



Index

A

- a Posteriori* design, 701–705
- a Posteriori* verification techniques. *See* Testing a priori design, 702
- AAFID. *See* Autonomous Agents for Intrusion Detection system
- Abbreviations of Access Control Lists, 382–384
- Academic computer security
 - e-mail policy and, 112–113
 - e-mail policy implementation and, 113–114, 977–988
 - University of California Davis policy and, 111–112, 959–998
- Acceptable Use Policy (AUP) of the University of California Davis, 111–112, 989–992
- Access
 - login procedure and, 848–850
 - passwords and, 846–848
- Access control mechanisms
 - Access Control Lists (ACLs)
 - abbreviations of, 382–384
 - creation and maintenance of, 384–385
 - propagated (PACLs), 402–404
 - ring-based, 400–402
 - capabilities, 390–391
 - access control lists *vs.*, 395–396
 - copying and amplifying, 392–393
 - implementation of, 391–392
 - limits of, 394–395
 - revocation of rights, 393–394
 - Locks and keys and, 396–400
- Access control(s)
 - by Boolean expression, 35–36
 - cryptographic key and, 4
 - discretionary, 103
 - file permissions and, 890–891
 - by history, 36–37
 - identity based (IBAC), 103–104
 - mandatory (MAC), 103–104
 - matrix model, 32–35
 - interpretation of, 58–60
 - role-based (RBAC), 182–183
 - types of, 103–104
 - validating entries of, 894
- Access Restriction Facility program, 35
- Account root, 870
- ACK packet, 793, 794, 795
- Active attack, 7
- Active wiretapping, 7
- Administrative assurance. *See* Operational assurance
- Advanced Encryption Standard, 232
- Agent, 742–743
 - autonomous, 752–753
 - goal of, 745–756
 - host-based, 744
 - network-based, 744
- AH protocol. *See* Authentication Header protocol
- AI. *See* Artificial Intelligence system
- Alteration of information threat. *See* Modification of information threat
- AM. *See* Audit Manager
- Anagramming, 219
- Analysis paralysis, 18
- Analyzer, 692–693
- Anomaly detection system, 727–733
- Anonymity on the Web, 371–375
 - purpose of, 375–377
- Anonymizing sanitizer, 698
- Anonymous Diffie-Hellman, 292
- Application data protocol, 298
- Application of auditing system design, 700–701
- Architecture, tagged. *See* Tagged architecture
- Architecture of an intrusion detection system, 742–748
- Architecture requirements for TCSEC, 576



1064 Index

- ARF program. *See* Access Restriction Facility program
- Artificial intelligence system, 463
- Aslam's model, 673, 676, 677
 - fingerd buffer overflow flaw and, 676–678
 - flaw classes of, 673
 - legacy and, 673–674
 - xterm log file flaw and, 674–676
- Assignment statements, 418
- Assumptions and Trust and Security, 11–12
- Assurance, 12–16, 478, 792
 - building secure and trusted system and, 484–492
 - implementation, 484
 - in implementation and integration, 531–541
 - implemented and operational system life cycle and, 482–484
 - during operation and maintenance, 541
 - operational and administrative, 484
 - policy, 482
 - in requirements, 497–509
 - security, 478
 - specifications and, 527–528
 - during system and software design, 510–531
 - system and software design and, 482–484
 - techniques, 478
 - trust and, 477–484
 - See also* Information assurance
- Asynchronous validation/inadequate serialization flaw, 675, 677
- Attack(s), 6–7
 - active, 7
 - anticipating, 796–798
 - containment phase of, 756–757
 - denial of service, 6, 8–9
 - detection, 10
 - dictionary, 312
 - distributed denial of service, 793, 798–799
 - eradication of, 757–760
 - follow up phase of, 760–765
 - handling, 755
 - login procedure and, 848–850
 - man-in-the-middle, 252
 - passive, 7
 - prevention, 10, 754–755
 - SYN flood, 793
 - tool, 724–726
 - Trojan horses, 614–615
 - types of, 218
 - virus, 615–617
 - worms, 623–624
- Attackers, 7
- Attenuation of privilege, 43
- Audit browsing tools, 715–717
- Audit design
 - a posteriori*, 701–702
 - a priori*, 702
- Audit manager, 156
- Auditing, 152, 690
- Auditing mechanisms, 705
 - nonsecure systems and, 707–708
 - secure systems and, 706–707
- Auditing system
 - components, 690–693
 - design of, 693–701
 - detecting violations and, 702–705
 - requirements for TCSEC, 576
- AUP. *See* Acceptable Use Policy (AUP) of the University of California Davis
- Authentication
 - basics of, 309–310
 - biometrics and, 328–331
 - challenge-response algorithm and, 324–328
 - classical key exchange protocols and, 247–250
 - definition of, 309
 - development network system and, 823–825
 - functions, 321–322
 - hardware support and, 326–327
 - by location, 331
 - multiple methods, 331–333
 - passwords and, 310–324
 - path, 256
 - process, 310
 - public key exchange protocols and, 251–252
 - Web server system in the DMZ and, 823
- Authentication Header protocol, 303–304
- Authenticator, 250
- Authorization and Authentication (A&A)
 - database, 128
- Authorized transfer of rights, 48
- Autokey cipher, 280
- Automated electronic mail processing, 865
- Automated penetration analysis tool, 682
- Autonomous Agents for Intrusion Detection system, 752–753
- Availability
 - common criteria and, 598

- computer security and, 6
 - military security policy and, 99
 - network flooding, 793–796
 - security policy and, 97
- B**
- Backoff technique, 321
 - Bacterium or a rabbit, 624–625
 - Basic block, 421
 - Basic security theorem, 134–136
 - McLean's \dagger -Property and, 143–146
 - Bell-LaPadula Model, 124–142, 187–191, 553–555
 - Basic Security model and, 144–145
 - Chinese Wall model and, 175–177
 - Clark-Wilson integrity model and, 179–180
 - composition of, 188–191
 - controversy over, 143–145
 - Lipner's integrity matrix model and, 156–158
 - outer firewall configuration and, 783–784
 - System Z and, 147
 - tranquility and, 143
 - Trusted Computer System Evaluation Criteria and, 546
 - Bernstein conditions, 34n
 - Biba integrity model, 153
 - low-water-mark policy in, 154
 - ring policy in, 155
 - strict integrity policy in, 155
 - Biconditional commands, 40–41
 - Binding of requirements, 582
 - Biometrics, 328–330
 - caution and, 330
 - by combination of characteristics, 330
 - by eye characteristics, 329
 - by face recognition, 329–330
 - by fingerprints, 328–329
 - by keystrokes, 330
 - by voices, 329
 - Black box testing. *See* Functional testing
 - Bledsoe theorem prover, 558–559
 - Block chaining cipher, 231
 - See also* Data Encryption Standard
 - Block ciphers, 281–283
 - Bombs, logic. *See* Logic bombs
 - Boolean expressions and access control, 35–36
 - Boot sector infectors, 617–618
 - Boundary condition error, 677
 - Boundary data tests, 915
 - Boyer-Moore theorem prover, 555–556
 - Brain (Pakistani) virus, 617
 - Branching time logic systems, 954
 - Breach of security, 96; *See also* attacks(s)
 - Bridge, 57
 - Building documentation and specifications, 521–524
 - Building or adding security, 501–505
 - Building secure and trusted systems
 - conception stage of, 485–486
 - deployment stage of, 487
 - fielded product life and, 488
 - life cycle stage and, 484–485
 - manufacturing stage and, 486–487
 - Burroughs B6700 system penetration study, 654–655
- C**
- CA. *See* Certification Authority
 - Caesar cipher, 217–218
 - Can share proof, 60–63
 - Can steal proof, 60–63
 - Canadian Trusted Computer Product Evaluation Criteria, 581
 - Can-create relations, 70–71
 - Capabilities, 390–391
 - Access Control Lists (ACLs) vs., 395–396
 - copying and amplifying, 392–393, 393
 - implementation of, 391–392
 - limits of, 394–395
 - revocation of rights and, 393–394
 - CAPSL. *See* Common Authentication Protocol Specification Language
 - CBC mode. *See* Cipher Block Chaining (CBC) mode
 - CC. *See* Common Criteria
 - CCIMB. *See* Common Criteria Interpretation Management Board
 - Certificate, 254
 - OpenPGP, 259–260
 - X.509.v3, 256–257
 - Certificate revocation list, 266
 - Certificate signature chains, 256–260
 - Certification Authority
 - authentication policy, 358–360
 - definition of, 258
 - issuance policy, 358–360
 - Certified, Licensed Evaluation Facilities, 584
 - CFB mode. *See* Cipher FeedBack (CFB) mode



