ISI/SR-78-10
March 1978

ARPA ORDER NO. 2223

PROTECTION ERRORS IN OPERATING SYSTEMS:

A Selected Annotated Bibliography and Index to Terminology

Jim Carlstedt

INFORMATION SCIENCES INSTITUTE
UNIVERSITY OF SOUTHERN CALIFORNIA
4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

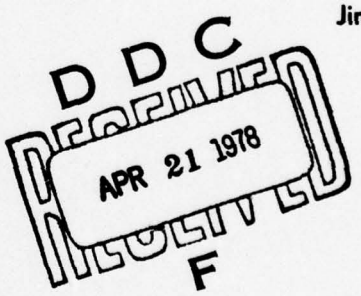
ISI/SR-78-10
March 1978

ARPA ORDER NO. 2223

PROTECTION ERRORS IN OPERATING SYSTEMS:

A Selected Annotated Bibliography and Index to Terminology

Jim Carlstedt



INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA

4676 Admiralty Way/ Marina del Rey/ California 90291
(213) 822-1511

THIS RESEARCH IS SUPPORTED BY THE ADVANCED RESEARCH PROJECTS AGENCY UNDER CONTRACT NO. DAHC15 72 C 0308, ARPA ORDER NO. 2223.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

(14) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ISI/SR-78-10 ✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* (6) Protection Errors in Operating Systems: A Selected Annotated Bibliography and Index to Terminology.		5. TYPE OF REPORT & PERIOD COVERED (9) Research Report.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Jim Carlstedt		8. CONTRACT OR GRANT NUMBER(s) DAHC 15 72 C0308 ✓
9. PERFORMING ORGANIZATION NAME AND ADDRESS USC/Information Sciences Institute ✓ 4676 Admiralty Way Marina del Rey, CA 90291		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order # 2223
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		(11) 12. REPORT DATE February 1978 Mar 78 13. NUMBER OF PAGES (12) 48 p.
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* ------		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release and sale; distribution unlimited.

(15) DAHC15-72-C-0308, ARPA Order-2223

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

------

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

operating systems protection, security, bibliography

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

(OVER)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

20. ABSTRACT

This report represents the current state of a bibliography on the subject of protection in computer operating systems. "Current state" means that the bibliography is incomplete; it is a byproduct of a research project in the field of protection, recently completed. The bibliography is being published in the belief that it may be useful as is, and that it might serve as the basis of a continuing effort to collect, annotate and index the more significant documents (reports, papers, articles, books, etc.) in the field. Ideally (especially in these days of computerized information bases and communication networks) workers in a research field will collaborate in developing and sharing their bibliographies--not only with simple annotations like this one but with more extensive comments and reviews. Perhaps this document can be a contribution in that direction and will stimulate owners of other "working" bibliographies to publish theirs. As noted below, this bibliography is online and may be accessed via the ARPANET (Defense Advanced Research Projects Agency network of interconnected scientific computers).

INTRODUCTION

This report represents the current state of a bibliography on the subject of protection in computer operating systems. "Current state" means that the bibliography is incomplete; it is a byproduct of a research project in the field of protection, recently completed. The bibliography is being published in the belief that it may be useful as is, and that it might serve as the basis of a continuing effort to collect, annotate and index the more significant documents (reports, papers, articles, books, etc.) in the field. Ideally (especially in these days of computerized information bases and communication networks) workers in a research field will collaborate in developing and sharing their bibliographies--not only with simple annotations like this one but with more extensive comments and reviews. Perhaps this document can be a contribution in that direction and will stimulate owners of other "working" bibliographies to publish theirs. As noted below, this bibliography is online and may be accessed via the ARPANET.

Because this bibliography was incidental to other work rather than an end in itself, litle time was spent in trying to make it comprehensive, or in supplying extensive annotations. It will also be noted that relatively little material has been added during the past year.

With respect to its subject, this bibliography was always intended to cover only a limited set of topics within the larger fields of operating system and computer security, specifically the following:

> Operating system requirements, policies and mechanisms, for insuring user and operating system program and data integrity and confidentiality.

The best way to get a feel for the topics included is to scan the index (see below). The following peripheral topics have been specifically excluded (except where they occur together with the main subject above):

> User identification and authentication
>
> Encryption
>
> Hardware reliability
>
> Error detection methods
>
> System generation and initialization
>
> Human integrity (e.g., administration and operations staff)
>
> Physical installation security

Communications security

Broader economic, social, and political issues

A reviewed document was also not included if the bulk of its subject matter, as a whole and by section, was other than that described above, even though it may have contained relevant fragments.

Gray areas exist in which the boundaries of the protection field are not well defined. Ultimately, probably no completely satisfactory definition of protection exists that stops short of including the entire constraintive aspect of programming and operating systems, i.e., of including policies and mechanisms for all kinds of unintended occurrences.

Selectivity has also been based on quality and significance. The attempt was made to exclude documents deemed not to be potentially useful for future reference personally, and which could not be recommended to students of protection. Exclusion of such entries from published bibliographies should be a professional ethic, even at the expense of exposing the often subjective judgments of their editors.

In general, a reviewed document has been included in this bibliography if it was judged to contribute original and potentially helpful observations, insights, ideas, or descriptions; or to express old ones in new and potentially helpful ways.

A document was rejected for any of the following "editorial" reasons:

It has been superseded by a revision, or its essence has been republished in a more accessible source.

It is totally obsolete or is of historical interest only.

Its editorial quality is inadequate.

As a result of the level of effort put into it and the restrictions applied, only 173 entries appear. Approximately 154 documents that qualified by topic (at least to some extent) were reviewed and rejected or included and later dropped. In addition, about 65 entries are currently on my list of unseen documents whose titles suggest relevance.

A unique feature of this bibliography is its index. An indication of the maturity of a field of research is the extent to which its technical terms are defined and used consistently. Access to definitions and usages is especially important to students in the field. Also, the technical terms used usually comprise a fair index to the content of a document. For these reasons, and to experiment with the actual usefulness of such an index, as each document was reviewed its key terms, concepts, and topics were noted as a set of keywords. Little attempt has been made to standardize the forms occurring in the index, or to insure consistency, for example by going back to revise the

keyword set of a document reviewed two or three years earlier. None of the massaging ordinarily needed to enhance the usefulness of an index has been applied to this one; like the bibliography itself, it is published in its current unfinished state.

This document corresponds to a file residing in the computer at ISI whose ARPANET hostname is ISIB, and which runs under the TENEX Operating System. The TENEX filename of this document is <REPORTS>PBIB.TXT; it may be accessed over the ARPANET via an FTP program.

Each entry in the bibliography has the following fields:

An identifier local to this bibliography.

Author's (authors') name(s) as appearing on the title page of the document.

Title of the document.

Source data fields: periodical issue, publishing agency, publisher's document identifier, date, pages, etc. In some cases an NTIS order number is also provided.

An annotation (usually), starting with the characters "[*" and ending with "*]", which attempts to summarize the most significant topics or features of the document in about one sentence.

Keywords or lists of keywords, enclosed by angle brackets "<" and ">" and separated (within lists) by semicolons. These can also occur as part of the annotation fields of an entry.

Fields of an entry are separated by double spaces, except that the local identifier and the author-name field are separated by a tab character. Entries themselves are separated by null lines.

