates the point at which the Kernel shifts from the Card reading phase to the Card writing phase. If *Proceed To First Write Flag* is not present or is present with non zero length and value different from zero, then the Kernel proceeds without waiting. If *Proceed To First Write Flag* is present with zero length, then the Kernel sends a DEK signal to the Terminal and waits for the DET signal. If *Proceed To First Write Flag* is present with non zero length and value equal to zero, then the Kernel waits for a DET signal from the Terminal without sending a DEK signal.

A.1.118 Protected Data Envelope 1

Tag:	'9F70'
Length:	var. 32-192
Format:	b
Update:	K/RA/ACT/DET
Implementations:	EMV/DE
Description:	The Protected Data Envelopes contain proprietary information from the issuer, payment system or third party. The Protected Data Envelope can be retrieved with the GET DATA command and updated with the PUT DATA command with secure messaging. The length is implementation specific and must have a value between 32 and 192 bytes.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 486

A.1.119 Protected Data Envelope 2

Tag: '9F71'
Length: var. 32-192
Format: b
Update: K/RA/ACT/DET
Implementations: EMV/DE
Description: Same as *Protected Data Envelope 1*.

A.1.120 Protected Data Envelope 3

Tag: '9F72'
Length: var. 32-192
Format: b
Update: K/RA/ACT/DET
Implementations: EMV/DE
Description: Same as *Protected Data Envelope 1*.

A.1.121 Protected Data Envelope 4

Tag: '9F73'
Length: var. 32-192
Format: b
Update: K/RA/ACT/DET
Implementations: EMV/DE
Description: Same as *Protected Data Envelope 1*.

A.1.122 Protected Data Envelope 5

Tag: '9F74'
Length: var. 32-192
Format: b
Update: K/RA/ACT/DET
Implementations: EMV/DE
Description: Same as *Protected Data Envelope 1*.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 487

A.1.123 PUNATC(Track1)

Tag:	'9F63'
Length:	6
Format:	b
Update:	K/RA
Implementations:	Always
Description:	*PUNATC(Track1)* indicates to the Kernel the positions in the discretionary data field of *Track 1 Data* where the *Unpredictable Number (Numeric)* digits and *Application Transaction Counter* digits have to be copied.

A.1.124 PUNATC(Track2)

Tag:	'9F66'
Length:	2
Format:	b
Update:	K/RA
Implementations:	Always
Description:	*PUNATC(Track2)* indicates to the Kernel the positions in the discretionary data field of *Track 2 Data* where the *Unpredictable Number (Numeric)* digits and *Application Transaction Counter* digits have to be copied.

A.1.125 Reader Contactless Floor Limit

Tag:	'DF8123'
Length:	6
Format:	n 12
Update:	K
Implementations:	EMV, EMV/DE
Description:	Indicates the transaction amount above which transactions must be authorized online.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 488

A.1.126 Reader Contactless Transaction Limit

Tag:	—
Length:	6
Format:	n 12
Update:	K
Implementations:	Always
Description:	Indicates the transaction amount above which the transaction is not allowed. This data object is instantiated with *Reader Contactless Transaction Limit (On-device CVM)* if on-device cardholder verification is supported by the Card and with *Reader Contactless Transaction Limit (No On-device CVM)* otherwise.

A.1.127 Reader Contactless Transaction Limit (No On-device CVM)

Tag:	'DF8124'
Length:	6
Format:	n 12
Update:	K
Implementations:	Always
Description:	Indicates the transaction amount above which the transaction is not allowed, when on-device cardholder verification is not supported.

A.1.128 Reader Contactless Transaction Limit (On-device CVM)

Tag:	'DF8125'
Length:	6
Format:	n 12
Update:	K
Implementations:	Always
Description:	Indicates the transaction amount above which the transaction is not allowed, when on-device cardholder verification is supported.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 489

A.1.129 Reader CVM Required Limit

Tag: 'DF8126'

Length: 6

Format: n 12

Update: K

Implementations: Always

Description: Indicates the transaction amount above which the Kernel instantiates the CVM capabilities field in *Terminal Capabilities* with *CVM Capability – CVM Required*.

A.1.130 Reference Control Parameter

Tag: 'DF8114'

Length: 1

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Working variable to store the reference control parameter of the GENERATE AC command.

Reference Control Parameter			
Byte 1	b8-7	AC type	
			00: AAC
			01: TC
			10: ARQC
			11: RFU
	b6	RFU	
	b5	CDA signature requested	
	b4-1	RFU	

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 490

A.1.131 Response Message Template Format 1

Tag: '80'
Length: var.
Format: b
Update: K/RA
Implementations: Always
Description: Contains the data objects (without tags and lengths) returned by the Card in response to a command.

A.1.132 Response Message Template Format 2

Tag: '77'
Length: var.
Format: b
Update: K/RA
Implementations: Always
Description: Contains the data objects (with tags and lengths) returned by the Card in response to a command.

A.1.133 Security Capability

Tag: 'DF811F'
Length: 1
Format: b
Update: K
Implementations: EMV, EMV/DE
Description: Indicates the security capability of the Kernel.

The *Security Capability* is coded according to Annex A.2 of [EMV Book 4].

Security Capability		
Byte 1	b8	SDA
	b7	DDA
	b6	Card capture
	b5	RFU
	b4	CDA
	b3-1	RFU

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 491

A.1.134 Service Code

Tag: '5F30'
Length: 2
Format: n 3
Update: K/RA
Implementations: EMV, EMV/DE
Description: Service code as defined in *Track 1 Data* and *Track 2 Data*.

A.1.135 Signed Dynamic Application Data

Tag: '9F4B'
Length: NIC
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: Digital signature on critical application parameters for CDA.

A.1.136 Static Data Authentication Tag List

Tag: '9F4A'
Length: var.
Format: b
Update: K/RA
Implementations: EMV, EMV/DE
Description: List of tags of primitive data objects defined in this specification for which the value fields must be included in the *Signed Dynamic Application Data*.

A.1.137 Static Data To Be Authenticated

Tag: —
Length: var. up to 2048
Format: b
Update: K
Implementations: EMV, EMV/DE
Description: Buffer used to concatenate records that are involved in offline data authentication.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 492

A.1.138 Tags To Read

Tag:	'DF8112'
Length:	var.
Format:	b
Update:	K/ACT/DET
Implementations:	DE, EMV/DE
Description:	List of tags indicating the data the Terminal has requested to be read. This data item is present if the Terminal wants any data back from the Card before the *Data Record*. This could be in the context of SDS, or for non data storage usage reasons, for example the PAN. This data item may contain configured data. This data object may be provided several times by the Terminal. Therefore, the values of each of these tags must be accumulated in the *Tags To Read Yet* buffer.

A.1.139 Tags To Read Yet

Tag:	—
Length:	var.
Format:	b
Update:	K
Implementations:	DE, EMV/DE
Description:	List of tags that contains the accumulated Terminal data reading requests received in *Tags To Read*. Requested data objects that are sent to the Terminal are spooled from this buffer. *Tags To Read Yet* is initiated when the Kernel is started with *Tags To Read* if present in the ACT signal. This list can be augmented with Terminal requested data items provided during Kernel processing in DET signals. The Kernel sends the requested data objects to the Terminal with the DEK signal in *Data To Send*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 493

A.1.140 Tags To Write After Gen AC

Tag:	'FF8103'
Length:	var.
Format:	b
Update:	K/ACT/DET
Implementations:	EMV/DE
Description:	Contains the Terminal data writing requests to be sent to the Card after processing the GENERATE AC command or the RECOVER AC command. The value of this data object is composed of a series of TLVs. This data object may be provided several times by the Terminal in a DET signal. Therefore, these values must be accumulated in *Tags To Write Yet After Gen AC*.

