

---

## Summary

[JLS76] writes, “The Folklore is replete with stories of ‘secure’ protection systems being compromised in a matter of hours. This is quite astounding since one is not likely to claim that a system is secure without some sort of proof to support the claim. In practice, proof is not provided, and one reason for this is clear: although the protection *primitives* are apparently quite simple, they may potentially interact in extremely complex ways. Vague and informal arguments, therefore, often overlook subtleties that an adversary can exploit. Precision is not merely desirable for protection systems, it is mandatory.”

In this thesis, we construct a formalism which provides the required precision.

We initially study the decidability of safety in specific protection systems using the common access-matrix model. We build upon this to the point where we can treat the systems themselves as computational elements and thus create a formalism which produces safety results for entire classes of protection systems.

Our study of the manipulation of rights encompasses a study of responsibility; we achieve a unification of the  $\lambda_{\text{sec}}$ -calculi for stack inspection with that for data inspection, and demonstrate an extremely close coupling between the calculi of responsibility and our formalism for decidability of safety in protection systems.

We use our formalism to model numerous existing systems and understand their strengths and their weaknesses; hence we are able to create powerful new systems without such weaknesses. Several such systems exist; we describe a sample implementation.

# Contents

<b>Acknowledgements</b>	<b>ii</b>
<b>Table of Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Algorithms</b>	<b>xiii</b>
<b>List of Music</b>	<b>xiv</b>
<b>Notation and Symbols</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 What questions can we answer? . . . . .	1
1.2 What questions do we not answer? . . . . .	3
1.3 Our Approach . . . . .	3
1.3.1 Protection . . . . .	4
1.3.2 Policy . . . . .	4
1.3.3 Formalisms . . . . .	5
1.3.4 Mathematics . . . . .	5
1.3.5 Classes . . . . .	5
1.3.6 The Current Principal . . . . .	6
1.3.7 Case Studies . . . . .	6
1.3.8 New Applications . . . . .	7
1.3.9 Previous Work and Bibliography . . . . .	7
1.4 The Organisation of this Thesis . . . . .	7
1.4.1 In Pictures . . . . .	7
1.4.2 In Words . . . . .	8
<b>2 Fundamentals of Protection Systems</b>	<b>10</b>
2.1 An Introduction to Protection . . . . .	10
2.1.1 The Motivation for Protection . . . . .	11
2.1.2 A Model for Protection . . . . .	12
2.1.3 The Basic Mechanisms of Protection . . . . .	13
2.1.4 From Static to Dynamic Systems . . . . .	14
2.2 Basic Definitions . . . . .	14
2.2.1 The Trusted Computing Base . . . . .	15
2.2.2 Objects . . . . .	16
2.2.3 Principals . . . . .	16
2.2.4 Privileges . . . . .	16

2.2.5	Superusers . . . . .	17
2.2.6	The Current Principal . . . . .	17
2.3	Design Choices . . . . .	18
2.3.1	Positive Tests . . . . .	18
2.3.2	Compound Objects . . . . .	19
2.3.3	Self protection . . . . .	19
2.3.4	Representation of Privileges . . . . .	20
2.3.5	Properties of Explicit Representations of Privileges . . . . .	20
2.3.6	Properties of Implicit Representations of Privileges . . . . .	22
2.3.7	Representation of Principals . . . . .	22
2.4	Design Considerations . . . . .	23
2.4.1	The Principle of Minimal Privilege . . . . .	23
2.4.2	The Principle of Attenuation of Privileges . . . . .	23
2.4.3	Other Requirements . . . . .	24
2.5	Summary . . . . .	25
<b>3</b>	<b>Policy</b>	<b>26</b>
3.1	An Introduction to Policy . . . . .	26
3.2	A Definition of Policy . . . . .	27
3.3	Making Use of Policy . . . . .	27
3.3.1	A Minor Restriction of Policy . . . . .	28
3.4	Some Considerations for Policy . . . . .	29
3.4.1	Inconsistency of Policy . . . . .	29
3.4.2	Incompleteness of Policy . . . . .	30
3.5	Satisfying a Policy . . . . .	31
3.6	Constructing a Protection System . . . . .	32
3.7	Summary . . . . .	33
<b>4</b>	<b>A Formal Model of Protection Systems</b>	<b>34</b>
4.1	Introduction to the Formalism . . . . .	34
4.2	Preliminary Definitions . . . . .	35
4.3	A Formal Model of Protection Systems . . . . .	36
4.4	The General Safety Problem . . . . .	39
4.4.1	An Example of the Safety Problem . . . . .	39
4.4.2	Undecidability of the General Safety Problem . . . . .	41
4.5	Consequences of the Definition . . . . .	44
4.5.1	Graph Representations of Configurations . . . . .	44
4.5.2	The Difficulty of Search . . . . .	45
4.5.3	The Logical Power of Access Matrix Commands . . . . .	46
4.5.4	The Simplicity of Object Protection Code . . . . .	46
4.5.5	Countability of Configurations . . . . .	47