The following abbreviations occur in the bibliography:

ACMnn Proceedings, ACM Conference 19nn

CACM Communications of the ACM

IBM74d Data Security and Data Processing, Vol. 4 Study Results: Massachusetts Institute of Technology IBM G320-1374 74.6

ICRS75 Proceedings, International Conference on Reliable Software, April 21-23, 1975

IWCA73 Proceedings, International Workshop on Computer Architecture, Grenoble, June 26-28, 1973

IWPOS74 Proceedings, International Workshop on Protection in Operating Systems, Rocquencourt, France, August 13-14, 1974. Institut de

Recherche d'Informatique et d'Automatique, BP 5 - Rocquencourt 78150 Le Chesnay, France

FJCCnn AFIPS Conf. Proceedings 19nn Fall Joint Computer Conference

NCCnn AFIPS Conf. Proceedings 19nn National Computer Conference

NTIS National Technical Information Service

PLOS73 Proc. of ACM SIGPLAN-SIGOPS Interface Meeting: Programming Languages - Operating Systems, April 9-12, 1973

SJCCnn AFIPS Conf. Proceedings 19nn Spring Joint Computer Conference

SOSP75 Proc. of the Fifth Symposium on Operating Systems Principles, November 19-21, 1975

BIBLIOGRAPHY

Abb+76 Abbott, R.P.; Chin, J.S.; Donnelley, J.E.; Konigsford, W.L.; Tokubo, S.; Webb, D.A. & Linden, T.A. (ed.) Security analysis and enhancements of computer operating systems. National Bureau of Standards Institute for Computer Sciences and Technology NBSIR 76-1041 76.4 68p [☆ Brief informal discussions of security flaws and enhancements; brief design overviews and descriptions of flaws found in IBM's OS/MVT, the UNIVAC 1100 series operating system, and the TENEX operating system for the DEC PDP-10. ☆] <security enhancements, types; errors, integrity, taxonomy/categories/examples>

AmbH77 Ambler, Allen L. & Hoch, Charles G. A study of protection in programming languages. SIGPLAN Notices 12,3(77.3) (Proc. ACM Conf. on Language Design for Reliable Software) 25-40 [☆ Compares protection features of Pascal, Concurrent Pascal, Euclid, CLU, and Gypsy, with respect to <abstract data types; modules; scope rules; parameter passing>, using the "prison mail system" problem as an example. ☆] <access control features, language; classes; protection principles, languages; selective access>

Ames74 Ames, Stanley Richard, Jr. File attributes and their relationship to computer security. Case Western Reserve Univ. Dept. of Computing and Information Sciences Report No. 1167 74.6 89p [☆ Successively refined models of a tree-structured file system to include security requirements governing observation and modification of file attributes. ☆] <security events/axioms; repositories; agents; security classes; observe relation; modify relation; clearance; executors; clusters; view relation; alter relation; manipulators; accumulations; looks-at relation; changes relation; mandatory/discretionary controls>

Ande72a Anderson, James P. Information security in a multi-user computer environment. Advances in Computers Vol. 12 Academic Press 1972 1-36 [☆ Includes a broad survey of problems, models, and mechanisms. ☆] <isolation mechanisms; two state operations; i/o channels; virtual machines; program identity; implied sharing; descriptors; classification, derived>

Ande72b Anderson, James P. Computer security technology planning study. ESD-TR-73-51, vol. II. 72.10 137p. [☆ Detailed research and development proposal, to include security models, design of a <security kernel>, and architecture, hardware, and systems studies. ☆] <reference monitor; descriptor-based architecture; vulnerabilities, classes; implied sharing vulnerability; scavenging problem; incomplete parameter checking; asynchronous interrupt vulnerability; asynchronous i/o vulnerability; trojan horse problem; file authorization; authorization, files; hierarchical access control; data management systems; aggregation; inference; procedural controls>

Andr74 Andrews, G.R. COPS--a mechanism for computer protection. IWPOS74 5-25 Also, Cornell Univ. Dept. of Computer Science CU-CSD-74-214 74.10 35p [☆ <Capability-based protection mechanism>. ☆] <actor, process/procedure; mechanism, def.; principle of control; manager problem; mutual suspicion; confined computation, def.; protection state; monitors; primitives, protection state/environment; information structures, control/computing; capabilities; attributes, access/control; environment; basic monitor; message confinement; protection, logical vs. physical>

Att+76 Attanasio, C.R.; Markstein, P.W. & Phillips, R.J. Penetrating an operating system: a study of VM/370 integrity. IBM Systems J. 15,1(1976) 102-116

Atta73 Attanasio, C.R. Virtual machines and data security. Proc. ACM SIGARCH-SIGOPS Workshop on Virtual Computer Systems, 73.3 ACM 1973 206-209 [☆ <Virtual machines, security advantages>. ☆] <intention, explicit>

Atta74 Attanasio, C.R. An additional protection ring for virtual machine systems. IBM Research RC 5617 (#22625) 74.11.14 12p [☆ Proposed method of allowing communication between virtual machines by extending each with a virtual control store (VCS) and virtual microprocessor, protected from the user. ☆]

Bar+67 Barron, D.W.; Fraser, A.G.; Hartley, D.F.; Landy, B. & Needham, R.M. File handling at Cambridge University. SJCC67 163-167 [☆ Early <file access control scheme>. ☆] <part owner>

Baum75 Baum, I.R. The architectural design of a secure data base management system. (Thesis) Ohio State Univ. Computer and Information Science Research Center 1975

BelL73a Bell, D.E. & LaPadula, L.J. Secure computer systems: mathematical foundations. Mitre Corp. MTR-2547, vol. I ESD-TR-73-278, vol. I 73.11 37p NTIS: AD-770 768 [☆ Formal model defined from the viewpoint of general systems theory. ☆] <security, def.; classifications; need-to-know categories; access attributes; request sequences; decision sequences; access matrices; state sequences>

BelL73b Bell, D.E. & LaPadula, L.J. Secure computer systems: a mathematical model. Mitre Corp. MTR-2547, vol II ESD-TR-73-278, vol. II 73.11 58p NTIS: AD-771 543 [☆ Enhancement of the model presented in [BelL73a]; <privacy restrictions> are shown to be enforced under certain state transformations. ☆] <access attributes; access matrix; control access; error decision; question decision; security condition; ☆-property; security principle; interactivity principle; tranquility principle; classifications; categories (access privileges); request sequences; decision sequences; state sequences; rule, def.; rules of operation>

Bell74 Bell, D.E. Secure computer systems: a refinement of the mathematical model. Mitre Corp. ESD-TR-73-278, Vol. III MTR-2547, Vol. III 74.5

BelL76 Bell, D.E. & LaPadula, L.J. Secure computer system: unified exposition and Multics interpretation. Mitre Corp. MTR-2997 ESD-TR-75-306 76.3 129p

BelW74 Belady, L.A. & Weissman, C. Experiments with secure resource sharing for virtual machines. IWPOS74 27-33 [☆ Results and methodology of VM/360 evaluation. ☆] <isolation; sharing, information vs. other resources; channel programs, vulnerabilities; asynchronous modification of instructions; security, def.>

Bing65 Bingham, Harvey W. Security techniques for EDP of multilevel classified information. Burroughs Corp. 4424-65-112 65.12 186p NTIS: AD-476 557 [☆ Includes broad treatment of early hardware and software mechanisms. ☆] <user control profile; program reference table; file control code>

Bis+75 Bisbey, Richard, II; Popek, Gerald & Carlstedt, Jim Protection errors in operating systems: inconsistency of a single data value over time. Univ. of Southern California Information Sciences Institute ISI/SR-75-4 75.12 16p [☆ Type of protection error in which the value of a variable can be changed between two operations for which it is assumed to remain constant. ☆] <consistency, single variables; time-of-check-to-time-of-use error; parameter passing; errors, examples>

Bran73 Branstad, Dennis K. Privacy and protection in operating systems. Computer 6,1(73.1) 43-46 Also, Operating System Review 7,1(73.1) 9-17 [☆ Report of Workshop on Privacy and Protection in Operating Systems, Princeton, NJ, June 12-14, 1972. ☆] <generic weaknesses; residue; parameter checking, incomplete; asynchronous i/o; trojan horse attack; covert channels>

Brat75 Bratt, Richard Glenn Minimizing the naming facilities requiring protection in a computing utility. MIT Project MAC TR-156 Also, Honeywell Information Systems Federal Systems Operations ESD-TR-76-161 75.9 129p [☆ Simplification of the Multics security kernel by removing <reference name space> management and directory pathname resolution. ☆] <segment as unit of protection; descriptors; security kernel, name management functions; directory initiation; detectability; directories, protection; rings, protection; access control lists; lying to prevent detectability>

BroS71 Browne, Peter S. & Steinauer, Dennis D. A model for access control. 1971 ACM SIGFIDET Workshop: Data Description, Access and Control 241-262 [☆ Model for enforcing privacy restrictions on the basis of object <classification> and category, active object <authority>, and <access lists>. ☆] <protection groups; access clique; authorization; access control; categories; high water mark; objects; contextual sensitivity; privilege list>

Brow71 Browne, Peter S. Data privacy and integrity: an overview. 1971 ACM SIGFIDET Workshop: Data Description, Access and Control 237-240 [☆ Basic distinctions. ☆] <privacy; integrity, data; input validation;

isolation; identification; authorization>

Burk74 Burke, Edmund L. Synthesis of a software security system. ACM74 648-658 [☆ Software verification schema based on four <levels of representation>--mathematical model, <formal specification>, algorithms, and machine language; sketch of its application to the MITRE security model and a <security kernel> for the PDP 11/45. ☆] <reference monitor>