A.1.141 Tags To Write Before Gen AC

Tag:	'FF8102'
Length:	var.
Format:	b
Update:	K/ACT/DET
Implementations:	EMV/DE
Description:	List of data objects indicating the Terminal data writing requests to be sent to the Card before processing the GENERATE AC command or the RECOVER AC command. This data object may be provided several times by the Terminal in a DET signal. Therefore, these values must be accumulated in *Tags To Write Yet Before Gen AC* buffer.

A.1.142 Tags To Write Yet After Gen AC

Tag:	—
Length:	var.
Format:	b
Update:	K
Implementations:	EMV/DE
Description:	List of data objects that contains the accumulated Terminal data writing requests received in *Tags To Write After Gen AC*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 494

A.1.143 Tags To Write Yet Before Gen AC

Tag:	—
Length:	var.
Format:	b
Update:	K
Implementations:	EMV/DE
Description:	List of data objects that contains the accumulated Terminal data writing requests received in *Tags To Write Before Gen AC*.

A.1.144 Terminal Action Code – Default

Tag:	'DF8120'
Length:	5
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Specifies the acquirer's conditions that cause a transaction to be rejected on an offline only Terminal.

A.1.145 Terminal Action Code – Denial

Tag:	'DF8121'
Length:	5
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Specifies the acquirer's conditions that cause the denial of a transaction without attempting to go online.

A.1.146 Terminal Action Code – Online

Tag:	'DF8122'
Length:	5
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Specifies the acquirer's conditions that cause a transaction to be transmitted online on an online capable Terminal.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 495

A.1.147 Terminal Capabilities

Tag: '9F33'

Length: 3

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Indicates the card data input, CVM, and security capabilities of the Terminal and Reader. The CVM capability (Byte 2) is instantiated with values depending on the transaction amount.

The Terminal Capabilities is coded according to Annex A.2 of [EMV Book 4].

Terminal Capabilities		
Byte 1	b8	Manual key entry
	b7	Magnetic stripe
	b6	IC with contacts
	b5-1	RFU
Byte 2	b8	Plaintext PIN for ICC verification
	b7	Enciphered PIN for online verification
	b6	Signature (paper)
	b5	Enciphered PIN for offline verification
	b4	No CVM required
	b3-1	RFU
Byte 3	b8	SDA
	b7	DDA
	b6	Card capture
	b5	RFU
	b4	CDA
	b3-1	RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 496

A.1.148 Terminal Country Code

Tag:	'9F1A'
Length:	2
Format:	n 3
Update:	K
Implementations:	Always
Description:	Indicates the country of the Terminal, represented in accordance with [ISO 3166-1].

A.1.149 Terminal Identification

Tag:	'9F1C'
Length:	8
Format:	an 8
Update:	K
Implementations:	Always
Description:	Designates the unique location of the Terminal.

A.1.150 Terminal Type

Tag:	'9F35'
Length:	1
Format:	n 2
Update:	K
Implementations:	EMV, EMV/DE
Description:	Indicates the environment of the Terminal, its communications capability, and its operational control. The *Terminal Type* is coded according to Annex A.1 of [EMV Book 4].

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 497

A.1.151 Terminal Verification Results

Tag: '95'

Length: 5

Format: b

Update: K

Implementations: EMV, EMV/DE

Description: Status of the different functions from the Terminal perspective.

The *Terminal Verification Results* is coded according to Annex C.5 of [EMV Book 3].

Terminal Verification Results		
Byte 1	b8	Offline data authentication was not performed
	b7	SDA failed
	b6	ICC data missing
	b5	Card appears on terminal exception file
	b4	DDA failed
	b3	CDA failed
	b2-1	RFU
Byte 2	b8	ICC and terminal have different application versions
	b7	Expired application
	b6	Application not yet effective
	b5	Requested service not allowed for card product
	b4	New card
	b3-1	RFU
Byte 3	b8	Cardholder verification was not successful
	b7	Unrecognised CVM
	b6	PIN Try Limit exceeded
	b5	PIN entry required and PIN pad not present or not working
	b4	PIN entry required, PIN pad present, but PIN was not entered
	b3	Online PIN entered
	b2-1	RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 498

Terminal Verification Results		
Byte 4	b8	Transaction exceeds floor limit
	b7	Lower consecutive offline limit exceeded
	b6	Upper consecutive offline limit exceeded
	b5	Transaction selected randomly for online processing
	b4	Merchant forced transaction online
	b3-1	RFU
Byte 5	b8	Default TDOL used
	b7	Issuer authentication failed
	b6	Script processing failed before final GENERATE AC
	b5	Script processing failed after final GENERATE AC
	b4-1	RFU

A.1.152 Third Party Data

Tag: '9F6E'

Length: 5-32

Format: b

Update: K/RA

Implementations: Always

Description: The *Third Party Data* contains proprietary information from a third party. If present in the Card, the *Third Party Data* must be returned in a file read using the READ RECORD command or in the *File Control Information Template*.

Third Party Data is not interpreted by the Kernel.

'Device Type' is present when the most significant bit of byte 1 of 'Unique Identifier' is set to 0b. In this case, the maximum length of 'Proprietary Data' is 26 bytes.

Data Field	Length	Format
Country Code	2	Country Code according to [ISO 3166-1]
Unique Identifier	2	b (value assigned by MasterCard)
Device Type	0 or 2	b
Proprietary Data	1-28	b

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 499

A.1.153 Time Out Value

Tag:	'DF8127'
Length:	2
Format:	b
Update:	K
Implementations:	DE, EMV/DE
Description:	Defines the time in ms before the timer generates a TIMEOUT signal.

A.1.154 Torn Entry

Tag:	—
Length:	var.
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Data object that is used to refer to a record in the Torn Transaction Log. This may be the record number, but the actual implementation is proprietary.

A.1.155 Torn Record

Tag:	'FF8101'
Length:	var.
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	A copy of a record from the Torn Transaction Log that is expired. *Torn Record* is sent to the Terminal as part of the *Discretionary Data*.

A.1.156 Torn Temp Record

Tag:	—
Length:	var.
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Holds a copy of a record from the Torn Transaction Log.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 500

A.1.157 Track 1 Data

Tag: '56'

Length: var. up to 76

Format: ans

Update: K/RA

Implementations: Always

Description: *Track 1 Data* contains the data objects of the track 1 according to [ISO/IEC 7813] Structure B, excluding start sentinel, end sentinel and LRC. The *Track 1 Data* may be present in the file read using the READ RECORD command during a mag-stripe mode transaction. It is made up of the following sub-fields:

Data Field	Length	Format
Format Code	1	'42'
Primary Account Number	var up to 19	digits
Field Separator	1	'5E'
Name	2-26	(see ISO/IEC 7813)
Field Separator	1	'5E'
Expiry Date	4	YYMM
Service Code	3	digits
Discretionary Data	var.	ans

A.1.158 Track 1 Discretionary Data

Tag: '9F1F'

Length: var.

Format: ans

Update: K/RA

Implementations: EMV, EMV/DE

Description: Discretionary part of track 1 according to [ISO/IEC 7813].

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 501

A.1.159 Track 2 Data

Tag: '9F6B'

Length: var. up to 19

Format: b

Update: K/RA

Implementations: Always

Description: *Track 2 Data* contains the data objects of the track 2 according to [ISO/IEC 7813], excluding start sentinel, end sentinel and LRC. The *Track 2 Data* is present in the file read using the READ RECORD command during a mag-stripe mode transaction. It is made up of the following sub-fields:

Data Field	Length	Format
Primary Account Number	var. up to 19 nibbles	n
Field Separator	1 nibble	b ('D')
Expiry Date	2	n (YYMM)
Service Code	3 nibbles	n
Discretionary Data	var.	n
Padded with 'F' if needed to ensure whole bytes.	1 nibble	b

A.1.160 Track 2 Discretionary Data

Tag: '9F20'

Length: var.

