Index

A
accelerators, 57–58
acceptable risks, documenting, 103, 114
adding customers, example code, 229–231
administrators
reasons for database security, 10
separation of duties, 122–123
Advanced Encryption Standard (AES), 20, 62–63, 163
algorithms. See engines and algorithms
aliases. See key aliases
American Express requirements, 15
APIs (application programming interfaces)
engine interface, 87
manifest interface, 87
service interface, 87
weak APIs, testing, 145
application crackers, reasons for database
security, 10
application penetration testing. See testing
application programming interfaces. See APIs
application tier, defined, 38
architecture, 40–41
asymmetric cryptography, 20–22
attacks against cryptosystems, 27–28
attack surfaces, 120–121
classifications of attackers, 113
provider subcomponents, attacks
against, 88
automated inspections of code and
binaries, 138

B
backup and restore
keys and key vaults, 76–77, 83
standards, 112
batch credit card payment processes, 29
block ciphers, 20
buffer overflows, common threats, 109
building and passing statements to
databases, 18
building cryptosystems, security-enhanced
project methodology, 93–156
Index

business analysts
managing security-enhanced cryptographic projects, defining roles, 100–101
requirements review, 110–112

c
California Information Practices Act, 14
CBC (Cipher Block Chaining) Mode, 63–64
changing and testing keys, 37–38
Children’s Online Privacy Protection Act, 14–15
chosen-plaintext attacks, 28
cipher, defined, 19
Cipher Block Chaining (CBC) Mode, 63–64
ciphertext
  defined, 19
  known-ciphertext attacks, 27–28
classes of threats, 109
classifications of attackers, 113
code samples (Java version 1.4.2), 157–254
column-spanning, 45
command injections, common threats, 109
command lines as entry points, 120–121
comment tokens, testing input, 143
common security risks, 109
  list, 98
  testing requirements, 144–145
common services, sample code, 160–163
components of applications
  failure, 124
  illustration of logical components (three data stores and four processes), 41
  initializers, 87–90
  key managers, 81
  separation of duties, 122–123
compound receipts
  defined, 69
  multiple alias ID exception, 243
confidentiality breaches. See privacy and confidentiality
configuration files as entry points, 120–121
consumers. See providers and consumers
containment of security incidents, 154
cookies as entry points, 120–121
“copy-and-waste” code, 99
corporate compliance agreements, 11–12, 15
Counter (CTR) Mode, 67–69
covet channels, testing, 148
CPUs dedicated to cryptography. See dedicated engines
cracking databases, reasons for, 3
credentials and permissions
  automated processes, 29
design problems, 128
development phase of security-enhanced cryptographic projects, 135
digital signatures, 21–22
high-level requirements, 106
key managers, 80, 195
network intruders, 10
obfuscation, 31
transparent encryption, 29, 31, 33
See also privileges
Index

credit cards
  batch credit card payment processes, 29
  designing security, 131
  example code, 225–226
  Payment Card Industry (PCI) Data Security Standard, 15
cross-application event managers, 155
cross-site scripting, common threat, 109
cryptographic architecture, 40–41
cryptographic consumers. See providers and consumers
cryptographic engines and algorithms. See engines and algorithms
cryptographic hashing, 12–20
cryptographic infrastructure. See infrastructure
cryptographic keys. See keys
cryptographic providers. See providers and consumers
cryptographic receipts. See receipts
cryptography, defined, 19
culture of security, 96–97
customers
  customer manager, 226–240
  customer not found exception, 243–244
  example code, 223–225
  managing security-enhanced cryptographic projects, 97
  working with customer information, prototype of database encryption system, 248–249

data conversions, example code, 160–163
data definition language (DDL), defined, 18
Data Encryption Standard (DES), 20
data flow diagrams (DFDs), designing security, 118–120
"data input" interactions, 107
data integrity. See integrity of data
data sanitization. See sanitizing inputs and outputs
data sensitivity grades, 114
database tier, defined, 38
decision trees, designing security, 124–125
decommissioning applications, 155–156
decryption
  defined, 19
  example code
    decrypting business data, 218–219
    providers, decryption results, 213–214
  receipts necessary for decryption. See receipts
dedicated engines, 57, 60–61
  key protection as primary purpose of, 29, 71
default behaviors, designing for, 124–125
defense of application
  after deployment, 153–155
  design decisions, 125
defensive threats, 109
Index