Car+71 Carroll, John M.; Martin, Robert; McHardy, Lorine & Moravec, Hans Multi-dimensional security program for a generalized information retrieval system. FJCC71 571-577 [☆ Scheme for controlling access at the file, record and item levels. ☆] <item key; item code; record key; record code; data base scheme; record level control; field level control>

Carl76 Carlstedt, Jim Protection errors in operating systems: validation of critical conditions. Univ. of Southern California Information Sciences Institute ISI/SR-76-5 76.5 33p [☆ Type of protection error in which insufficient validation exists to insure that an assumed condition holds. ☆] <validation errors; errors, examples; validity vs. integrity; criticality; influencability/influentiality>

Cha+75 Chamberlin, D.D.; Gray, J.N. & Traiger, I.L. Views, authorization, and locking in a relational data base system. NCC75 425-430 [☆ "Views prescribe what can be seen. Authorization prescribes what can be done to what is seen. Locks are a dynamic kind of authorization which prescribe what can be done to what is seen at this instant." ☆] <views, data base; authorization types, data base; grant; revocation; access control, data base; value dependent authorization>

ChaS76 Chandersekaran, C.S. & Shankar, K.S. On virtual machine integrity. (Letter) IBM Systems Journal 15,3(76) 264-269 [☆ Criticism of the imprecise terminology and the conclusion of [DonM75]. ☆] <hierarchically structured systems; security, def.; privacy, def.; integrity, def.; mechanism requirements; isolation, def.; security kernel>

ClaR75 Clark, David D. & Redell, David D. Protection of information in computer systems. Tutorial, Compcon 75 Fall IEEE Computer Society 1975 260p

CohJ75 Cohen, Ellis & Jefferson, David Protection in the Hydra Operating System. Operating Systems Review 9,5(1975) (SOSP75) 141-160 [☆ Description of the Hydra protection mechanisms and their application to several well-known <protection problems>. ☆] <protection, access/control; decisions, prior/future; decisions, unilateral/negotiated; procedural embedding; capabilities; ownership; subsystems; amplification; rights, generic/auxiliary; C-list; templates; domains; seal/unseal; rights, required/new; mutual suspicion; modification; propagation; conservation; confinement; initialization; revocation, immediate/permanent/selective/partial/temporal/sharing; freezing; accounting; lost objects; aliases>

Con+72a Conway, R.W.; Maxwell, W.L. & Morgan, H.L. On the implementation of security measures in information systems. CACM 15,4(72.4) 211-220

[☆ How access control decisions based on <static privacy conditions> can be made more efficiently at compile time. ☆] <access matrix; virtual user; data dependent conditions; compile time access control; security vs. privacy; decision rule; column system; diagonal system; compile-time checking>

Con+72b Conway, R.W.; Maxwell, W.L. & Morgan, H.L. Selective security capabilities in ASAP--a file management system. SJCC72 1181-1185 [☆ <Data base scheme> incorporating both <record level control> via <access control lists> and <field level control> via a <lock-and-key mechanism>. ☆] <directory of authorized users; security class, field; compile-time checking>

Coss72 Cosserat, D.C. A capability oriented multi-processor system for real-time applications. Proc. First Intern. Conf. on Computer Communication. IEEE Computer Society or ACM 282-289 [☆ Includes a brief description of the <capability architecture> of the Plessey System 250. ☆] <protected subroutines>

Coss74 Cosserat, D.C. A data model based on the capability protection mechanism. IWPOS74 35-53 [☆ Capabilities are treated as protected data structure pointers that can be freely copied. ☆] <capability segments; segment as unit of protection; control, transfers of; capabilities vs. pointers>

DalD74 Daley, Robert C. & Donohue, James P. Security and authorization --semantics and examples. IBM74d 135-149 [☆ Suggestions for improving IBM's Resource Security System, among other things by allowing disseminated control of authorization and providing for <protected subsystems>, including <execute-only access> to programs. ☆] <security officer; user groups; file groups; authorization classes; access types>

DalN65 Daley, R.C. & Neumann, P.G. A general-purpose file system for secondary storage. FJCC65 213-229 [☆ Early description of the Multics file system. ☆] <hierarchical access control; file directories; access paths; usage attributes; access control lists; trap list>

Den+74 Denning, D.E.; Denning, P.J. & Graham, G.S. Selectively confined subsystems. IWPOS74 55-61 [☆ Implementation of a <confinement mechanism> satisfying several necessary properties, and its insufficiency. ☆] <confinement problem; error conditions; protected subsystems; selective confinement; engagement; mutual exclusion, customer processes; closure, confidentiality; nonleakage; transitivity of engagement; declassification; disengagement; nonretention; covert channels; compile-time checking>

DenD77 Denning, Dorothy E. & Denning, Peter J. Certification of programs for secure information flow. CACM 20,7(77.7) 504-513 [☆ <Lattice model> of <information flow>, and <certification mechanism> based on it that verifies the security of information flow in a program, including the <confinement property>. ☆] <security classes; flow relation/policies; property lattice>

Denn64 Dennis, J.B. Program structure in a multi-access computer. MIT Project MAC TR-11 1964 [☆ Introduces the notion of <spheres of protection>. ☆]

Denn65 Dennis, Jack B. Segmentation and the design of multiprogrammed computer systems. J. of the ACM 12,4(65.10) 589-602 [☆ "...Hence it is preferable that a protection mechanism operate in the name space of a computation..." ☆] <segment as unit of protection; spheres of protection>

Denn75 Denning, Dorothy E. Secure information flow in computer systems. (Thesis) Purdue Univ. Dept. of Computer Science CSD TR-145 75.5 177p

Denn76a Denning, Dorothy E. A lattice model of secure information flow. CACM 19,5(76.5) 236-243 [☆ Formulation of information flow security requirements in terms of lattice properties; run-time and compile-time enforcement mechanisms for static and dynamic security environments. ☆] <information flow, explicit/implicit; security classes; security clearances; binding, static/dynamic; data mark machine; compile-time/ run-time enforcement>

Denn76b Denning, Peter J. Fault tolerant operating systems. ACM Computing Surveys 8,4(76.12) 359-389 [☆ Broad tutorial and survey: how <capability architectures> facilitate the techniques required for <confinement of errors>. ☆] <process isolation; environment, open/closed; capability list; access code; name, local/system; access list; storage capability; enter capability; multiple domain processes; protected entry; domain changing; attenuation of privilege; privilege state mechanism; privilege number; descriptors, privacy; virtual machines; resource control; decision verification; encapsulation, process>

DenV66 Dennis, Jack B. & Van Horn, Earl C. Programming semantics for multiprogrammed computations. CACM 9,3(66.3) 143-155 [☆ Introduces the notion of <capability list>. ☆] <spheres of protection; c-list; segment as unit of protection>

DonM75 Donovan, J.J. & Madnick, S.E. Hierarchical approach to computer system integrity. IBM Systems J. 14,2(1975) 188-202 [☆ Argues probabilistically that a <hierarchically structured operating system> on a <virtual machine monitor> can be more secure than a conventional multiprogramming system. ☆] <integrity, def.; security, def.; redundant security>

Down73 Downey, Peter J. Secure military computing systems. In Schell, Roger R.; Downey, Peter J.; and Popek, Gerald J. Preliminary Notes on the Design of Secure Military Computer Systems. Air Force Systems Command Electronic Systems Division Directorate of Information Systems Technology MCI-73-1 73.1 47p [☆ <Military security model> focusing on the structure of <access functions> and of a <security kernel>. ☆] <objects; subjects; access attributes (types); access relation; access monitor; security state; update monitor; constraints, updation (policy); consistency, policy; access control lists; locks and keys; compartments; need to know; user responsibility; control attribute; owner attribute;

accesses, normal vs. security update; breaches, internal vs. external>

Elli74 Ellis, Clarence A. Analysis of some abstract measures of protection in computer systems. Univ. of Colo. Department of Computer Science. CU-CS-043-74 74.5 46p NTIS: PB-235 297 [☆ Defines absolute, relative, and minimum <degree of protection> in terms of assignments of <access codes> to subjects and objects, gives assignments that maximize these, and applies this to hierarchical systems. ☆] <isolation level>

Engl72 England, D.M. Architectural features of System 250. Infotech State of the Art Report on Operating Systems. 1972 12p [☆ Description of the <capability architecture> of the Plessey System 250. ☆] <privileged mode, nonnecessity; capability registers; enter access>