Format: cn

Update: K/RA

Implementations: EMV, EMV/DE

Description: Discretionary part of track 2 according to [ISO/IEC 7813].

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 502

A.1.161 Track 2 Equivalent Data

Tag: '57'

Length: var. up to 19

Format: b

Update: K/RA

Implementations: EMV, EMV/DE

Description: Contains the data objects of the track 2, in accordance with [ISO/IEC 7813], excluding start sentinel, end sentinel, and LRC as follows:

Data Field	Length	Format
Primary Account Number	var. up to 19 nibbles	n
Field Separator	1 nibble	b ('D')
Expiration Date (YYMM)	2	n (YYMM)
Service Code	3 nibbles	n
Discretionary Data	var.	n
Padded with 'F' if needed to ensure whole bytes	1 nibble	b

A.1.162 Transaction Category Code

Tag: '9F53'

Length: 1

Format: an

Update: K

Implementations: EMV, EMV/DE

Description: This is a data object defined by MasterCard which indicates the type of transaction being performed, and which may be used in card risk management.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 503

A.1.163 Transaction Currency Code

Tag:	'5F2A'
Length:	2
Format:	n 3
Update:	K/ACT/DET
Implementations:	EMV, EMV/DE
Description:	Indicates the currency code of the transaction, in accordance with [ISO 4217].

A.1.164 Transaction Currency Exponent

Tag:	'5F36'
Length:	1
Format:	n 1
Update:	K/ACT/DET
Implementations:	EMV, EMV/DE
Description:	Indicates the implied position of the decimal point from the right of the transaction amount represented, in accordance with [ISO 4217].

A.1.165 Transaction Date

Tag:	'9A'
Length:	3
Format:	n 6 (YYMMDD)
Update:	K/ACT/DET
Implementations:	EMV, EMV/DE
Description:	Local date that the transaction was performed.

A.1.166 Transaction Time

Tag:	'9F21'
Length:	3
Format:	n 6 (HHMMSS)
Update:	K/ACT/DET
Implementations:	EMV, EMV/DE
Description:	Local time at which the transaction was performed.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 504

A.1.167 Transaction Type

Tag:	'9C'
Length:	1
Format:	n 2
Update:	K/ACT/DET
Implementations:	EMV, EMV/DE
Description:	Indicates the type of financial transaction, represented by the first two digits of [ISO 8583:1993] Processing Code.

A.1.168 UDOL

Tag:	'9F69'
Length:	var.
Format:	b
Update:	K/RA
Implementations:	Always
Description:	The *UDOL* is the DOL that specifies the data objects to be included in the data field of the COMPUTE CRYPTOGRAPHIC CHECKSUM command. The *UDOL* must at least include the *Unpredictable Number (Numeric)*. The *UDOL* is not mandatory for the Card. If it is not present in the Card, then the *Default UDOL* is used.

A.1.169 Unpredictable Number

Tag:	'9F37'
Length:	4
Format:	b
Update:	K
Implementations:	EMV, EMV/DE
Description:	Contains a Kernel challenge (random) to be used by the Card to ensure the variability and uniqueness to the generation of a cryptogram during an EMV mode transaction.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 505

A.1.170 Unpredictable Number (Numeric)

Tag:	'9F6A'
Length:	4
Format:	n 8
Update:	K
Implementations:	Always
Description:	Unpredictable number generated by the Kernel during a mag-stripe mode transaction. The *Unpredictable Number (Numeric)* is passed to the Card in the data field of the COMPUTE CRYPTOGRAPHIC CHECKSUM command. The 8-*nUN* most significant digits must be set to zero.

A.1.171 Unprotected Data Envelope 1

Tag:	'9F75'
Length:	var. 32-192
Format:	b
Update:	K/RA/ACT/DET
Implementations:	EMV/DE
Description:	The Unprotected Data Envelopes contain proprietary information from the issuer, payment system or third party. Unprotected Data Envelopes can be retrieved with the GET DATA command and can be updated with the PUT DATA (CLA='80') command without secure messaging. The length of the Unprotected Data Envelopes is implementation specific and must have a value between 32 and 192 bytes.

A.1.172 Unprotected Data Envelope 2

Tag:	'9F76'
Length:	var. 32-192
Format:	b
Update:	K/RA/ACT/DET
Implementations:	EMV/DE
Description:	Same as *Unprotected Data Envelope 1*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 506

A.1.173 Unprotected Data Envelope 3

Tag:	'9F77'
Length:	var. 32-192
Format:	b
Update:	K/RA/ACT/DET
Implementations:	EMV/DE
Description:	Same as *Unprotected Data Envelope 1*.

A.1.174 Unprotected Data Envelope 4

Tag:	'9F78'
Length:	var. 32-192
Format:	b
Update:	K/RA/ACT/DET
Implementations:	EMV/DE
Description:	Same as *Unprotected Data Envelope 1*.

A.1.175 Unprotected Data Envelope 5

Tag:	'9F79'
Length:	var. 32-192
Format:	b
Update:	K/RA/ACT/DET
Implementations:	EMV/DE
Description:	Same as *Unprotected Data Envelope 1*.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 507

A.1.176 User Interface Request Data

Tag: 'DF8116'
Length: 22
Format: b
Update: K
Implementations: Always
Description: Combines all parameters to be sent with the MSG signal.

Data Field	Length	Format
Message Identifier	1	b (see below)
Status	1	b (see below)
Hold Time	3	n 6
Language Preference	8	an
Value Qualifier	1	b (see below)
Value	6	n 12
Currency Code	2	n 3

Message Identifier			
Byte 1	b8-1	Message Identifier	
			00010111: CARD READ OK
			00100001: TRY AGAIN
			00000011: APPROVED
			00011010: APPROVED - SIGN
			00000111: DECLINED
			00011100: ERROR - OTHER CARD
			00100000: SEE PHONE
			00011011: AUTHORISING – PLEASE WAIT
			00011110: CLEAR DISPLAY
			11111111: N/A
			Other values: RFU

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 508

Status			
Byte 1	b8-1	Status	
			00000000: NOT READY
			00000001: IDLE
			00000010: READY TO READ
			00000011: PROCESSING
			00000100: CARD READ SUCCESSFULLY
			00000101: PROCESSING ERROR
			11111111: N/A
			Other values: RFU

Value Qualifier			
Byte 1	b8-5	Value Qualifier	
			0000: NONE
			0001: AMOUNT
			0010: BALANCE
			Other values: RFU
	b4-1	RFU	

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 509

A.2 Data Objects by Tag

Tag	Data Object
'50'	Application Label
'56'	Track 1 Data
'57'	Track 2 Equivalent Data
'5A'	Application PAN
'5F24'	Application Expiration Date
'5F25'	Application Effective Date
'5F28'	Issuer Country Code
'5F2A'	Transaction Currency Code
'5F2D'	Language Preference
'5F30'	Service Code
'5F34'	Application PAN Sequence Number
'5F36'	Transaction Currency Exponent
'5F57'	Account Type
'6F'	File Control Information Template
'77'	Response Message Template Format 2
'80'	Response Message Template Format 1
'82'	Application Interchange Profile
'84'	DF Name
'87'	Application Priority Indicator
'8C'	CDOL1
'8E'	CVM List
'8F'	CA Public Key Index (Card)
'90'	Issuer Public Key Certificate
'92'	Issuer Public Key Remainder
'94'	Application File Locator
'95'	Terminal Verification Results
'9A'	Transaction Date