definitions
application tier, 38
availability of data, 4
block ciphers, 20
ciphers, 19
ciphertext, 19
compound receipt, 69
confidentiality of data, 4
customer, 89
cryptographic consumer, 40
cryptographic engine, 40
cryptographic hashes, 19
cryptographic provider, 40
cryptography, 19
data definition language (DDL), 18
database tier, 38
decryption, 19
dedicated engines, 57
expired keys, 47
foreign keys, 18
Hardware Security Model (HSM), 57
indexes, 18
information leakage, 51
integrity of data, 4
key alias, 55
key family, 43
key manager, 40
key manifest, 40, 55
key vault, 40
key zone, 80
keys, 17–18, 40
live keys, 47
local engines, 57
message digest, 22
obfuscation, 30
pending keys, 47
presentation tier, 38
primary keys, 17
profile, 131
protected data, 40
provider, 87
receipt table, 89
relational database, 17
retired keys, 47
scrambling, 19
stored procedures, 184
stream ciphers, 20
structured query language (SQL), 18
terminated keys, 47
threat model, 9
vulnerabilities, 146
zeroing, 75
deletions
delete statements, 18
hardware, deleting data from, 155–156
keys, deleting. See key managers
deployment of applications
main discussion, 151–152
deployment profile, defense of applications, 153
separation of duties, 152
designing security
main discussion, 117–132
attack surfaces, 120–121
checking design against security standards, 117
component failure, 124
context diagrams, 119
credit card numbers, 131
data flow diagrams (DFDs), 118–120
decision trees, 124–125
default behavior, 124–125
defense plan, 125
design guidelines, 120–125
e-mail addresses, 131
ever points, 120–121
event logging, 125
exceptional states of application, 125
failing securely, 124
goals, 117
guideline approach, 118
layers of security, 123–124
names, 131–132
new functionality to address common threats, 144–145
phone numbers, 131
practical implementations, 157–254
privileges, 121
requirements
review, 110–112
See also requirements, documenting
roles in managing security-enhanced cryptographic projects, 100–101
searching and profiles, 130–132
security analyst’s role, 117–118
security patterns, applying to weak spots, 118
self-monitoring systems, 124
separation of duties, 122–123
SHA-1-based profiles, 131
social security numbers, 131
threat modeling, 118, 125–127

developers
role of. See programmers

security-enhanced development.
See development phase of security-enhanced projects
development phase of security-enhanced projects
main discussion, 133–140
automated inspections of code and binaries, 138
input type, size, and composition, 134
language security guides, 139–140
logging security events, 137–138
manual security inspections of code and binaries, 138
permissions issues, 135
platform security guides, 139–140
privilege management, 135–136
protocols, objects sent via, 135
RMI protocols, objects sent via, 135
sanitizing inputs and outputs, 134–135
security analyst’s role, 133–134
security guides, 133, 139–140
security inspections of code and binaries, 138
SOAP protocols, objects sent via, 135
testing functions and procedures, 138–139
unit tests, 138–139
URLs as input, 135
values read from files, 135
variables containing input data, 134–135
wiping data from memory, 136–137
workshop approach, 134
zeroing, 137
digital signatures, 21–22
downloading databases, classifications of attackers, 113
Index

E

e-mail addresses
designing security, 131
test code, 221–242

electronic code book (ECB) mode, 17, 65, 167

engines and algorithms
main discussion, 57–69
accelerators, 57–58
Advanced Encryption Standard (AES), 62–63
APIs (application programming interfaces), engine interface, 87
Cipher Block Chaining (CBC) Mode, 63–64
Counter (CTR) Mode, 67–69
dedicated engines. See dedicated engines
definition of cryptographic engine, 40
test code, 163–164

