# LONDON MATHEMATICAL SOCIETY LECTURE NOTE SERIES

Managing Editor: Professor J.W.S. Cassels, Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, 16 Mill Lane, Cambridge CB2 1SB, England

The books in the series listed below are available from booksellers, or, in case of difficulty, from Cambridge University Press.

17 Differential germs and catastrophes, Th. BROCKER & L. LANDER
34 Representation theory of Lie groups, M.F. ATIYAH *et al*
36 Homological group theory, C.T.C. WALL (ed)
39 Affine sets and affine groups, D.G. NORTHCOTT
40 Introduction to $H_p$ spaces, P.J. KOOSIS
43 Graphs, codes and designs, P.J. CAMERON & J.H. VAN LINT
45 Recursion theory: its generalisations and applications, F.R. DRAKE & S.S. WAINER (eds)
46 p-adic analysis: a short course on recent work, N. KOBLITZ
49 Finite geometries and designs, P. CAMERON, J.W.P. HIRSCHFELD & D.R. HUGHES (eds)
50 Commutator calculus and groups of homotopy classes, H.J. BAUES
51 Synthetic differential geometry, A. KOCK
54 Markov processes and related problems of analysis, E.B. DYNKIN
57 Techniques of geometric topology, R.A. FENN
58 Singularities of smooth functions and maps, J.A. MARTINET
59 Applicable differential geometry, M. CRAMPIN & F.A.E. PIRANI
60 Integrable systems, S.P. NOVIKOV *et al*
62 Economics for mathematicians, J.W.S. CASSELS
65 Several complex variables and complex manifolds I, M.J. FIELD
66 Several complex variables and complex manifolds II, M.J. FIELD
68 Complex algebraic surfaces, A. BEAUVILLE
69 Representation theory, I.M. GELFAND *et al*
74 Symmetric designs: an algebraic approach, E.S. LANDER
76 Spectral theory of linear differential operators and comparison algebras, H.O. CORDES
77 Isolated singular points on complete intersections, E.J.N. LOOIJENGA
78 A primer on Riemann surfaces, A.F. BEARDON
79 Probability, statistics and analysis, J.F.C. KINGMAN & G.E.H. REUTER (eds)
80 Introduction to the representation theory of compact and locally compact groups, A. ROBERT
81 Skew fields, P.K. DRAXL
82 Surveys in combinatorics, E.K. LLOYD (ed)
83 Homogeneous structures on Riemannian manifolds, F. TRICERRI & L. VANHECKE
85 Solitons, P.G. DRAZIN
86 Topological topics, I.M. JAMES (ed)
87 Surveys in set theory, A.R.D. MATHIAS (ed)
88 FPF ring theory, C. FAITH & S. PAGE
89 An F-space sampler, N.J. KALTON, N.T. PECK & J.W. ROBERTS
90 Polytopes and symmetry, S.A. ROBERTSON
91 Classgroups of group rings, M.J. TAYLOR
92 Representation of rings over skew fields, A.H. SCHOFIELD
93 Aspects of topology, I.M. JAMES & E.H. KRONHEIMER (eds)
94 Representations of general linear groups, G.D. JAMES
95 Low-dimensional topology 1982, R.A. FENN (ed)
96 Diophantine equations over function fields, R.C. MASON
97 Varieties of constructive mathematics, D.S. BRIDGES & F. RICHMAN
98 Localization in Noetherian rings, A.V. JATEGAONKAR
99 Methods of differential geometry in algebraic topology, M. KAROUBI & C. LERUSTE
100 Stopping time techniques for analysts and probabilists, L. EGGHE