1066 Index

- Challenge-response algorithm, 324–328
- Change flaw, 674, 676
- Checksum, 237
- Chinese Wall model, 169–175
 - Bell-LaPadula Model and, 175–177
 - Clark-Wilson integrity model and, 177
- Choice of initial protection domain flaw, 675, 676
- Choice of operand or operation flaw, 676, 909–911
- Chosen plaintext attack, 218
- CIF. *See* Common Internal Form
- Cipher Block Chaining (CBC) mode, 231, 282
- Cipher FeedBack (CFB) mode, 231, 281
- Cipher(s)
 - autokey, 280
 - block, 281–283
 - classical, 232–233
 - FEAL block, 232
 - IDEA, 233
 - interchange, 293
 - one-time pad, 227
 - period of the, 222
 - polyalphabetic, 222
 - problems, 275–277
 - RSA, 234–237
 - self-synchronous stream, 280–281
 - substitution, 220–221
 - synchronous stream, 278–280
 - transposition, 219–220
 - Vigenère, 221–227
- Ciphertext only attack, 218
- CISR. *See* Commercial International Security Requirements
- Clark-Wilson integrity model, 160–166
 - Bell-LaPadula Model and, 179–180
 - Chinese Wall model and, 177
- Classes of variables, 432–433
- Classical cryptographic key exchange, 257–250
- Classical cryptography, 233
- Classical cryptosystems, 218–233
- Classical signatures, 267
- CLEFs. *See* Certified, Licensed Evaluation Facilities
- Clients process, 859
- Clinical information systems security policy, 177–179
- Clipper chip, 263
- Closed access, 35
- Closure alert, 297
- Code book mode, 276
- Coding faults, 673
- Colored Petri Automation, 733–735
- COM virus, 618
- Commands
 - conditional, 40–41
 - mono-operational, 48–53
 - primitive, 38–40
- Commercial International Security Requirements:1991, 586–587
- Commercial system
 - data integrity and, 151–152
 - military system vs., 152
 - security policy, 100
- Common Authentication Protocol Specification Language, 566
- Common Criteria, 591–592
 - assurance requirements, 599
 - conformance claim, 595
 - evaluation assurance levels, 599–601
 - evaluation process of, 601
 - future of, 602–604
 - impacts, 602
 - methodology, 592–596
 - protection profile (PP), 592–593
 - requirements, 596–597
 - security functional requirements, 597–598
- Common Criteria Interpretation Management Board, 602–603
- Common Internal Form, 553
- Company dataset, 170
- Compiler-based mechanisms, 415–429
- Complementation property, 229
- Complete mediation principle. *See* Principle of complete mediation
- Component. *See* Subsystem
- Compound statements, 419
- Computer security
 - components of, 3–6
 - cost-benefit analysis and, 16–17
 - distribution and, 917–918
 - feedback phase, 23
 - laws and customs, 18–19
 - maintenance and operation phase, 23
 - people and, 21–22
 - risk analysis and, 17–18
 - systems administrators and, 21
 - testing, maintenance and operation, 913–917
 - training and, 21



- trust and, 101–103
 - See also* Security policy
- Computer Security Act of 1987, 580
- Computer system
 - assurance, 12–16, 792
 - design, 14, 461
 - fielding, 490
 - isolation, 442–446
 - low, 156
 - maintenance, 462
 - specifications, 13–14
 - testing, 490, 534
 - Z, 146–148
- Computer system security
 - authentication and, 822–825
 - files and, 831–836
 - networks and, 811–817
 - policy and, 806–811
 - processes and, 825–831
 - users and, 817–822
- Computer virus(es), 615–617
 - boot sector infectors, 617–618
 - encrypted, 620–621
 - executable infectors, 618–619
 - macro, 622–623
 - multipartite, 619
 - polymorphic, 621–622
 - stealth, 620
 - theory of, 626–630
 - TSR, 620
- Computer worms, xxxi, 623–624
- Concurrency control mechanisms, 426–428
- Condition, 105
- Conditional and joint probability, 935–937
- Conditional commands, 40–41
- Conditional statements, 419–420
- Confidential security level, 124
- Confidentiality classification system, 124
- Confidentiality policy, 101
 - Bell-LaPadula Model and, 124–133
 - commercial security and, 100
 - computer security and, 4
 - data and, 4
 - goals of, 123
 - military security and, 99
 - resource hiding and, 4
 - RSA cipher and, 234–237
 - security and, 97
- Configuration management, 532–533
 - requirements for TCSEC, 576
- Confinement flow model, 411–412
- Confinement problem, 439–442
- Conflict of interest, 170
- Conflicts and access control lists, 386–387
- Conspiracy and Take-Grant Protection model, 63–65
- Control rights, 66
- Control Tree Logic, 534
 - semantics of, 955–956
 - syntax of, 954–955
- Controlled access protection for TCSEC, 577
- Cookies, 369–371
- Copy flag
 - Schematic protection model and, 66
 - Windows NT and, 42
- Copy Right, 42
- Copying and moving files processes, 860–861
- Corporate computer system, penetration study of, 655–656
- Cost-benefit analysis, 16–17
- Counter method, 280
- Counterattacks, 763–764
- Covert channel(s), 115, 440, 446
 - analysis of, 462–467
 - capacity analysis, 462–467
 - measuring of, 464–465
 - noisy, 465–467
 - noninterference, 462–464
 - detection of, 448–462
 - existence and, 447
 - mitigation of, 467–470
 - noiseless, 447
 - noisy, 447
 - storage of, 446
- Covert flow trees, 454–462
- Create rules, 54, 70–71, 79
- Creation graph of MTAM model, 89
- Creation of access control lists, 384–385
- Cryptanalysis, 217
- Cryptographic checksum function, 237
- Cryptographic hash, 255
- Cryptographic key, 4
 - infrastructures, 254–255
 - process, 861–862
- Cryptographic mechanisms, support of, 292–294
- Cryptographically pseudorandom numbers, sequence of, 253

1068 Index

- Cryptographically random numbers, sequence of, 252–253
- Cryptography
 - access control and, 4
 - definition and goal of, 217–218
 - implementation of access control lists capabilities and, 391–392
 - laws and customs and, 18–19
 - network and, 283–286
 - public key, 233–237
 - translucent, 265
 - See also* Classical cryptographic key
 - exchange; Data encryption standard; Diffie-Hellman, scheme; Digital signatures; Public key cryptographic key exchange
- Cryptosystem, 217
 - classical, 218–233
- CTL. *See* Control Tree Logic
- CTPEC. *See* Canadian Trusted Computer Product Evaluation Criteria
- CuD. *See* Customer data
- Current security level, 127
- Customer data, 775
- Cypherpunk (or type 1) remailer, 372
- D**
- DAC. *See* Discretionary Access Control
- Data and confidentiality, 4
- Data and instruction change
 - files contents, 898
 - memory and, 895–898
 - race conditions in file accesses, 898–899
- Data classes, 775
- Data Encipherment Key, 288–289
- Data Encryption Standard, 228–232, 253
- Data integrity, 96
 - commercial systems and, 151–152
 - detection mechanism and, 5
- Data Mark Machine, 430–432
- Data recovery component, 263
- DDEP. *See* Development Data for Existing Products
- DDFP. *See* Development Data for Future Products
- DDoS. *See* Distributed Denial of Service attack
- Deallocation and deletion of information, 901–902
- Deallocation or deletion flaw, 674
- Deception threat, 7
- Deception Tool Kit (DTK), 757
- Decipherment keys, 233
- Declassification problem, 142
- Decoy servers, 756–757
- Deducibly secure systems, 204
 - composition of, 204–205
- Default permissions and access control lists, 387
- DEK. *See* Data Encipherment Key
- Delay of service, 8–9
- Delegation, 7
- Deliverable state, 72
- Demilitarized zone. *See* DMZ
- Denial of receipt, 8
- Denial of service attack, 8–9
- Deployment of secure and trusted systems, 487
- Derived rules, 950
- DES. *See* Data Encryption Standard
- Design
 - assurance, 483–484
 - documentation, 512–513
 - internal description, 515–520
 - phase of a system, 14
 - principles, 343–349
 - program security, 873–880
 - specification, internal, 520–521
 - for validation, 907
- Design specification and verification requirements for TCSEC, 576
- Designing an auditing system, 693–701
- Detection mechanism, 10
 - data integrity and, 5
- Detection of covert channels, 448–462
- Detection of intrusions, 724–726
- Deterministic non-interference secure system, 191–195
 - composition of, 201–202
- Development Data for Existing Products, 775
- Development Data for Future Products, 775
- Development subnet. *See* Devnet
- Development system, 808–810, 838
 - authentication and, 823–825
 - consequences, 809–810
 - DMZ Web server system *vs.*, 810–811, 816–817, 822, 825, 830–831, 835–836
 - files and, 833–835
 - networks and, 814–816
 - policy and, 807–810
 - processes and, 829–830

