

Index

A

`abort()` method in `JAAS LoginModules`,
 292–296
 Access-control model. *See also* `Permission`
 class/permissions; Policy files
 `AccessController`. *See*
 `AccessController` class
 authentication and authorization services,
 18
 flow of access control, 283
 JAAS, 257
 jar utility, 256
 jarsigner utility, 257
 Java support for standards, 20
 keytool utility, 256–257
 performance issues, 286–287
 policy files, 254
 policy management, 340
 Policy Tool utility, 257
 privileges, privileged blocks, 256
 privileges, lexical scoping of privilege
 modifications, 254–256
 privileges, principle of least privilege, 275
 privileges, `PrivilegedAction` class,
 277–279
 privileges, `PrivilegedException-`
 `Action` class, 277–279
 `SecurityManager`. *See*
 `SecurityManager` class
`AccessController` class
 basics, 238, 242
 and `SecurityManager` class, 273–274
 ACID (atomicity, consistency, isolation,
 and durability), assurance with EJB,
 158
 Actions, permissions, 258–259

Active Directory (Microsoft Windows),
 authentication and authorization
 services, 18
 Adapters. *See* Resource adapters
`addProvider()` method in `Security` class,
 388
 Advanced Encryption Standard. *See* AES
 Advanced Peer-to-Peer Communication. *See*
 APPC
 AES (Advanced Encryption Standard), 353
`AlgorithmParameters` class, 413
`AlgorithmParameterSpec` class, 414–415
`aliases()` method in `KeyStore` class,
 396
`AllPermission` class, 258, 260–261
 APPC (Advanced Peer-to-Peer
 Communication), CICS client/server
 applications, 27–28
 Application Assembler platform role
 authorization, 125
 basics, 64–69, 73, 84, 176–181
 deployment descriptors, 111–112
 EJB, basics, 176–177
 EJB, linking roles to role references,
 179–180
 EJB, method authorizations, 178–179
 EJB, principal delegation, 180–181
 Application client modules, JAR files, 57,
 60
 Application Component Provider platform role,
 64, 65, 66–67, 73
 Application domains, 266
 assembly-descriptor XML element, 177
 Asynchronous messaging, JMS, 20
 Atomicity, consistency, isolation, and
 durability. *See* ACID

Auditing

- ClassLoaders, 221–223
- SecurityManagers, 247–249
- servlet filters, 151–154

Authentication Markup Language. *See***AuthML**

- authentication-mechanism-type XML element, 88

Authentication services

- basics, 112–113, 529–532
- certificate-based, 78, 81–82, 114, 120–122, 456–457, 536–537
- connection policies, 88–90
- cryptography, 343–344
- data-origin, 356–359
- EJB, 183–184
- form-based, 78, 80–81, 114, 117–120
- HTTP methods, basic and digest modes, 77–79
- JAAS, 8, 20
- JAAS, basics, 291–292
- JAAS, examples, 296–313
- JAAS, LoginModule interface, 292–296, 531
- Java security support, 8–9
- Kerberos, 18, 88, 90
- login-configuration policies, 76–82, 113–122
- QoS, secure communication, 95
- secure-channel constraints, 82
- SSL, 450–451
- SSL, certificate-based authentication, 456–457
- SSL, example, with no authentication, 469–476
- SSL, example, with server and client authentication, 484–493
- SSL, example, with server authentication, 476–484
- SSL, mutual WAS authentication, 457
- SSL, protecting communication channels, 459
- SSL, protecting user ID and password, 455–456
- SSL, reverse proxy servers, 457
- SSL, single sign-on, certificate-based, 458–459
- SSL, single sign-on, cookie-based, 457–458

- SSO, 123–124, 457–459, 531–532
- terminology, 289–291
- user information, 531

AuthML (Authentication Markup Language), and SAML, 502–503**Authorization services. *See also* Permissions**

- application policies, 85
- authorization bindings, 85
- authorization tables, 84, 533
- basics, 124–125, 532–535
- declarative security policies, 82–85
- EJB, 184–185
- invocation chains, 126–127
- J2SE, 253–288
- JAAS, 8, 20, 313–337
- Java security support, 8–9
- Kerberos, 18
- method-permission table, 533
- precedence rules, 130–132
- proactive authorization, 92–94
- reactive authorization, 136–140
- URL protection, extensions, 129–130
- URL protection, patterns, 128–129
- URL protection, specific URLs, 127–128
- user data constraints, over SSL, 132–133
- Web Services, 523–525

B**Basic HTTP authentication**

- basics, 77–79
- connection policies, 88–90
- login-configuration policies, 114–117

Bean-Managed Persistence. *See* BMP**Binding, HTTP and SOAP, 12, 499****Block ciphers**

- CBC mode of operation, 353–355
- DES, 351
- ECB mode of operation, 353–354
- Feistel, 350–351
- IDEA, 353
- Rijndael, 353
- Triple-DES, 352

BMP (Bean-Managed Persistence), 58, 161**Brute-force attacks, 245****Bytecode**

- basics, 224–225
- magic number, 47, 228
- verifier, 229, 231–235

C**CA (Certificate Authority)**

- basics, 373–374
- client certificates, 536–537
- PKCS#10, 438
- SSL, 456–457

Caesar Cipher, 346–347, 355**Call-by-reference. *See* CBR****Call-by-value. *See* CBV****Canonicalization. *See* C14N****CBC (cipher block chaining), 353–355****CBR (call-by-reference), 163****CBV (call-by-value), 163****Certificate Authority. *See* CA****Certificate-based (X.509) authentication**

- basics, 78, 81–82, 114, 120–122
- SSL, 140–142

Certificate class, 415**Certificate revocation lists. *See* CRLs****CertificateFactory class, 415****CertificateFactorySpi class, 415****CertificationRequest object, 440****CertPath**

- cryptographic services, 20
- Java security technologies, 8–9

C14N (Canonicalization), 501–502**CGI (Common Gateway Interface)**

- alternatives, 102
- emergence of Java Servlet programming model, 13–14
- and HTTP servers, 101–102
- versus* servlets, 106–107