main discussion, 209–212
EngineWrapper, 209, 210
local engine, 209, 210

Federal Information Processing Standard (FIPS) 140–2 [18], 61–62
implied access to vaults from access to engines, 72
infrastructure, 40–41
initialization vectors (IVs), 27, 63, 69
key manifest, 77
key vaults, access to, 71
local engines, 57–59, 72
memory attacks, 59–60
obfuscation, 60
providers and consumers, engine interface, 87
SHA algorithms, 23
standards, 112
symmetric algorithms, 62–63
test code, 141
entrance points
designing security, 120–121
test code, 142
erasing data
from hardware, 155–156
zeroing. See zeroing
See also deletions

events. See logging and monitoring security events
exception handling, 241–244
expense of encryption, 113
expired keys, 47, 53–54, 78
external policies requiring encryption, reasons for database security, 11
extranets, 38–39
requirements documents, locations of stored data, 114

F

failing securely, design decisions, 124
false positives, defense of application after deployment, 153
Federal laws and regulations. See United States
filtering and validating output, 107
Financial Modernization Act of 1999, 13
firewalls, 38
flexibility of infrastructure, 40
foreign keys, defined, 18
Index

form fields, designing security, entry points, 120–121
format string overflows, common threats, 109
functional security testing, 142–146
functional threats, 109

G
government regulations. See United States Gramm-Leach-Bliley Act (GLBA), 13, 15
guidelines
design guidelines, 118, 120–125
development guides, 133, 139–140
See also standards and policies

H
hacking databases, reasons for, 3
hardware issues
decommissioning applications, 155–156
dedicated engines. See dedicated engines
Federal Information Processing Standard (FIPS) 140–2 [18], 61–62
keys, locking in dedicated tamper-proof hardware, 29
security model. See Hardware Security Model (HSM)
separate CPUs dedicated to cryptography. See dedicated engines
Hardware Security Model (HSM)
defined, 57
example engine and key vault, 163
standards, 112
“virtual HSM,” 159, 164
hashes
ciphertext, 19
cryptographic hashes, 19
SHAs (secure hash algorithms), 23, 131
Health Insurance Portability and Accountability Act (HIPAA), 12–13
hexadecimal strings, keys stored as, example code, 165, 173
hidden fields, designing security, entry points, 120–121
honeycombing, 29
HSM. See Hardware Security Model (HSM)
HTTPS encryption, 3
implementation bugs, 27
indexes, defined, 18
information leakage, defined, 51
infrastructure, 35–92
algorithms. See engines and algorithms
budget issues, 40
cryptographic keys. See keys
general. See engines and algorithms
engines. See engines and algorithms
firewalls, 38
flexibility, 40
illustration of logical components (three data stores and four processes), 41
information leakage, 51
large data sets and key fatigue, 52
location of consumers, 40
modularity, 40
receipts, 41–42, 46, 48–49, 51, 53–54
tiers, 38–40
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>initialization vectors (IVs), 27, 63, 69</td>
</tr>
<tr>
<td>functions of initializers, 87</td>
</tr>
<tr>
<td>testing, 145</td>
</tr>
<tr>
<td>initializers, as provider subcomponents, 87–90</td>
</tr>
<tr>
<td>input data</td>
</tr>
<tr>
<td>data sanitization. See sanitizing inputs and outputs</td>
</tr>
<tr>
<td>event logging, input flaws, 143</td>
</tr>
<tr>
<td>testing, 142</td>
</tr>
<tr>
<td>type, size, and composition, 134, 143</td>
</tr>
<tr>
<td>URLs as input, 120–121, 135, 143</td>
</tr>
<tr>
<td>variables containing input data, 134–135, 143</td>
</tr>
<tr>
<td>insert statements, 18</td>
</tr>
<tr>
<td>integer overflows, common threats, 109</td>
</tr>
<tr>
<td>integrating security into overall project plans, 93–156</td>
</tr>
<tr>
<td>integrity of data</td>
</tr>
<tr>
<td>applying cryptography, 25–26</td>
</tr>
<tr>
<td>attacks against, reasons for database security, 6–8</td>
</tr>
<tr>
<td>defined, 4</td>
</tr>
<tr>
<td>message authentication codes (MACs), 26</td>
</tr>
<tr>
<td>reasons for database security, 6–8</td>
</tr>
<tr>
<td>See also sanitizing inputs and outputs</td>
</tr>
<tr>
<td>interfaces</td>
</tr>
<tr>
<td>APIs. See APIs</td>
</tr>
<tr>
<td>engine interface, 87</td>
</tr>
<tr>
<td>manifest interface, 87</td>
</tr>
<tr>
<td>service interface, 87</td>
</tr>
<tr>
<td>small computer serial interface (SCSI) connections, 57</td>
</tr>
<tr>
<td>Internet addresses as input, 120–121, 135, 143</td>
</tr>
<tr>
<td>intranets, 38–39</td>
</tr>
<tr>
<td>requirements documents, locations of stored data, 114</td>
</tr>
<tr>
<td>invalid key state exception, keys, 241–242</td>
</tr>
<tr>
<td>IP addresses, log testing, 144</td>
</tr>
<tr>
<td>IVs. See initialization vectors (IVs)</td>
</tr>
<tr>
<td>J</td>
</tr>
<tr>
<td>join statements, 18</td>
</tr>
<tr>
<td>K</td>
</tr>
<tr>
<td>Kerberos protocol, key vaults, 73</td>
</tr>
<tr>
<td>Kerckhoffs’ Principle, 19, 62</td>
</tr>
<tr>
<td>“key admin” interactions, 107</td>
</tr>
<tr>
<td>key. See keys</td>
</tr>
<tr>
<td>key aliases, 55</td>
</tr>
<tr>
<td>example code, 183–189, 191</td>
</tr>
<tr>
<td>exceptions, 240</td>
</tr>
<tr>
<td>in key manifest, 77</td>
</tr>
<tr>
<td>See also key manifests</td>
</tr>
<tr>
<td>key zones, 80</td>
</tr>
<tr>
<td>multiple alias ID exception, 243</td>
</tr>
<tr>
<td>providers and consumers, 87–90</td>
</tr>
<tr>
<td>key families, 43–44</td>
</tr>
<tr>
<td>in key manifest, 77</td>
</tr>
<tr>
<td>one-to-one relationship between families and protected columns, 44</td>
</tr>
<tr>
<td>providers and consumers, 87–90</td>
</tr>
<tr>
<td>setting key families, 82</td>
</tr>
<tr>
<td>standards, 112</td>
</tr>
<tr>
<td>key managers</td>
</tr>
<tr>
<td>access credentials, 80</td>
</tr>
<tr>
<td>building once, 99</td>
</tr>
<tr>
<td>core subcomponents, 81</td>
</tr>
</tbody>
</table>
Index