- users and, 817–819
 - See also* Devnet system
 - Devices
 - monitors and window systems, 859
 - smart terminals, 857–858
 - writable terminals, 857
 - Devnet system, 807–808, 830–831
 - See also* Development system
 - Dictionary attack(s), 312
 - challenge response and, 327–328
 - reusable passwords and, 320–321
 - DIDS. *See* Distributed Intrusion Detection System
 - Differential cryptanalysis, 230
 - Diffie-Hellman
 - cipher, 292–293
 - scheme, 233–234
 - Digital signatures, 266–270
 - See also* Public key signatures
 - Direct alias, 855
 - Director, 746–747
 - Directory authentication framework, 256–258
 - Disabling technique, 322
 - Disclosure threat, 7
 - Disconnection technique, 321
 - Discretionary Access Control (DAC), 103
 - Bell-LaPadula security model and, 124
 - TCSEC requirements, 575
 - Discretionary protection for TCSEC, 577
 - Disruption threat, 7
 - Distributed Denial of Service attack, 793, 798–799
 - Distributed Intrusion Detection System, 750–752
 - Distributed Program Execution Monitor, 740
 - DMZ, 779
 - DNS server, 785, 786, 789
 - log server, 789–790
 - mail server, 783, 786–787
 - Web server system, 783, 787–788, 806–807
 - authentication and, 823
 - development network system authentication *vs.*, 825
 - development system files *vs.*, 833–836
 - development system networks *vs.*, 816–817
 - development system policy *vs.*, 810–811
 - development system processes *vs.*, 829–830
 - development system users *vs.*, 822
 - devnet system policy *vs.*, 810–811, 830–831
 - files and, 831–833
 - networks and, 812–814
 - policy and, 806–807
 - processes and, 825–829
 - users and, 817–819
 - DNS. *See* Domain Name System
 - DNS server, DMZ, 789
 - Documentation
 - building of specification and, 521–531
 - design, 512–513
 - Domain, 66
 - Domain Name System, 367
 - security issues with, 368–369
 - Domain-Type Enforcement Language, 106–109
 - DPEM. *See* Distributed Program Execution Monitor
 - DTEL. *See* Domain-Type Enforcement Language
 - Dynamic identifiers, 367–368
 - Dynamic security policies, 200–201
- E**
- Early formal verification techniques, 551–559
 - ECB mode. *See* Electronic Code Book (ECB) mode
 - Economy of mechanism principle. *See* Principle of economy of mechanism
 - EDE mode. *See* Encrypt-Decrypt-Encrypt (EDE) mode
 - Edge adding operations, 84–85
 - EES. *See* Escrowed Encryption Standard
 - EHDM. *See* Enhanced Hierarchical Development Methodology
 - El Gamal digital signature, 269–270
 - Electronic Code Book (ECB) mode, 231
 - Electronic communications, 865–866
 - Electronic mail. *See* E-mail
 - Elimination rule, 949
 - E-mail, 113–114
 - anonymous, 371–375
 - automated processing of, 865
 - failure to check certificates and, 865–866
 - PGP certificate signature chains and, 258–260
 - policy at the University of California Davis, 112
 - RSA cipher and, 234–237
 - secure (PEM), 286–290
 - sending unexpected content and, 866
 - University of California Davis computer security policy, 112–113, 959–988

1070 Index

- E-mail (*continued*)
 - University of California Davis implementation policy, 977–988
 - Emergent faults, 673
 - Encapsulating Security Payload (ESP) protocol, 304–305
 - Encipherment keys, 233
 - Encrypt-Decrypt-Encrypt (EDE) mode, 231, 232, 283
 - Encrypted key exchange, 327–328
 - Encrypted viruses, 620–621
 - Encryption. *See* Cryptography
 - End-to-end encryption, 284–285
 - Enforcement mechanism, 116
 - Enhanced Hierarchical Development
 - Methodology, 551–553
 - Entity, 66, 353, 356
 - Entropy and uncertainty, 937–938
 - Entropy-based analysis, 408–409
 - Ephemeral Diffie-Hellman, 292
 - Error alert, 297
 - Error checking, 904
 - Error handling tests, 916
 - Escrow system, 262–263
 - Escrowed Encryption Standard, 263
 - ESP protocol. *See* Encapsulating Security Payload (ESP) protocol
 - ESPM. *See* Extended Schematic Protection model
 - Euclidean algorithm, 929–930
 - extended, 930–932
 - Evaluation methodology
 - formal, 572
 - historical perspective of, 573–574
 - Trusted Computer System Evaluation Criteria and, 574–581
 - Evaluation process
 - common criteria, 601
 - Information Technology Security Evaluation Criteria and, 583–584
 - Trusted Computer System Evaluation Criteria, 578
 - Event system, 203
 - Example information flow controls, 433–436
 - Exception tests, 916
 - Exceptions and infinite loops, 424–425
 - EXE virus, 618
 - Executable infectors, 618–619
 - Execution trace, 739
 - Execution-based mechanisms, 429–433
 - Executive privilege, 123
 - Exploitable logic error flaw, 675
 - Exploiting the vulnerability, 645
 - Exploratory programming, 491
 - Exponential backoff, 321
 - Extended Schematic Protection model, 79–83
 - External functional specification, 486
 - Extreme programming, 492
 - Eye characteristics authentication, 329
- F**
- Face recognition authentication, 329
 - Fail-safe defaults principle. *See* Principle of, fail-safe defaults
 - Father Christmas worm, 626
 - FC. *See* Federal Criteria
 - FEAL block cipher, 232
 - Federal Criteria, 587–589
 - Feedback chaining cipher, 231
 - Feedback phase and computer security, 23
 - Fielded product life, 460
 - Fielding, system, 490
 - Fifth postulate of Euclid, 147n
 - File(s)
 - contents changes in, 898
 - deletion, 855–856
 - group access and, 854–855
 - permission, 853, 890–891
 - system security and, 831–836
 - Filter function, 68
 - Filtering firewalls, 781, 791
 - Finger buffer overflow flaw, 661, 667, 676–678
 - Finger protocol, 661
 - Fingerprint authentication, 328–329
 - Finite-state machine, 96 (Fig. 4–1)
 - FIPS 140, requirements of, 589–590
 - FIPS 140-2 security levels, 590
 - Firewall(s), 780, 791
 - filtering, 781
 - inner configuration, 785–786
 - outer configuration, 783–785
 - proxies and, 780–782
 - First order logic. *See* Predicate logic
 - Flaw
 - classes of
 - asynchronous validation/inadequate serialization, 663, 664
 - coding faults, 673
 - emergent faults, 673

