

# Contents

<b>Acknowledgments</b>	<b>ix</b>
<b>Introduction</b>	<b>xxvii</b>
<b>Chapter 1: The JCA and the JCE</b>	<b>1</b>
<b>Basic Architecture</b>	<b>1</b>
<b>Provider Signing</b>	<b>4</b>
<b>Jurisdiction Policy Files</b>	<b>4</b>
Installing the Unrestricted Policy Files	4
Troubleshooting Other Issues	7
How Do You Know the Policy Files Really Behave as Sun Says They Do?	7
<b>Installing the Bouncy Castle Provider</b>	<b>7</b>
Installing by Configuring the Java Runtime	8
Install the JAR File Containing the Provider	8
Enable the Provider by Adding It to the java.security File	8
Installing During Execution	10
<b>How Provider Precedence Works</b>	<b>10</b>
<b>Examining the Capabilities of a Provider</b>	<b>12</b>
<b>Summary</b>	<b>13</b>
<b>Exercises</b>	<b>14</b>
<b>Chapter 2: Symmetric Key Cryptography</b>	<b>15</b>
<b>A First Example</b>	<b>15</b>
A Basic Utility Class	16
The SecretKeySpec Class	19
The Cipher Class	19
Cipher.getInstance()	19
Cipher.init()	20
Cipher.update()	20
Cipher.doFinal()	20
<b>Symmetric Block Cipher Padding</b>	<b>21</b>
PKCS #5/PKCS #7 Padding	21
Other Padding Mechanisms	24

<b>Symmetric Block Cipher Modes</b>	<b>24</b>
ECB Mode	25
CBC Mode	26
Inline IVs	28
Creating an IV	30
Random IVs	31
Creating a SecureRandom Object	31
Pseudorandom IVs	32
A Look at Cipher Parameter Objects	34
The AlgorithmParameters Class	34
CTS Mode: A Special Case of CBC	34
Streaming Symmetric Block Cipher Modes	35
CTR Mode	35
OFB Mode	37
CFB Mode	38
<b>Symmetric Stream Ciphers</b>	<b>39</b>
<b>Generating Random Keys</b>	<b>40</b>
The Key Interface	42
Key.getAlgorithm()	42
Key.getEncoded()	42
Key.getFormat()	42
The KeyGenerator Class	42
KeyGenerator.getInstance()	43
KeyGenerator.init()	43
KeyGenerator.generateKey()	43
<b>Password-Based Encryption</b>	<b>43</b>
Basic PBE	44
The Password	45
The Salt	45
The Iteration Count	45
PBE in the JCE	45
The PBEPParameterSpec Class	48
The PBEKeySpec Class	48
The SecretKeyFactory Class	48
<b>Key Wrapping</b>	<b>50</b>
<b>Doing Cipher-Based I/O</b>	<b>52</b>
<b>Summary</b>	<b>55</b>
<b>Exercises</b>	<b>55</b>

---

<b>Chapter 3: Message Digests, MACs, and HMACs</b>	<b>57</b>
<b>Getting Started</b>	<b>57</b>
<b>The Problem of Tampering</b>	<b>60</b>
<b>Message Digests</b>	<b>62</b>
The MessageDigest Class	64
MessageDigest.update()	65
MessageDigest.digest()	65
MessageDigest.isEqual()	65
<b>Tampering with the Digest</b>	<b>66</b>
<b>MACs Based on Digests — the HMAC</b>	<b>68</b>
The Mac Class	71
Mac.init()	71
Mac.update()	71
Mac.doFinal()	71
<b>MACs Based on Symmetric Ciphers</b>	<b>72</b>
<b>Digests in Pseudorandom Functions</b>	<b>73</b>
PBE Key Generation	74
Mask Generation	77
<b>Doing Digest-Based I/O</b>	<b>79</b>
<b>Summary</b>	<b>81</b>
<b>Exercises</b>	<b>82</b>
<b>Chapter 4: Asymmetric Key Cryptography</b>	<b>83</b>
<b>Getting Started</b>	<b>84</b>
<b>The PublicKey and PrivateKey Interfaces</b>	<b>85</b>
<b>The RSA Algorithm</b>	<b>85</b>
The KeyFactory Class	88
RSAPublicKeySpec and RSAPublicKey	88
RSAPrivateKeySpec and RSAPrivateKey	89
Creating Random RSA Keys	89
The KeyPair Class	90
The KeyPairGenerator Class	91
The RSAKeyGenParameterSpec Class	91
Improving RSA Performance	91
Chinese Remainder Theorem	92
RSAPrivateCrtKeySpec and RSAPrivateCrtKey	92
Multi Prime Chinese Remainder Theorem	93