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 510

Tag	Data Object
'9C'	Transaction Type
'9F01'	Acquirer Identifier
'9F02'	Amount, Authorized (Numeric)
'9F03'	Amount, Other (Numeric)
'9F07'	Application Usage Control
'9F08'	Application Version Number (Card)
'9F09'	Application Version Number (Reader)
'9F0D'	Issuer Action Code – Default
'9F0E'	Issuer Action Code – Denial
'9F0F'	Issuer Action Code – Online
'9F10'	Issuer Application Data
'9F11'	Issuer Code Table Index
'9F12'	Application Preferred Name
'9F15'	Merchant Category Code
'9F16'	Merchant Identifier
'9F1A'	Terminal Country Code
'9F1C'	Terminal Identification
'9F1E'	Interface Device Serial Number
'9F1F'	Track 1 Discretionary Data
'9F20'	Track 2 Discretionary Data
'9F21'	Transaction Time
'9F26'	Application Cryptogram
'9F27'	Cryptogram Information Data
'9F32'	Issuer Public Key Exponent
'9F33'	Terminal Capabilities
'9F34'	CVM Results
'9F35'	Terminal Type
'9F36'	Application Transaction Counter
'9F37'	Unpredictable Number
'9F38'	PDOL

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 511

Tag	Data Object
'9F40'	Additional Terminal Capabilities
'9F42'	Application Currency Code
'9F44'	Application Currency Exponent
'9F46'	ICC Public Key Certificate
'9F47'	ICC Public Key Exponent
'9F48'	ICC Public Key Remainder
'9F4A'	Static Data Authentication Tag List
'9F4B'	Signed Dynamic Application Data
'9F4C'	ICC Dynamic Number
'9F4D'	Log Entry
'9F4E'	Merchant Name and Location
'9F50'	Offline Accumulator Balance
'9F51'	DRDOL
'9F53'	Transaction Category Code
'9F54'	DS ODS Card
'9F55'	Mobile Support Indicator
'9F5B'	DSDOL
'9F5C'	DS Requested Operator ID
'9F5D'	Application Capabilities Information
'9F5E'	DS ID
'9F5F'	DS Slot Availability
'9F60'	CVC3 (Track1)
'9F61'	CVC3 (Track2)
'9F62'	PCVC3(Track1)
'9F63'	PUNATC(Track1)
'9F64'	NATC(Track1)
'9F65'	PCVC3(Track2)
'9F66'	PUNATC(Track2)
'9F67'	NATC(Track2)
'9F69'	UDOL

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 512

Tag	Data Object
'9F6A'	Unpredictable Number (Numeric)
'9F6B'	Track 2 Data
'9F6D'	Mag-stripe Application Version Number (Reader)
'9F6E'	Third Party Data
'9F6F'	DS Slot Management Control
'9F70'	Protected Data Envelope 1
'9F71'	Protected Data Envelope 2
'9F72'	Protected Data Envelope 3
'9F73'	Protected Data Envelope 4
'9F74'	Protected Data Envelope 5
'9F75'	Unprotected Data Envelope 1
'9F76'	Unprotected Data Envelope 2
'9F77'	Unprotected Data Envelope 3
'9F78'	Unprotected Data Envelope 4
'9F79'	Unprotected Data Envelope 5
'9F7C'	Merchant Custom Data
'9F7D'	DS Summary 1
'9F7F'	DS Unpredictable Number
'A5'	File Control Information Proprietary Template
'BF0C'	File Control Information Issuer Discretionary Data
'DF4B'	POS Cardholder Interaction Information
'DF60'	DS Input (Card)
'DF61'	DS Digest H
'DF62'	DS ODS Info
'DF63'	DS ODS Term
'DF8104'	Balance Read Before Gen AC
'DF8105'	Balance Read After Gen AC
'DF8106'	Data Needed
'DF8107'	CDOL1 Related Data
'DF8108'	DS AC Type

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 513

Tag	Data Object
'DF8109'	DS Input (Term)
'DF810A'	DS ODS Info For Reader
'DF810B'	DS Summary Status
'DF810C'	Kernel ID
'DF810D'	DSVN Term
'DF810E'	Post-Gen AC Put Data Status
'DF810F'	Pre-Gen AC Put Data Status
'DF8110'	Proceed To First Write Flag
'DF8111'	PDOL Related Data
'DF8112'	Tags To Read
'DF8113'	DRDOL Related Data
'DF8114'	Reference Control Parameter
'DF8115'	Error Indication
'DF8116'	User Interface Request Data
'DF8117'	Card Data Input Capability
'DF8118'	CVM Capability – CVM Required
'DF8119'	CVM Capability – No CVM Required
'DF811A'	Default UDOL
'DF811B'	Kernel Configuration
'DF811C'	Max Lifetime of Torn Transaction Log Record
'DF811D'	Max Number of Torn Transaction Log Records
'DF811E'	Mag-stripe CVM Capability – CVM Required
'DF811F'	Security Capability
'DF8120'	Terminal Action Code – Default
'DF8121'	Terminal Action Code – Denial
'DF8122'	Terminal Action Code – Online
'DF8123'	Reader Contactless Floor Limit
'DF8124'	Reader Contactless Transaction Limit (No On-device CVM)
'DF8125'	Reader Contactless Transaction Limit (On-device CVM)
'DF8126'	Reader CVM Required Limit

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 514

Tag	Data Object
'DF8127'	Time Out Value
'DF8128'	IDS Status
'DF8129'	Outcome Parameter Set
'DF812A'	DD Card (Track1)
'DF812B'	DD Card (Track2)
'DF812C'	Mag-stripe CVM Capability – No CVM Required
'DF812D'	Message Hold Time
'FF8101'	Torn Record
'FF8102'	Tags To Write Before Gen AC
'FF8103'	Tags To Write After Gen AC
'FF8104'	Data To Send
'FF8105'	Data Record
'FF8106'	Discretionary Data

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 515

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Annex B Data Exchange

B.1 Introduction

The full power of Data Exchange is achieved when the Terminal and Kernel process concurrently. The design of the Kernel aims at maximizing this concurrency by allowing it to provide data to the Terminal in parallel to reading data from the Card and analyzing the results from the previous read instruction.

The Kernel prioritizes sending GET DATA requests over sending READ RECORD commands and permits multiple updates from the Terminal. It is designed to send data to the Terminal when it has completed the (currently outstanding) requests from the Terminal rather than sending data piecemeal-wise for each request.

As a result of the above, most use cases can be addressed by a single DEK/DET exchange. Three examples of such use cases are given below.

B.2 Example 1 – Generic Data Exchange

The Terminal wants to make a simple purchase transaction that reads out the Third Party Data (if any) and modifies the transaction amount accordingly.

The configuration settings of the Kernel TLV Database (for the selected AID and for the purchase transaction type) are then as follows:

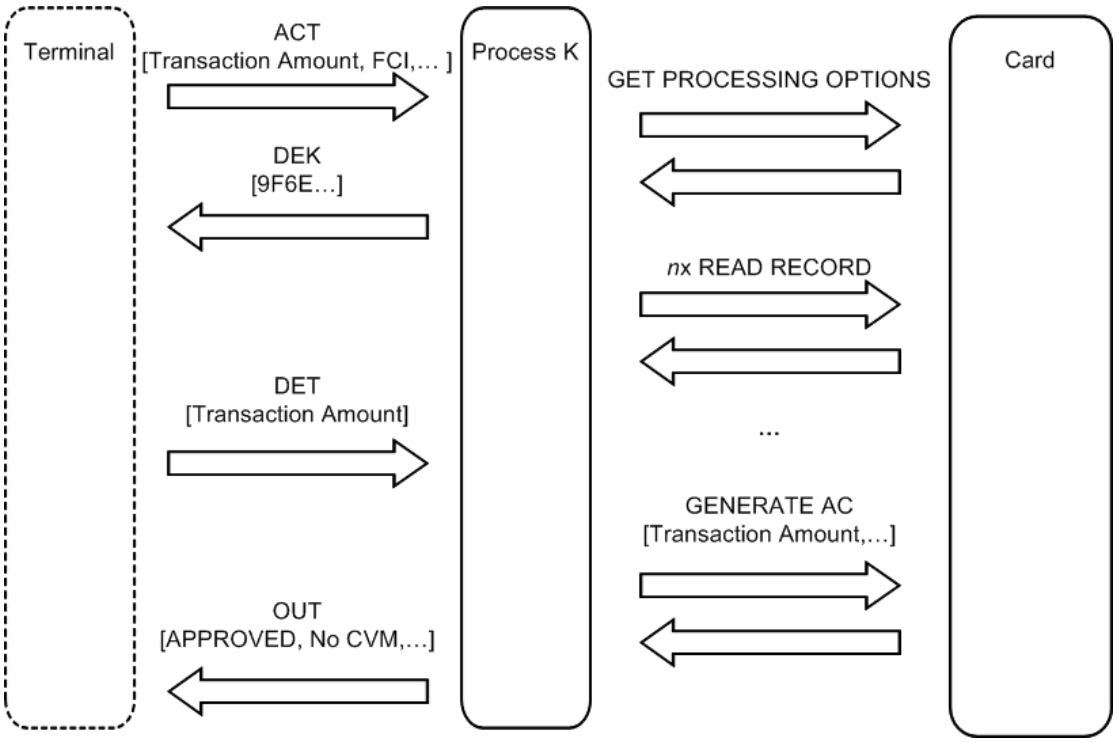
- The following tags are present:
  - *Tags To Read* with one entry: '9F6E'
- The following tags are absent:
  - *Tags To Write Before Gen AC*
  - *Tags To Write After Gen AC*
  - *DS Requested Operator ID*
  - *Proceed To First Write Flag*

The preferred setting is for the *Proceed To First Write Flag* to be not present and for the design of the system to be such that the Terminal will always respond in a timely fashion. If this is not the case then the *Proceed To First Write Flag* should be present with a value of 0 and in the example that follows the Terminal must respond to the Kernel with a *Proceed To First Write Flag* with a value of 1. The example below is written assuming a quick Terminal.

The resulting flow is illustrated in Figure B.1.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 517

Figure B.1—Data Exchange Example



The transaction amount (*Amount, Authorized (Numeric)*) is included in the ACT Signal, so that it is populated in the Kernel database with a length different from zero. Therefore, it will not be requested from the Terminal.

The *File Control Information Template* is included in the ACT Signal as well.

If the Third Party Data (tag '9F6E') is part of the *File Control Information Template*, its length is now set to a value different from zero in the Kernel database. As *Tags To Read* does not include a tag that is to be retrieved using a GET DATA command, no GET DATA command is sent prior to the first READ RECORD command. As the Third Party Data is the only entry data object in *Tags To Read*, the Kernel has all the data requested by the Terminal and sends the DEK Signal. The data objects *Proceed To First Write Flag*, *Tags To Write Before Gen AC*, and *Tags To Write After Gen AC* are not included in the DEK Signal as they are absent from the Kernel database.

It will take the Kernel and card something in the region of 100 to 150 milliseconds to complete the GET PROCESSING OPTIONS command and the READ RECORD commands, giving ample time to the Terminal to analyze the Third Party Data.

If the Third Party Data is not included in the *File Control Information Template* and after processing all the READ RECORD commands, the Third Party Data is still not available, the Kernel sends the DEK Signal, with the length of tag '9F6E' set to zero – as an indication that the data object was not available.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 518

Upon receipt of the DEK Signal, the Terminal now knows the Third Party Data or has an indication that the Third Party Data are not available. In case of the former, the Terminal can send a DET Signal with an updated Transaction Amount.

The Kernel, after completing its read sequence, moves to its write sequence.

With *Proceed To First Write Flag* absent from the Kernel database, the Kernel will not wait for a confirmation that it can proceed with the GENERATE AC command.

Upon receipt of the Card response, the Kernel sends an OUT Signal to the Terminal that includes the outcome of the transaction.

B.3 Example 2 – Stand Alone Data Storage

The Terminal wants to configure the Reader to make a simple purchase transaction that reads out a tagged data object '9F75' from the Card, update this data object and write it back to the Card. The information contained in this data object has no impact on the transaction amount.

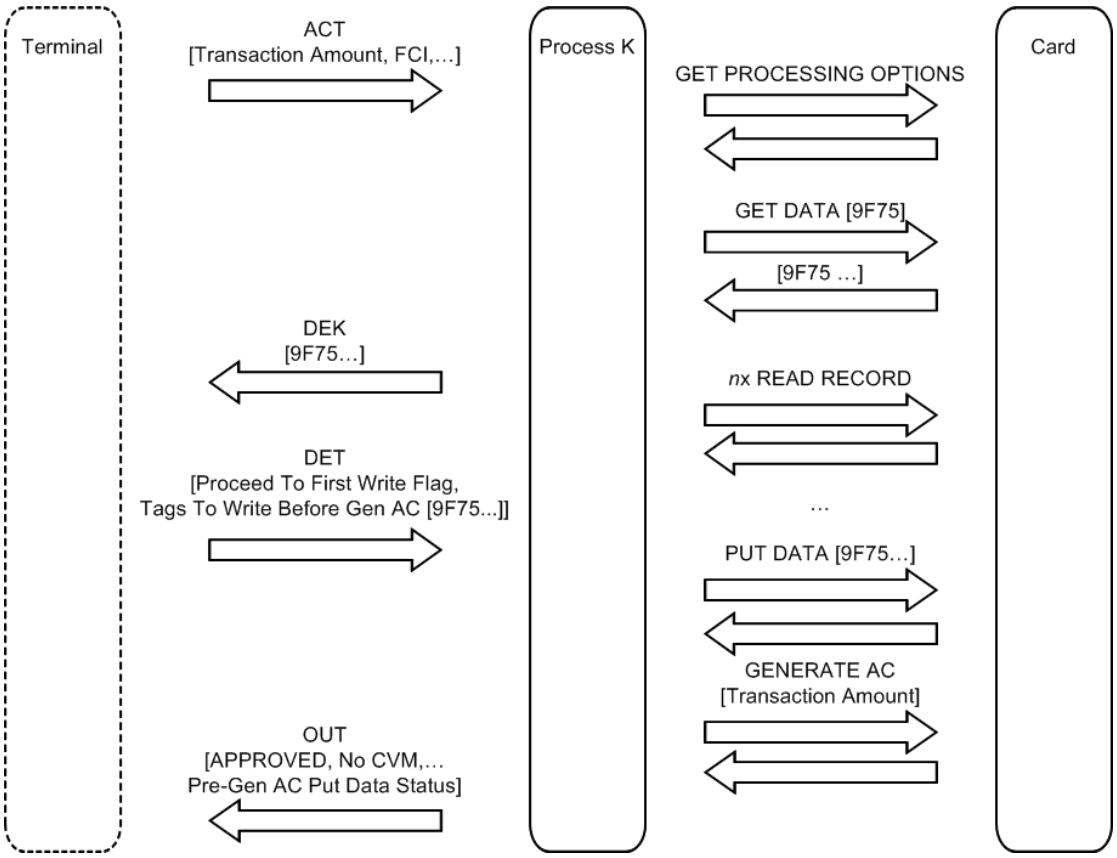
The configuration settings of the Kernel TLV Database (for the selected AID and for the purchase transaction type) are then as follows:

- The following tags are present:
  - *Tags To Read*, with one entry: '9F75'
  - *Proceed To First Write Flag*, with value '00'
- The following tags are absent:
  - *Tags To Write Before Gen AC*
  - *Tags To Write After Gen AC*
  - *DS Requested Operator ID*

The resulting flow is illustrated in Figure B.2.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 519

Figure B.2—SDS Example



The transaction amount (*Amount, Authorized (Numeric)*) is included in the ACT Signal, so that it is populated in the Kernel database with a length different from zero. Therefore, it will not be requested from the Terminal.