- exploitable logic error, 663, 665
 - implicit sharing of privileged/confidential data, 663, 664
 - improper change, 666, 667
 - improper choice of initial protection domain, 666, 668
 - improper deallocation or deletion, 666, 667
 - improper indivisibility, 666, 668
 - improper isolation of implementation detail, 666
 - improper naming, 666, 667
 - improper protection domain initialization and enforcement, 666
 - improper sequencing, 666, 668
 - improper synchronization, 666, 667
 - improper validation, 666, 667
 - inadequate identification/authentication/authorization, 663, 664
 - inadvertent, 671
 - incomplete parameter validation, 663, 664
 - inconsistent parameter validation, 663, 664
 - intentional, 671
 - violable prohibition/limit, 663, 665
 - hypothesis methodology, 649–650
 - elimination step of, 649
 - generalization step and, 649
 - information gathering and, 650–651
 - testing, 651
 - malicious and nonmalicious, 671–672
 - security, 661–662
 - Flow function, 72–73
 - Flow-based model of penetration analysis, 679–682
 - Formal assurance techniques, 478
 - Formal evaluation methodology, 572
 - Formal model, 132
 - Formal proof mechanisms, 528
 - Formal specification(s), 523, 548–551
 - Formal transformation model, 491–492
 - Formal verification technique, 545–548, 646
 - Fortezza cryptographic token, 293
 - 4096 virus. *See* Stealth virus
 - Frameworks, 662
 - Aslam's model, 673–674
 - flaw classes, 673
 - legacy, 673–674
 - NRL Taxonomy, 671–673
 - flaw classes, 671–672
 - legacy, 672–673
 - Protection Analysis model, 665–666
 - analysis procedure, 668–670
 - flaw classes, 666–668
 - legacy of, 670
 - RIOSIS study and
 - flaw classes, 665
 - legacy of, 665
 - Functional testing, 533
- ## G
- Gate, 401
 - Generalized noninterference system, 205
 - composition of, 206–208
 - Generic right, 48
 - Get-read rule, 140–141
 - Give-read rule, 141–142
 - Global identifier, 367
 - Global Positioning System, 331
 - Goto statements, 421–423
 - GPS. *See* Global Positioning System
 - Grant right, 53. *See also* Copy Right
 - Grant rule, 54, 56
 - Graph rewriting rules, 53–54
 - Graphical Intrusion Detection System, 747–748
 - GrIDS. *See* Graphical Intrusion Detection System
 - Groups, 356–357
 - wildcards and, 386
 - Gupta and Gilgor's theory of penetration analysis, 678–683
 - GVE. *See* Gypsy Verification Environment
 - Gypsy language, 557–558
 - Gypsy Verification Environment, 557–559
- ## H
- HMAC, 239
 - HAM. *See* Hierarchical Development Methodology
 - Hardware-supported challenge-response procedures, 326–327
 - Harrison-Ruzzo-Ullman model, 47
 - Schematic Protection model *vs.*, 78–79
 - Hash function(s), 294, 311
 - HMAC and, 239
 - key crunching and, 315
 - Hierarchical Development Methodology, 547–548
 - enhanced, 556–557
 - verification in, 553–555

1072 Index

Hierarchy consistency checker, 552
Hierarchy Specification Language, 552
High-level policy languages, 104–109
History and access control, 36–37
Honey pots, 756–757
 technique, 322
Host and network monitoring, 750–752
Host identity, 366–367
Host-based agents, 744
Hosts, intermediate, 793–794
HRU. Model. *See* Harrison-Ruzzo-Ullman model
HSL. *See* Hierarchy Specification Language
Human factors in computer security, 19–22
Hybrid policies, 169–185

I

IBAC. *See* Identity-Based Access Control
IDEA cipher, 233
Identification and authentication requirements for
 TCSEC, 575
Identifiers, 367–368
Identity
 conflicts and, 360–361
 definition of, 353–354
 of files and objects, 354–355
 groups and roles, 356–357
 meaning of, 363–364
 naming and certificates for, 357–360
 trust and, 364–365
 users and, 355–356
 uses of, 354
 on the Web, 366–377
Identity-Based Access Control, 103–104
IDES. *See* Intrusion Detection Expert System
IDF virus. *See* Stealth virus
IDIOT. *See* Intrusion Detection In Our Time
IDIP. *See* Intrusion Detection and Isolation
 Protocol
IDS. *See* Intrusion Detection System
Immediate forward dominator, 423
Implementation assurance, 484
Implementation detail, improper isolation of, 666
 resource exhaustion and user identifiers and,
 893–894
 restricting the protection domain of the role
 process and, 894–895
 validating the access control entries and, 894
Implementation management, assurance through,
 532–533

Implementation phase
 auditing system design and, 696
 capability of access control lists, 391–392
 computer program, 14–16
 computer security and, 22–23
 design and, 533–536
 program security and, 880–887
 unit testing and, 489–490
 University of California Davis E-mail policy,
 114, 977–988
Implicit sharing privileged/confidential data flow,
 664
Inadequate identification/authorization/authenti-
 cation flow, 675, 677
Inadequate serialization error, 676
Inadvertent flaw of serialization/aliasing, 675
Incomplete parameter validation, 664
Inconsistent parameter validation flaw, 664, 675
Index of coincidence, 223
Indivisibility flaw, 674, 907–908
Inductive verification techniques, 547
Inert rights, 66
Infinite loops and exceptions, 424–425
Informal arguments, 526–527
Informal assurance techniques, 478
Informal correspondence, 524–526
Information assurance, 478
 See also Assurance
Information flow, 97
 compiler-based mechanisms and,
 415–429
 entropy-based analysis, 408–409
 execution-based mechanisms and,
 429–433
 information flow controls and, 433–436
 metrics, 631–632
 models and mechanisms, 409–410
 nonlattice, 410–415
 policies, 407–410
Information Technology Security Evaluation
 Criteria, 581–582
 assurance requirements, 582
 evaluation levels and process, 583–584
 impacts, 585
 international efforts and, 581–582
Information transfer path, 153
Infrastructure analysis
 network and, 782–783

- Initial protection domain, improper choice of, 666
 - access control file permissions and, 890–891
 - memory protection and, 891–892
 - process privileges and, 888–890
 - trust in the system and, 892–893
 - Initial state operations, 84
 - Input checking, 905–907
 - Insiders, 21
 - Instantiation, 105
 - Integration, 490
 - Integrity
 - Biba model and, 153–155
 - Clark-Wilson, 160–166
 - commercial security policy and, 100
 - computer security and, 5
 - data and, 96
 - Lipner's matrix model and, 156–160
 - military security policy and, 99
 - policy, 101
 - goals of, 151–152
 - security policy and, 97, 100
 - Integrity Value Check, 303
 - Interchange key, 246, 288
 - Interference, 191
 - Intermediate hosts, 793–794
 - Internal design description, 515–520
 - Internal design specification, 520–521
 - Internal mail server, 791
 - Internal network organization, 790–792
 - Internal Web server, 791
 - Internet
 - anonymity on the, 371–377
 - host identity and, 366–367
 - security issues with the domain name system and, 368–369
 - state and cookies and, 369–371
 - static and dynamic identifiers and, 367–368
 - Internet Policy Registration Authority, 359
 - Internet Research Task Force on Privacy, 287
 - Introduction rule, 949
 - Intrusion Detection and Isolation Protocol, 748
 - Intrusion Detection Expert System, 728, 730
 - Intrusion Detection In Our Time, 733–735
 - Intrusion detection system
 - architecture of, 742–748
 - basics of, 724–726
 - distributed, 750–752
 - Intrusion response and, 754–764
 - models of, 727–742
 - organization of, 748–753
 - principles of, 723–724
 - See also* Attack(s)
 - Invocation, 105
 - IPRA. *See* Internet Policy Registration Authority
 - IPsec, 298–299
 - architecture, 299–303
 - Island, 56
 - Isolation, system, 442–446
 - Isolation and implementation detail flaw, 676
 - Iterated tunneling, 301
 - Iterative statements, 420–421
 - ITSEC. *See* Information Technology Security Evaluation Criteria
 - IVC. *See* Integrity Value Check
- J**
- Jailing technique, 322
 - Java programs
 - Pandey and Hashii's policy constraint language and, 105–106
 - Jerusalem (Israeli) virus, 619
 - Joint and conditional entropy, 938–940
- K**
- Kasiski and Vigenère cipher, 223
 - KEDP. *See* Key Escrow Decrypt Processor
 - Kemmerer methodology, 450–452, 454
 - Kerberos, 250–251
 - Key crunching, 315
 - Key Escrow, 262–263
 - clipper chip and, 263
 - component, 263
 - system, 262
 - Key Escrow Decrypt Processor, 263
 - Key exchange, 246–247
 - Key exchange protocols, 247–250
 - Key management, defined, 245
 - Key storage, 261
 - Keyed cryptographic checksum function, 238
 - Keyed hash function, 239
 - Keyless hash function, 239
 - Keys and locks. *See* Locks and keys
 - Keystroke authentication, 330
 - Klein's password guessing experiments, 316
 - Known plaintext attack, 218