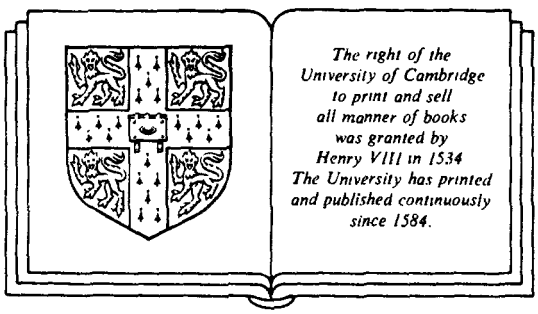
101 Groups and geometry, ROGER C. LYNDON
103 Surveys in combinatorics 1985, I. ANDERSON (ed)
104 Elliptic structures on 3-manifolds, C.B. THOMAS
105 A local spectral theory for closed operators, I. ERDELYI & WANG SHENGWANG
106 Syzygies, E.G. EVANS & P. GRIFFITH
107 Compactification of Siegel moduli schemes, C-L. CHAI
108 Some topics in graph theory, H.P. YAP
109 Diophantine Analysis, J. LOXTON & A. VAN DER POORTEN (eds)
110 An introduction to surreal numbers, H. GONSHOR
111 Analytical and geometric aspects of hyperbolic space, D.B.A.EPSTEIN (ed)
112 Low-dimensional topology and Kleinian groups, D.B.A. EPSTEIN (ed)
113 Lectures on the asymptotic theory of ideals, D. REES
114 Lectures on Bochner-Riesz means, K.M. DAVIS & Y-C. CHANG
115 An introduction to independence for analysts, H.G. DALES & W.H. WOODIN
116 Representations of algebras, P.J. WEBB (ed)
117 Homotopy theory, E. REES & J.D.S. JONES (eds)
118 Skew linear groups, M. SHIRVANI & B. WEHRFRITZ
119 Triangulated categories in the representation theory of finite-dimensional algebras, D. HAPPEL
121 Proceedings of *Groups - St Andrews 1985*, E. ROBERTSON & C. CAMPBELL (eds)
122 Non-classical continuum mechanics, R.J. KNOPS & A.A. LACEY (eds)
123 Surveys in combinatorics 1987, C. WHITEHEAD (ed)
124 Lie groupoids and Lie algebroids in differential geometry, K. MACKENZIE
125 Commutator theory for congruence modular varieties, R. FREESE & R. MCKENZIE
126 Van der Corput's method for exponential sums, S.W. GRAHAM & G. KOLESNIK
127 New directions in dynamical systems, T.J. BEDFORD & J.W. SWIFT (eds)
128 Descriptive set theory and the structure of sets of uniqueness, A.S. KECHRIS & A. LOUVEAU
129 The subgroup structure of the finite classical groups, P.B. KLEIDMAN & M.W.LIEBECK
130 Model theory and modules, M. PREST
131 Algebraic, extremal & metric combinatorics, M-M. DEZA, P. FRANKL & I.G. ROSENBERG (eds)
132 Whitehead groups of finite groups, ROBERT OLIVER
133 Linear algebraic monoids, MOHAN S. PUTCHA
134 Number theory and dynamical systems, M. DODSON & J. VICKERS (eds)
135 Operator algebras and applications, 1, D. EVANS & M. TAKESAKI (eds)
136 Operator algebras and applications, 2, D. EVANS & M. TAKESAKI (eds)
137 Analysis at Urbana, I, E. BERKSON, T. PECK, & J. UHL (eds)
138 Analysis at Urbana, II, E. BERKSON, T. PECK, & J. UHL (eds)
139 Advances in homotopy theory, S. SALAMON, B. STEER & W. SUTHERLAND (eds)
140 Geometric aspects of Banach spaces, E.M. PEINADOR and A. RODES (eds)
141 Surveys in combinatorics 1989, J. SIEMONS (ed)
142 The geometry of jet bundles, D.J. SAUNDERS
143 The ergodic theory of discrete groups, PETER J. NICHOLLS
144 Uniform spaces, I.M. JAMES
145 Homological questions in local algebra, JAN R. STROOKER
146 Maximal Cohen-Macaulay modules over Henselian rings, Y. YOSHINO
147 Continuous and discrete modules, S.H. MOHAMED & B.J. MÜLLER
154 Number theory and cryptography, J. LOXTON (ed)

London Mathematical Society Lecture Note Series. 154

# Number Theory and Cryptography

Edited by

J.H. Loxton
Professor of Mathematics,
Macquarie University




CAMBRIDGE UNIVERSITY PRESS
Cambridge
New York Port Chester Melbourne Sydney

Published by the Press Syndicate of the University of Cambridge
The Pitt Building, Trumpington Street, Cambridge CB2 1RP
40 West 20th Street, New York, NY 10011, USA
10, Stamford Road, Oakleigh, Melbourne 3166, Australia



First published 1990

Printed in Great Britain at the University Press, Cambridge

*Library of Congress cataloguing in publication data available*

*British Library cataloguing in publication data available*

ISBN 0 521 39877 0

# CONTENTS

## PART III. DIOPHANTINE ANALYSIS

# CONTRIBUTORS

*Richard P. Brent*, Computer Sciences Laboratory, Research School of Physical Sciences, Australian National University, GPO Box 4, Canberra, ACT 2601, Australia.