# Contents

---

RSA Padding Mechanisms	93
PKCS #1 V1.5 Padding	94
OAEP Padding	96
Wrapping RSA Keys	101
<b>Secret Key Exchange</b>	<b>103</b>
<b>Key Agreement</b>	<b>106</b>
The Diffie-Hellman Algorithm	106
The DHParameterSpec Class	109
Specification Objects for Diffie-Hellman Keys	109
Interfaces for Diffie-Hellman Keys	110
Diffie-Hellman with Elliptic Curve	110
ECField, ECFieldFp, and ECFieldF2m	112
The EllipticCurve Class	113
The ECPoint Class	113
The ECParameterSpec Class	113
The ECGenParameterSpec Class	114
Elliptic Curve Cryptography Before JDK 1.5	115
Diffie-Hellman for More Than Two Parties	115
<b>The El Gamal Algorithm</b>	<b>116</b>
The AlgorithmParameterGenerator Class	118
AlgorithmParameterGenerator.init()	118
AlgorithmParameterGenerator.generateParameters()	118
The DHGenParameterSpec Class	120
<b>Digital Signatures</b>	<b>121</b>
The Signature Class	121
Using the Signature Class in Signature Creation Mode	121
Using the Signature Class in Signature Verification Mode	122
Signature.setParameter() and Signature.getParameters()	122
The Digital Signature Algorithm	122
Regular DSA	123
Elliptic Curve DSA	127
RSA-Based Signature Algorithms	128
PKCS #1 1.5 Signatures	129
PSS Signatures	130
<b>Summary</b>	<b>132</b>
<b>Exercises</b>	<b>133</b>
<b>Chapter 5: Object Description in Cryptography Using ASN.1</b>	<b>135</b>
<b>What Is ASN.1?</b>	<b>135</b>
<b>Getting Started</b>	<b>136</b>

---

<b>Basic ASN.1 Syntax</b>	<b>136</b>
Comment Syntax	137
Object Identifiers	137
The Module Structure	138
<b>ASN.1 Types</b>	<b>140</b>
Simple Types	140
Bit String Types	141
Character String Types	142
Structured Types	143
Type Annotations	144
Tagging	144
EXPLICIT Tagging	146
IMPLICIT Tagging	147
AUTOMATIC Tagging	148
CHOICE	148
CLASS	149
<b>Encoding Rules</b>	<b>150</b>
BER Encoding	150
The Primitive Definite-Length Method	151
The Constructed Definite-Length Method	151
The Constructed Indefinite-Length Method	152
DER Encoding	152
<b>The Bouncy Castle ASN.1 API</b>	<b>153</b>
Creating the Basic ASN.1 Types	154
Dealing with Tagging	155
Defining Your Own Objects	156
Analyzing an Unknown Encoded Object	162
<b>Using ASN.1 in Java: Some Real Examples</b>	<b>164</b>
Some Basic ASN.1 Structures	164
The AlgorithmIdentifier Structure	164
The Attribute Structure	165
Encoding an IV	165
Inside a PKCS #1 V1.5 Signature	166
Encoding PSS Signature Parameters	169
Encoding Public and Private Keys	171
The X509EncodedKeySpec Class	171
The PKCS8EncodedKeySpec Class	174
The EncryptedPrivateKeyInfo Class	176
<b>Summary</b>	<b>181</b>
<b>Exercises</b>	<b>182</b>