L

Label requirements for TCSEC, 575
Labeled security protection for TCSEC, 577
LAFS. *See* Logging and auditing file system
Language
 domain-set enforcement, 106–109
 high-level policy, 104–109
 low-level policy, 109–111
 Pandey and Hashii's policy constraint, 105
Lattice(s), 925–927
Law Enforcement Access Field, 263
Laws and customs and technology, 18–19
LEAF. *See* Law Enforcement Access Field
Leaked right, 48
Least common mechanism principle, 348
Least privilege principle. *See* Principle of least privilege
Legal transition, 72
Legitimate handshake, 793
LFSR. *See* Linear Feedback Shift Register
Life cycle of system development
 assurance throughout the, 482–484
 models of software development and, 491–492
 stages of, 484–488
 Waterfall model and, 488–491
Linear Feedback Shift Register, 278–279
Linear time logic systems, 954
Link encryption, 284–285
Link predicate, 66–67
Link protocol, 283
Linux
 obtaining location, 884–885
 process privileges and, 889–890
Lipner's integrity matrix model, 158–160
 Bell-LaPadula Model use in, 156–158
 Biba's integrity model and, 160
 commercial requirements and, 151–152
Local area network and access control matrix, 34
 (Fig. 2–2)
Local identifier, 367
Location and authentication, 331
Location Signature Sensor, 331
Locks and keys, 396–397
 sharing secrets and, 399–400
 type checking and, 397–398
Log sanitization and auditing system design, 698–700
Log server, DMZ, 789–790
Logger, 690–692

Logging, 689
Logging and auditing file system (LAFS), 685–686, 713–714
Logic, malicious. *See* Malicious logic
Logic bombs, 625
Login procedure, 848–850
 access and, 848–850
 leaving the system and, 850–852
 trusted hosts and, 850
Lower layer security protocols, 294
Low-level policy languages, 109–111
Low-Water-Mark policy, 154
LSS. *See* Location Signature Sensor

M

MAC. *See* Mandatory Access Control
MAC labels
 assigning of, 128–131
 using, 131–132
Machine catdog, 206–208
MacMag Peace virus, 617
Macro virus, 622–623
Mail proxy, 786–787
Mail server
 DMZ, 786–787
 internal, 791
Maintenance and operations
 assurance during, 541
 computer security and, 23
Maintenance of access control lists, 384–385
Malicious flaws. *See* Flaws, malicious
Malicious logic
 acting as both data and instructions, 630–631
 altering files, 637–638
 altering statistical characteristics, 639
 assuming the identity of a user, 631–636
 crossing protection domain boundaries by
 sharing, 636
 defenses against, 630–635
 definition of, 613
 other forms of, 624–630
 performing actions beyond specification, 638
 reducing the rights of, 632–635
 sandboxing and, 635–636
 theory of, 626–630
 Trojan horses as, 614–615
 trust notion and, 640
 user security and, 864
 viruses as, 615–617



- worms as, 623–624
 - Mandatory Access Control, 7, 61, 103–104
 - Assurance and, 499, 504, 505, 506, 507, 522
 - Bell-LaPadula security model and, 124
 - Data General 132 UNIX system and, 128–132
 - Malicious logic and, 617, 636
 - Originator-Controlled Access Control and, 181
 - Role-Based Access Control and, 183
 - TCSEC requirements, 575, 577, 579
 - Man-in-the-middle attack, 252
 - Manipulation Detection Codes, 637–638
 - Manufacturing secure and trusted systems, 486–487
 - Mapping specifications, 552
 - Masquerading, 7–8
 - Mathematical induction, 951–952
 - Maximal state notion, 72, 73–74
 - Maximally precise mechanism, 118
 - Maximum security level, 127
 - McLean's †-Property and the Basic Security Theorem, 143–146
 - McLean's System Z, 146–148
 - MDCs. *See* Manipulation Detection Codes
 - Memory
 - allocations and TCP server, 794–796
 - data and instruction change and, 895–898
 - programming problems and change of, 895–898
 - protection, 391, 891–892
 - Merkle's tree authentication scheme, 255–256
 - Message digest. *See* Checksum
 - Message Integrity Check, 289
 - See also* Integrity Value Check
 - Message Transport (Transfer) Agent, 286–287
 - Method, 105
 - MIC. *See* Message Integrity Check
 - Military system
 - commercial systems vs., 152
 - security policy, 99
 - Miller and Baldwin model, 35–37
 - Misordered blocks, 276
 - Misuse detection, 733
 - Misuse modeling, 733–738
 - See also* Attack(s)
 - Mixmaster (or type 2) remailer, 374–375
 - MLS. *See* Multilevel Security tool
 - Model checker, 528, 547
 - Model-based techniques
 - proof-based techniques vs., 546
 - Modification of information threat, 7
 - Module, 511
 - Monoconditional commands, 40
 - Mono-operational commands, 48–53
 - Monotonic models
 - multiparent, 87–88
 - single-parent, 87–88
 - Monotonic systems, defined, 52
 - Moving and copying files processes, 860–861
 - MTA. *See* Message Transport (Transfer) Agent
 - Multics system, 139
 - Multilevel directory, 129
 - Multilevel Security tool, 553–555
 - Multipartite viruses, 619
 - Multiple encryption, 282–283
 - Mutually exclusive authorization, 183
- ## N
- Naming objects, 899–901
 - National Computer Security Center, 579
 - National Institute of Standards and Technology, 232
 - Natural deduction, 948
 - rules of, 949
 - Naval Protocol Analyzer (NPA)
 - experience, 567
 - languages, 566
 - Naval Protocol Analyzer Temporal Requirements Language (NPATRL), 566
 - Naval Research Laboratory (NRL) Protocol Analyzer, 566
 - NCSC. *See* National Computer Security Center
 - Needham-Schroeder protocol, 247–250
 - Netscape Corporation, Secure Socket Layer (SSL) and, 291
 - Network and host monitoring, 750–752
 - Network File System protocol, 785
 - Network Flight Recorder, 737–738
 - Network flooding, availability and, 793
 - Network Identification Number (NID), 750
 - Network layer and security, 298–305
 - Network(s)
 - cryptography and, 283–286
 - host identity and, 366–367
 - infrastructure analysis and, 782–783
 - organization, 779–780
 - policy development and, 774–775
 - system security and, 811–817

1076 Index

Network Security Monitor, 749–750
 Network-based agents, 744
 New DES algorithm, 232
 NFR. *See* Network Flight Recorder
 NFS protocol. *See* Network File System protocol
 NFS v.2 protocol auditing analysis, 709–713
 NID. *See* Network Identification Number
 NLFSR. *See* Nonlinear Feedback Shift Register
 Node creation operations, 84
 Noiseless covert channel, 447
 Noisy covert channel, 447
 capacity analysis of, 465–467
 Nonces, 248
 Noninterference model, 448–450
 Nonlattice information flow policies, 410–411
 confinement flow model, 411–412
 nontransitive, 413–415
 transitive nonlattice, 412–413
 Nonlinear Feedback Shift Register, 279–280
 Nonmalicious flaws. *See* Flaws, malicious and nonmalicious
 Nonsecure systems, auditing mechanisms and, 707–708
 Nontransitive information flow policies, 413–415
 Normal data tests, 915
 Notifier, 693, 747–748
 NRL taxonomy
 flaw classes
 malicious, 671–672
 nonmalicious, 671–672
 flaws by location, 672–673
 flaws by time of introduction, 671–672
 legacy and, 672–673
 NSM. *See* Network Security Monitor