credentials, 80, 195
defined, 40
textbook code, 195–208
how vaults, manifests, and managers
work together, 71
key zones, 80
single or few key managers,
manageability and consistency of,
99–100
dystems, 80
example code, 195–208
how vaults, manifests, and managers
work together, 71
key zones, 80
single or few key managers,
manageability and consistency of,
99–100
key manifests
compromised keys, 79–80
cryptographic architecture, 40–41
defined, 40, 55
textbook code, 183–194
how vaults, manifests, and managers
work together, 71
manifest interface, 87
online attacks, 79
provider, interaction with, 87
timeline indicating relationship between
states and activation dates, 78
See also key aliases
key masks, 74
key stores. See key vaults
key vaults
main discussion, 71–77
backup and restore, 76–77
cryptographic architecture, 40–41
defined, 40
textbook code
main discussion, 165–181
accessing local keys, 179–180
Advanced Encryption Standard
(AES), 163
compromised keys, replacing
key-encrypting keys, 176–179
creating new aliases, 186
determining key states, 192
electronic code book (ECB), 167
encrypting keys, 172–175
generating key-encrypting keys, 199
generating new key-encrypting keys,
170–171
hexadecimal strings, keys stored as,
165, 173
key aliases, 183–189, 191
key managers, 195–208
key manifests, 183–194
key tool, 195–199
live key-encrypting key, 166
loading new keys into key stores, 199
local key stores and LocalKeyStore class
courier, 169–178
local keys and LocalKey class courier,
166–168
manual zeroing, 173–174
optimized state checks, 193
periodic key replacement, 176–179
reading aliases from manifests,
188–189
reading the current live key, 189–191
replacing key-encrypting keys, 176–179
retiring keys, 202
saving aliases, 191
saving keys to key stores, 175–176
SecretKeySpec courier, 166–167
terminating keys, 203
updating pending keys, 204–207
viewing keys, 200–202
wipe courier, 173–174
how vaults, manifests, and managers
work together, 71
### Index