Chaining servlets

- basics, 105
- invocation chains, 126–127

checkConnect() method in

- SecurityManager class, 238, 241

checkCreateClassLoader() method in

- SecurityManager class, 244

checkPermission() method in

- AccessController class, 238–239, 267–270, 272–275, 287, 315–316

checkPermission() method in

- SecurityManager class, 238–239, 247–249, 267–268

checkRead() method in SecurityManager

- class, 238, 250–251

checkWrite() method in SecurityManager

- class, 238, 251–252

CICS (Customer Information Control Systems), IBM. *See* IBM CICS**Cipher block chaining. *See* CBC****Cipher class, 390, 417–419****Cipher suites, 140–142****CipherInputStream and**

- CipherOutputStream classes, 419–421

Ciphertext, 343**Class-confusion attacks, 226****Class editor attacks, 223****Class file verifier**

- basics, 223–226
- checks, 227–228
- checks, bytecode-integrity, 229
- checks, class-integrity, 228
- checks, file-integrity, 228
- checks, runtime-integrity, 230–231
- definition, 203–204
- duties, 226–227
- example, 235–237
- interdependence with class loaders and the SecurityManager class, 252
- limitations, 233–235

Class loaders/ClassLoader class

- application class path/loader, 211–212
- basics, 206
- boot class path, 210
- customizing ClassLoaders, 219–223
- definition, 203
- extension class path/loader, 212
- interdependence with class file verifier and SecurityManager class, 252
- internal class loader, 209
- loading process, delegation model, 216–219
- loading process, from untrusted sources, 211–214
- loading process, search order, 207, 214–216
- loading process, from trusted sources, 209–211
- name spaces, 206, 207
- name spaces, isolation, 209
- primordial class loader, 209
- security responsibilities, 207
- system classes, 209
- trustworthiness levels, 207–208

Classes

- default-access members, 205
- private members, 205
- protected members, 205–206
- public members, 206

CMP (container-managed persistence), 58, 161

CodeSource object, 264

commit() method in JAAS LoginModules,
291–296

Common Gateway Interface. *See* CGI

Common Object Request Broker Architecture.
See CORBA

Common Secure Interoperability. *See* CSIV2

Computing models

- four-tier, 190
- three-tier, 10–14, 25–29
- two-tier, 23–25

Connect methods, proxy servers, 49

Connection policies, declarative security
model, 87–90

Connectors. *See also* Resource adapters

- Java Connector Architecture, 61–62
- system-level and security contracts, 61

Container-managed persistence. *See* CMP

Cookies, 148–149

CORBA (Common Object Request Broker
Architecture)

- basics, 16
- EJB authentication, 183–184
- Java support for standards, 20

CRL class, 415

CRLs (certificate revocation lists), 121, 374

Cryptographic service providers. *See* CSP

Cryptography. *See also* CSP; JCA; JCE; PKCS

- AES, 353
- authentication, 343–344
- authentication, data-origin, 356–359, 369
- CA, 373–374
- CBC mode of operation, 353–355
- ciphertext, 343
- CRL, 374
- data confidentiality, 344, 356, 367–369
- data integrity, 344, 356–359, 369
- DES, 351
- digital certificates, 372–375
- digital signatures, basics, 370–371
- discrete logarithm, 362
- DH (Diffie-Hellman), 362–363

DSA digital signatures, 372

ECB mode of operation, 353–354

elliptic curve, 364–367

Feistel, 350–351

hash/message digest functions, 356

IDEA, 353

Java security support, 8–9, 377–432

keystream, 350

message randomization, 368–369

nonrepudiation, 344–345, 356, 369

one-time pad, 350

one-way and one-way trapdoor functions,
359

Pollard rho method, 367

prime numbers/relative primes, 360

primitive roots, 362

public-key, 345

public-key, basics, 359

public-key, combining with secret-key,
375–376

Rijndael, 353

root certificates, 373–374

RSA digital signatures, 371

RSA public-key algorithm, 360–362, 371

secret-key, 345–346

secret-key, block ciphers, 350–353

secret-key, confidentiality, 356

secret-key, key space, 355

secret-key, stream ciphers, 350

secret-key, substitutions and transpositions,
346–348

Triple-DES, 352

XOR (eXclusive OR) operation, 348–350

and XML, 503–505

CSIV2 (Common Secure Interoperability
version 2) protocol, 189

CSP (Cryptographic service provider), 379,
382. *See also* Cryptography; JCA; JCE;
PKCS

D

Data confidentiality, cryptography

- basics, 344
- public-key, 367–369
- secret-key, 356

Data Encryption Standard. *See* DES

Data integrity, cryptography
basics, 344

INDEX

- public-key, 369
 - secret-key, 356–359
 - SSL, 450
 - Declarative security model
 - authentication/login-configuration policies, 76–82
 - authorization policies, 82–85
 - basics, 75
 - connection policies, 87–90
 - delegation policies, 85–87
 - policy configuration in two-tier architecture, 25
 - Decompiler attacks, 223, 225
 - Delegation model, class loaders, 216–219
 - Delegation policies
 - declarative security model, 85–87
 - Deployer, 70
 - EJB specification, 185
 - J2EE Product Provider, 72
 - principal delegation, EJB, 180–181
 - principal delegation, J2EE, 133–134
 - Demilitarized zone. *See* DMZ
 - Deployer platform role, 64–66, 70–75, 83, 87, 181–182
 - Deployment descriptors
 - EAR files, 58
 - EJB, 14
 - login-config XML element, 77–80
 - URIs, 60
 - user-data-constraint XML element, 77, 82
 - Web modules, 111–112
 - DER (Distinguished Encoding Rules), 407
 - DES (Data Encryption Standard), 351, 359
 - destroy() method in Servlet objects, 108–111
 - DH (Diffie-Hellman) public-key cryptography algorithm, 362–363
 - Digest HTTP authentication
 - basics, 77–79
 - login-configuration policies, 114–117
 - digest() method in MessageDigest class, 398
 - Digital certificates, 372–375
 - Digital Signature Algorithm. *See* DSA
 - Digital signatures
 - basics, 370–371
 - DSA, 372
 - Java Cryptography Architecture, 8
 - RSA, 371
 - Disassembler attacks, 223, 225
 - Discrete logarithm, 362
 - Distinguished Name. *See* DN
 - DMZ (demilitarized zone), network security
 - physical setup, 30–31
 - DN (Distinguished Name), 121–122
 - DNS (Domain Name Service), 124
 - doAs() method in Subject class, 316–321
 - doAsPrivileged() method in Subject class, 316–317
 - doPrivileged() method in
 - AccessController class, 275–277, 284–285, 316–321
 - DSA (Digital Signature Algorithm), 372
 - DSAKey class, 416
 - DSAPrivateKey class, 416
 - DSAPublicKey class, 416
 - Dynamic content
 - four-tier architecture, 190
 - three-tier architecture, 10–14, 25–29
 - two-tier architecture, 23–25
- ## E
- E-business, secure request flows, 97–100
 - ECB (electronic codebook), 353–354
 - ECI (External Call Interface), CICS client/server applications, 27–28
 - EIS (enterprise information system), sign-on approaches
 - declarative, 61–62
 - programmatic, 61–62, 94–95
 - EJB (Enterprise JavaBeans). *See also* EJB roles
 - ACID assurance, 158
 - authentication, 183–184
 - authorization, 184–185
 - basics, 14
 - CMP and BMP, 58, 161
 - deployment descriptors, 14
 - entity beans, 58, 161
 - identify information retrieval, 91–92
 - interoperability with RMI-IIOP, 158–159
 - JAR files, 57
 - linking roles to role references, 179–180
 - marshalling/demarshalling RMI requests, 159, 549
 - message-driven beans, 58, 59, 161