O

Object reuse requirements for TCSEC, 575
 Object set, 66
 Objects, 170, 354–355
 Oblivious transfer, 265
 Observability postulate, 114, 117
 One-time pad, 227
 One-time passwords, 325–326
 Open access, 35
 Open design principle. *See* Principle of open design
 OpenPGP public key packet, 259
 Operational and administrative assurance, 484
 Operational assurance, 452, 456

Operational issues and computer security, 16–19
 Operations and maintenance
 assurance during, 541
 computer security, 22–23
 Orange Book. *See* Trusted Computer System Evaluation Criteria
 ORCON. *See* Originator Controlled Access Control
 Organizational certificate, 360
 Organizations and computer security, 20–21
 ORGCON. *See* Originator Controlled Access Control
 Origin integrity, 5, 96
 Originator Controlled Access Control, 104, 180–181
 Otway-Rees protocol, 249
 Outer firewall configuration, 783–785
 Output feedback (OFB) mode, 279–280
 Outsiders, 21
 Overwriting files, 861
 Own right, 33, 42–43
 Owner-based policy, 68–69

P

PA model. *See* Protection Analysis model
 Packet, 259
 PACL. *See* Propagated Access Control List mechanism
 Paging and virtual machines, 945–946
 Pandey and Hashii's policy constraint language
 Java programming language and, 105–106
 Parameter validation flaw, 677
 Pascal programming language, 557
 Pass algorithms, 324–325
 Passive attack, 7
 Passive wiretapping. *See* Wiretapping
 Password(s), 310, 846–848
 access and, 846–848
 aging, 322–324
 attacking, 312
 authentication of, 310–312
 guessing, 313, 321–322
 one-time, 325–326
 process, 861–862
 pronounceable and computer-generated, 315
 random selection of, 314–315
 reusable, 320–321
 selection of, 316–320
 PAT. *See* Process Action Team Guidance Working Group

- Pattern-directed protection evaluation, 665–666
- PCC. *See* Proof-Carrying Code
- PD. *See* Public Data
- PEM. *See* Privacy-Enhanced Electronic Mail
- Penetration analysis, 646
 - automated tool, 682
 - flow-based model of, 679–682
 - Gupta and Gligor's theory of, 678–679
- Penetration studies
 - Burroughs B6700 system and, 654–655
 - corporate computer study and, 655–656
 - flaw hypothesis methodology and, 649–652
 - goals of, 647–648
 - layering of tests and, 648–649
 - methodology at each layer and, 649
 - University of Michigan terminal system and, 652–654
 - UNIX system and, 656–658
 - validity of, 659
 - Windows NT system and, 658–659
 - See also* Penetration testing
- Penetration testing, 646
- People and computer security, 21–22
- Period of the cipher, 222
- PGP Certificate signature chains, 258–260
 - privacy-enhanced electronic mail (PEM) *vs.*, 290
 - X.509: Certification signature chains *vs.*, 260
- PGWG methodology, 536
- Physical characteristics and authentication, 328–330
- Pigeonhole principle, 238
- Policy
 - assurance, 482
 - definition and requirements specification, 505–508
 - development, 774–775
 - availability and, 778
 - consistency check and, 778–779
 - data classes and, 775
 - user classes and, 776–777
 - language(s)
 - definition of, 104
 - high-level, 104–109
 - low-level, 109–111
 - mechanism and, 9–10
 - security, 95–122
 - system security, 808–811
- University of California Davis e-mail, 113, 959–998
 - user security, 845–846
- Polyalphabetic cipher, 222
- Polymorphic viruses, 621–622
- PP. *See* Protection Profile
- Precision and security, 114–119
- Predevelopment *vs.* postdevelopment technique, 546
- Predicate calculus. *See* Predicate logic
- Predicate logic, 952–953
 - natural deduction in, 953–954
- Prevention mechanism, 10
 - data integrity and, 5
- Prime number, 147
- Primitive commands, 38–40
- Principal, 353
- Principle of
 - attenuation of privilege, 43
 - autonomy, 189
 - complete mediation, 345–346
 - economy of mechanism, 344–345
 - fail-safe defaults, 344
 - least common mechanism, 348
 - least privilege, 343–344
 - open design, 346–347
 - psychological acceptability, 348–349
 - security, 189
 - separation of privilege, 347–348, 777
- Privacy Act of the United States, 99, 123
- Privacy research group. *See* Internet Research Task Force on Privacy
- Privacy-Enhanced Electronic Mail, 286–289
 - PGP Certificate signature chains *vs.*, 290–291
- Privilege(s)
 - attenuation, 43
 - limiting, 863
 - process, 888–890
 - virtual machines and, 943–944
- Proactive password checker, 318–320
- Proactive password selection, 316
- Procedure calls, 424
- Process Action Team Guidance Working Group, 536
- Process of performance, 605
- Process(s)
 - capability, 605
 - copying and moving files, 860–861
 - development system, 829–830

1078 Index

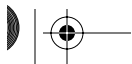
- Process(s) (*continued*)
 - encryption, cryptographic keys and passwords, 861–862
 - limiting privileges, 863
 - malicious logic, 864
 - maturity, 605
 - overwriting files, 861
 - start-up settings, 863
 - Web server system in the DMZ, 825–828
 - Product cipher, 228
 - Product retirement, 460
 - Program
 - design, 461
 - distribution, 917–918
 - implementation, 14–16
 - statements, 418
 - Program security
 - design and, 873–880
 - programming problems and, 887–913
 - refinement and implementation of, 880–887
 - requirements and policy of, 870–873
 - Programming problems
 - choice of operand and operation and, 909–911
 - data and instruction change and, 895–899
 - deallocation and deletion and, 901–902
 - indivisibility and, 907–908
 - isolation of implementation detail and, 893–895
 - naming objects and, 899–901
 - sequencing and, 908–909
 - validation and, 902–911
 - Proof checkers. *See* Formal proof mechanisms
 - Proof of concept, 486
 - Proof rules, 948
 - Proof-based techniques, 527–529
 - model-based techniques *vs.*, 546
 - Proof-Carrying Code, 638–639
 - Propagated Access Control List mechanism, 402–404
 - Propagating Trojan horse, 615
 - Property verification *vs.* full verification, 518
 - Propositional calculus. *See* Propositional logic
 - Propositional logic, 947–948
 - mathematical induction and, 951–952
 - natural deduction in, 948
 - rules of, 949
 - truth tables and, 950–951
 - well-formed formulas and, 90
 - Protection Analysis model, 665–666
 - analysis procedure and, 668–670
 - flaw classes and, 666–668
 - legacy and, 670
 - Protection mechanism, 115, 116
 - Protection Profile, 588
 - Protection state
 - system interpretation of, 31–32, 58
 - transitions, 37–41
 - Protection system, 31
 - Protection type. *See* Schematic Protection model
 - Protocol Analyzer (NPA) and languages, 538
 - Prototype Verification System, 559
 - experience with, 562
 - proof checker, 561–562
 - specification language, 559–561
 - Prototyping, 491
 - Proxies, 780–782, 791
 - Pseudo-anonymous remailer, 372–373
 - Pseudonymizing sanitizer, 698
 - Psychological acceptability principle, 348–349
 - Public Data, 775
 - Public key cryptographic key exchange, 251–252
 - Public key cryptography
 - Diffie-Hellman scheme and, 233–234
 - RSA cipher and, 234–237
 - Public key revocation, 265–266
 - Public key signatures, 267
 - PVS. *See* Prototype Verification System
 - PVS proof checker, 561–562
 - PVS specification language, 531–533
- Q**
- Query-set-overlap control, 36–37
- R**
- Rabbits and bacteria, 624–625
 - Rail fence cipher, 219
 - RAMP. *See* Ratings Maintenance Program
 - Random data tests, 916
 - Random variable, 935
 - Range, 131
 - Ratings Maintenance Program, 578
 - RC. *See* Control rights
 - Receipt denial. *See* Denial of receipt
 - Record, 885–886
 - Recovery mechanism, 10–11
 - Red team attack. *See* Penetration studies
 - Reference monitor, 502
 - Reference Validation Mechanism, 502
 - Refinement, program security and, 880–887
 - Remove rule, 54