implied access to vaults from access to engines, 72
Kerberos protocol, 73
key masks, 74
key servers, 73
key stores, 74–76
master key, 73, 112
obfuscation keys, 74
protecting, 73–76
protocols, 73
services provided by, 72
SSL (Secure Sockets Layer) or TLS (Transport Layer Security), 73–74
zeroing, defined, 75

**key zones, 80**  
**keys**
main discussion, 42–55
activation date, in key manifest, 77
activation dates, 83
backup, 83
changing and testing, 37–38
changing and testing keys, 37–38
column-spanning, 45
creating, 82
dedicated engines for key protection, 29, 71
defined, 17–18, 40
deleting, and altering. See key managers encrypting
example code, 172–173, 172–175
prototype of database encryption system keys, 245, 253
example code
engine and key vault, 163–164
key vaults. See key vaults
exception when key not found, 242
expired keys, 47, 53–54, 78
families. See key families
foreign keys, 18
good security, requiring that old keys be retired and new keys brought into service, 38, 43
indirect access of keys, 29–30
information leakage, 51
invalid key state exception, 241–242
key encrypting keys, prototype of database encryption system, 245, 253
key families. See key families
key fatigue, 42–55
key IDs, 77, 83
key managers. See key managers
key manifests. See key manifests
key migration, 53
key not found exception, 242
key replacements, 54, 176–179, 249
key scopes, 48–50, 112
key separation, 42–43
key vaults. See key vaults
known-ciphertext attacks, 43
lengths, standards for, 112
life cycle of keys, 46–48, 112
listing, 82
live keys. See live keys
locking in dedicated tamper-proof hardware, 29
lost keys, 26
managers. See key managers
manifests. See key manifests
one-to-one relationship between families and protected columns, 44
## Index

pending keys. See pending keys
primary keys, 17
prototype of database encryption system, 245–249
public-key cryptography, 20–22
rekeying, 54
retiring. See retired keys
row IDs, 46
setting key families, 82
setting up keys, prototype of database encryption system, 245–248
single purpose, 42–43, 83
striping, 45–46, 112
symmetric keys, 21
terminating. See terminated keys
testing key protection, 141
vaults. See key vaults
known defects and vulnerabilities, 151
known-ciphertext attacks, 27–28
known-plaintext attacks, 28

### L
language security guides, 139–140
large data sets and key fatigue, infrastructure, 52
layers of security, design decisions, 123–124
least privilege principle, 106
legislation related to security
main discussion, 11–14
project methodology for building database cryptosystems, 97
See also United States
legitimate users, reasons for database security, 10

life cycle of keys, 46–48, 112
live keys
defined, 47
example code, 166
live key not found exception, 242
timeline indicating relationship between states and activation dates, 78
local engines
defined, 57
engines and algorithms, 57–59, 72
example code, 209–210
locations of stored data, 114
logging and monitoring security events
main discussion, 151–156
defending applications after deployment, 153–154
design decisions, 125
development phase, 137–138
input flaws, 143
log testing, 144
managing security-enhanced cryptographic projects, defining roles, 100–101
requirements, documenting, 108
requirements review, 110–112
lost keys, 26

