

# Contents

<b>List of Tables</b> .....	xxvii
<b>List of Figures</b> .....	xxix
<b>List of Algorithms</b> .....	xxx
<b>List of Abbreviations</b> .....	xxxi
<b>1 Introduction: How to Use this Book</b> .....	1
<b>Part I Cryptanalysis</b>	
<b>2 The Block Cipher Keeloq and Algebraic Attacks</b> .....	9
2.1 What is Algebraic Cryptanalysis? .....	10
2.1.1 The CSP Model .....	10
2.2 The Keeloq Specification .....	10
2.3 Modeling the Non-linear Function .....	11
2.3.1 I/O Relations and the NLF .....	12
2.4 Describing the Shift-Registers .....	12
2.4.1 Disposing of the Secret Key Shift-Register .....	13
2.4.2 Disposing of the Plaintext Shift-Register .....	13
2.5 The Polynomial System of Equations .....	13
2.6 Variable and Equation Count .....	14
2.7 Dropping the Degree to Quadratic .....	14
2.8 Fixing or Guessing Bits in Advance .....	15
2.9 The Failure of a Frontal Assault .....	16
<b>3 The Fixed-Point Attack</b> .....	17
3.1 Overview .....	17
3.1.1 Notational Conventions .....	17
3.1.2 The Two-Function Representation .....	17
3.1.3 Acquiring an $f_k^{(8)}$ -oracle .....	18

3.2	The Consequences of Fixed Points . . . . .	18
3.3	How to Find Fixed Points . . . . .	19
3.4	How far must we search? . . . . .	20
3.4.1	With Analytic Combinatorics . . . . .	21
3.4.2	Without Analytic Combinatorics . . . . .	23
3.5	Comparison to Brute Force . . . . .	23
3.6	Summary . . . . .	24
3.7	Other Notes . . . . .	25
3.7.1	A Note about Keeloq's Utilization . . . . .	25
3.7.2	RPA vs KPA vs CPA . . . . .	26
3.8	Wagner's Attack . . . . .	26
3.8.1	Later Work on Keeloq . . . . .	27
<b>4</b>	<b>Iterated Permutations . . . . .</b>	<b>29</b>
4.1	Applications to Cryptography . . . . .	29
4.2	Background . . . . .	30
4.2.1	Combinatorial Classes . . . . .	30
4.2.2	Ordinary and Exponential Generating Functions . . . . .	30
4.2.3	Operations on OGFs . . . . .	31
4.2.4	Examples . . . . .	34
4.2.5	Operations on EGFs . . . . .	36
4.2.6	Notation and Definitions . . . . .	39
4.3	Strong and Weak Cycle Structure Theorems . . . . .	40
4.3.1	Expected Values . . . . .	41
4.4	Corollaries . . . . .	43
4.4.1	On Cycles in Iterated Permutations . . . . .	45
4.4.2	Limited Cycle Counts . . . . .	46
4.4.3	Monomial Counting . . . . .	47
4.5	Of Pure Mathematical Interest . . . . .	47
4.5.1	The Sigma Divisor Function . . . . .	48
4.5.2	The Zeta Function and Apéry's Constant . . . . .	48
4.5.3	Greatest Common Divisors and Cycle Length . . . . .	49
4.6	Highly Iterated Ciphers . . . . .	49
4.6.1	Distinguishing Iterated Ciphers . . . . .	50
4.6.2	A Key Recovery Attack . . . . .	52
<b>5</b>	<b>Stream Ciphers . . . . .</b>	<b>55</b>
5.1	The Stream Ciphers Bivium and Trivium . . . . .	55
5.1.1	Background . . . . .	55
5.1.2	Bivium as Equations . . . . .	61
5.1.3	An Excellent Trick . . . . .	64
5.1.4	Bivium-A . . . . .	65
5.1.5	A Notational Issue . . . . .	65
5.1.6	For Further Reading . . . . .	65
5.2	The Stream Cipher QUAD . . . . .	66

5.2.1	How QUAD Works	66
5.2.2	Proof of Security	67
5.2.3	The Yang-Chen-Bernstein-Chen Attack against QUAD	72
5.2.4	Extending to $\mathbb{GF}(16)$	75
5.2.5	For Further Reading	77
5.3	Conclusions for QUAD	78