<b>Chapter 6: Distinguished Names and Certificates</b>	<b>183</b>
<b>Getting Started</b>	<b>184</b>
<b>Distinguished Names</b>	<b>184</b>
The X500Principal Class	186
X500Principal.getEncoded()	186
X500Principal.getName()	186
<b>Public Key Certificates</b>	<b>187</b>
The Certificate Class	187
Certificate.getType()	187
Certificate.getPublicKey()	188
Certificate.verify()	188
Certificate.getEncoded()	188
<b>X.509 Certificates</b>	<b>188</b>
The X509Certificate Class	189
X509Certificate.getTBSCertificate()	189
X509Certificate.getVersion()	190
X509Certificate.getSerialNumber()	190
X509Certificate.getIssuerX500Principal()	191
X509Certificate.getNotBefore() and X509Certificate.getNotAfter()	191
X509Certificate.checkValidity()	192
X509Certificate.getSubjectX500Principal()	192
X509Certificate.getIssuerUniqueID()	192
X509Certificate.getSubjectUniqueID()	192
X509Certificate.getSignature()	193
X509Certificate.getSigAlgOID(), and X509Certificate.getSigAlgParams()	193
X509Certificate.getSigAlgName()	193
X.509 Extensions	195
The X509Extension Interface	196
X509Extension.getCriticalExtensionsOIDs()	197
X509Extension.getExtensionValue()	197
X509Extension.getNonCriticalExtensionOIDs()	197
X509Extension.hasUnsupportedCriticalExtension()	197
Extensions Supported Directly by X509Certificate	198
X509Certificate.getKeyUsage()	198
X509Certificate.getSubjectAlternativeNames()	199
X509Certificate.getIssuerAlternativeNames()	200
X509Certificate.getBasicConstraints()	200
X509Certificate.getExtendedKeyUsage()	200
<b>Reading and Writing Certificates</b>	<b>204</b>
The CertificateFactory Class	204
CertificateFactory.generateCertificate()	204
CertificateFactory.generateCertificates()	204

---

<b>Certification Requests</b>	<b>208</b>
<b>Writing a Simple Certificate Authority</b>	<b>214</b>
<b>Certificate Paths and Stores</b>	<b>219</b>
The CertPath Class	219
CertPath.getType()	220
CertPath.getCertificates()	220
CertPath.getEncoded()	220
CertPath.getEncodings()	220
The CertStore Class	221
The X509CertSelector Class	222
X509CertSelector.setCertificate()	222
X509CertSelector.setIssuer()	222
X509CertSelector.setSerialNumber()	223
X509CertSelector.setSubject()	223
<b>Summary</b>	<b>224</b>
<b>Exercises</b>	<b>225</b>
<b>Chapter 7: Certificate Revocation and Path Validation</b>	<b>227</b>
<b>Getting Started</b>	<b>228</b>
<b>Certificate Revocation Lists</b>	<b>230</b>
The CRL Class	231
CRL.getType()	231
CRL.isRevoked()	231
<b>X.509 Certificate Revocation Lists</b>	<b>232</b>
The X509CRL Class	233
X509CRL.getTBSCertList()	233
X509CRL.getVersion()	234
X509CRL.getIssuerX500Principal()	234
X509CRL.getThisUpdate() and X509CRL.getNextUpdate()	234
X509CRL.getRevokedCertificates()	235
X509CRL.getRevokedCertificate()	235
X509CRL.getSignature()	235
X509CRL.getSigAlgOID(), and X509CRL.getSigAlgParams()	235
X509CRL.getSigAlgName()	235
X509CRL.verify()	235
X509CRL.getEncoded()	236
The X509CRLEntry Class	236
X509CRLEntry.getCertificateIssuer()	236
X509CRLEntry.getRevocationDate()	236
X509CRLEntry.getSerialNumber()	236
X509CRLEntry.hasExtensions()	236