### M
managing security-enhanced cryptographic projects
main discussion, 95–102
“copy-and-waste” code, 99
culture of security, 96–97
customer value and security, 97
defining roles, 100–101
Index

determining and mitigating security risks, 98–99
functionality of applications vs. security, 97
number of entry points to system, 100
requirements review, 110–112
single or few key managers, manageability and consistency of, 99–100
skills needed, 101
manifests. See key manifests
manual security inspections of code and binaries
development phase, 138
testing, 146
manual zeroing, 173–174
MasterCard compliance requirements, 15
memory attacks, 59–60
message authentication codes (MACs), integrity of data, 26
message digest, defined, 22
methodology, practical implementations, 157–254
mitigating strategies
defense of released applications, 154
testing, common-threats requirements, 144–145
modes, standards for, 112
modularity, infrastructure, 40
monitoring security events. See logging and monitoring security events
multiple alias ID exception, key aliases, 243
MySQL, testing of code samples, 157–254
N
names
designing security, 131–132
example code, 221–241
networks
network communication leaks, testing, 148
network connections, eavesdropping on, 3
network intruders, reasons for database security, 10
network tiers, separation of duties, 122–123
new functionality to address common threats, 144–145
O
obfuscation
main discussion, 30–31
engines and algorithms, 60
key vaults, 74
one-to-one relationship between families and protected columns, 44
operations staff
defense of application after deployment, 153–154
managing security-enhanced cryptographic projects, defining roles, 100–101
requirements review, 110–112
Index

P

packet sniffing, 3
padding schemes, 85, 87
passing statements to databases, 18
PCI (Payment Card Industry)
  PCI Data Security Standard, 15
  PCI connections in HSM communications, 57
pending keys
  defined, 47
  timeline indicating relationship between states and activation dates, 78
penetration testing
  main discussion, 146–149
  defining roles in security-enhanced cryptographic projects, 100–101
  requirements review, 110–112
periodic key replacement, example code, 176–179
permissions. See credentials and permissions
personally identifiable information. See privacy and confidentiality
plaintext attacks, 28
platform security guides, 139–140
policies. See standards and policies
practical implementations of design and methodology, 157–254
presentation tier, defined, 38
primary keys, 17
privacy and confidentiality
  applying cryptography, 23–25
  Children’s Online Privacy Protection Act, 14–15
  common threats, 109
  credit card numbers, 131
  definition of confidentiality, 4
  e-mail addresses, 131
  example code, personally identifiable information, 221–240
  legislation related to privacy, 11–14
  names, 131
  phone numbers, 131
  reasons for database security, 5–6
  requirements, documenting, 109–110
  SHA-1-based profiles, 131
  social security numbers, 131
  testing, 145
privileges
  design phase, 121
  development phase, 135–136
  digital signatures, 21–22
See also credentials and permissions
profiles
  defined, 131
  designing security, searching and profiles, 130–132
program memory, testing, 148
programmers
  in development phase. See development phase of security-enhanced projects
  reasons for database security, 10
  separation of duties, 122–123
project managers
  managing security-enhanced cryptographic projects, defining roles, 100–101
  requirements review, 110–112
project methodology for building database cryptosystems, 93–156
protected data, defined, 40
Index

prototype of database encryption system, 245–254

providers and consumers
main discussion, 85–91
attacks against provider subcomponents, 88
cryptographic architecture, 40–41
definition of consumer, 89
definition of provider, 87
encoder, 87
engine interface, 87
example code
main discussion, 213–240
adding customers, 229–231
credit card information, 225–226
customer information, 223–225
customer manager, 226–240
decrypting business data, 218–219
encrypting business data, 217–218
encryption requests and decryption results, 213–214
key replacements, 236–240
receipts, 214–217
replacing keys, 219–220
searching for customers, 235–236
viewing customer records, 231–235
initializers, 87–90
key aliases, 87–90
key families, 87, 89–90
manifest interface, 87
padding, 85, 87
receipts necessary for decryption.
See receipts
service interface, 87
strict adherence to secure coding practices, why required, 88, 90
third-party applications, base tables as part of, 90

public-key cryptography, 20–22

Q
quality assurance testing. See testing
queries, testing input, 143
quick encryption solutions, 38
quote characters, testing input, 143