Engl74 England, D.M. Capability concept mechanism and structure in System 250. IWPOS74 63-82 [☆ "...everything...is reducible to data structures...built up from...an interconnected network of capabilities..." ☆] <capability architecture; capability registers, real/virtual; load/store, capabilities; central capability segment; enter capability; privileged mode, nonnecessity; resources as protected data structures>

EvaL67 Evans, David C. & Leclerc, Jean Yves Address mapping and the control of access in an interactive computer. SJCC67 23-30 [☆ Proposal for integrating access control into the (virtual) addressing and parameter-binding mechanisms of an operating system. ☆] <parameter spaces; entry, protected; access path control; address map, control information>

Fabr68 Fabry, R.S. Preliminary description of a supervisor for a machine oriented around capabilities. Univ. of Chicago Institute of Computer Research Quarterly Progress Report No. 18, I-B 1-97 68.8 [☆ Sketch of design and implementation. ☆] <supervisor as interaction controller; capabilities; capability registers; segment as unit of protection; virtual machines; capability segments; access codes; capabilities, user; enter access>

Fabr71 Fabry, R.S. List-structured addressing. (Thesis) Univ. of Chicago 1971

Fabr73 Fabry, R.S. Dynamic verification of operating system decisions. CACM 16,11(73.11) 659-668 [☆ An independent consistency check for every decision involving process interactions. ☆] <isolation; message system>

Fabr74 Fabry, R.S. Capability-based addressing. CACM 17,7(74.7) 403-412 [☆ Rationale, comparison, and implementation considerations of schemes in which addresses are capabilities containing unique segment identifiers. ☆] <capability-based addressing; capabilities, representation integrity, approaches; capabilities and storage allocation; enter access/instruction>

Fent73 Fenton, J.S. Information protection systems. (Thesis) Cambridge

Univ. Computer Laboratory 1973

Fent74 Fenton, J. S. Memoryless subsystems. Computer J. 17,2(74.5) 143-147

Fer+74 Ferrie, J.; Kaiser, C.; Lanciaux, D. & Martin, B. An extensible structure for protected systems' design. IWPOS74 83-105 [☆ Object and type construction/destruction and call/return operations in a <capability-based system, extensible>. ☆] <agents; objects; mutual suspicion; transition type; descriptor (capability); encapsulated type; case/uncase operators; domains; environment of domain; operators, strong/weak; kernel; type transformation; call/return operations, domain>

Fer+75 Fernandez, E.B.; Summers, R.C. & Coleman, C.D. An authorization model for a shared data base. Proc. 1975 SIGMOD Intern. Conf. 9p [☆ Access matrix is extended to allow entries to contain predicates that depend on any data in the data base; most checking is done at compile time. ☆] <data base access control; access matrix, extended; access predicates; allocate (access type); functional access; administer (access type); inherit (access type); use (access type); compile-time checking>

FerS76 Fernandez, Eduardo B. & Summers, Rita C. Integrity aspects of a shared data base. NCC76 819-827

Frie70 Friedman, T.D. The authorization problem in shared files. IBM Systems J. 9,4(1970) 258-280 [☆ Broad treatment of access control requirements and considerations, recommending <compartmentalization> and data element <tagging>. ☆] <integrity, system; authorization, def.; privilege structure, complexity; names, concealment; passwords, file; privileges, program/user; field, protected; isolation, authorization mechanism; single-tag rule; group (protection category); levels of classification, undesirability; authority hierarchies, undesirability; clique, user; matrix, user-privilege; user profile>

Gain72 Gaines, R. Stockton An operating system based on the concept of a supervisory computer. CACM 15,3(72.03) 150-156 [☆ <File access control> via a <lock-and-key scheme>. ☆] <protection categories, file/process, read/write>

GeoS73 George, James E. & Sager, Gary R. Variables--bindings and protection. SIGPLAN Notices 8,12(73.12) 18-29 [☆ Six "mini-languages" differing in their schemes for controlling the scope and accessibility of names. ☆] <names, scope/sharing/protection>

Gla+75 Gladney, H.M.; Worley, E.L. & Myers, J.J. An access control mechanism for computing resources. IBM Systems J. 14,3(1975) 212-228

Grah68 Graham, Robert M. Protection in an information processing utility. CACM 11,5(68.5) 365-369 [☆ Protection requirements, and a model featuring <rings of protection>. ☆] <privileged instructions, deficiency; need-to-know principle; two-mode systems, deficiency; layers of protection, advantages; segment as unit of protection; access bracket; call bracket;

ring bracket; user descriptor; gatekeeper; calls, cross-ring; asynchronous modification of arguments>

GraD72 Graham, G.Scott & Denning, Peter J. Protection--principles and practice. SJCC72 417-429 [☆ <Access matrix model>: enforcement rules, protection commands, applicability, and implementation considerations. ☆] <levels of protection; mutually suspicious subsystems; memoryless subsystems; completeness requirement; objects; subjects; access rules, requirements; protection state; access attributes; monitor, object type; identification, subjects; update rules, access matrix; copy flag; owner attribute; control attribute; transfer command; grant command; copy flag, transfer-only; domains; hierarchies, subject, advantages; universal subject; untrustworthy subsystems; indirect access; debugging problem; trust; trustworthy subjects; revocability; c-list; access control list; authority list; lock list; key>

Grah71 Graham, G.Scott Protection structures in operating systems. (Masters Thesis) Univ. of Toronto Department of Computer Science 71.8

GriW76 Griffiths, Patricia P. & Wade, Bradford W. An authorization mechanism for a relational data base system. IBM Research RJ 1721 (#25154) 76.2.11 32p [☆ Mechanism providing for restricted sharing of data bases by allowing the granting and revocation of privileges and the definition and sharing of <views>. ☆] <data base privacy mechanisms; revocation, recursive; grants, labeled>

Har+75 Harrison, Michael A.; Ruzzo, Walter L. & Ullman, Jeffrey D. Protection in operating systems. CACM 19,8(76.8) 461-471 [☆ <Safety> of a protection system modelled by a given general <access matrix model> is undecidable. ☆] <rights, generic; configuration, protection system; own right; leaking, rights; undecidability, safety>

HarH75 Hartson, H.R. & Hsiao, D.K. Languages for specifying protection requirements in data base systems. Part I. Ohio State Univ. Computer and Information Sciences Research Center OSU-CISRC-TR-74-10 75.1 67p NTIS: AD/A-006 280 [☆ Models the <authorization process> and the <enforcement process> in a <security space> of users, authorizers, resources, operations, and value states. ☆] <access control model; authorization specification, validity/consistency; access condition; protection specification language; access matrix model, limitations; ownership, absolute/conditional; system administrator; withdrawal of rights; subownership; memoryless specifications; access history; authorized subspace; dominance, access history; access decision timing; domain of authorization; auxiliary invocation, domain; validation of inputs; progressive authorization; amplification; extended resources; franchise, user/operation/resource unit; partial rejection>

Harr75 Harrison, Michael A. On models of protection in operating systems. Lecture Notes in Computer Science. Vol 32: Mathematical Foundations of Computer Science 1975 (J. Becvar, ed.) Springer-Verlag 1975 46-60

Hart75 Harston, R. Rex Languages for specifying protection requirements

in data base systems--a semantic model. Ohio State Univ. Computer and Information Science Research Center CISRC-TR-75-6 75.8 248p NTIS: AD-018 284/0GA

Hel+75 Held, G.D.; Stonebraker, M.R. & Wong, E. INGRES--a relational data base system. NCC75 409-416 [☆ Access is controlled and integrity assured by means of <query modification>. ☆] <access control, data base; integrity, data base; restrict statement; integrity constraint>

HinS75 Hinke, T.H. & Schaefer, Marvin Secure data management system. System Development Corp. TM-(L)-5407/007/00 RADC-TR-75-266 75.11 197p NTIS: AD-A019 201

Hoff71 Hoffman, Lance J. The formulary model for flexible privacy and access controls. FJCC71 587-601 [☆ Proposes a <formulary> procedure to intercept all accesses to the data element to which it is attached and to determine the right of access, possibly via a dialogue with the user. ☆] <control profile, user; authority item; data dependent decisions>

HolB76 Hollingworth, Dennis & Bisbey, Richard, II Protection errors in operating systems: <allocation/deallocation residuals>. Univ. of Southern California Information Sciences Institute ISI/SR-76-7 76.6 17p [☆ Type of protection error in which a residual from a previous allocation remains accessible after a new allocation. ☆] <residual, content/access; errors, examples>