# Contents

---

X.509 CRL Entry Extensions	237
The ReasonCode Extension	237
The HoldInstructionCode Extension	237
The InvalidityDate Extension	238
The CertificateIssuer Extension	238
X.509 CRL Extensions	238
The AuthorityKeyIdentifier Extension	239
The IssuerAlternativeName Extension	239
The CRLNumber Extension	239
The DeltaCRLIndicator Extension	239
The IssuingDistributionPoint Extension	239
The FreshestCRL Extension	240
Reading CRLs using the CertificateFactory Class	243
CertificateFactory.generateCRL()	243
CertificateFactory.generateCRLs()	243
The X509CRLSelector Class	245
X509CRLSelector.addIssuer() and X509CRLSelector.addIssuerName()	245
X509CRLSelector.setDateAndTime()	246
X509CRLSelector.setMaxCRL() and X509CRLSelector.setMinCRL()	246
<b>Online Certificate Status Protocol</b>	<b>248</b>
The CertificateID Class	248
CertificateID.getHashAlgOID()	249
CertificateID.getIssuerNameHash()	249
CertificateID.getIssuerKeyHash()	249
CertificateID.getSerialNumber()	249
The OCSPReq Class	249
OCSPReq.getTBSRequest()	250
OCSPReq.getVersion()	250
OCSPReq.getRequestorName()	250
OCSPReq.getRequestList()	250
OCSPReq.isSigned()	251
OCSPReq.getSignature() and OCSPReq.getSignatureAlgOID()	251
OCSPReq.getCertificates()	251
OCSP Request Extensions	251
The Nonce Extension	251
The Acceptable Response Types Extension	252
The Service Locator Extension	252
The OCSPResp Class	255
The BasicOCSPResp Class	256
BasicOCSPResp.getTBSResponseData()	256
BasicOCSPResponse.getVersion()	257
BasicOCSPResponse.getResponderID()	257

---

BasicOCSPResponse.getProducedAt()	257
BasicOCSPResponse.getResponses()	257
OCSP Response Extensions	258
The CRL References Extension	258
The Archive Cutoff Extensions	258
X.509 CRL Entry Extensions	259
<b>Certificate Path Validation</b>	<b>264</b>
The TrustAnchor Class	264
The PKIXParameters Class	265
PKIXParameters.addCertStore() and PKIXParameters.setCertStores()	266
PKIXParameters.setDate()	266
PKIXParameters.setTargetCertConstraints()	266
PKIXParameters.setRevocationEnabled()	266
The CertPathValidator Class	266
CertPathValidator.getDefaultType()	267
CertPathValidator.getAlgorithm()	267
CertPathValidator.validate()	267
The PKIXCertPathValidatorResult Class	267
The PKIXCertPathChecker Class	270
PKIXCertPathChecker.init()	270
PKIXCertPathChecker.isForwardCheckingSupported()	271
PKIXCertPathChecker.getSupportedExtensions()	271
PKIXCertPathChecker.check()	271
<b>Building a Valid Path from a CertStore</b>	<b>275</b>
The CertPathBuilder Class	275
The PKIXBuilderParameters Class	275
<b>Summary</b>	<b>278</b>
<b>Exercises</b>	<b>279</b>
<b>Chapter 8: Key and Certificate Management Using Keystores</b>	<b>281</b>

---

<b>Getting Started</b>	<b>281</b>
<b>The KeyStore Class</b>	<b>283</b>
Keystore Types	284
Standard JDK Keystore Types	284
Bouncy Castle Keystore Types	284
The Basic KeyStore API	285
KeyStore.aliases()	285
KeyStore.containsAlias()	285
KeyStore.deleteEntry()	285
KeyStore.getCertificate()	285
KeyStore.getCertificateAlias()	285