*B. Brindza*, School of Mathematics, Physics, Computing and Electronics, Macquarie University, NSW 2109, Australia.

*Johannes Buchmann*, FB-10 Informatik, Universität des Saarlandes, D-6600 Saarbrücken, West Germany.

*Bill Caelli*, Information Security Research Centre, Queensland University of Technology, GPO Box 2434, Brisbane, Queensland 4001, Australia.

*Ed Dawson*, School of Mathematics, Queensland University of Technology, GPO Box 2434, Brisbane, Queensland 4001, Australia.

*Frank G. Garvan*, School of Mathematics, Physics, Computing and Electronics, Macquarie University, NSW 2109, Australia.

*H. Gustafson*, School of Mathematics, Queensland University of Technology, GPO Box 2434, Brisbane, Queensland 4001, Australia.

*Teturo Kamae*, Department of Mathematics, Osaka City University, Sugimoto-cho, Osaka, 558 Japan.

*H. W. Lenstra, Jr.*, Department of Mathematics, University of California, Berkeley, California 94720, USA.

*Rudolf Lidl*, Department of Mathematics, University of Tasmania, GPO Box 252C, Hobart, Tasmania 7001, Australia.

*J. H. Loxton*, School of Mathematics, Physics, Computing and Electronics, Macquarie University, NSW 2109, Australia.

*R. A. Mollin*, Department of Mathematics and Statistics, University of Calgary, Calgary, Alberta, Canada T2N 1N4.

*G. Myerson*, School of Mathematics, Physics, Computing and Electronics, Macquarie University, NSW 2109, Australia.

*Y.-N. Nakai*, Department of Mathematics, Faculty of Education, Yamanashi University, Kofu, 400 Japan.

*Bill Newman*, Department of Mathematics, James Cook University of North Queensland, Townsville, Queensland 4811, Australia.

*Lindsay Peters*, Technical Computing, Plessey Australia, Railway Road, Meadowbank NSW 2114, Australia.

*A. Pollington*, Department of Mathematics, Brigham Young University, Provo, Utah 84602, USA.

*Mark Rudolph*, Department of Communication and Electronic Engineering, Royal Melbourne Institute of Technology, PO Box 2476C, Melbourne, Victoria 3000, Australia.

*R. S. Safavi-Naini*, Department of Computer Science, University College, University of New South Wales, Australian Defence Forces Academy, Canberra, ACT 2600, Australia.

*J. R. Seberry*, Department of Computer Science, University College, University of New South Wales, Australian Defence Forces Academy, Canberra, ACT 2600, Australia.

*I. Shiokawa*, Department of Mathematics, Keio University, Hiyoshi, Yokohama, 223 Japan.

*Alice Silverberg*, Thomas J. Watson Research Centre, IBM, PO Box 218, Yorktown Heights, New York 10598, USA.

*John Snare*, Secure Communication Systems Section, Telecom Australia Research Laboratories, 770 Blackburn Road, Clayton, Victoria 3168, Australia.

*A. J. Stephens*, Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2.

*A. J. van der Poorten*, School of Mathematics, Physics, Computing and Electronics, Macquarie University, NSW 2109, Australia.

*Michael Warner*, Secure Communication Systems Section, Telecom Australia Research Laboratory, 770 Blackburn Road, Clayton, Victoria 3168, Australia.

*H. C. Williams*, Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2.

*R. T. Worley*, Department of Mathematics, Monash University, Clayton, Victoria 3168, Australia.

# INTRODUCTION

Number theory and cryptography would have seemed unlikely partners only a few years ago. Indeed, in *A Mathematician's Apology*, Hardy professes that no use at all can be made of real mathematics:*

> If the theory of numbers could be employed for any practical and obviously honourable purpose, if it could be turned directly to the furtherance of human happiness or the relief of human suffering, as physiology and even chemistry can, then surely neither Gauss nor any other mathematician would have been so foolish as to decry or regret such applications. But science works for evil as well as for good (and particularly, of course, in time of war); and both Gauss and lesser mathematicians may be justified in rejoicing that there is one science at any rate, and that their own, whose very remoteness from ordinary human activities should keep it gentle and clean.