With *DS Requested Operator ID* absent from the Kernel database, IDS will not be activated. If the PDOL of the Card includes the tag of the *DS Requested Operator ID*, the corresponding field in the GET PROCESSING OPTIONS command will be zero filled.

As *Tags To Read* contains a single entry '9F75', which is a tag of a data object to be retrieved through a GET DATA command, the Kernel sends the GET DATA prior to the first READ RECORD command. The TLV data object returned by the Card is sent to the Terminal in a DEK Signal. The *Proceed To First Write Flag* is not included in the DEK Signal, as it has a length different from zero. *Tags To Write Before Gen AC* and *Tags To Write After Gen AC* are not included in the DEK Signal either as they are absent from the Kernel database.

While the Kernel continues with the READ RECORD commands, the Terminal is presented with the content of tag '9F75'.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 520

It will take the Kernel and card something in the region of 100 milliseconds to complete the READ RECORD commands, so as long as the Terminal responds in less than this, the transaction proceeds without interruption.

The Terminal replies with a single DET Signal that contains both *Tags To Write Before Gen AC* with a single entry for '9F75' with the new data and with the *Proceed To First Write Flag* set to a value different from zero.

The Kernel, after completing its read sequence, moves to its write sequence. As *Proceed To First Write Flag* has now a value different from zero, there is no need for the Kernel to wait and the Kernel sends a single PUT DATA command with tag '9F75', followed by the GENERATE AC command. Upon receipt of the Card response, the Kernel sends an OUT Signal to the Terminal that includes the outcome of the transaction and the flags indicating completion of the writing of the data to '9F75' (i.e. the *Pre-Gen AC Put Data Status*).

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 521

B.4 Example 3 – Integrated Data Storage

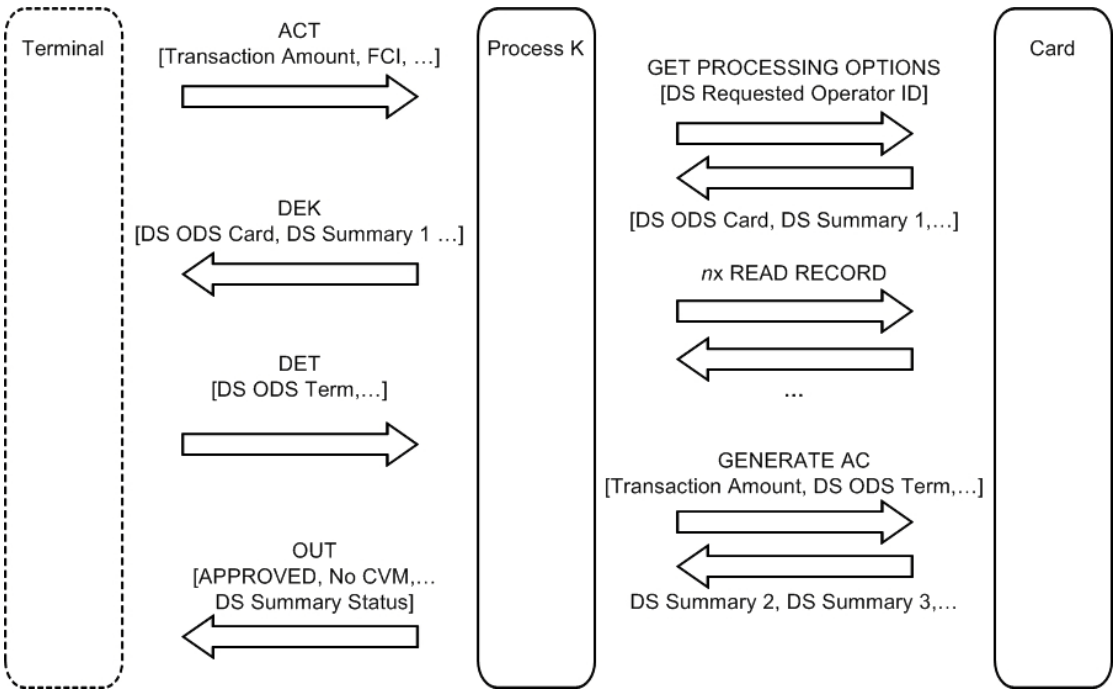
The Terminal wants to make a simple purchase transaction that reads out a slot from the Card for a particular operator identifier. The Terminal then updates the slot data object and writes the updated slot data back to the Card. The information contained in the slot data has no impact on the transaction amount.

The configuration settings of the Kernel TLV Database (for the selected AID and for the purchase transaction type) are then as follows:

- The following tags are present:
  - *DS Requested Operator ID*, with length different from zero
  - *Proceed To First Write Flag*, with value '00'
- The following tags are absent:
  - *Tags To Read*
  - *Tags To Write Before Gen AC*
  - *Tags To Write After Gen AC*

The resulting flow is illustrated in Figure B.3.

Figure B.3—IDS Example



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 522

The transaction amount (*Amount, Authorized (Numeric)*) is included in the ACT Signal, so that it is populated in the Kernel database with a length different from zero. Therefore, it will not be requested from the Terminal.

If *DS Requested Operator ID* is present in the Kernel database with a length different from zero, it will not be requested from the Terminal. If the PDOL of the Card includes the tag of the *DS Requested Operator ID*, IDS will be activated and the corresponding field in the GET PROCESSING OPTIONS command will be filled with the value of *DS Requested Operator ID* (and padding, if needed).

With the IDS data available, the Kernel has all the data requested by the Terminal and sends the DEK Signal.

The DEK Signal sent to the Terminal does not include the *Proceed To First Write Flag*, as this data object has a length different from zero. *Tags To Read*, *Tags To Write Before Gen AC*, and *Tags To Write After Gen AC* are not included in the DEK Signal either as they are absent from the Kernel database.

Upon receipt of the DEK Signal, the Terminal is now presented with the IDS data from the Card.

The Terminal replies with a single DET Signal that contains *DS ODS Term* (and other IDS related data) if the Terminal wants to update the data. The DET Signal also includes the *Proceed To First Write Flag* with a value different from zero, indicating that the Kernel no longer has to wait before proceeding with the GENERATE AC.

The Kernel, after completing its read sequence, moves to its write sequence. As *Proceed To First Write Flag* has now a length different from zero, the Kernel sends the GENERATE AC command, including *DS ODS Term* (and other IDS related data) appended after the CDOL1 data. Upon receipt of the Card response, the Kernel checks the Summaries and sends an OUT Signal to the Terminal that includes the outcome of the transaction and the flags indicating completion of the writing of the data (i.e. the *DS Summary Status*).

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 523

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Annex C Offline CAM Optimization

C.1 Introduction

Cryptographic processing and hashing of data are time-consuming but necessary operations. The design of the Reader should aim to minimise the processing time after the Card has completed the GENERATE AC command. However it should do this without slowing down the critical time period when the Card is still in the field and interacting with the Reader. Ideally recovery of the ICC key should be completed before the CDA response from the Card is available so that as little time is wasted as is possible.

The time needed will depend on the hardware design of the Reader. Performing an RSA operation using the public exponent on a fast implementation should only take a few milliseconds, but may take significantly longer on simpler hardware. Performing the SHA-1 hashing operations may also take several tens of milliseconds.

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 525

C.2 Optimization Techniques

The simplest tactic to use is to perform recovery of the ICC key while the Card is processing the GENERATE AC command because a budget of over 100 ms, typically over 200 ms, will be available to the Reader. The time budget for RECOVER AC will be less, perhaps 100 ms. It is also possible to begin the processing earlier, for example when the *Issuer Public Key Certificate* is first available from a READ RECORD command.

The flow charts shown in this section illustrate one way in which this might be achieved. They illustrate how to perform the cryptographic operations sequentially, as a procedure that is called at specific points in the main state model of the Kernel.