# Contents

---

KeyStore.getCertificateChain()	286
KeyStore.getCreationDate()	286
KeyStore.getKey()	286
KeyStore.getType()	286
KeyStore.isCertificateEntry()	286
KeyStore.isKeyEntry()	286
KeyStore.load()	286
KeyStore.setCertificateEntry()	287
KeyStore.setKeyEntry()	287
KeyStore.size()	287
KeyStore.store()	287
<b>KeyStore Nested Classes and Interfaces</b>	<b>290</b>
The KeyStore.ProtectionParameter Interface	290
KeyStore.CallbackHandlerProtection	291
KeyStore.PasswordProtection	291
The KeyStore.Entry Interface	291
KeyStore.getEntry()	291
KeyStore.setEntry()	291
KeyStore.entryInstanceOf()	292
KeyStore.PrivateKeyEntry	292
KeyStore.SecretKeyEntry	292
KeyStore.TrustedCertificateEntry	292
The KeyStore.Builder Class	295
KeyStore.Builder.getKeyStore()	295
KeyStore.Builder.getProtectionParameter()	295
KeyStore.Builder.newInstance()	295
The KeyStore.LoadStoreParameter Interface	297
<b>The PKCS #12 Format</b>	<b>298</b>
Using PKCS #12 with the KeyStore API	300
<b>The Keytool</b>	<b>304</b>
Keytool Commands	304
General Command Options	304
Commands and Their Options	305
The JVM's CA Keystore	307
Some Keytool Experiments	308
Generating Some Sample Keystore Files	308
<b>Jarsigning and Java Policy</b>	<b>312</b>
The Jarsigner	312
Java Policy Files	312
<b>Summary</b>	<b>313</b>
<b>Exercises</b>	<b>314</b>

---

<b>Chapter 9: CMS and S/MIME</b>	<b>315</b>
<b>Getting Started</b>	<b>315</b>
<b>Cryptographic Message Syntax</b>	<b>318</b>
Basic CMS	318
The Data Content Type	319
The CMSProcessable Interface	320
<b>CMS Signed-Data</b>	<b>320</b>
ASN.1 Structure	321
The DigestAlgorithms Field	322
The EncapContentInfo Field	322
The Certificates and Crls Fields	322
The SignerInfos Field	323
The Version Field	325
The SignerInformation Class	325
SignerInformation.getDigestAlgOID()	325
SignerInformation.getDigestAlgParams()	325
SignerInformation.getEncryptionAlgOID()	326
SignerInformation.getEncryptionAlgParams()	326
SignerInformation.getSID()	326
SignerInformation.getSignature()	326
SignerInformation.getSignedAttributes()	326
SignerInformation.getUnsignedAttributes()	326
SignerInformation.verify()	326
SignerInformation.replaceUnsignedAttributes()	327
The SignerInformationStore Class	327
SignerInformationStore.get()	327
SignerInformationStore.getSigners()	327
SignerInformationStore.size()	327
The CMSSignedData Class	327
CMSSignedData.getCertificatesAndCRLs()	328
CMSSignedData.getEncoded()	328
CMSSignedData.getSignedContent()	328
CMSSignedData.getSignedContentOID()	328
CMSSignedData.getSignerInfos()	328
CMSSignedData.replaceSigners()	328
<b>CMS Enveloped-Data</b>	<b>332</b>
ASN.1 Structure	333
The OriginatorInfo Field	333
The RecipientInfos Field	333
The EncryptedContentInfo Field	334
The UnprotectedAttrs Field	335
The Version Field	335