Hsi+74 Hsiao, D.K.; Kerr, D.S. & McCauley, E.J., III A model for data secure systems (Part I). Ohio State Univ. Computer and Information Science Research Center OSU-CISRC-TR-73-8 74.2 44p [☆ General protection specifications can be used to define users' views (and accessibility domains) of a data base, since they have the same form as retrieval specifications (Boolean functions). ☆]

Hsia68 Hsiao, David K. A file system for a problem solving facility. (Thesis) University of Pennsylvania The Moore School of Electrical Engineering Report No. 68-33 68.5 157p NTIS: AD-671 826 [☆ Both <file access control> and <record level control>, based on information in the <authority item> associated with each user. ☆] <login program, file; owner; authentication, file login program>

HsiM74 Hsiao, D.K. & McCauley, E.J., III A model for data secure systems. (Part II) Ohio State Univ. Computer & Information Science Research Center OSU-CISRC-TR-74-7 74.10 45p

IBMx71 (anon) OS/MVT with resource security: general information and planning manual. IBM GH20-1058-0 71.12 28p [☆ Description of Resource Security System features. ☆] <security levels; access category; need to know; security officer; definition of controlled resources; grouping of users and resources; authorization; program-restricted data sets; associative programs>

Jans74 Janson, Philippe Arnaud Removing the dynamic linker from the

security kernel of a computing utility. (Thesis) MIT Project MAC TR-132 74.6 128p [☆ Protection implications and design of a <dynamic linker> external to the security kernel, in terms of an abstract storage, name space, and protection model; implementation in Multics. ☆] <principle of least privilege; security kernel, criteria; object; subject; capability; gate; protected subsystem; name space model; system initialization; link fault handling; static storage allocation; rings, protection>

Jone73 Jones, Anita K. Protection in programmed systems. (Thesis) Carnegie-Mellon Univ. Department of Computer Science 73.6 145p NTIS: AD-765 535 [☆ Model of a protection mechanism based on <environments>: collections of <constituent rights>, associated with objects or their components; <declared rights>, associated with procedures; and <dynamic rights>, associated with procedure invocations. ☆] <environment crossing operators; protection, def.; object; access; type; enforcement rule; right transfer rule/primitives; capability; ownership; access matrix; environment binding rule; environment representations; checking, dynamic vs. static; parameter passing; amplification; protection state, transformations, closure; memoryless procedure; dynamic type creation; subsystem; suitability factor; need-to-know principle; accuracy measure; protection systems, comparison; policy vs. mechanism>

JonL75 Jones, Anita K. & Lipton, Richard J. The enforcement of security policies for computation. Operating Systems Review 9,5(75) (SOSP75) 197-206 [☆ For any viewing program and any security policy there exists a sound and <maximally complete mechanism>, but which cannot always be constructed; construction of sound <surveillance mechanisms>. ☆] <v functions; o operators; high water mark; policy, def.; mechanism, def.; soundness; completeness; confinement; memoryless subsystems; data security; observability postulate; information control vs. access control>

JonL76a Jones, Anita & Lipton, Richard J. (Letter) Operating Systems Review 10,2(76.4) 7-8 [☆ Reply to [Rote76] on the <soundness> of the privacy restriction mechanism at issue. ☆]

JonL76b Jones, Anita K. & Liskov, Barbara H. A language extension for controlled access to shared data. IEEE Trans. on Software Engineering SE-2,4(76.12) 277-285 [☆ Rules enforceable at compile time governing the binding of <capabilities (variables)> to strongly typed, structured objects, and implementing a wide class of controlled sharing policy. ☆] <compile-time checking; type-module; qualified type; binding rules; amplification>

JonW75 Jones, Anita K. & Wulf, William A. Towards the design of secure systems. Software--Practice and Experience 5,4(75.10-12) 321-336 [☆ Distinction between policies and mechanisms; capability-based mechanisms; the Hydra mechanism; types and examples of policies that can be implemented in Hydra. ☆] <objects; security policy, def.; appropriateness, policy; requirements, protection mechanisms; data base protection; capabilities, extended; mechanisms, functions; amplification; local name space; rights, kernel/auxiliary; capability transfer operations; environment crossing operations; checkrights; amplifyrights; template;

courier policy; monitoring policy; revocation policy; confinement policy; filtering policy; gatekeeper>

KamU77 Kam, John B. & Ullman, John D. A model of statistical databases and their security. ACM Trans. on Database Systems 77.3

KarS74 Karger, Paul A. & Schell, Roger R. MULTICS security evaluation: vulnerability analysis. USAF Electronic Systems Division ESD-TR-74-193, Vol. II 74.6 155p [☆ Overview of Multics security controls; detailed descriptions of several vulnerabilities identified, confirmed, and exploited during an early 1973 analysis. ☆] <multi-level security; master mode; access control lists; rings, protection; subverter program; vulnerabilities, descriptions; validation of arguments, insufficient; penetration techniques; trap doors, classes; procedural vulnerabilities; utility programs, dump/patch; compiler trap doors; gatekeeper>

KohG75 Kohout, L. & Gaines, B.R. The logic of protection. Lecture Notes in Computer Science, Vol. 34 GI--5. Jahrestagung Springer-Verlag 1975 736-751

Lack74 Lackey, R.D. Penetration of computer systems. An overview. Honeywell Computer J. 8,2(74) 81-85 [☆ Categories of <penetration techniques> and some common <error types>. ☆]

Lam+77 Lampson, B.; Needham, R.; Randall, R. & Schroeder, M. Protection, security, reliability. Operating Systems Review 11,1(77.1) 12-14 [☆ Things to be done; misconceived problems. ☆] <absolute vs. defensive protection; numerical measure of security>

Lamp67 Lampson, Butler Wright Scheduling and protection in an interactive multi-processor system. (Thesis) Univ. of Calif., Berkeley 67.3 82p [☆ <Control protection> via a per-process vector of allowable instructions; <memory protection> implications of various addressing schemes. ☆]

Lamp68 Lampson, Butler W. A scheduling philosophy for multiprocessing systems. CACM 11,5(68.5) 347-360. [☆ Includes the scheme for <control protection> presented in [Lamp67]. ☆]

Lamp69 Lampson, B.W. Dynamic protection structures. FJCC69 27-38 [☆ Detailed treatment of <capabilities> and <domains>, especially with respect to problems of <control transfers> and sharing. ☆] <access key; rings; gates; proprietary programs; passwords, file; access control lists>

Lamp71 Lampson, Butler W. Protection. Operating Systems Review 8,1(74.1) 18-24 [☆ Use and possible implementations of the <access matrix>. ☆] <domain; message system; identification; object system; access attributes; copy flag; revocation; ownership; capability list; access key; access lock list; access control procedure, per object>

Lamp73 Lampson, Butler W. A note on the <confinement problem>. CACM 16,10(73.10) 613-615 [☆ The problem of preventing a <service program>

from leaking information. ☆] <covert channels; memorylessness; transitivity of confinement; masking, channels>

Lamp74 Lampson, Butler W. Redundancy and robustness in memory protection. Proc. IFIP Congress 74, Vol. 1 North-Holland/American Elsevier 1974 128-132 [☆ Approaches to protection as the control of efficient communication between memory-sharing domains, as opposed to protection as isolation of message-sending/receiving domains. ☆] <domain; chattels; protection, def.; protection, absolute/defensive; isolation; message model; memory environment; scope of domain; communication by value vs. pointer; authentication; trademarks; registrar; memory protection; sharing, slow/fast; capabilities>

LamS75 Lampson, Butler W. & Sturgis, Howard E. Reflections on an operating system design. CACM 19,5(76.5) 251-265 [☆ Strengths and weaknesses of the capability-based Cal operating system in retrospect. ☆] <capabilities; domains; c-lists; seals; access keys>

Laue74 Lauer, H.C. Protection and hierarchical addressing structures. IWPOS74 137-148 [☆ Advantages/disadvantages and protection implications of <nested address space systems> versus <global object name systems>. ☆] <scope of names>

Less68 Lesser, V.R. A multi-level computer organization designed to separate data-accessing from computation. Stanford Univ. Computer Science Department Tech. Rep. CS90 68.3

Lind73 Lindsay, Bruce Suggestions for an extensible capability-based architecture. IWCA73 20p [☆ Possibilities for the representation, protection, passing, creation, interpretation, & encapsulation/retrieval of <capabilities>.☆]