EJB (Enterprise JavaBeans) (*continued*)

- method authorizations, 178–179
- platform roles, 159–183
- platform roles, Application Assembler, 64–69, 73, 84, 176–181
- platform roles, Deployer, 64–66, 70–75, 83, 87, 181–182
- platform roles, EJB Container Provider, 182–183
- platform roles, Enterprise Bean Provider, 161–176
- platform roles, System Administrator, 64–66, 71–72, 75, 83, 87, 182
- principal delegation, 180–181
- proactive authorization, 93–94
- programmatically security, 167–172
- runtime restrictions, 172–176
- RMI-IIOP communication, 160, 161–167
- as server-side technology, 32–33
- session beans, 58–59, 161
- skeletons, 159, 548–549
- stubs, 159, 548–549
- WAS components, 34–35
- wire protocol, 159

EJB Container Provider platform role, 182–183

EJBContext interface, 167, 182–183

EJBHome interface, 178

EJBLocalHome interface, 178

EJBLocalObject interface, 178

EJBObject interface, 178

Electronic codebooks. *See* ECBs

Elliptic curve, public-key cryptography algorithm, 364–367

Encryption. *See also* Cryptography

- basics, 343–346
- JCE, 8, 416–229

Enterprise Bean Provider platform role

- RMI-IIOP communication, 160, 161–167
- runtime restrictions, 172–176
- security APIs, 167–172

Enterprise information system. *See* EIS

Enterprise Java environment, 56–57

Enterprise JavaBeans. *See* EJB

Enterprise resource planning. *See* ERP

Entity beans, 58, 161

EntityBean interface, 171

EnvelopeData object, 440–444

EPI (External Presentation Interface), CICS client/server application, 27–28

ERP (enterprise resource planning) systems, Java Connector Architecture, 11

exclude-list XML element, 184

eXclusive OR operation. *See* XOR

eXtensible Access Control Markup Language. *See* XACML

eXtensible Markup Language. *See* XML

External Call Interface. *See* ECI

External Presentation Interface. *See* EPI

F

Fat clients, 194

Feistel ciphers, 350–351

File Transfer Protocol. *See* FTP

FilePermission class, 258, 260

Firewalls

- definition, 28
- downloading applets via HTTP, 45
- Java network connections through firewalls, 48–49
- network security physical setups, 30
- proxy servers, 40–42
- RMI connections through firewalls, 49–51
- SOCKS, 19–20, 40, 42–44
- SOCKS gateways *versus* proxy servers, 44
- standards, 36–37
- stopping Java downloads, 45–48
- TCP/IP packets, 37–39

Fixed-template data/text in JSP applications, 104

Form-based authentication, 78, 80–81, 114, 117–120

Four-tier computing models

- basics, 188–189
- caching reverse proxy server, 195–196
- clustered environment, 193–194
- connector layer, 197–198
- entry-level topology, 192–193
- portal servers as presentation layer, 194–195

FTP (File Transfer Protocol), proxy servers, 41

G

Generator point of an elliptic curve, 367

Generic Security Services. *See* GSS

GenericServlet class, 107–111

INDEX

getCallerPrincipal() method in
 EJBContext interface, 91, 167–172,
 531
 getInstance() factory methods in engine
 classes, 383, 386, 393
 getRemoteUser() method in
 HttpServletRequest interface, 92,
 135–136, 531
 getUserPrincipal() method in
 HttpServletRequest interface,
 91–92, 135–136, 531
 Gopher, proxy servers, 41
 Granted security roles, 83
 GSS (Generic Security Services),
 authentication
 basics, 340
 Java security support, 8–9

H

Handshake protocol, SSL, 120, 452–454
 HMAC, 359
 Home interface, EJB specification, 68
 HTML (Hypertext Markup Language), Java
 support for standards, 20
 HTTP (Hypertext Transfer Protocol)
 authentication methods, basic and digest
 modes, 77–79, 113–117
 binding with SOAP, 12, 499
 communication with component software,
 15
 connections to external servers via HTTPS,
 145–147
 connections to WASs behind firewalls, 198
 cookies, 148–149
 downloading applets via HTTP, 45
 encapsulation of requests by SOCKS,
 42–44
 HTTP servers and CGI, 101–102
 proxy servers, 41
 SSL sessions, 149–150
 stateless protocols, 147–148
 HTTPS (HTTP over SSL)
 authentication, 116
 basics, 454–455
 connections to external HTTP servers,
 145–147
 proxy servers, 41
 servlet containers, 34

HttpServlet class
 basics, 107–111
 HTTP methods doGet(), doPost(),
 doPut(), and doDelete(), 109,
 111
 server-side technology, 32
 HttpServletRequest interface
 basics, 91–93, 109–110
 isUserInRole() method, 91, 137–139,
 531
 getRemoteUser() method, 92, 135–136,
 531
 getUserPrincipal() method, 91–92,
 135–136, 531
 programmatic security, 135, 137
 HttpServletResponse interface, 109–110
 HttpSession interface, 149
 Hypertext Markup Language. *See* HTML
 Hypertext Transfer Protocol. *See* HTTP