R
reads, database reads as entry points, 120–121
reasons for database security
main discussion, 3–16
administrators, 10
application crackers, 10
availability attacks, 8–9
corporate compliance agreements, 11–12, 15
developers, 10
external policies requiring encryption, 11–12
integrity attacks, 6–8
legitimate users, 10
network intruders, 10
privacy and confidentiality, 5–6, 11–14
reputation, damage to, 11–12, 15
thieves, 11
threat models, 8–9
three principles of security, 4
trade regulations, 11–12, 15
types of attacks, 4–5
See also risks and vulnerabilities
Index

receipts
compound receipts, multiple alias ID exception, 243
example code, 214–217
infrastructure, 42, 46, 48–49, 51, 53–54
log testing, receipt information, 144
receipt manager, 87–88
receipt tables, 89–90
testing, 141
See also providers and consumers
recovery procedures in defense plan, 154
rekeying, 54
relational database, defined, 17
released software, security considerations for, 151–156
Remote Method Invocation (RMI) calls, entry points, 120–121
replacing keys
example code, 219–220
key-encrypting, 176–179
reputation, damage as reason for database security, 11–12, 15–16
requirements, documenting
main discussion, 103–115
acceptable risks, 103, 114
access controls, 106
classes of threats, 109
confidentiality of data, 109–110
“data input” and “key admin” interactions, 107
data sanitization standards, 107
data sensitivity grades, 115
defensive threats, 109
expense of encryption, 113

filtering and validating output, 107
functional threats, 109
key vaults, 107
least privilege principle, 106
locations of stored data, 114
logging and monitoring security events, 108–109
policies specifying organizational security goals, 105–106
requirements review, 110–112
sensitive tags on data, 113
separation of duties, 106
single-sign-on, 106
three-tier data classification, 113–114
restore. See backup and restore
repaired keys, 83
defined, 47
example code, 202
timeline indicating relationship between states and activation dates, 78
risks and vulnerabilities
associated with cryptography, 26–27
definition of vulnerability, 146
determining and mitigating security risks, 98–99
list of common security risks, 98
specifying acceptable security risks, 103, 114
testing, 146
threat model. See threat model vulnerability matrix, 149
See also reasons for database security
RMI (Remote Method Invocation)
entry points, RMI calls, 120–121
objects sent via development phase, RMI protocols, 135
Index

roles in managing security-enhanced cryptographic projects, application designers, 100–101
row IDs, keys, 46

development phase, 133–134
testing, determining sufficiency, 141
security events. See logging and monitoring security events
security guides, 133, 139–140
security inspections of code and binaries during development phase, 138
security officers, access to key vaults, 71
self-monitoring systems, designing security, 124
sensitive tags on data, 113
separate CPUs dedicated to cryptography. See dedicated engines
separation of duties, 106, 122–123
deployment, 152
user acceptance and quality assurance testing, 141
service interface, 87
session hijacking, common threats, 109
SHAs (secure hash algorithms), 23
SHA-1-based profiles, designing security, 131
signatures, digital. See digital signatures
single purpose, keys, 42–43, 83
single-sign-on, 106
small computer serial interface (SCSI) connections, 57
sniffing, 3
SOAP (simple object access protocol)
entry points, SOAP objects, 120–121
objects sent via development phase, 135
SQL (structured query language)
defined, 18
Index

MySQL, testing of code samples, 157–254
SQL injection, common threats, 109
SQL queries, testing, 143
SSH encryption, 3
SSL (Secure Sockets Layer), 57, 73–74
standards and policies
main discussion, 103–115
backup and restore, 112
checking design against security standards, 117
for data erasure, 156
documenting. See requirements, documenting
engines and algorithms, 112
external policies requiring encryption, reasons for database security, 11–12
federal standards. See U.S. government standards
Hardware Security Model (HSM), 112
key families, 112
organizational security goal policies, 105–106
See also guidelines
stored procedures, defined, 184
stream ciphers, defined, 20
striping, 45, 112
structured query language. See SQL (structured query language)
subcomponents of applications. See components of applications
suspicious behavior
defending applications after deployment, 153–154
See also logging and monitoring security events
swap files, testing, 148
symmetric algorithms, 62–63
symmetric cryptography, 20–21
symmetric keys, 21
system design. See designing security systems analysts
managing security-enhanced cryptographic projects, defining roles, 100–101
requirements review, 110–112
T
terminated keys, 83
defined, 47
example code, 203
timeline indicating relationship between states and activation dates, 78
testing
main discussion, 141–150
access control, 142
algorithms, 141
comment tokens, testing input, 143
common-threats requirements, 144–145
confidentiality of information, 145
covert channels, 148
data sanitization, 142
defining roles, testing team, 100–101
development phase, testing functions and procedures, 138–139
entry points, 142
functional security testing, 142–146
initialization vectors (IVs), 145
input, 142
input type, size, and composition, 143
key protection, 141
keys, testing and changing, 37–38
log testing, 144
manual and automated code inspections, 146
network communication leaks, 148
penetration testing, 146–148
program memory, 148
queries, testing input, 143
quote characters, testing input, 143
receipts, 141
requirements review, testing team, 110–112
sanitizing inputs and outputs, 143
security analyst’s role, 141
separation of duties, 141
SQL queries, 143
sufficiency, 141
swap files, 148
third-party HSMs, 149
threat models and penetration testing, 146
union statements, testing input, 143
URLs as input, 143
variables containing input data, 143
vulnerabilities, 146
weak APIs, 145
zeroing, 145