Lind74 Linden, T.A. Different goals for protection. IWPOS74 149-153 [☆ A protection mechanism should (1) provide for rigorous data security, (2) facilitate the construction of reliable software, (3) support the implementation of special protection mechanisms, and (4) aid in guaranteeing the protection mechanism's own integrity and correctness. ☆] <goals, protection mechanisms>

Lind76a Linden, Theodore A. Protection: a nuisance or an opportunity? Digest of Papers, COMPCON Fall 76 IEEE 30-35

Lind76b Linden, Theodore A. Operating system structures to support security and reliable software. ACM Computing Surveys 8,4(76.12) 409-445 [☆ Tutorial on small protection domains implemented via <capability-based addressing>; <extended-type objects>; and the applicability of these to reliable software and system security. ☆] <security, def.; subjects; objects; access modes; access right; domains; capabilities; access matrix model; domain switching; protected procedure; enter right; mutual suspicion; principle of least privilege; defensive programming; Trojan horse problem; intermediaries; directories; revocation; amplification; indirection; extended-type manager; modularity;

discretionary/nondiscretionary controls; classification systems>

Lipn72 Lipner, Steven B. Computer security research and development requirements. The MITRE Corporation MTR-142 73.2 <reference monitor; need to know; segment, basis for access control>

Lipn75 Lipner, Steven B. A comment on the confinement problem. Operating Systems Review 9,5(1975) (SOSP75) 192-196 [☆ Formalization of confinement requirements in terms of the <☆-property> and an approach to proving that this property hold for all <storage channels> and <legitimate channels> of an operating system; comment on the use of <virtual time> to eliminate <covert channels>. ☆] <confinement problem; hiding, information; high water mark; o-functions; v-functions>

LisZ74 Liskov, Barbara & Zilles, Stephen Programming with <abstract data types>. SIGPLAN Notices 9,4(74.4) (Proc. Symp. Very High Level Languages) 50-59 [☆ A form of <encapsulation> is provided by defining classes of objects in terms of the <cluster of operations> available on them. ☆] <type checking>

Mano71 Manola, Frank A. An extended data management facility for a general-purpose time sharing system. (Master's thesis) Univ. of Pennsylvania 71.05 NTIS: AD-724 801 160p [☆ Extensions to scheme of [Hsia68]: <deny/allow descriptions> for <record level control>; <field level control> by deleting restricted keywords and corresponding fields from accessed records. ☆]

ManW75 Manola, Frank A. & Wilson, Stanley H. Data security implications of an extended subschema concept. Naval Research Laboratory Report 7905 75.7.15 16p

Mart73 Martin, James Security, accuracy, and privacy in computer systems. Prentice-Hall 1973 640p [☆ Includes descriptions of a variety of authorization schemes, including IBM's Resource Security System. ☆] <data security, def.; authorization structures; stratification; compartmentalization (isolation); authorization tables, user/data; transaction types; categories, user/data; locks, data record; passwords, file; capability (training); lockwords; zones, data; authorization level; security levels; access categories; need to know; integrity, system; security officer; program restricted data; security by association>

Mcph74 McPhee, W.S. Operating system integrity in OS/VS2. IBM Systems J. 13,3(74) 230-252 [☆ Seven types of <system integrity> errors in forerunner systems, and their solutions in OS/VS2. ☆] <time-of-check-to-time-of-use problem; validity checking; identification, objects; storage protection; restricted names; authorized program facility; program authorization; user-supplied addresses; serialization mechanisms; user data passed as system data; errors, types/examples>

Mill75 Millen, Jonathan K. Security kernel validation in practice. CACM 19,5(76.5) 244-250 [☆ Detailed description of method and techniques used to prove security properties of a kernel for the PDP-11/45. ☆]

<security kernel; ☆-property; simple security condition; descretionary access control matrix; tranquility principle; indirect information paths; O-functions; V-functions; elimination rules; channels, legitimate/covert; time problems>

Mins76a Minsky, Naftaly Intentional resolution of privacy protection in database systems. CACM 19,3(76.3) 148-159 [☆ Operations on data retrieved from a data base by a user program (assumed written in a strongly typed language) can be restricted more strongly than with access control alone, by requiring the program to interact via a subschema containing <application rules> augmenting those of the given language. ☆] <intentional resolution; privacy (security); access control limitations; database protection; database subschema; right (part of application rule); brands; data analysis, prevention; intermediate results, hiding; confinement; confidence modules>

Mins76b Minsky, N. An activator based protection mechanism. Rutgers Univ. Dept. of Computer Science Technical Report No. 25 76.6 40p

MooC74 Moore, Charles G., III & Conway, Richard Program predictability and data security. Cornell Univ. Dept. of Computer Science TR 74-212 74.9 12p [☆ Capabilities can be controlled at the source language level, allowing identification of potential <indirect access> via analysis of the <information flow graph> as well as some <compile-time enforcement>, provided that the name interpretation and accessing semantics of that language are correctly defined and implemented.☆] <security matrix>

Moor73 Moore, B.J. A classification of central processor architecture. IWCA73 19p [☆ Basic architectural concepts and the relationships between them, including processes, address spaces, and protection domains and mechanisms. ☆]

Moor74 Moore, Charles G., III Potential capabilities in Algol-like programs. Cornell Univ. Dept. of Computer Science TR 74-211 74.9 19p [☆ Potential flow of information in either direction between a given uninterpreted block B and a given variable may be determined from the <information flow graph> (for which the construction algorithm is given) of the program in which B is contained. ☆] <path condition; flow condition; capabilities, potential>

Morr73a Morris, James H., Jr. Protection in programming languages. CACM 16,1(73.01) 15-21 [☆ Proposes <seals> and <trademarks> for maintaining privacy and integrity of program module data. ☆]

Morr73b Morris, James H., Jr. Types are not sets. Conf. Record of ACM Symposium on Principles of Programming Languages, 73.10.1-3 120-124 [☆ Introduction of <seal operators, opaque/transparent> to guarantee the integrity of operators on and representations of objects of dynamic types. ☆] <authentication (type checking); secrecy (of representations)>

Need72 Needham, R.M. Protection systems and protection implementations. FJCC72 571-578 [☆ Per process <capability segments>; addressing via

<indirection tables> in which segment names are bound. ☆]

Need73 Needham, R.M. Protection--a current research area in operating systems. International Computing Symposium 1973. North-Holland/American Elsevier Publishing Company 1974 123-126 [☆ Motivation for and description of the capability/domain approach. ☆] <mechanisms, def.; security of information, def.; Trojan horse problem; file access via given program; regime of protection; segment as unit of protection; capabilities; enter capability; asynchronous events>

NeeW74 Needham, R.M. & Walker, R.D.H. Protection and process management in the "CAP" computer. IWPOS74 155-160 [☆ Non-hierarchical protection in a hierarchically-structured system. ☆] <capability segment; indirection table; ENTER capability; protected procedure; hierarchical process structure>

Neu+76 Neumann, Peter G.; Feiertag, Richard J.; Levitt, Karl N. & Robinson, Lawrence Software development and proofs of multi-level security. Proc. 2nd Intern. Conf. on Software Engineering IEEE 1976 421-428

Neu+77 Neumann, Peter G.; Boyer, Robert S.; Feiertag, Richard J.; Levitt, Karl N. & Robinson, Lawrence A provably secure operating system. Stanford Research Institute Project 4332 Final Report 77.2.11 483p [☆ A five-stage design methodology for general-purpose operating systems, with assertions stated and proved in an assertion language common to all stages; and the design of a secure operating system achieved via that methodology, structured as a hierarchy of abstract machines. ☆] <alteration principle; detection principle; denial of service; leakage, information; capabilities; objects; access code; type manager; revocation, selective; lost-object problem; Trojan horse attacks; mediated access; mutually suspicious subjects; memoryless operation; military security classification; need to know; inference; confinement principle; ☆-property; security kernel; C-list; subjects; domains; rings; call and return mechanism; capability channels; revocable copy; distinguished entry>

Neum73 Neumann, Peter G. (reporter) Report of evening session on protection. SIGPLAN Notices 8,9(73.9) (PLOS73) p27 <hidden channels; move-without-reading; read-without-copying; user interface; principle of maximum security; protection as restriction>

Owen71a Owens, Richard C. Primary access control in large-scale time-shared decision systems. (M.S. Thesis) MIT Project MAC TR-89 71.07 91p

Owen71b Owens, R. Evaluation of access authorization characteristics of derived data sets. Proc. 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control 263-278