# Contents

---

The RecipientInformation Class	335
RecipientInformation.getContent()	336
RecipientInformation.getKeyEncryptionAlgOID()	336
RecipientInformation.getKeyEncryptionAlgorithmParameters()	336
RecipientInformation.getKeyEncryptionAlgParams()	336
RecipientInformation.getRID()	336
The KeyTransRecipientInformation Class	336
The RecipientInformationStore Class	337
RecipientInformationStore.get()	337
RecipientInformationStore.getRecipients()	337
RecipientInformationStore.size()	337
The CMSEnvelopedData Class	337
CMSEnvelopedData.getEncoded()	338
CMSEnvelopedData.getEncryptionAlgOID()	338
CMSEnvelopedData.getEncryptionAlgorithmParameters()	338
CMSEnvelopedData.getEncryptionAlgParams()	338
CMSEnvelopedData.getRecipientInfos()	338
CMSEnvelopedData.getUnprotectedAttributes()	338
The KEKRecipientInformation Class	342
<b>Data Compression in CMS</b>	<b>345</b>
ASN.1 Structure	345
The CMSCompressedData Class	345
CMSCompressedData.getContent()	345
CMSCompressedData.getEncoded()	345
<b>Secure/Multipurpose Internet Mail Extensions (S/MIME)</b>	<b>347</b>
The CMSProcessableBodyPart Class	348
The SMIMEUtil Class	348
SMIMEUtil.toMimeBodyPart()	348
SMIMEUtil.createIssuerAndSerialNumberFor()	348
<b>S/MIME Signed Messages</b>	<b>348</b>
The CMSProcessableBodyPartInbound Class	349
The CMSProcessableBodyPartOutbound Class	349
The SMIMESigned Class	349
SMIMESigned.getContent()	350
SMIMESigned.getContentAsMimeMessage()	350
SMIMESigned.getContentWithSignature()	350
<b>S/MIME Enveloped Messages</b>	<b>354</b>
The SMIMEEnveloped Class	355
Combining Signing with Encryption	357
<b>S/MIME Compressed Messages</b>	<b>361</b>
The SMIMECompressed Class	361
<b>Summary</b>	<b>363</b>
<b>Exercises</b>	<b>364</b>

---

<b>Chapter 10: SSL and TLS</b>	<b>365</b>
<b>The SSL and TLS Protocols</b>	<b>365</b>
<b>Getting Started</b>	<b>366</b>
<b>A Basic SSL Client and Server</b>	<b>369</b>
The SSLSocketFactory Class	370
SSLSocketFactory.createSocket()	370
SSLSocketFactory.getDefault()	370
SSLSocketFactory.getDefaultCipherSuites()	370
SSLSocketFactory.getSupportedCipherSuites()	370
The SSLServerSocketFactory Class	370
SSLServerSocketFactory.createServerSocket()	371
SSLServerSocketFactory.getDefault()	371
SSLServerSocketFactory.getDefaultCipherSuites() and SSLServerSocketFactory.getSupportedCipherSuites()	371
The SSLSocket Class	371
SSLSocket.setEnabledCipherSuites()	371
SSLSocket.setEnabledProtocols()	372
SSLSocket.setEnableSessionCreation()	372
SSLSocket.setUseClientMode()	372
SSLSocket.startHandshake()	372
The SSLServerSocket Class	373
SSLServerSocket.setEnabledCipherSuites()	373
SSLServerSocket.setEnabledProtocols()	373
SSLServerSocket.setEnableSessionCreation()	373
SSLServerSocket.setUseClientMode()	373
The HandshakeCompletedListener Interface	377
<b>Client-Side Authentication</b>	<b>378</b>
SSLServerSocket Configuration	379
SSLServerSocket.setNeedClientAuth()	379
SSLServerSocket.setWantClientAuth()	379
Server-Mode SSLSocket Configuration	379
The setNeedClientAuth() Method	379
The setWantClientAuth() Method	379
The SSLContext Class	379
SSLContext.init()	380
SSLContext.getClientSessionContext()	380
SSLContext.getProtocol()	380
SSLContext.getServerSessionContext()	381
SSLContext.getServerSocketFactory()	381
SSLContext.getSocketFactory()	381