## Part II Linear Systems Mod 2

<b>6</b>	<b>Some Basic Facts about Linear Algebra over <math>\mathbb{GF}(2)</math></b>	81
6.1	Sources	81
6.2	Boolean Matrices vs $\mathbb{GF}(2)$ Matrices	81
6.2.1	Implementing with the Integers	82
6.3	Why is $\mathbb{GF}(2)$ Different?	82
6.3.1	There are Self-Orthogonal Vectors	82
6.3.2	Something that Fails	83
6.3.3	The Probability a Random Square Matrix Singular or Invertible	84
6.4	Null Space from the RREF	85
6.5	The Number of Solutions to a Linear System	86
<b>7</b>	<b>The Complexity of <math>\mathbb{GF}(2)</math>-Matrix Operations</b>	89
7.1	The Cost Model	89
7.1.1	A Word on Architecture and Cross-Over	90
7.1.2	Is the Model Trivial?	91
7.1.3	Counting Field Operations	91
7.1.4	Success and Failure	92
7.2	Notational Conventions	92
7.3	To Invert or to Solve?	93
7.4	Data Structure Choices	94
7.4.1	Dense Form: An Array with Swaps	94
7.4.2	Permutation Matrices	94
7.5	Analysis of Classical Techniques with our Model	96
7.5.1	Naïve Matrix Multiplication	96
7.5.2	Matrix Addition	96
7.5.3	Dense Gaussian Elimination	96
7.5.4	Back-Solving a Triangulated Linear System	98
7.6	Strassen's Algorithms	99
7.6.1	Strassen's Algorithm for Matrix Multiplication	100
7.6.2	Misunderstanding Strassen's Matrix Inversion Formula	101
7.7	The Unsuitability of Strassen's Algorithm for Inversion	101
7.7.1	Strassen's Approach to Matrix Inversion	102
7.7.2	Bunch and Hopcroft's Solution	103
7.7.3	Ibara, Moran, and Hui's Solution	103

<b>8</b>	<b>On the Exponent of Certain Matrix Operations</b>	107
8.1	Very Low Exponents	107
8.2	The Equicomplexity Theorems	108
8.2.1	Starting Point	109
8.2.2	Proofs	109
8.3	Determinants and Matrix Inverses	118
8.3.1	Background	118
8.3.2	The Baur-Strassen-Morgenstern Theorem	120
8.3.3	Consequences for the Determinant and Inverse	132
<b>9</b>	<b>The Method of Four Russians</b>	133
9.0.4	The Fair Coin Assumption	134
9.1	Origins and Previous Work	134
9.1.1	Strassen's Algorithm	135
9.2	Rapid Subspace Enumeration	135
9.3	The Four Russians Matrix Multiplication Algorithm	137
9.3.1	Role of the Gray Code	137
9.3.2	Transposing the Matrix Product	138
9.3.3	Improvements	138
9.3.4	A Quick Computation	139
9.3.5	M4RM Experiments Performed by SAGE Staff	139
9.3.6	Multiple Gray-Code Tables and Cache Management	141
9.4	The Four Russians Matrix Inversion Algorithm	141
9.4.1	Stage 1:	141
9.4.2	Stage 2:	142
9.4.3	Stage 3:	142
9.4.4	A Curious Note on Stage 1 of M4RI	143
9.4.5	Triangulation or Inversion?	145
9.5	Exact Analysis of Complexity	145
9.5.1	An Alternative Computation	146
9.5.2	Full Elimination, not Triangular	147
9.5.3	The Rank of $3k$ Rows, or Why $k + \epsilon$ is not Enough	148
9.5.4	Using Bulk Logical Operations	149
9.6	Experimental and Numerical Results	149
9.7	M4RI Experiments Performed by SAGE Staff	151
9.7.1	Determination of $k$	151
9.7.2	The Transpose Experiment	151
9.8	Pairing With Strassen's Algorithm for Matrix Multiplication	151
9.8.1	Pairing M4RI with Strassen	152
9.9	Higher Values of $q$	152
9.9.1	Building the Gray Code over $\mathbb{GF}(q)$	152
9.9.2	Other Modifications	153
9.9.3	Running Time	153
9.9.4	Implementation	154

<b>10 The Quadratic Sieve</b>	159
10.1 Motivation	159
10.1.1 A View of RSA from 60,000 feet	160
10.1.2 Two Facts from Number Theory	161
10.1.3 Reconstructing the Private Key from the Public Key	161
10.2 Trial Division	163
10.2.1 Other Ideas	165
10.2.2 Sieve of Eratosthenes	167
10.3 Theoretical Foundations	169
10.4 The Naïve Sieve	170
10.4.1 An Extended Example	171
10.5 The Gödel Vectors	171
10.5.1 Benefits of the Notation	172
10.5.2 Unlimited-Dimension Vectors	173
10.5.3 The Master Stratagem	173
10.5.4 Historical Interlude	173
10.5.5 Review of Null Spaces	174
10.5.6 Constructing a Vector in the Even-Space	175
10.6 The Linear Sieve Algorithm	176
10.6.1 Matrix Dimensions in the Linear & Quadratic Sieve	176
10.6.2 The Running Time	178
10.7 The Example, Revisited	178
10.8 Rapidly Generating Smooth Squares	180
10.8.1 New Strategy	181
10.9 Further Reading	183
10.10 Historical Notes	183