I

IBM Access Manager (Tivoli), WAS security
 environment, 36
 IBM CICS (Customer Information Control
 System), Internet Gateway, 14, 26–28
 IBM Host On-Demand, two-tier architecture,
 24
 IBM WebSphere MQ (Message Queuing)
 JMS connections, 199
 resource adapters, 12
 ICAPI (Internet Connection API), alternative to
 CGI, 102
 IDEA (International Data Encryption
 Algorithm), 353
 Identity assertion, 530
 Identity mapping, 89–90, 540–542
 IEEE (Institute of Electrical and Electronics
 Engineers), and TLS, 449
 IIOP (Internet Inter-Object Request Broker
 Protocol)
 communication with component software,
 15
 connections between WASs, 198
 RMI-IIOP, 158–159, 551
 implies() method
 in Permission objects, 260
 in PermissionCollection objects,
 260

implies() method (*continued*)
 in Permissions objects, 260, 266
 in ProtectionDomain objects, 266
 init() method in HttpServlet class,
 108–111
 initSign() and initVerify() methods in
 Signature class, 400, 413
 insertProviderAt() method in Security
 class, 388
 Institute of Electrical and Electronics
 Engineers. *See* IEEE
 Integrity, QoS, secure communication, 95
 Interface definition languages. *See* IDLs
 International Business Machines. *See* IBM
 International Data Encryption Algorithm. *See*
 IDEA
 Internet Connection API. *See* ICAP
 Internet Engineering Task Force. *See* IETF
 Internet Inter-Object Request Broker Protocol.
 See IIOP
 isCallerInRole() method in EJBContext
 interface, 90–91, 167–172, 531
 isSecure() method in ServletRequest
 interface, 140
 isUserInRole() method in
 HttpServletRequest interface, 91,
 137–139, 531

J

JAAS (Java Authentication and Authorization
 Service)
 authentication, basics, 291–292
 authentication, examples, 296–313
 authentication, LoginModule interface,
 292–296, 531
 authentication, support, 20
 authentication, considerations in J2EE,
 338–340
 authorization, basics, 313
 authorization, policy files, 321–324
 authorization, ProtectionDomain class,
 313–316
 authorization, subject-based examples,
 324–336
 authorization, subjects and principals,
 290–291, 316–321, 339
 authorization, support, 20

 authorization, considerations in J2EE,
 338–340
 basics, 7, 289–291
 Java security support, 8–9
 stacked authenticators, 294
 terminology, 289–291
JAR (Java Archive) files
 J2EE applications, application client
 modules, 57, 60
 J2EE applications, EJB modules, 57, 58–59
 jar utility, 256
 jarsigner utility, 257
 Java Archive. *See* JAR
 Java Authentication and Authorization Service.
 See JAAS
 Java Community Process. *See* JCP
 Java Connector Architecture. *See* JCA
 Java Cryptography Architecture. *See* JCA
 Java Cryptography Extension. *See* JCE
 Java Database Connectivity. *See* JDBC
 Java Generic Security Services. *See* JGSS
 Java keystore. *See* JKS
 Java Messaging Service. *See* JMS
 Java Plug-in, 24
 Java Runtime Environment. *See* JRE
 Java Secure Socket Extension, JSSE
 Java Specification Request. *See* JSR
 Java 2, Standard Edition. *See* J2SE
 Java 2, Enterprise Edition. *See* J2EE
 Java 2 Runtime Environment. *See* J2RE
 Java 2 Software Development Kit. *See* J2SDK
 Java Virtual Machine. *See* JVM
 JavaServer Pages. *See* JSP
JCA (Java Connector Architecture)
 identity mapping, 540–542
 resource adapters, 11–12
JCA (Java Cryptography Architecture). *See*
 also Cryptography; CSP; JCE; PKCS
 algorithms, 378–379, 380–382
 basics, 8–9, 19–20, 379–380
 classes, AlgorithmParameters, 413
 classes, CertificateFactory, 415
 classes, CertificateFactorySpi, 415
 classes, CRL, 415
 classes, engine, 378, 380–381, 389–390
 classes, KeyFactory, 380–381, 389, 394
 classes, KeyPair, 394

- classes, `KeyPairGenerator`, 389, 394–395
- classes, `KeyStore`, 395–397
- classes, `MessageDigest`, 397–400
- classes, `Provider`, 379, 382–388
- classes, `SecureRandom`, 392–393
- classes, `Signature`, 380–381, 389, 400–412
- classes, `SignedObject`, 413–414
- classes, `X509Certificate`, 415
- classes, `X509CRL`, 415
- classes, `X509CRLEntry`, 415
- cryptographic services, 8–9
- deep copies, 413
- interfaces, `AlgorithmParameterSpec`, 414–415
- interfaces, `Certificate`, 415
- interfaces, `DSAPublicKey`, 416
- interfaces, `DSAPrivateKey`, 416
- interfaces, `Key`, 393
- interfaces, `PrivateKey`, 393
- interfaces, `PublicKey`, 393
- interface, `X509Extension`, 415
- and JSSE, 460–461
- keystores, 378
- and PKCS, 434
- providers, 379, 382–388
- security interfaces, 8
- JCE (Java Cryptography Extensions). *See also* Cryptography; CSP; JCA; PKCS
 - algorithms, 378–379, 380–382
 - basics, 8–9, 19–20, 379–380, 416–432
 - cipher modes, 418
 - classes, `Cipher`, 390, 417–419
 - classes, `CipherInputStream` and `CipherOutputStream`, 419–421
 - classes, engine, 378, 380–381, 390–391
 - classes, `KeyAgreement`, 391, 424–425
 - classes, `KeyGenerator`, 391, 422
 - classes, `Mac`, 391, 425–426
 - classes, `Provider`, 379, 382–388
 - classes, `SealedObject`, 423–424
 - classes, `SecretKeyFactory`, 422–423
 - classes, `SecretKeySpec`, 422
 - encryption, 8, 417
 - examples, 426–431
 - interfaces, `SecretKey`, 421
 - and JSSE, 460–461
 - key agreements, 417
 - keystores, 378
 - MACs, 417
 - padding schemes, 418
 - and PKCS, 434
 - providers, 416–417
 - security interfaces, 8
 - software components, 416–417
 - transformations, 418
- JDBC (Java Database Connectivity), 8
- JGSS (Java Generic Security Services), 8–9
- JIT (just-in-time) compilers, JRE, 6–7
- JKS (Java keystore), 395
- JMS (Java Messaging Service)
 - enterprise Java environment, 62–63
 - messaging, 199
 - messaging, asynchronous, 20
 - messaging, reliable point-to-point model, 62–63
 - publish/subscribe model, 63
- JRE (Java Runtime Environment), JIT
 - COMPILERS, 6–7. *See also* Class file verifier; Class loaders; Security-Manager class
- JSP (JavaServer Page)
 - basics, 104–105
 - identity information retrieval, 91–92
 - proactive authorization, 92–93
 - runtime restrictions, 143–145
 - server-side technology, 32–33
- JSRs (Java Specification Requests)
 - JAAS, 340
 - XML and Web Services security, 503
- JSSE (Java Secure Socket Extension)
 - basics, 459–493
 - considerations for container providers, 536
 - cryptographic services, 20
 - SSL, 460–461
- J2EE (Java 2, Enterprise Edition)
 - basics, 5
 - connections to non-J2EE systems, 199
 - containers, 61
 - platform roles, Application Assembler, 64–69, 73, 84, 176–181
 - platform roles, Application Component Provider, 64, 65, 66–67, 73