# Contents

---

The KeyManagerFactory Class	381
KeyManagerFactory.init()	381
KeyManagerFactory.getAlgorithm()	382
KeyManagerFactory.getDefaultAlgorithm()	382
KeyManagerFactory.getKeyManagers()	382
The TrustManagerFactory Class	386
TrustManagerFactory.init()	386
TrustManagerFactory.getAlgorithm()	386
TrustManagerFactory.getDefaultAlgorithm()	386
TrustManagerFactory.getTrustManagers()	387
<b>Managing SSL Session Information</b>	<b>389</b>
The SSLSession Interface	389
SSLSession.getCipherSuite()	389
SSLSession.getCreationTime()	389
SSLSession.getId()	390
SSLSession.getLastAccessedTime()	390
SSLSession.getLocalCertificates()	390
SSLSession.getLocalPrincipal()	390
SSLSession.getPeerCertificates()	390
SSLSession.getPeerHost()	390
SSLSession.getPeerPort()	390
SSLSession.getPeerPrincipal()	390
SSLSession.getProtocol()	391
SSLSession.getSessionContext()	391
SSLSession.invalidate()	391
SSLSession.isValid()	391
SSLSession.putValue()	391
<b>Dealing with HTTPS</b>	<b>394</b>
The HTTPSURLConnection Class	394
HTTPSURLConnection.getCipherSuite()	395
HTTPSURLConnection.getHostnameVerifier()	395
HTTPSURLConnection.getLocalCertificates()	395
HTTPSURLConnection.getLocalPrincipal()	395
HTTPSURLConnection.getPeerPrincipal()	395
HTTPSURLConnection.getServerCertificates()	396
HTTPSURLConnection.getSSLSocketFactory()	396
HTTPSURLConnection.setDefaultSSLSocketFactory()	396
HTTPSURLConnection.setDefaultHostnameVerifier()	396
The HostnameVerifier Interface	396
<b>Summary</b>	<b>401</b>
<b>Exercises</b>	<b>402</b>

---

<b>Appendix A: Solutions to Exercises</b>	<b>403</b>
<b>Chapter 1 Solutions</b>	<b>403</b>
<b>Chapter 2 Solutions</b>	<b>403</b>
<b>Chapter 3 Solutions</b>	<b>404</b>
<b>Chapter 4 Solutions</b>	<b>405</b>
<b>Chapter 5 Solutions</b>	<b>406</b>
<b>Chapter 6 Solutions</b>	<b>407</b>
<b>Chapter 7 Solutions</b>	<b>408</b>
<b>Chapter 8 Solutions</b>	<b>409</b>
<b>Chapter 9 Solutions</b>	<b>412</b>
<b>Chapter 10 Solutions</b>	<b>415</b>
<b>Appendix B: Algorithms Provided by the Bouncy Castle Provider</b>	<b>417</b>
<b>Asymmetric Ciphers</b>	<b>417</b>
<b>Certificate Path Validation</b>	<b>417</b>
<b>Key Agreement Algorithms</b>	<b>417</b>
<b>Key Stores</b>	<b>418</b>
<b>MAC Algorithms</b>	<b>418</b>
<b>Signature Algorithms</b>	<b>418</b>
<b>Message Digests</b>	<b>418</b>
<b>Symmetric Block Ciphers</b>	<b>418</b>
<b>Symmetric Stream Ciphers</b>	<b>419</b>
<b>Appendix C: Using the Bouncy Castle API for Elliptic Curve</b>	<b>421</b>
<b>Elliptic Curve Interfaces</b>	<b>421</b>
The EKey Interface	421
The ECPrivateKey Interface	422
The ECPublicKey Interface	422
The ECPointEncoder Interface	422
<b>Elliptic Curve Classes</b>	<b>422</b>
The ECNamedCurveParameterSpec Class	422
The ECNamedCurveSpec Class	423
The ECParameterSpec Class	423
The ECPrivateKeySpec Class	423
The ECPublicKeySpec Class	423

<b>Appendix D: Bibliography and Further Reading</b>	<b>425</b>
<b>ASN.1 Standards</b>	<b>425</b>
<b>IETF Working Group Charter Pages</b>	<b>425</b>
<b>NIST Publications</b>	<b>426</b>
<b>PKCS Standards</b>	<b>426</b>
<b>RFCs</b>	<b>427</b>
<b>Other Useful Standards</b>	<b>428</b>
<b>Useful References</b>	<b>428</b>
<b>Useful Web Links</b>	<b>429</b>
<b>Index</b>	<b>431</b>