thieves, reasons for database security, 11
third-party HSMs, application penetration tests, 149
threat model, 8–9
defined, 9
designing security, 118, 125–126
penetration testing, 146
three principles of security, 4
tiers
infrastructure, 38–40
three-tier data classification, 113–114
timeline indicating relationship between states and activation dates, 78
TLS (Transport Layer Security), key vaults, 73–74
trade regulations
project methodology for building database cryptosystems, 97
reasons for database security, 11–12, 15
transparent encryption, securing databases with cryptography, 31–33
undoing encryption, 26
union statements, testing input, 143
unit tests, development phase, 138–139
United States
Federal Trade Commission, 15
Financial Modernization Act of 1999, 13
Gramm-Leach-Bliley Act (GLBA), 13, 15
Health Insurance Portability and Accountability Act (HIPAA), 12–13
Sarbanes-Oxley Act, 13
See also U.S. government standards
update statements, 18
updating pending keys, example code, 204–207
URLs as input, 120–121, 135, 143
Index

U.S. government standards
for data erasure, 156
Federal Information Processing
Standard (FIPS) 140–2 [18],
61–62

user acceptance testing. See testing
utilities, example code, 160–163

validating output, 107

variables containing input data
development phase, 134–135
testing, 143

vault. See key vaults

virtual HSM, 159, 164

viruses, 154

Visa compliance requirements, 15

VPN encryption, 3

vulnerabilities. See risks and
vulnerabilities

W
Web addresses and Web forms as input,
120–121, 135, 143

wiping data from memory. See zeroing
workshop approach, development
phase, 134

worms, 154

Z

zeroing
defined, 75
development phase, 136–137
example code, 173–174
testing, 145
Register
Your Book
at www.awprofessional.com/register

You may be eligible to receive:
• Advance notice of forthcoming editions of the book
• Related book recommendations
• Chapter excerpts and supplements of forthcoming titles
• Information about special contests and promotions throughout the year
• Notices and reminders about author appearances, tradeshows, and online chats with special guests

Contact us
If you are interested in writing a book or reviewing manuscripts prior to publication, please write to us at:

Editorial Department
Addison-Wesley Professional
75 Arlington Street, Suite 300
Boston, MA 02116 USA
Email: AWPro@aw.com

Visit us on the Web: http://www.awprofessional.com
The Executive Guide to Information Security
Mark Egan with Tim Mather

Mapping Security
Tom Patterson with Scott Gleeson Blue

(SCTS) Symantec Certified Technical Specialist
Nik Alston, Mike Chapple, and Kalani Kirk Hausman

The Art of Computer Virus Research and Defense
Peter Szor

The Symantec Guide to Home Internet Security
Andrew Conry-Murray and Vincent Weafer
YOUR GUIDE TO IT REFERENCE

Articles
Keep your edge with thousands of free articles, in-depth features, interviews, and IT reference recommendations—all written by experts you know and trust.

Online Books
Answers in an instant from InformIT Online Book’s 600+ fully searchable online books. For a limited time, you can get your first 14 days free.

Catalog
Review online sample chapters, author biographies and customer rankings and choose exactly the right book from a selection of over 5,000 titles.