J2EE (Java 2, Enterprise Edition) (*continued*)
 platform roles, Deployer, 64–66, 70–75, 83, 87, 181–182
 platform roles, Enterprise Bean Provider, 161–176
 platform roles, EJB Container Provider, 182–183
 platform roles, J2EE Product Provider, 64, 72–73, 85
 platform roles, System Administrator, 64–66, 71–72, 75, 83, 87, 182
 support for engineering software in heterogeneous world, 21–22
 J2EE Product Provider platform role, 64, 72–73, 85
 J2SDK (Java 2 Software Development Kit), 5–6
 J2SE (Java 2, Standard Edition)
 basics, 4–5
 basis for J2EE, 3
 security architecture, 203–252
 permission model, 253–287
 JAAS, 289–340
 JCA and JCE, 377–432
 JSSE, 459–462
 Just-in-time compilers. *See* JIT compilers
 JVM (Java Virtual Machine), components, 204.
See also Class file verifiers; Class loaders; SecurityManager class

K

Kerberos
 authentication, connection policies, 88–90
 authentication and authorization services, 18
 Java security support, 8–9
 Java support for standards, 20
 Key interface, 393
 Key space, secret-key cryptography, 355
 KeyAgreement class, 391, 424–425
 KeyFactory class, 380–381, 389, 394
 KeyFactorySpi class, 380–381
 KeyGenerator class, 391, 422
 KeyManagerFactory class, 484–485
 KeyPair class, 394
 KeyPairGenerator class, 389, 394–395
 KeyStore class, 395–397, 461
 keytool utility, 256–257, 396

L

LDAP (Lightweight Directory Access Protocol), WAS security environment, 36, 65–66, 88, 97, 121–122, 151, 154, 160, 199, 530–531, 536
 Lexical scoping of privilege modifications, 254–256
 Lightweight Directory Access Protocol. *See* LDAP
 Local home interface, EJB specification, 68
 Local interface, EJB specification, 68
 login-config deployment descriptor element, 77–80
 Login-configuration policies, authentication, 76–82, 113–122
 login() method in LoginModule interface, 292–296
 Login servlet filters, 151–154
 LoginContext class, 294–296
 LoginModule interface, 292–296, 531
 logout() method in LoginModule interface, 292–296

M

Mac class, 391, 425–426
 MAC (media access control), adapters, 38
 MAC (message authentication code)
 basics, 356–359
 HMAC, 359
 Magic number, bytecode, 47, 228
 Malicious compilers, 223, 244–245
 Many-to-many identity mapping, 90
 Many-to-one identity mapping, 89–90, 541
 MD5 (Message Digest V5), 357, 379–381
 Media access control. *See* MAC
 Message authentication code. *See* MAC
 Message Digest V5. *See* MD5
 Message-driven beans, 58, 59, 161
 Message queuing. *See* MQ
 Message randomization, 368–369
 Message Digest
 basics, 356–359
 MD5, 357, 379–381
 MD5 and SHA-1, 357
 MessageDigest class, 380–382, 389, 397–400
 MessageDigestSpi class, 380–382
 SHA-1, 357, 381

INDEX

Messaging systems access, JMS, 8
 method-permission XML element
 basics, 179
 authorization, 82–83
 Method-permission tables, 83–84
 Middleware, three-tier architecture, 10–14,
 25–29
 Model-View-Controller. *See* MVC
 MQ (Message Queuing), IBM WebSphere
 JMS connections, 199
 resource adapters, 12
 MVC (Model-View-Controller), 33

N

National Center for Supercomputing
 Applications. *See* NCSA
 NCSA (National Center for Supercomputing
 Applications), firewall standards, 36
 NetBios (Network Basic Input Output System),
 27–28
 Netscape Server Applications Programming
 Interface. *See* NSAPI
 Network Basic Input Output System. *See*
 NetBios
 Network security, physical setups, 29–31
 Non-type-safe code, *versus* type-safe code, 5–7
 Nonrepudiation
 cryptography, 344–345, 356
 SSL, 451
 NSAPI (Netscape Server Application
 Programming Interface), alternative to
 CGI, 102

O

Object Management Group. *See* OMG
 Object serialization, 548
 OMG (Object Management Group)
 basics, 16
 Java support for standards, 20
 One-to-one identity mapping, 90, 541
 Opaque key representation, 389–390, 413
 Open Systems Interconnection. *See* OSI
 OSI (Open Systems Interconnection) model,
 TCP/IP packets, 37–38