### Part III Polynomial Systems and Satisfiability

<b>11 Strategies for Polynomial Systems</b>	187
11.1 Why Solve Polynomial Systems of Equations over Finite Fields?	187
11.2 Universal Maps	189
11.3 Polynomials over $\mathbb{GF}(2)$	191
11.3.1 Exponents: $x^2 = x$	191
11.3.2 Equivalent versus Identical Polynomials	191
11.3.3 Coefficients	192
11.3.4 Linear Combinations	192
11.4 Degree Reduction Techniques	192
11.4.1 An Easy but Hard-to-State Condition	193
11.4.2 An Algorithm that meets this Condition	194
11.4.3 Interpretation	195
11.4.4 Summary	196
11.4.5 Detour: Asymptotics of the “Choose” Function	196
11.4.6 Complexity Calculation	197
11.4.7 Efficiency Note	198

11.4.8	The Greedy Degree-Dropper Algorithm	198
11.4.9	Counter-Example for Linear Systems	199
11.5	NP-Completeness of MP	199
11.6	Measures of Difficulty in MQ	203
11.6.1	The Role of Over-Definition	203
11.6.2	Ultra-Sparse Quadratic Systems	203
11.6.3	Other Views of Sparsity	205
11.6.4	Structure	205
11.7	The Role of Guessing a Few Variables	206
11.7.1	Measuring Infeasible Running Times	206
11.7.2	Fix-XL	207
<b>12</b>	<b>Algorithms for Solving Polynomial Systems</b>	<b>209</b>
12.1	A Philosophical Point on Complexity Theory	209
12.2	Gröbner Bases Algorithms	210
12.2.1	Double-Exponential Running Time	210
12.2.2	Remarks about Gröbner Bases	210
12.3	Linearization	211
12.4	The XL Algorithm	213
12.4.1	Complexity Analysis	215
12.4.2	Sufficiently Many Equations	216
12.4.3	Jumping Two Degrees	216
12.4.4	Fix-XL	217
12.5	ElimLin	219
12.5.1	Why is this useful?	220
12.5.2	How to use ElimLin	221
12.5.3	On the Sub-Space of Linear Equations in the Span of a Quadratic System of Equations	223
12.5.4	The Weight of the Basis	224
12.5.5	One Last Trick for $\mathbb{GF}(2)$ -only	225
12.5.6	Notes on the Sufficient Rank Condition	226
12.6	Comparisons between XL and F4	227
12.7	SAT-Solvers	228
12.8	System Fragmentation	228
12.8.1	Separability	229
12.8.2	Gaussian Elimination is Not Enough	230
12.8.3	Depth First Search	230
12.8.4	Nearly Separable Systems	231
12.8.5	Removing Multiple vertices	232
12.8.6	Relation to Menger's Theorem	232
12.8.7	Balance in Vertex Cuts	233
12.8.8	Applicability	233
12.9	Resultants	234
12.9.1	The Univariate Case	234
12.9.2	The Bivariate Case	235

12.9.3	Multivariate Case	236
12.9.4	Further Reading	238
12.10	The Raddum-Semaev Method	238
12.10.1	Building the Graph	238
12.10.2	Agreeing	239
12.10.3	Propagation	240
12.10.4	Termination	240
12.10.5	Gluing	240
12.10.6	Splitting	242
12.10.7	Summary	242
12.11	The Zhuang-Zi Algorithm	243
12.12	Homotopy Approach	243
<b>13</b>	<b>Converting MQ to CNF-SAT</b>	<b>245</b>
13.1	Summary	245
13.2	Introduction	246
13.2.1	Application to Cryptanalysis	247
13.3	Notation and Definitions	247
13.4	Converting MQ to SAT	248
13.4.1	The Conversion	248
13.4.2	Measures of Difficulty	250
13.4.3	Preprocessing	252
13.4.4	Fixing Variables in Advance	253
13.4.5	SAT-Solver Used	254
13.5	Experimental Results	255
13.5.1	The Source of the Equations	255
13.5.2	Note About the Variance	255
13.5.3	The Log-Normal Distribution of Running Times	256
13.5.4	The Optimal Cutting Number	257
13.6	Cubic Systems	258
13.6.1	Do All Possible Monomials Appear?	258
13.6.2	Measures of Efficiency	260
13.7	Further Reading	260
13.7.1	Previous Work	260
13.7.2	Further Work	261
13.8	Conclusions	262
<b>14</b>	<b>How do SAT-Solvers Operate?</b>	<b>263</b>
14.1	The Problem Itself	263
14.1.1	Conjunctive Normal Form	264
14.2	Solvers like Walk-SAT	264
14.2.1	The Search Space	265
14.2.2	Papadimitriou's Algorithm	265
14.2.3	Greedy SAT or G-SAT	266
14.2.4	Walk-SAT	267