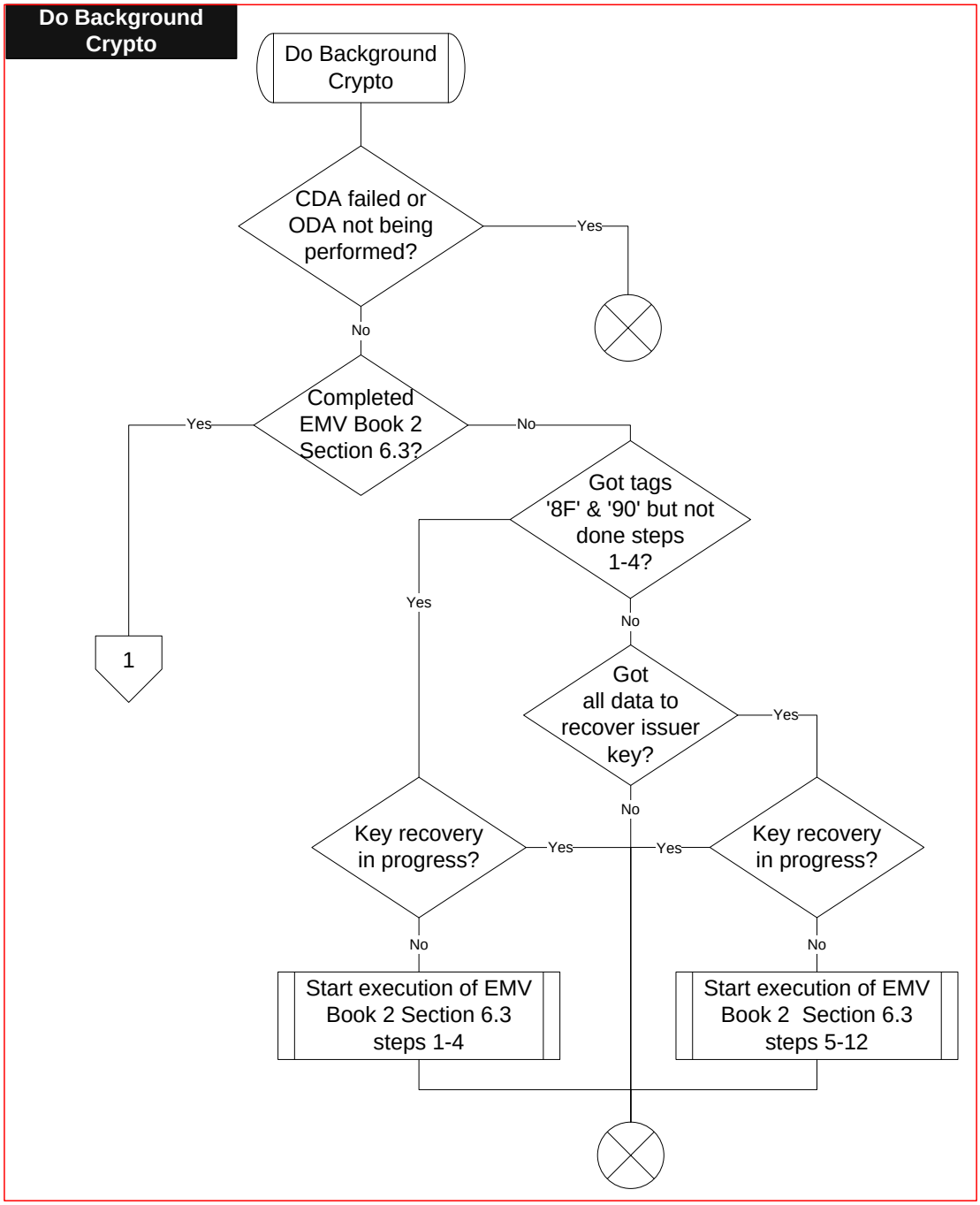
The performance benefit obtained (if any) depends on the hardware of the Reader and the personalisation of the Card.

The procedure "Do Background Crypto" would be called at the following points with the proviso that the process must either launch a parallel process, for example with a crypto coprocessor, or must return before the next card response is available (perhaps just 10 ms for a READ RECORD command)

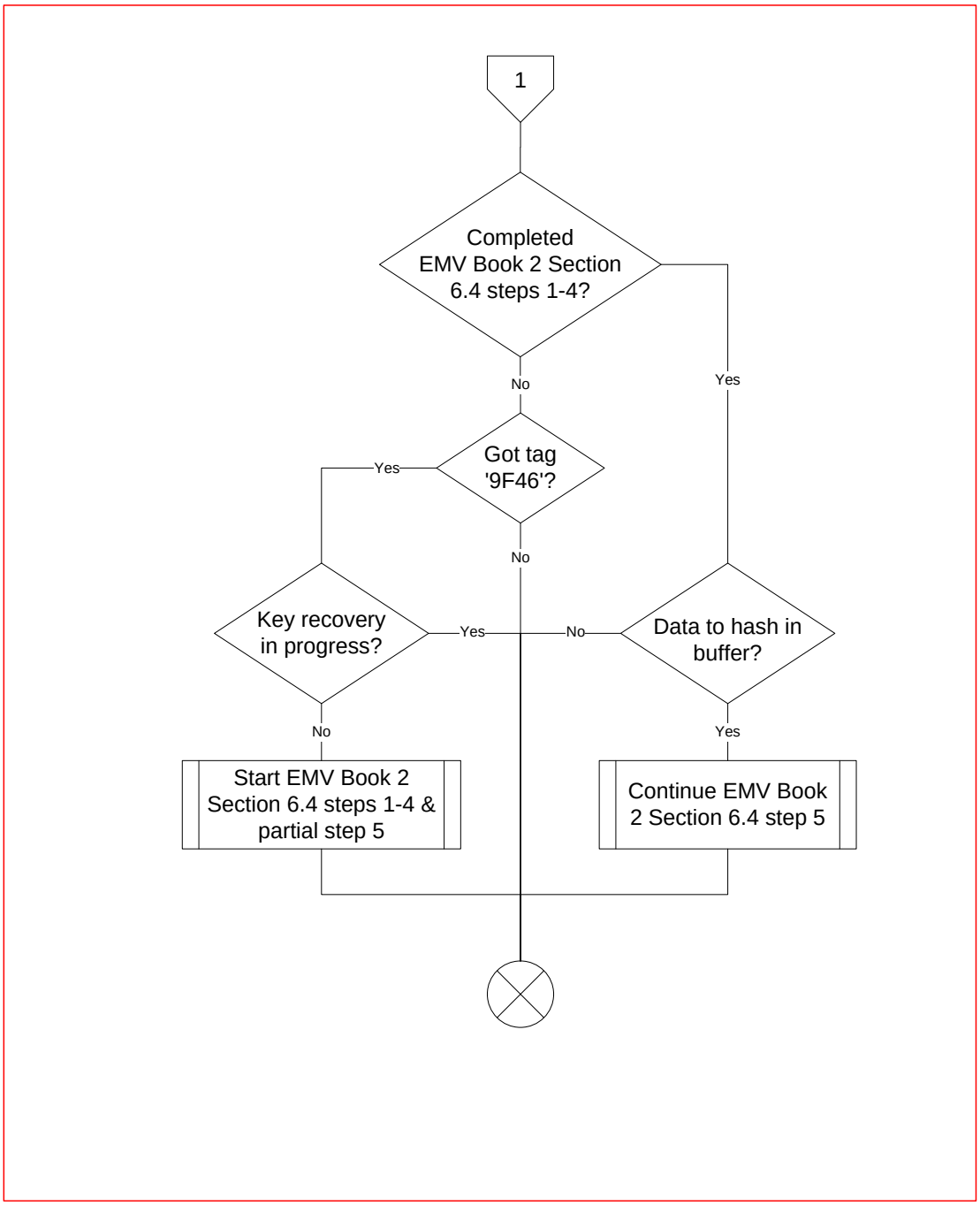
- After S456.E1 (GET DATA decision)
- After S456.ED4, the DE (‘No’) branch from before S456.ED2 and the ‘No’ branch of S456.ED3
- After S456.ED10
- After S456.ED51
- After S12.ED11