P

PAM (Pluggable Authentication Module), 291
 Parent class loaders, 216

Passivated/activated EJB components,
 173
 Permission class/permissions. *See also*
 Access-control model; Authorization
 services; Policy files
 AllPermission class, 258, 260–261
 authorization policies, 82
 basics, 258–259
 Codesource object, 264
 FilePermission class, 258, 260
 inheritance tree, 258
 methods, 260, 266
 Permission classes equivalent to
 AllPermission, 260–261
 PermissionCollection class, 259–260
 Permissions class, 259–260
 policy files, basics, 261–264
 ProtectionDomain class, 265–270
 ProtectionDomain class, inheritance,
 285–286
 ProtectionDomain class, removing
 duplicates, 286–287
 RuntimePermission class, 258, 261
 SecurityPermission class, 258, 261
 SocketPermission class, 258
 targets and actions, 258–259
 PermissionCollection class, 259–260
 Permissions class, 259–260
 Permutations, secret-key cryptography,
 346–348
 PKCS (Public-Key Cryptography Standards).
 See also Cryptography; CSP; JCA; JCE;
 Public-key cryptography
 and JCA and JCE, 434
 overview, 434–435
 PKCS#1, RSA Cryptography Standard,
 435, 439–440
 PKCS#5, Password-Based Cryptography
 Standard, 435–436
 PKCS#7, Cryptographic Message Syntax
 Standard, 436, 439–444
 PKCS#8, Private-Key Information Syntax
 Standard, 436–437
 PKCS#9, Selected Attribute Types, 437
 PKCS#10, Certification Request Syntax
 Standard, 437, 440
 PKCS#12, Personal Information Exchange
 Syntax Standard, 437–438

- PKCS (*continued*)
 - S/MIME, encrypting transactions with, 445 and S/MIME, example, 444–445
 - S/MIME, signing and verifying transactions with, 439–442
- PKI (Public Key Infrastructure)
 - security support, 8–9
 - standards support, 20
- Pluggable Authentication Module. *See* PAM
- Point-to-point messaging model, JMS, 62–63
- Policy files. *See also* Access-control model; *Permission* class/permissions
 - basics, 254, 261–264
 - declarative policy configuration, 25
- Policy Tool utility, 257
- Pollard rho method, 367
- Portability, in heterogeneous computing environments, 10
- Precedence rules, authorization services, 130–132
- Prime numbers/relative primes, 360
- Primitive roots, 362
- Principal delegation
 - Application Assembler, EJB, 64–69, 73, 84, 176–181
 - basics, 133–134
- Principal interface
 - basics, 291
 - identity information, 92
- Private members, 205
- PrivateKey interface, 393
- Privilege modifications, lexical scoping of, 254–256
- Privileged blocks, 256
- Privileged/nonprivileged ports, 39
- Privileges
 - blocks, 256
 - doPrivileged() method in
 - AccessController class, 275–277, 284–285, 316–321
 - lexical scoping of privilege modifications, 254–256
 - principle of least privilege, 275
 - PrivilegedAction interface, 277–279, 316
 - PrivilegedExceptionAction interface, 277–279, 316
- PRNGs (Pseudorandom Number Generators), 392
- Proactive authorization
 - basics, 92–94
 - programmatic security, 136–140
- Programmatic security
 - application-managed sign-on to EIS, 94–95
 - getCallerPrincipal() method in
 - EJBContext interface, 91, 167–172, 531
 - getRemoteUser() method in
 - HttpServletRequest interface, 92, 135–136, 531
 - getUserPrincipal() method in
 - HttpServletRequest interface, 91–92, 135–136, 531
 - identity information retrieval, 91–92
 - isCallerInRole() method in
 - EJBContext interface, 90–91, 167–172, 531
 - isUserInRole() method in
 - HttpServletRequest interface, 91, 137–139, 531
 - logins, 142–143
 - principal information, 135–136
 - proactive authorization, 92–94
 - proactive/reactive authorization, 136–140
 - sign-on, EIS, 62
 - SSL, certificates and cipher suites, 140–142
- Protected members, 205–206
- Protection matrix, 84, 533
- ProtectionDomain class
 - basics, 265–266
 - inheritance, 285–286
 - and Permission class, 266–267
 - removing duplicates, 286–287
- Provider class, 383–388
- Proxy servers
 - caching reverse, 195–196
 - connect methods, 49
 - firewalls, 40–42
 - reverse, 193
 - SOCKS gateways *versus* proxy servers, 44
- Pseudorandom Number Generator. *See* PRNG
- Public-key cryptography. *See also* PKCS
 - basics, 345, 359
 - combining with secret-key, 375–376
 - DH (Diffie-Hellman), 362–363

digital certificates, 372–375
 digital signatures, basics, 370–371
 digital signatures, DSA, 372
 digital signatures, RSA, 371
 elliptic curve, 364–367
 RSA, 360–362, 371
 Public-Key Cryptography Standards. *See*
 PKCS
 Public Key Infrastructure. *See* PKI
 Public members, 206
 PublicKey interface, 393
 Publish/subscribe model, JMS (Java Messaging
 Service), 63

Q

QoS (quality of service), secure
 communication, 95

R

RACF (Resource Access Control Facility)
 authentication and authorization services,
 18
 WAS security environment, 36
 RDBMS (relational database management
 system), JDBC and SQLJ support, 20
 Relational database management systems. *See*
 RDBMSs
 Remote interface, EJB specification, 68
 Remote Method Invocation. *See* RMI
 Remote Method Invocation over Internet Inter-
 Object Request Broker Protocol. *See*
 RMI-IIOP
 removeProvider() method in Security
 class, 388
 Replay attacks, 150
 Request objects, servlets, 109
 RequestDispatcher interface, invocation
 chains, 126
 res-auth XML element, 87
 Resource Access Control Facility. *See* RACF
 Resource adapters
 Java Connector Architecture, 11–12
 media access controls, 38
 three-tier computing models, 191–192
 Response objects, servlets, 109
 Reverse proxy servers, 193
 Rijndael block cipher, 353
 Rivest-Shamir-Adleman. *See* RSA

RMI-IIOP (Remote Method Invocation over
 Internet Inter-Object Request Broker
 Protocol)
 communication with component software,
 15
 distributed systems, 20
 EJB interoperability, 158–159
 security, 551
 support for OMG CORBA model, 16

RMI (Remote Method Invocation)
 basics, 547–548
 communication with component software,
 15
 connections through firewalls, 49–51
 registry, 549–550
 security issues, 550–551
 RMIClassLoader class, 220
 RMISecurityManager class, 551
 role-link element, 93, 180

Role references
 basics, 67, 138
 EJB Application Assembler platform role,
 64–69, 73, 84, 176–181