4.6	The Introduction of the Current Principal . . . . .	47
4.7	Example: An Inverse Relation . . . . .	50
4.7.1	An Introduction to the Inverse Relation . . . . .	50
4.7.2	The Construction of an Inverse Relation . . . . .	50
4.8	Summary . . . . .	54
<b>5</b>	<b>Mathematical Properties of Protection Systems</b>	<b>55</b>
5.1	Simulation of Protection Systems . . . . .	56
5.2	Equivalence of Protection Systems . . . . .	58
5.2.1	Introduction to Equivalence . . . . .	58
5.2.2	Properties of Equivalence . . . . .	60
5.2.3	Examples of Equivalence . . . . .	61
5.2.4	Restricted Equivalence . . . . .	62
5.3	Expressiveness of Protection Systems . . . . .	62
5.4	Safety Equivalence of Configurations . . . . .	63
5.5	Summary . . . . .	65
<b>6</b>	<b>Classes of Protection System</b>	<b>67</b>
6.1	Bounded Systems . . . . .	68
6.1.1	Construction of Bounded Protection Systems . . . . .	68
6.1.2	Decidability of the Class Safety Problem for Object Bounded Protection Systems . . . . .	73
6.1.3	Decidability of the Class Safety Problem for Subject Bounded Protection Systems . . . . .	74
6.2	Monoconditional Protection Systems . . . . .	84
6.2.1	Construction of the Monoconditional Protection System . . . . .	84
6.2.2	Decidability of the Class Safety Problem for Monoconditional Protection Systems . . . . .	84
6.3	Monooperational Protection Systems . . . . .	86
6.3.1	Construction of the Monooperational Protection System . . . . .	86
6.3.2	Decidability of the Class Safety Problem for Monooperational Protection Systems . . . . .	86
6.4	Boolean Protection Systems . . . . .	88
6.4.1	Construction of the Boolean Protection System . . . . .	89
6.4.2	Undecidability of the Class Safety Problem for Boolean Protection Systems . . . . .	90
6.5	Monotonic Protection Systems . . . . .	93
6.5.1	Construction of the Monotonic Protection System . . . . .	94
6.5.2	Undecidability of the Class Safety Problem for Monotonic Protection Systems . . . . .	94
6.6	Grammatical Protection Systems . . . . .	101
6.6.1	Construction of Grammatical Protection Systems . . . . .	101