Palm73 Palme, Jacob Protected program modules in Simula 67. Research Institute of National Defense, Stockholm, Sweden FOA P Report C 8372 73.7 26p NTIS: PB-224 776 [☆ Requirements for <inter-module protection>

in programming systems and languages; most of them satisfied by Simula 67. A <HIDDEN specification> is proposed to prevent external access to class attributes.☆] <classes, Simula 67>

Palm74 Palme, Jacob Software security. Datamation 20,1(74.1) 51-55 [☆ Brief tutorial. ☆] <deduction problem; file keys; error patterns>

ParP74 Parnas, D.L. & Price, W.E. Using memory access control as the only protection mechanism. IWPOS74 177-181 [☆ Argues that a virtual memory that provides a separate address space for each process is the only protection mechanism needed for basic access limitation. ☆] <virtual memory as protection mechanism; need-to-know principle>

Pope73a Popek, Gerald J. Access control models. (Thesis) Harvard Univ. Center for Research in Computing Technology 73.2 153p NTIS: AD-761 807 [☆ Model based on a <restriction graph> whose nodes are <security objects> and edges are <access paths>; an <access map> from <actors> to nodes is implemented as a map from actors to boolean <key variables> and a map from edges to boolean <lock functions> of these variables; implementation algorithms. ☆] <locks and keys; collusion; filtering property; access graph; access control list>

Pope73b Popek, Gerald J. Correctness in access control. ACM73 236-241 [☆ <Set-theoretic model> of a <security system>, with proof of correctness. ☆] <inference problem; statistical access problem; kernel>

Pope74 Popek, Gerald J. Protection structures. Computer 7,6(74.6) 22-33 [☆ Brief but comprehensive survey. ☆] <mutually suspicious subsystems; memoryless process; statistical access; inference; errors, examples; parameter checking; synchronization; higher level errors; hidden channels; security object; passive object; active object; protection data; update program; enforcement rule; access matrix; rights transfer primitives; capabilities; environment binding; domain; sphere of protection; c-list; revocation; locked boxes; access control list; rings; seal/unseal; data-dependent access decisions; formularies; principle of least privilege; grain of protection; security kernel; virtual machines; locks and keys; scope of names; redundant checking; penetration>

PopK74a Popek, Gerald J. & Kline, Charles S. A verifiable protection system. SIGPLAN Notices 10,6(75.6) (ICRS75) 294-304 [☆ <Security kernels>, <virtual machines>, and program verification; design considerations of the UCLA-VM System. ☆] <security sensitive instructions; security kernel, functions; input/output; data security, def.; security objects; access predicates; updater; capability faulting; interruptibility; privileged code, exclusion from user process; scheduling; file management kernel; existence, knowledge of; kernels, levels; levels of mechanism; direct access predicate; indirect access predicate; objecthood; Morse Code problem; vulnerabilities, multilevel>

PopK74b Popek, Gerald J. & Kline, Charles S. Verifiable secure operating system software. NCC74 145-151 [☆ Approach to providing verifiable security; the applicability of <security kernel> designs, virtual machine

work, and program verification techniques; the UCLA-VM system. ☆] <constraints; security objects; accessible sets; policy, def.; modularity; owner; virtual machines; security primitives; Morse Code problem; updater>

Pric73 Price, William Robert Implications of a virtual memory mechanism for implementing protection in a family of operating systems. (Thesis) Carnegie-Mellon Univ. Computer Science Dept. 73.6 251p NTIS: AD-766 292 [☆ Operating system design featuring local working subsets of virtual address spaces as domains of protection; Parnas specifications of the operating system modules; formal verifications. ☆]

Rede74 Redell, David D. Naming and protection in extendible operating systems. (Thesis) MIT Project MAC TR-140 74.11 166p [☆ Features and goals of typical capability systems, especially those related to <revocation> and <type extendibility>, and a proposed implementation for a system that handles both these problems using <sealed capabilities>. ☆] <base level (kernel); domains and processes; communication, interprocess/interdomain; lost object problem; capabilities, protection of; identifiers, unique; caretaker domain; process control as capability privilege; ownership privilege; capabilities, indirect, chains; amplification, authorization; operators, non-monadic; capabilities, sealed; capability systems, goals; subletting; revoker capability; revocability, revocable; trust, changing levels; locker capability; extender capability; parameters, revocable; directories in capability system; comprehensibility of protection mechanisms; propagation, knowledge>

RedF74 Redell, D.D. & Fabry, R.S. Selective <revocation of capabilities>. IWPOS74 197-209 [☆ Goals for revocation schemes; two domain-independent schemes based on the notions of <indirection and control>; base-level implementation. ☆] <revocability of revocability; copy of capability, literal/revocable>

Rhod75 Rhode, R. Secure multilevel virtual computer systems. Mitre Corp. MTR-2890 ESD-TR-74-370 75.2 33p NTIS: AD/A-007 059

Rob+75 Robinson, Lawrence; Levitt, Karl N.; Neumann, Peter G. & Saxena, Ashok R. On attaining reliable software for a secure operating system. SIGPLAN Notices 10,6(75.6) (ICRS75) 267-284 [☆ Design methodology for and features of an operating system, with security and concurrent verification as goals. ☆] <type manager; protection theorems; security problems, special; security, def.; security kernel; capabilities; access vector; capability manager>

Rote74 Rotenberg, Leo J. Making computers keep secrets. (Thesis) MIT Project MAC TR-115 74.2 393p [☆ Mechanisms designed to solve classes of problems associated with <proprietary services; privacy restrictions>; and <authority hierarchies>. ☆] <segments; access control lists; capabilities; capability lists; domains, postulates; protection, abstract formulation; message system; caretaker program; encapsulation; argument passing; name spaces; naming hierarchy; access control packet; revocation; ownership of domains; call-out; call-in; capabilities, passing of; hidden data; argument spying; blind service; benign domain; billing; accounting

channel; proprietary services, maintenance; locksmith; call packet;>

Rote76 Rotenberg, Leo J. (Letter) Operating Systems Review 10,2(76.4) 5-6 [☆ On the <soundness> of the privacy restriction mechanism described in [Rote74]. ☆]

Saal77 Saal, Harry J. A hardware architecture for controlling information flow. IBM Research RC 6414 (#27675) 77.2.28 30p [☆ Proposes an enhanced <capability vector mechanism> that improves on traditional capability schemes, which are shown to be inconsistent with the <least privilege principle> and with <information flow requirements>. (Abstract) ☆]

SalS75 Saltzer, J.H. & Schroeder, M.D. The protection of information in computer systems. Proc. of the IEEE 63,9(75.9) 1278-1308 [☆ Tutorial survey: basic principles; <list-oriented mechanisms> vs. <ticket-oriented mechanisms>; <dynamic authorization>, <authority structures>, and <protected subsystems>. ☆] <access control list; capabilities; design principles; failsafe principle; separation of privilege; least privilege; psychological acceptability; descriptor; principal; segment; domain; revocation; propagation; review of access; access controller; protection group; self control; hierarchical control; prescript; discretionary controls; sensitivity levels; confinement; operator-type restrictions; encapsulation>

Salt73 Saltzer, J.H. Protection and the control of information sharing in Multics. CACM 17,7(74.7) 388-402 [☆ Detailed description of the protection mechanisms associated with the Multics file system and virtual memory, with design principles and insights. ☆] <access control list; segment descriptor; permission principle; least privilege principle; protected subsystem; principals; compartments; access modes, primitive; intent specification; trap extension; file backup; system administrator; directories; authority hierarchy; locksmith; propagation of protection specifications; gates; rings of protection; argument checking; i/o operations; channel programs; system initialization; storage residue>

Salt74 Saltzer, Jerome H. Ongoing research and development on information protection. Operating Systems Review 8,3(74.7) 8-24 [☆ Informal survey of work at about 30 sites. ☆]

Salt76 Saltzer, J.H. Technical possibilities and problems in protecting data in computer systems. In R. Dierstein, H. Fiedler, and A. Schulz, Datenschutz and Datensicherung. J.P. Bachem Verlag, Cologne, Germany 76.9 27-36 [☆ A brief survey of protection methods and problems. ☆] <policy, informality; access control lists vs. capabilities; access control, container vs. data; limited-use systems; protected subsystems; classification systems; information flow, control; confinement; sharing, arbitrary patterns; inference>

Scha75 Schaefer, Marvin Secure data management system preliminary mathematical model. System Development Corporation RADC-TR-74.352 75.2 42p NTIS: AD-A007 748 [☆ Description and indicated proof

of a model of a <data access monitor> for a militarily secure data management system to be implemented in the Multics operating system. ☆] <classification; clearance; category; compartment; need to know; ☆-property; domination; high water mark; ownership; delete access; classification basis, record/field>