- Replicating Trojan horse. *See* Propagating Trojan horse
 - Representation correspondence. *See* Informal correspondence
 - Repudiation of origin, 8
 - Requirement(s)
 - definition of, 481
 - justifying, 508–509
 - program security, 870–873
 - role in assurance, 453–454
 - specification, 488
 - policy definition and, 477–480
 - tracing, 523–526
 - Trusted Computer System Evaluation Criteria, 554–555
 - Research Into Secure Operating Systems, 662–663, 677–678
 - Residential certificate, 360
 - Resources
 - hiding and confidentiality, 4
 - system security and, 20–21
 - Restrictive systems
 - composition of, 209–210
 - definition of, 209
 - Review mechanism, assurance and, 528–531
 - Revocation of rights and access control lists, 387–389
 - RI. *See* Inert rights
 - RIACS file system checker, 110
 - Rights
 - in a program, 34 (Fig. 2–3)
 - sharing, 55–58
 - Ring Policy, 155
 - Ring-based access control, 400–402
 - Risk analysis, 17–18
 - RISOS. *See* Research Into Secure Operating Systems
 - Role, 357
 - Role account(s), 869
 - access to, 870
 - authorized users and, 872–873
 - design access to, 875–877
 - settings of, 870
 - unauthorized users and, 871–872
 - Role-based Access Control. *See* Access controlled, role-based
 - Root, 871
 - Root account, 882
 - Rootkit* attack tool, 724–725
 - Round keys, 228
 - RSA
 - cipher, 234–237
 - digital signatures, 267–269, 293
 - Rule of acyclic creates, 70–71
 - Rule of transitive confinement, 441
 - Rule-based Access Control. *See* Mandatory Access Control
 - Rules of natural induction, 949
 - Rules of transformation, 136–139
 - RVM. *See* Reference Validation Mechanism
- S**
- SA. *See* Security Association
 - SA bundle. *See* Security Association, bundle
 - SAD. *See* Security Association, Database
 - Safe system, 48
 - Safety analysis model, 72–77
 - Salting technique, 320–321
 - Saltzer and Schroeder's principles, xxxiv
 - Sandboxes, 444–446, 635–636
 - Schematic Protection model, 66
 - demand and create operations and, 69–71
 - extending the, 79–83
 - filter function and, 68
 - Harrison-Ruzzo-Ullman model *vs.*, 78–79
 - link predicate and, 66–67
 - owner-based policy and, 68–69
 - safety analysis and, 72–77
 - Take-Grant Protection model and, 68–69
 - Scheme, 85
 - Search path, 864
 - Secret security level, 124
 - Secure Network Server Mail Guard, 434–436
 - Secure Shell protocol, 785, 788
 - login procedure and, 849
 - Secure Socket Layer, 291
 - alert protocol, 297–298
 - application data protocol, 298
 - change cipher spec protocol, 297
 - handshake protocol, 295–297
 - login procedure and, 849
 - record protocol, 294
 - session, 291
 - supporting cryptographic mechanisms and, 292–294
 - Secure systems
 - auditing mechanisms and, 706–707
 - definition of, 95, 125, 127



1080 Index

- Secured and trusted systems building, 456–464
- Security
 - clearances, 124
 - levels, 124–125
 - audit manager, 156
 - maximum and current, 127
 - system low, 156
 - unclassified, 124
 - policy, 194. *See also* Computer security
 - academic computer and, 111–114
 - access control types and, 103–104
 - breach of, 96
 - Chinese Wall model and, 169–177
 - clinical information systems, 177–179
 - definition of, 9, 95–99
 - dynamic, 200–201
 - mechanism, 9, 11–12, 98, 116, 117
 - model, 99
 - precision and, 114–119
 - trust and, 103
 - types of, 99–101
 - violation, 585
 - testing, 533–536
 - test assertions and, 538–539
 - test matrices and, 536–538
 - test specifications and, 539–540
 - using PGWG, 536
- Security Association, 300
 - bundle, 301
 - Database, 300
- Security assurance. *See* Assurance
- Security at the network layer: IPsec, 298–305
- Security building or adding, 501–505
- Security domains for TCSEC, 577–578
- Security Features User's Guide, 577
- Security flaws, 661–662
 - See also* Aslam's model; Vulnerability
- Security functional requirements, common criteria, 597–598
- Security functional testing, 534
- Security functions summary specification, 513
- Security kernel, 502
- Security mechanisms, layered architecture and, 500–501
- Security Parameter Index, 300
- Security Pipeline Interface, 433
- Security Policy Database, 299–300
- Security requirements testing, 534
- Security specification(s), 494–495, 520
- Security target, 594
- Self-healing property, 231
- Self-synchronous stream ciphers, 280–281
- Semi-informal assurance techniques, 478
- Sensitive data structure, 943
- Sensitive instruction, 943
- Separation of duties, 97, 152
- Separation of function, 152
- Separation of privilege principle. *See* Principle of, separation of privilege
- Sequencing flaw, 675, 908–909
- Service denial. *See* Denial of service
- Session key, 246
- Settings, start-up, 863
- SFUG. *See* Security Features User's Guide
- Shared Resource Matrix (SRM) methodology, 450–452
- Sharing of rights, 55–58
- Shoulder surfing technique, 849
- Signature packet, OpenPGP, 259–260
- Signatures, digital. *See* Digital signatures
- Simple security property, 127
- Simplicity and design, 341–343
- Simulation and expressiveness, 83–88
- Single-key. *See* Classical cryptosystems
- SMV. *See* Symbolic Model Verifier
- Sniffers program, xxxi
- Snooping, 7
- SNSMG. *See* Secure Network Server Mail Guard
- Social engineering, 21
- Software development models
 - assembly of, 492
 - exploratory programming, 491
 - extreme programming, 492
 - formal transformation, 491–492
 - prototyping, 491
 - system assembly from reusable components, 492
- SPD. *See* Security Policy Database
- Speaker recognition (verification), 329
- SPECIAL language, 548, 549, 550
- Specification(s), 13–14, 505, 520
 - external functional, 513–514
 - formal, 523
 - languages, 549
 - modeling, 738–740
 - modification, 521–522
 - security, 522–523
- Specification-based detection, 738