6.6.2	Decidability of the Class Safety Problem for Grammatical Protection Systems . . . . .	102
6.7	Summary . . . . .	103
<b>7</b>	<b>Practical Protection Systems</b>	<b>104</b>
7.1	Introduction . . . . .	104
7.2	Designing for Simplicity . . . . .	105
7.3	A Hierarchy of Models . . . . .	105
7.4	Level 0 Primitive . . . . .	107
7.4.1	Formalisation . . . . .	107
7.4.2	The Class Safety Problem . . . . .	108
7.4.3	Examples . . . . .	108
7.4.4	Summary . . . . .	108
7.5	The Problem of Vetting . . . . .	108
7.6	Level 1 Primitive . . . . .	109
7.6.1	Formalisation . . . . .	109
7.6.2	The Class Safety Problem . . . . .	110
7.6.3	Examples . . . . .	110
7.6.4	Summary . . . . .	110
7.7	The Problem of Layers . . . . .	110
7.8	Level 1 Role Based . . . . .	111
7.8.1	Development . . . . .	111
7.8.2	Properties of Roles . . . . .	112
7.8.3	Formalisation . . . . .	113
7.8.4	The Class Safety Problem . . . . .	114
7.8.5	Specifics of Implementation . . . . .	115
7.8.6	Summary . . . . .	115
7.9	Level 1 Transitive . . . . .	116
7.9.1	Development . . . . .	116
7.9.2	Properties of Transitive Systems . . . . .	117
7.9.3	Formalisation . . . . .	118
7.9.4	Consequences of the Design . . . . .	119
7.9.5	The Class Safety Problem . . . . .	119
7.9.6	Specifics of Implementation . . . . .	119
7.9.7	Partial Ordering of Roles . . . . .	121
7.9.8	Exploiting and Extending the Transitive Model . . . . .	121
7.9.9	Summary . . . . .	122
7.10	Level 2 Transitive: The First Ideal System . . . . .	123
7.10.1	Development . . . . .	123
7.10.2	Formalisation . . . . .	126
7.10.3	Consequences of the Design . . . . .	127

7.10.4	Specifics of Implementation . . . . .	127
7.10.5	The Class Safety Problem . . . . .	128
7.10.6	Summary of the Formalism . . . . .	129
7.10.7	Summary of The Ideal Model . . . . .	129
7.11	Level 3 Transitive and Higher . . . . .	130
7.12	Other Basic Models . . . . .	130
7.12.1	Level 2 Role Based . . . . .	131
7.12.2	Level 0 Transitive . . . . .	131
7.12.3	Level 2 Primitive . . . . .	132
7.13	Summary . . . . .	132
<b>8</b>	<b>The Current Principal</b> . . . . .	<b>134</b>
8.1	Introduction and Preliminaries . . . . .	134
8.2	A Descriptive Introduction to Some Models . . . . .	135
8.3	The Immediate Principal Model . . . . .	136
8.3.1	Construction . . . . .	136
8.3.2	Explanation . . . . .	136
8.3.3	Examples . . . . .	137
8.4	The Root Principal Model . . . . .	137
8.4.1	Construction . . . . .	137
8.4.2	Explanation . . . . .	138
8.4.3	Examples . . . . .	138
8.5	Aside: Combining Principals . . . . .	138
8.6	The Stack Model . . . . .	139
8.6.1	Construction . . . . .	139
8.6.2	Explanation . . . . .	140
8.6.3	Examples . . . . .	140
8.7	The Data Model . . . . .	140
8.7.1	Construction . . . . .	140
8.7.2	Examples . . . . .	141
8.8	Aside: The Dangers of Ad-Hocery . . . . .	141
8.9	Indemnification . . . . .	142
8.10	Partial Principals: A New Foundation for Computation . . . . .	143
8.11	The $\lambda_{\text{sec}}$ -calculus . . . . .	145
8.11.1	Basic Expressions in $\lambda_{\text{sec}}$ -calculus . . . . .	146
8.11.2	Partial Context . . . . .	147
8.11.3	Basic Operational Semantics . . . . .	148
8.11.4	Overall Context of Execution . . . . .	148
8.11.5	Responsibility in $\lambda$ -calculus . . . . .	149
8.12	Data Inspection Calculus . . . . .	149
8.12.1	Construction . . . . .	149