Root certificates, 373–374

RSA (Rivest-Shamir-Adleman) public-key
 cryptography
 basics, 360–362, 371
 PKCS#1, 435

run-as element, 133–134, 180–181

Runtime
 restrictions, for EJB components, 172–176
 restrictions, for Web components, 143–145
 security enforcement, 73

RuntimePermission class, 241

S

S/MIME (Secure/Multipurpose Internet Mail
 Extensions)
 overview, 439
 PKCS, encrypting transactions with, 445
 PKCS, signing/verifying transactions with,
 439–445
 security technologies, 8–9

SAML (blending of AuthML and S2ML),
 502–503, 504

SASL (Simple Authentication and Security
 Layer), 340

SDK (Software Development Kit). *See* J2SDK

- SealedObject class, 423–424
- Secret-key cryptography
 - basics, 345–346
 - block ciphers, CBC mode of operation, 353–355
 - block ciphers, DES, 351
 - block ciphers, ECB mode of operation, 353–354
 - block ciphers, Feistel, 350–351
 - block ciphers, IDEA, 353
 - block ciphers, Rijndael, 353
 - block ciphers, Triple-DES, 352
 - combining with public-key, 375–376
 - data confidentiality, 356
 - Java security support, 8–9
 - key space, 355
 - stream ciphers, 350
 - substitutions and transpositions, 346–348
 - XOR (eXclusive OR) operation, 348–350
- SecretKey interface, 421
- SecretKeyFactory class, 391, 422–423
- SecretKeySpec class, 422
- Secure-channel constraints, authentication, 82
- Secure/Multipurpose Internet Mail Extensions. *See* S/MIME
- Secure Sockets Layers. *See* SSL
- SecureClassLoader class, 220–221
- SecureRandom class, 392–393
- Security audit trails, 73
- security-constraint XML element, authorization, 82–83
- security-identity XML element, 86
- Security-mechanism agnostic applications, 75
- security-role-ref element, 138–140
- Security roles, 73–76. *See also* EJB roles
- Security Services Markup Language. *See* S2ML
- Security subjects, 84
- Security unaware applications, 75
- SecurityManager class
 - and AccessController class, 273–274
 - attacks, levels, 237–238
 - attacks, types, 242–246
 - basics, 267–270
 - checkPermission() method, 267–270
 - cycle stealing attacks, 245–246
 - definition, 204
 - duties, 238–240
 - extensions, 246–252
 - interdependence with class loaders and class file verifier, 252
 - methods, 239
 - operation, 240–242
- SecurityPermission class, 258, 261
- Server-side Java, basics, 32–33
- service() method, servlets, 108–111
- Service provider interface. *See* SPI
- Servlet class, 107–111
- Servlet-tag technique, 105–106
- <SERVLET> tags, 105–106
- ServletRequest interface
 - programmatic security, 140–141
 - SSL attributes, 140–142
- Servlets, runtime restrictions, 143–145
- Servlets
 - advantages, 105–107
 - basics, 13
 - versus CGI programs, 106–107
 - filters, 151–154
 - HTTPS connections to external HTTP servers, 145–147
 - identify information retrieval, 91–92
 - pooling, 13
 - pre- and post-servlet processing, 150–154
 - proactive authorization, 92–93
 - process flow sequence, 103
 - server-side technology, 32–33
 - servlet deployment descriptors, 13, 59–60
 - servlet forwards and includes, 126
 - WAS components, 33–34
 - Web modules, 60
- Session beans, 58–59, 161
- SessionBean interface, 171
- SHA-1 (Secure Hash Algorithm V1), 357, 381
- sign() method in Signature class, 407
- Signature class, 380–381, 389, 400–412
- SignatureSpi class, 380–381
- SignedData object, 440–445
- SignedObject class, 413–414
- Simple Authentication and Security Layers. *See* SASL
- Simple Mail Transfer Protocol. *See* SMTP
- Simple Object Access Protocol. *See* SOAP
- Single sign-on. *See* SSO
- Singleton patterns, 107

INDEX

- Skeletons
 - EJBs, 159
 - RMI, 549
 - SMTP (Simple Mail Transfer Protocol), e-mail impersonation, 246
 - SNA (Systems Network Architecture)
 - CICS client/server applications, 27–28
 - three-tier architecture, 31
 - SOAP (Simple Object Access Protocol)
 - basics, 499–500, 509
 - binding with HTTP, 12, 499
 - communication with component software, 15
 - Java support for standards, 20
 - Socket class, 241
 - SOCKEt Secure. *See* SOCKS
 - SocketPermission class, 239, 258
 - SOCKS (SOCKEt Secure)
 - firewalls, 19–20, 40, 42–44
 - gateways *versus* proxy servers, 44
 - Source/destination addresses, TCP/IP packets, 38, 39
 - SPI (service provider interface), 380–381
 - SQLJ (Structured Query Language for Java)
 - Java support for standards, 20
 - resource adapters, 12
 - SSL (Secure Sockets Layer)
 - authentication, certificate-based, 120–122, 140–142, 456–457
 - authentication, protecting communication channels, 459
 - authentication, protecting user IDs and passwords, 455–456
 - authentication, reverse proxy servers, 457
 - authentication, SSO, certificate-based, 458–459
 - authentication, SSO, cookie-based, 457–458
 - authentication, WAS mutual authentication, 457
 - authorization, user data constraints, 132–133
 - basics, 449–451
 - example, with SSL, with no authentication, 469–476
 - example, with SSL, with server and client authentication, 484–493
 - example, with SSL, with server authentication, 476–484
 - example, without SSL, 463–468
 - handshake protocol, 120, 452–454
 - HTTP and SSL sessions, 149–150
 - JSSE, 460–461
 - QoS requirements, 95–97
 - record protocol, 451
 - secure communication, 16–17
 - secure communication, combinations, 96–97
 - support for standards, 20
 - support via JSSE, 8–9, 460–461
 - trust managers, 461
 - truststores, 461–462
 - SSO (single sign-on)
 - authentication, 531–532
 - basics, 123–124
 - SSL, certificate-based authentication, 458–459
 - SSL, cookie-based authentication, 457–458
 - Stream ciphers, 350
 - Structured Query Language for Java. *See* SQLJ
 - Stubs
 - EJB, 159
 - RMI, 548, 549
 - S2ML (Security Services Markup Language), and SAML, 502–503
 - Substitutions, secret-key cryptography, 346–348
 - System Administrator platform role, 64–66, 71–72, 75, 83, 87, 182
 - System domain, 266, 287, 538–540
 - System-level contracts, resource adapters, 61
 - Systems Network Architecture. *See* SNA
- ## T
- Targets, permissions, 258–259
 - TCP/IP (TCP/Internet Protocol)
 - firewalls, 37–39
 - headers, 37–39
 - CICS client/server applications, 27–28
 - and SOCKS, 19
 - TCP (Transmission Control Protocol), transport layer headers, 38–39
 - Thick/thin clients, 189
 - Three-tier computing models. *See also* Four-tier computing models
 - basics, 10–12, 25–29, 188–189
 - business-logic layer, 191