- SPI. *See* Security Parameter Index; Security Pipeline Interface
- SPM. *See* Schematic Protection model
- Spoofing. *See* Masquerading
- SQL. *See* Structured Query Language
- SRI model, 553–555
- SRM methodology. *See* Shared Resource Matrix (SRM) methodology
- SSE-CMM. *See* System Security Engineering Capability Maturity Model
- SSH protocol. *See* Secure Shell protocol
- SSL. *See* Secure Socket Layer
- ST. *See* Security Target
- Standards, Data Encryption, 282–32
- State Machine model, 208–209
- State of a system, defined, 31
- State transition, 32
function, 191
protection, 37–41
- State-based logging mechanism, 702
- Static identifiers, 367–368
- Stealing notion, 60
- Stealth virus, 620
- Storage of data, 877–880
- Stream ciphers, 277–281
- Strong hash function. *See* Cryptographic checksum function
- Strong mixing function, 253
- Strong noninterference, 204
- Strong one-way function. *See* Cryptographic checksum function
- Strong tranquility, 143
- Structural testing, 534
- Structured protection for TCSEC, 577
- Structured Query Language, 35
- Subcomponent, 511
- Subject set, 66
- Subject telegraph, 34
- Substitution ciphers, 220–221
- Subsystem, 510–511
- Suitability of requirements, 582
- Symbolic Model Verifier, 562
experience, 566
language, 562–564
proof theory, 564–565
- Symmetric cryptosystems. *See* Classical cryptosystems
- SYN flood attack, 762, 793–796
- Synchronous stream ciphers, 278–280
- Syntactic issues of auditing system design, 696–697
- System administrators and computer security, 21
- System logging, auditing system design, 672–673
- System program
design of, 873–880
requirements and, 870–871
threats against, 871–873
- System Security Engineering Capability Maturity Model, 604–606
using the, 606–607
- ## T
- Tagged architecture, 393
- Take right, 66
- Take rule, 53, 56
- Take-Grant Protection model, 53–54, 69
conspiracy and, 63–65
graph of, 64 (Fig. 3–4)
interpretation of, 58–60
sharing of rights and, 55–58
theft in, 60–63
- TAM model. *See* Typed Access Matrix model
- Target of Evaluation, 581
- TCB. *See* Trusted Computing Base
- TCP server and memory allocations, 794–796
- TCSEC. *See* Trusted Computer System Evaluation Criteria
- Technical review mechanism, 528–531
- Technology, laws and customs and, 18–19
- Telnet protocol, 284
- Temporal logic systems, 954–956
- Termite and stay resident virus, 620
- Testing, 15–16
flaw, 651
formal verification and property-based, 618
functional, 533
program security and, 913–917
requirements for TCSEC, 576
structural, 534
system, 534
third party, 534
unit, 534
- TFM. *See* Trusted Facility Manual
- Tg-path, 55
- Theft in Take-Grant Protection model, 60–63
- Theory of computer viruses, 626–630
- Theory of malicious logic, 626–630

1082 Index

- Theory of penetration analysis. *See* Gupta and Gilgor's theory of penetration analysis
- Third-party testing, 534
- Threat(s)
 computer security and, 6–9
 security objectives and, 498–499
 system program and, 871–873
 See also Attack(s)
- Threshold scheme, 399
- Thumbprinting, 760–761
- Ticket, 66, 250
- Tiger team attack. *See* Penetration studies
- Time
 Escrowed Encryption Standard (EES) and, 265
 Yaksha security system and, 265
- Time-based inductive learning, 729
- Timestamps, 249–250
- TLS. *See* Transport Layer Security protocol
- TO. *See* Object set
- TOE. *See* Target of Evaluation
- TOE Security Functions, 592
- TOE Security Policy, 592
- Token, 326
- Top secret security level, 124
- Total isolation, 440
- Training and computer security, 21
- Tranquility principle, 142–143
- Transaction authorization rule, 183
- Transaction-oriented integrity security policies, 100
- Transformation procedures, 38
- Transition-based auditing, 703–704
- Transition-based logging mechanism, 703
- Transitions, 37–41
- Transitive confinement, rule of, 441
- Transitive nonlattice information flow policies, 412–413
- Translucent cryptography, 265
- Transport adjacency, 302
- Transport Layer Security Protocol, 291
- Transport mode, 298–299
- Transposition ciphers, 219–220
- Triple DES mode, 231–232
- Triple encryption mode, 283
- Tripwire program, 110
- Trojan horse(s), 614–615, 848
- Trust
 assurance and, 477–479
 computer security and, 101–103, 892–893
 definition of, 477
 notion of, 640
- Trusted Computer System Evaluation Criteria, 571
 assurance requirements, 576–577
 evaluation classes, 577–578
 evaluation process, 578
 functional requirements of, 575–576
 impacts, 578–581
 requirements in ITSEC not found in, 583
- Trusted Computing Base, 129n, 502
- Trusted distribution requirement for TCSEC, 576
- Trusted entities, 142–143
- Trusted Facility Manual, 577
- Trusted host mechanism, 850
- Trusted path requirements for TCSEC, 576
- Trusted system, 572
- Truth tables, 950–951
- TS. *See* Subject Set
- TSF. *See* TOE Security Functions
- TSP. *See* TOE Security Policy
- TSR. *See* Termite and Stay Resident virus
- TSR viruses, 620
- Tunnel mode, 298–299
- Turing machine, 49–53
- Type checking, 397–398, 903–904
- Typed Access Matrix model, 88–90
 authorization scheme, 89
 monotonic, 89
- ## U
- UID. *See* Unique Identifier for Device; User(s), Identification Number
- Unauthorized access to information. *See* Disclosure threat
- Unclassified security level, 124
- Uniform Resource Locator, 355
- Union notion, 117–118
- Unique Identifier for Device, 263
- Unit testing, 534
 implementation and, 489–490
- University of California Davis
 acceptable use policy of, 111–112, 989–992
 electronic mail policy and, 112–113, 959–988
 implementation of the policy at, 114, 977–988
 policy of, 113–114
- University of Michigan terminal system, penetration study of, 652–654
- UNIX
 abbreviations of access control lists and, 383–384

- absolute path names, 354
- obtaining location in, 884–885
- password, 311
- penetration study and, 656–658
- process privileges and, 889–890
- rdist (remote distribution), 740
- relative path names, 354
- salts and, 321
- users and, 818–819
- Unsafe system, 48
- Unused mechanism, 19
- Unwinding theorem, 195–197
- Upper layer security protocols, 295–298
- URL. *See* Uniform Resource Locator
- User identifiers and resource exhaustion, 893–894
- User interface and design, 874
- User(s), 355–356
 - access and, 846–852
 - access control lists and, 385
 - accessing role accounts, 871–873
 - classes, 776–777
 - development system and, 817–819
 - devices and, 857–859
 - DMZ Web server system *vs.* development system and, 822
 - group access and, 854–855
 - identification number, 355–356, 818–819
 - login procedure and, 848–850
 - security
 - processes and, 860–866
 - security policy, 845–846
 - Web server system in the DMZ and, 817–819
- Usurpation threat, 7

V

- Validation
 - bounds checking and, 902–903
 - checking for valid, not invalid, data, 904–905
 - checking input, 905–906
 - designing for, 907
 - error checking and, 904
 - fingerd buffer overflow flaw and, 676
 - flaw, 676
 - type checking and, 903–904
- VAX/VMS system
 - virtual machine monitor and, 945
- VCs. *See* Verification Conditions
- Verbs, 35
- Verification Conditions, 554–555
- Verification systems, 559–567

- Verification techniques
 - early formal, 551–559
 - formal, 545–548
 - full *vs.* property, 546
- Verified protection for TCSEC, 578
- Vertices, 55
- Vigenère cipher, 221–227
- Violable prohibition/limit flaw, 675, 677
- Virtual machine(s), 442–444
 - monitor, 942–943
 - paging and, 945–946
 - physical resources and, 944–945
 - privilege and, 943–944
 - structure, 941
- Virtual Private Networks, 799
- Virus(es). *See* Computer virus(es)
- Voice authentication, 329
- VPNs. *See* Virtual Private Networks
- Vulnerability, 498, 645
 - classification, 660
 - exploiting the, 645
 - See also* Threat(s)

W

- Waterfall life cycle model, 488–491
- WCs. *See* Web Server Consequences
- Weak tranquility, 143
- Web
 - anonymity on the, 371–375
 - identity on the, 366–377
 - See also* Internet
- Web server. *See* WWW server
- Web Server Consequences, 806–807
- Web server system in the DMZ. *See* DMZ, Web server system
- Well-formed formulas, 950
- WFFs. *See* Well-formed formulas
- While loop, 75
- Wildcards, groups and, 386
- Window manager process, 859
- Windows NT
 - access control lists and, 388–389
 - copy flag and, 42
 - logs of, 691–692
 - penetration study of, 658–659
- Wiretapping, 7
- Witness, 54
- Worms. *See* Computer worms
- Wrappers, 756–757
- WWW server

1084 Index

DMZ, 787–788, 812–814
DMZ Web server system and, 837
internal, 791
policy, 806–807
WWW-clone, 788

X

Xterm log file flaw, 674–676
xterm program, 661–662
X.509: Certification signature chains, 256–258

Y

Yaksha security system, 264–265