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 526

Figure C.1—Do Background Crypto



© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 527



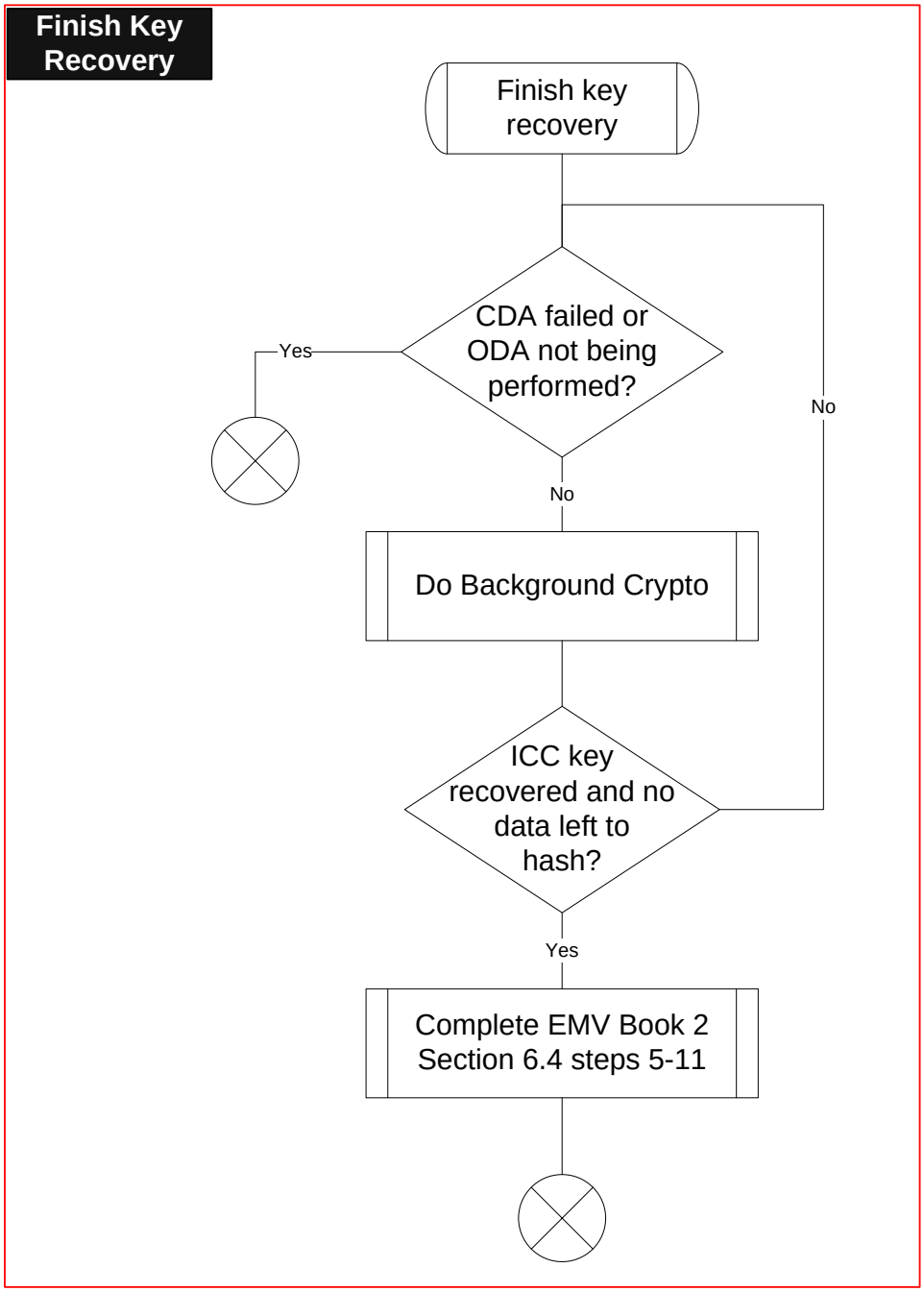
© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 528

The procedure "Finish Key Recovery" would be called at the following points, with the proviso that such processing must not delay completion of the Card / Reader interaction.

- After S456.E46
- After S456.E49
- After S12.ED16
- After S12.ED19

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 529

Figure C.2—Finish Key Recovery



© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 530

In designing a system to operate in this way, although it is important to minimise the processing time after the Card has finished its interaction with the Reader, this is less critical than minimising the processing time when the Card is interacting with the Reader.

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 531

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.

Mastercard, Ex1026, p. 532

Annex D Glossary

The following abbreviations are used in this document. For information on terms used in this specification, see section 1.5, Terminology.

Abbreviation	Description
AAC	Application Authentication Cryptogram
AC	Application Cryptogram
ADF	Application Definition File
AES	Advanced Encryption Standard
AFL	Application File Locator
AID	Application Identifier
AIP	Application Interchange Profile
an	Alphanumeric characters
ans	Alphanumeric and Special characters
APDU	Application Protocol Data Unit
ARQC	Authorization Request Cryptogram
ATC	Application Transaction Counter
b	Binary
BCD	Binary Coded Decimal
BER	Basic Encoding Rules
CA	Certification Authority
C-APDU	Command APDU
CDA	Combined DDA/AC Generation
CDOL	Card Risk Management Data Object List
CID	Cryptogram Information Data
CLA	Class byte of command message
cn	Compressed Numeric
CRL	Certification Revocation List
CVC	Card Verification Code

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 533

Abbreviation	Description
CVM	Cardholder Verification Method
DE	Data Exchange
DEK	Data Exchange Kernel
DES	Data Encryption Standard
DET	Data Exchange Terminal
DF	Dedicated File
DOL	Data Object List
DRDOL	Data Recovery Data Object List
DSDOL	Data Storage Data Object List
FIFO	First In First Out
IAD	Issuer Application Data
ICC	Integrated Circuit Card
IDS	Integrated Data Storage
INS	Instruction byte of command message
ISO	International Organization for Standardization
n	Numeric
N_{CA}	Length of CA Public Key Modulus
N_I	Length of Issuer Public Key Modulus
N_{IC}	Length of ICC Public Key Modulus
ODA	Offline Data Authentication
OWF	One Way Function
PAN	Primary Account Number
PCII	POS Cardholder Interaction Information
PDOL	Processing Options Data Object List
POS	Point of Sale
PIN	Personal Identification Number
R/CNS	Rejected, Conditions Not Satisfied
RFU	Reserved for Future Use
R-APDU	Response APDU
RID	Registered Application Provider Identifier

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 534

Abbreviation	Description
SFI	Short File Identifier
SHA	Secure Hash Algorithm
SW12	Status bytes 1-2
SDAD	Signed Dynamic Application Data
TC	Transaction Certificate
TL	Tag Length
TLV	Tag Length Value
TTQ	Terminal Transaction Qualifiers
TVR	Terminal Verification Results
UDOL	Unpredictable Number Data Object List
UN	Unpredictable Number
var.	Variable

© 2011 EMVCo, LLC ("EMVCo"). All rights reserved. Any and all uses of the EMV Specifications ("Materials") shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 535

(End of Document)

© 2011 EMVCo, LLC (“EMVCo”). All rights reserved. Any and all uses of the EMV Specifications (“Materials”) shall be permitted only pursuant to the terms and conditions of the license agreement between the user and EMVCo found at http://www.emvco.com/specifications.aspx.
Mastercard, Ex1026, p. 536