14.2.5	Walk-SAT versus Papadimitriou	268
14.2.6	Where Heuristic Methods Fail	268
14.2.7	Closing Thoughts on Heuristic Methods	269
14.3	Back-Tracking	269
14.4	Chaff and its Descendants	272
14.4.1	Variable Management	272
14.4.2	Unit Propagation	273
14.4.3	The Method of Watched Literals	273
14.4.4	Absent Literals	274
14.4.5	Summary	274
14.5	Enhancements to Chaff	275
14.5.1	Learned Clauses	275
14.5.2	The Alarm Clock	275
14.5.3	The Third Finger	276
14.6	Economic Motivations	276
14.7	Further Reading	277
<b>15</b>	<b>Applying SAT-Solvers to Extension Fields of Low Degree</b>	<b>279</b>
15.1	Introduction	279
15.2	Solving $\mathbb{GF}(2)$ Systems via SAT-Solvers	280
15.2.1	Sparsity	280
15.3	Overview	281
15.4	Polynomial Systems over Extension Fields of $\mathbb{GF}(2)$	281
15.4.1	Extensions of the Coefficient Field	282
15.4.2	Difficulty in Bits	282
15.5	Finding Efficient Arithmetic Representations via Matrices	282
15.6	Using the Algebraic Normal Forms	286
15.6.1	Remarks on the Special Forms	287
15.6.2	Remarks on Degree	287
15.6.3	Remarks on Coefficients	288
15.6.4	Solving with Gröbner Bases	288
15.7	Experimental Results	289
15.7.1	Computers Used	291
15.7.2	Polynomial Systems Used	291
15.8	Inverses and Determinants	292
15.8.1	Determinants	292
15.8.2	Inverses	292
15.8.3	Rijndael and the Para-Inverse Operation	293
15.9	Conclusions	294
15.10	Review of Extension Fields	295
15.10.1	Constructing the Field	295
15.10.2	Regular Representation	297
15.11	Reversing the Isomorphism: The Existence of Dead Give-Aways	298



<b>A</b>	<b>On the Philosophy of Block Ciphers With Small Blocks</b>	301
A.1	Definitions	301
A.2	Brute-Force Generic Attacks on Ciphers with Small Blocks	302
A.3	Key Recovery vs. Applications of Ciphers with Small Blocks	303
A.4	The Keeloq Code-book—Practical Considerations	306
A.5	Conclusions	307
<b>B</b>	<b>Formulas for the Field Multiplication law for Low-Degree Extensions of <math>\mathbb{GF}(2)</math></b>	309
B.1	For $\mathbb{GF}(4)$	309
B.2	For $\mathbb{GF}(8)$	309
B.3	For $\mathbb{GF}(16)$	310
B.4	For $\mathbb{GF}(32)$	311
B.5	For $\mathbb{GF}(64)$	312
<b>C</b>	<b>Polynomials and Graph Coloring, with Other Applications</b>	315
C.1	A Very Useful Lemma	315
C.2	Graph Coloring	316
C.2.1	The $c \neq p^n$ Case	316
C.2.2	Application to $\mathbb{GF}(2)$ Polynomials	316
C.3	Related Applications	317
C.3.1	Radio Channel Assignments	317
C.3.2	Register Allocation	318
C.4	Interval Graphs	318
C.4.1	Scheduling an Interval Graph Scheduling Problem	319
C.4.2	Comparison to Other Problems	320
C.4.3	Moral of the Story	321
<b>D</b>	<b>Options for Very Sparse Matrices</b>	323
D.1	Preliminary Points	323
D.1.1	Accidental Cancellations	323
D.1.2	Solving Equations by Finding a Null Space	324
D.1.3	Data Structures and Storage	324
D.2	Naïve Sparse Gaussian Elimination	325
D.2.1	Sparse Matrices can have Dense Inverses	326
D.3	Markowitz's Algorithm	326
D.4	The Block Wiedemann Algorithm	326
D.5	The Block Lanczos Algorithm	327
D.6	The Pomerance-Smith Algorithm	327
D.6.1	Overview	328
D.6.2	Inactive and Active Columns	329
D.6.3	The Operations	329
D.6.4	The Actual Algorithm	331
D.6.5	Fill-in and Memory Management	332
D.6.6	Technicalities	333

D.6.7	Cremona's Implementation .....	334
D.6.8	Further Reading .....	335
<b>E</b>	<b>Inspirational Thoughts, Poetry and Philosophy .....</b>	<b>337</b>
	<b>References .....</b>	<b>339</b>
	<b>Index .....</b>	<b>351</b>



<http://www.springer.com/978-0-387-88756-2>

Algebraic Cryptanalysis

Bard, G.

2009, XXXIII, 356 p., Hardcover

ISBN: 978-0-387-88756-2