8.12.2	Formalism . . . . .	150
8.12.3	Consequences . . . . .	152
8.13	Explicit Context Passing Form . . . . .	153
8.13.1	Construction . . . . .	153
8.13.2	Formalism . . . . .	154
8.13.3	Consequences . . . . .	157
8.14	Implicit Context Passing Calculus . . . . .	158
8.14.1	Construction . . . . .	158
8.14.2	Formalism . . . . .	159
8.14.3	Consequences . . . . .	160
8.15	Dependency Tracking Calculus . . . . .	161
8.15.1	Construction . . . . .	161
8.15.2	Consequences . . . . .	161
8.16	Stack Inspection Calculus . . . . .	161
8.16.1	Construction . . . . .	162
8.16.2	Formalism . . . . .	162
8.16.3	Consequences . . . . .	163
8.17	Consequences of Calculi for Identification of the Current Principal . . . . .	163
8.17.1	Relationships between our Calculi . . . . .	163
8.17.2	Partial Principals and Subterm Reduction . . . . .	164
8.17.3	Lazy Evaluation with Partial Principals . . . . .	164
8.17.4	Abstract Interpretation with Partial Principals . . . . .	166
8.17.5	Consequences of Enforced Security . . . . .	167
8.18	Other Comparisons with Previous Work . . . . .	168
8.18.1	Security Beyond Dynamic Context . . . . .	168
8.18.2	Nonfatal Exceptions . . . . .	169
8.19	Summary of Vulnerabilities . . . . .	169
8.20	Summary . . . . .	170
<b>9</b>	<b>Case Studies</b>	<b>172</b>
9.1	UID Based Mechanisms . . . . .	172
9.1.1	Introduction to UIDs . . . . .	172
9.1.2	Construction of UID Based Mechanisms . . . . .	173
9.2	Case Study: Unix (1970) . . . . .	174
9.2.1	Identification of the Current Principal . . . . .	174
9.2.2	Access to Protected Objects . . . . .	176
9.2.3	Modification of Rights . . . . .	177
9.2.4	Specifics of Implementation . . . . .	177
9.2.5	Summary of Unix . . . . .	178
9.3	Summary of UID Based Mechanisms . . . . .	178
9.4	Stack Inspection Based Mechanisms . . . . .	178

9.5	Case Study: Java (1991)	179
9.5.1	Origins of Java	179
9.5.2	Evolution of an Architecture for Protection	179
9.5.3	Construction and Representation of Principals and Permissions	180
9.5.4	Identification of the Current Principal	181
9.5.5	Modification of Rights	182
9.5.6	Specifics of Implementation	184
9.5.7	Vulnerabilities	185
9.5.8	Summary of Java	186
9.6	Case Study: .NET and the Common Language Runtime (1999)	186
9.7	Case Study: Multics Ring System (1965)	186
9.7.1	The Aged Aged Man	186
9.7.2	A Very Curious Thing	187
9.7.3	Identification of the Current Principal	187
9.7.4	Vulnerabilities	189
9.7.5	Modification of Rights	189
9.7.6	Summary of Multics	189
9.8	Data Inspection Based Mechanisms	189
9.9	Case Study: Perl (1987)	190
9.9.1	The History of Perl	190
9.9.2	Perl for System Administration	190
9.9.3	Identification of the Current Principal	191
9.9.4	Specifics of Implementation	191
9.9.5	Summary of Perl	192
9.10	More than Tainting	192
9.11	Case Study: EROS and Capabilities	193
9.11.1	History of Capabilities	193
9.11.2	Identification of the Current Principal	193
9.11.3	Specifics of Implementation	194
9.11.4	Summary of Capabilities	194
9.12	On the Granting of Rights	195
9.13	Anarres II	195
9.13.1	History of Anarres II	195
9.13.2	Modification of Rights	196
9.13.3	Specifics of Implementation	197
9.13.4	Summary of Anarres II	197
9.14	Summary of Case Studies	198
<b>10</b>	<b>A New Model for Computation in Protection Systems</b>	<b>199</b>
10.1	The Lattice of Principals	200
10.1.1	Background to the Lattice of Principals	200