Three-tier computing models (*continued*)
 client-side access, 189
 gateway software guidelines, 29
 JSP, 13–14
 legacy applications, 191–192
 presentation layer, static and dynamic
 content, 190
 resource adapters, 191–192
 servlet containers and pooling, 13
 servlet deployment descriptors, 13

TLS (Transport Layer Security)
 basics, 449–451
 support via JSSE, 8–9, 460–461

Transmission Control Protocol. *See* TCP

Transparent key representation, 390, 413

Transport layer, TCP/IP packet headers, 38–39

Transport Layer Security. *See* TLS

Transpositions, secret-key cryptography,
 346–348

Triple-DES block ciphers, 352, 359

TRNG (true random number generator), 392

Trust managers/*TrustManager* interface,
 461

TrustManagerFactory class, 476, 484–485

Truststores, 461–462

Two-tier architecture, 23–25

Type-safe code, *versus* non-type-safe code,
 5–7

U

UDP (User Datagram Protocol), transport layer
 headers, 38–39

unchecked XML element, 184

Uniform resource locator. *See* URL

url-pattern element, 128

URLClassLoader class, 220–221

URL (uniform resource locator), protection in
 authorization, 127–130

use-caller-identity XML element, 82

user-data-constraint XML element
 authorization over SSL, 132–133
 basics, 82

User Datagram Protocol. *See* UDP

V

verify() method in *Signature* class, 407,
 413–414

Virtual hosting/virtual hosting providers, 154

W

WAR (Web Archive), J2EE applications, Web
 modules, 57, 58, 59–60

WAS (Web application server)
 client and server objects, 35–36
 EJB containers, 34–35
 partitioning applications, 154–155
 secure communication, 95–97
 security environment, 35–36
 servlet containers, 33–34
 Web containers, 33–34

Web application server. *See* WAS

Web Archive. *See* WAR

Web modules
 deployment descriptors, 111–112
 WAR files, 57, 58, 59–60

web-resource-collection XML element,
 127–128

Web Services
 application patterns, browser-to-server,
 517
 application patterns, example, 518–519
 application patterns, modular design,
 517–518
 authorization enforcement, 523–525
 Java support for standards, 20
 messaging security model, 508, 510–511
 proof-of-possession claims, 511
 resource adapters, 12
 security, reasons needed, 501–502
 security, transport-protocol-specific and
 -agnostic handlers, 520
 security-token services, 508
 signed security tokens, 510
 WS-Addressing, 500
 WS-Authorization, 509, 510, 515
 WS-Federation, 509, 510, 515
 WS-Policy, 509, 511–513
 WS-PolicyAssertions, 512
 WS-PolicyAttachment, 512
 WS-Privacy, 509, 510, 515
 WS-SecureConversation, 509, 510,
 514–515
 WS-Security, 503, 504, 507–510
 WS-Security, authentication, 520–523
 WS-Security, example, 516–517
 WS-SecurityPolicy, 512, 513
 WS-Trust, 509, 510, 513–514

INDEX

- Web Services Addressing specification. *See* WS-Addressing
 - Web Services Description Language. *See* WSDL
 - Web Services General Policy Assertions Language specification. *See* WS-PolicyAssertions
 - Web Services Message Security, 503
 - Web Services Policy Attachment specification. *See* WS-PolicyAttachment
 - Web Services Policy Framework specification. *See* WS-Policy
 - Web Services Secure Conversation Language specification. *See* WS-SecureConversation
 - Web Services Security Assertions, 503
 - Web Services Security Policy Language specification. *See* WS-SecurityPolicy
 - Web Services Security specification. *See* WS-Security
 - Web Services Trust Language specification. *See* WS-Trust
 - WebSphere MQ (Message Queuing), IBM
 - JMS connections, 199
 - resource adapters, 12
 - WS-Addressing (Web Services Addressing specification), 500
 - WS-Authorization (Web Services Authorization specification), 509, 510, 515
 - WS-Policy (Web Services Policy Framework specification), 509, 511–513
 - WS-PolicyAssertions (Web Services General Policy Assertions Language specification), 512
 - WS-PolicyAttachment (Web Services Policy Attachment specification), 512
 - WS-SecureConversation, (Web Services Secure Conversation Language specification), 509, 510, 514–515
 - WS-Security (Web Services Security specification), 503, 504, 507–510
 - authentication, 520–523
 - example, 516–517
 - WS-SecurityPolicy (Web Services Security Policy Language specification), 512, 513
 - WS-Trust (Web Services Trust Language specification), 509, 510, 513–514
 - WSDL (Web Services Description Language), 500–501
- X**
- X.509 (certificate-based) authentication
 - basics, 78, 81–82, 114, 120–122
 - certificate contents, 553–554
 - international standard, 372
 - SSL, 140–142
 - versions, 554
 - XACML (eXtensible Access Control Markup Language), 504
 - X509Certificate class, 415
 - X509CRL class, 415
 - X509CRLEntry class, 415
 - X509Extension interface, 415
 - XKMS (XML Key Management Specification), 504
 - XML Digital Encryption standard, 503, 504
 - XML Digital Signature standard, 503, 504–505
 - XML Encryption standard, 503, 504–505
 - XML (eXtensible Markup Language)
 - basics, 498–499
 - and cryptography, 503–505
 - Java support for standards, 20
 - XML Key Management Specification. *See* XKMS
 - XML Signature standard, 502, 504–505
 - XOR (eXclusive OR) operation, 348–350