Schi75 Schiller, W.L. The design and specification of a security kernel for the PDP-11/45. Mitre Corp. MTR-2934 ESD-TR-75-69 75.5 117p NTIS: AD-A011 712 [☆ Design approach (levels of abstraction) and details of process management, segment management, security functions, and i/o. ☆]

Schr72 Schroeder, Michael D. Cooperation of mutually suspicious subsystems in a computer utility. (Thesis) MIT Project MAC TR-104 72.9 167p [☆ Design of a processor to provide controlled passing of parameters in <cross-domain calls> within a single computation. ☆] <protection, def./objectives; domain; capability; protected subsystem; segment as unit of protection; gates; access, static/dynamic; access control lists>

Schr75 Schroeder, Michael D. Engineering a security kernel for Multics. Operating Systems Review 9,5(1975) (SOSP75) 25-32 [☆ Notion of <security kernel>; motivation, method and activities involved in simplifying the Multics security kernel to make it easier to verify. ☆] <least common mechanism; rings, protection>

SchS72 Schroeder, Michael D. & Saltzer, Jerome H. A hardware architecture for implementing protection rings. CACM 15,3(72.3) 157-170 [☆ Description of the Multics ring structure. ☆] <segment as unit of protection; domains; gates; ring structure/crossing; ring brackets, read/write/execute>

SevT74 Sevcik, K.C. & Tsichritzis, D.C. Authorization and access control within overall system design. IWPOS74 211-224 [☆ How the kernel of the SUE operating system solves various protection problems. ☆] <capabilities, control of transfer; rights, quantitative; isolation/interaction; suspicion; confinement; sponsor; excessive use of resources; volume, foreign/local; revocation; indirect capabilities>

SevT75 Sevcik, K.C. & Tsichritzis, D.C. Operating system design with security as an objective. INFOR J. 13,2(75.6) 159-174

Smit74 Smith, Grant N. The state of practice of computer security. IBM74d 163-178 [☆ Protection mechanisms found in current systems. ☆] <integrity, data/system; privilege levels; subfile access control; user categories; access types; access control list; virtual memory; virtual machines; virtual I/O>

Spi+74 Spier, Michael J.; Hastings, Thomas N. & Cutler, David N. A storage mapping technique for the implementation of protective domains. Software--Practice and Experience 4(74) 215-230 [☆ Domain is a gated <caretaker procedure> together with the data encapsulated with it; implementation of an <activation/incarnation mechanism>. ☆] <domain

(protected subsystem); kernel>

Spie73 Spier, Michael J. A model implementation for protective domains. Intern. J. Computer & Info. Sciences 2,3(73.9) 201-229 [☆ A domain architecture in which domains are represented by sets of descriptors specifying not only allowable primitive access modes but also the mapping of information structures (objects of processing) onto storage areas (objects of ownership). ☆]

Spie74 Spier, M.J. A system theoretic look at the complexity of access control mechanisms. IWPOS74 225-241 [☆ The <complexity of an access control mechanism> is measured in terms of inter-module connectivity. ☆] <isolation; memorylessness and complexity>

Srin72 Srinivasan, C.V. A framework for a theory of protection. Rutgers Univ. Dept. of Computer Science Technical Report #16 72.05

Stor75 Stork, D.F. Downgrading in a secure multilevel computer system: the formulary concept. Mitre Corp. MTR-2924 ESD-TR-75-62 75.59p NTIS: AD-A011 696

StoW74 Stonebraker, Michael & Wong, Eugene Access control in a relational data base management system by query modification. ACM74 180-186 [☆ Retrieve access is controlled by automatically ANDing access restrictions to the user's stated retrieval conditions prior to their interpretation. ☆] <relational data base, access control; query modification>

Tsic72 Tsichritzis, Denis System Security. IBM Thomas J. Watson Research Center RC 3989 72.8.17

Tsic73 Tsichritzis, D. Reliability. Lecture Notes in Computer Science Vol. 30: Advanced Course on Software Engineering F.L. Bauer (ed.) Springer-Verlag 1973 319-373 [☆ Introductions to basic concepts of protection and security. ☆] <protection vs. security; domain; monitor; unique names; capabilities; revocation; indirect capabiiities; mutually suspicious processes; access matrix; access control list; data dependent access>

Tsic74 Tsichritzis, D. A note on protection in data base systems. IWPOS74 243-248 [☆ Problem of <data base protection> is too complex to be solved by current operating system protection mechanisms. ☆] <capabilities, inadequacies; capability procedures; protection rings, inadequacies>

Ulha75 Ul Haq, Mohammed Inam Insuring individual's privacy from statistical data base users. NCC75 941-946

Vand69 Vanderbilt, Dean Hanawalt Controlled information sharing in a computer utility. (Thesis) MIT Project MAC TR-67 69.10.24

Wal+74a Walter, K.G.; Ogden, W.F.; Rounds, W.C.; Bradshaw, F.T.; Ames,

S.R. & Shumway, D.G. Primitive models for computer security. Case Western Reserve Univ. Department of Computing and Information Sciences ESD-TR-74-117 74.1.23 36p NTIS: AD-778 467 [☆ Model of <governmental security>, and modified version in which the <repositories> are directories and files of a tree-structured file system. ☆] <agents; need to know; security classes; classification; clearance; information transfer path; need to modify; discretionary/mandatory security>

Wal+74b Walter, K.G.; Ogden, W.F.; Rounds, W.C.; Bradshaw, F.T.; Ames, S.R.; Biba, K.J.; Gilligan, J.M.; Shaeffer, D.D.; Shaen, S.I. & Shumway, D.G. Modeling the security interface. Case Western Reserve Univ. Dept. of Computing and Information Sciences Report No. 1158 74.8 130p [☆ Approach to the design of a <security kernel> for <military security> in Multics, by a process of successive refinement of models, and corresponding proofs that security assumptions continue to hold for each successive model. ☆] <security system; security perimeter; repositories; agents; classification; clearance; information transfer path; discretionary/mandatory security>

Wal+75 Walter, K.G.; Schaen, S.I.; Ogden, W.F.; Rounds, W.C.; Shumway, D.G.; Schaeffer, D.D.; Biba, K.J.; Bradshaw, F.T.; Ames, S.R. & Gilligan, J.M. Structured specification of a security kernel. SIGPLAN Notices 10,6(75.6) (ICRS75) 285-293 [☆ Design methodology for a <governmental security> system: successive refinement of models, about which basic theorems can be proved. ☆] <security kernel; information security, models; information flow monitoring; security system; classification; clearance; sensitivity level; repositories; agents; transfer relation/path; executors; alter list; view list>

Walk73 Walker, R.D.H. The structure of a well-protected computer. (Thesis) Cambridge Univ. 1974

Weis69 Weissman, C. Security controls in the ADEPT-50 time-sharing system. FJCC69 119-133 [☆ File access is granted if and only if the (accumulated) <authority level> of both the user and his terminal is greater than that of the file, and the <category> of both user and terminal includes that of the file, and the user is a member of the <franchise> of the file. ☆] <authority history; high water mark>

Weis75 Weismann, Clark Secure computer operation with virtual machine partitioning. NCC75 929-934

Whit75 White, J.C.C. Design of a secure file management system. Mitre Corp. MTR-2931 ESD-TR-75-57 75.4 29p NTIS: AD-A010 590 [☆ Preliminary design of a file management system intended to operate under the Mitre PDP-11/45 <security kernel>. ☆] <access control list; ☆-property; access control, hierarchical directory structure; semaphores, access restrictions>

Wul+73 Wulf, W.; Cohen, E.; Corwin, W.; Jones, A.; Levin, R.; Pierson, C. & Pollack, F. Hydra: the kernel of a multiprocessor operating system. CACM 17,6(74.6) 337-345 [☆ Salient features are the dynamic type/object hierarchy and the protection mechanisms for procedure activations and

parameter passing. ☆] <kernel; object; local name space; capability; call mechanism; protection vs. security; ownership; privilege hierarchy; type; kernel rights; auxiliary rights; parameter template; subsystem; walk right>

Zill73 Zilles, Stephen N. Procedural encapsulation: a linguistic protection technique. SIGPLAN Notices 8,9(73.9) (PLOS73) 142-146 [☆ Objects and types are characterized only by (the procedures that implement) the operators defined on them; the protection features required for such <procedural encapsulation> are satisfied by <domain architectures>. ☆]

INDEX