Despite this, the hard problems of arithmetic involving factorisation and diophantine equations have revolutionised the theory and practice of cryptology in recent times. Problems in arithmetic are simple to pose and hard to solve. These qualities have drawn mathematicians to number theory since the time of the Greeks and are at the root of the applications in cryptology. Even simple arithmetic, such as

$$\sqrt{2} = 1.41421\ 35623\ 73095\ 04880\ 16887\ 24209\ 69807\ 85697\ldots,$$

introduces an unpredictability which is mysterious, tantalising and a useful source of pseudo-random numbers for the cryptographer. Again, this happens to go against the rhetoric of a great man, in this case von Neumann, who remarked that 'anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin'. The history of mathematics is full of such sobering insights. The recent history of coding theory and cryptography may seem to be a paradox, but it illustrates again the 'miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics'.**

This book contains a number of papers presented at the 33rd Annual Meeting of the Australian Mathematical Society and at a Workshop on Number Theory and Cryptography in Telecommunications held at Macquarie University in Sydney from 29 June to 7 July 1989. The papers are organised into three sections dealing, respectively, with the mathematical side of cryptography, studies of shift register sequences and other problems in cryptanalysis, and related topics in number theory.

---

* G. H. Hardy, *A Mathematician's Apology* (Cambridge University Press, 1969).

** E. G. Wigner, 'The unreasonable effectiveness of mathematics in the natural sciences', *Communications on Pure and Applied Mathematics* XIII (1960), 1–14.

Perhaps, the most important idea in public-key cryptography is the *RSA* code of Rivest, Shamir and Adleman. The secret key consists of two large primes $p$ and $q$. Encryption of a message $z$ is done by the $z \to z^e \pmod{n}$, using the public keys $n = pq$ and $e$. Decryption is done by $z \to z^d \pmod{n}$, where $de \equiv 1 \pmod{(p-1)(q-1)}$ and, presumably, it is impossible to find $d$ without factorising $n = pq$. Thus the code relies on the fact that it is easy to find very large primes of, say, 100 digits, but it is computationally infeasible to factorise a number of 200 digits. This code has revived interest in factorisation and records continue to tumble under the impact of new techniques such as the elliptic curve algorithm and new technologies such as parallel processing. The current state of the art is described in the paper by Richard Brent. Further aspects of factorisation are detailed by Stephens and Hugh Williams in their paper on number sieves. These are specially tuned 'computers' for number theory which have been used to find primes and factors and to gather data on many other arithmetical problems. The parameters $p$, $q$ and $e$ in the *RSA* code must be carefully chosen to avoid an attack based on diophantine approximation. Rudolf Lidl in his paper describes how continued fractions can be used to break an *RSA* code in which the encrypting exponent, $e$, is too short. He also shows how continued fractions in a different context provide new measures for the complexity of sequences generated by stream ciphers. The mathematical background to continued fractions and stream ciphers and their significance in number theory is given in the paper by Alf van der Poorten. Another class of difficult problems useful in cryptography centres on discrete logarithms: given a group $G$ and elements $g$ and $u$ in $G$, find the exponent $x$ in $u = g^x$, if it exists. Lidl describes the key distribution scheme of Diffie and Hellman based on discrete logarithms and some recent refinements of their idea. Many of the groups in coding theory are based on finite fields and efficient algorithms for computation in finite fields are very important. Hendrik Lenstra in his paper describes where these stand and how the gap between existence theorems and practical algorithms is sometimes uncomfortably wide. Recently, cryptographers have turned to other groups and even further afield. Elliptic curves and number fields present difficult problems which have been proposed as the basis for cryptosystems. Johannes Buchmann and Hugh Williams in their paper discuss arithmetic in quadratic number fields, $\mathbf{Q}(\sqrt{D})$, and describe how it may be possible to set up a code using the decision problem for principal ideals in these fields. This paper concludes with a number of open problems.

The second part begins with a paper by John Snare giving an overview of the challenges in telecommunications which need cryptographic solutions. The diversity of applications for public key cryptography adds point to the search for more algorithms to supplement the *RSA* code. The papers by Ed Dawson, Helen Gustafson, Bill Caelli, Safavi-Naini and Jenny Seberry describe the theory and use of stream ciphers in cryptography and in the generation of pseudo-random numbers. In particular, various methods of increasing the complexity of the output from stream ciphers are discussed. The earlier papers by Rudolf Lidl and Alf van der Poorten comment on some of these matters from a different perspective. There follow a number of papers dealing with particular cryptographic problems. Michael Warner examines the security of the *MACNET* shared fibre access network and describes an encryption scheme for it based on stream ciphers. Bill Newman explains the theory of authentication and describes a practical authentication scheme based on the *RSA* system. Rod Worley

describes a variant of the knapsack cryptosystem and presents an attack on the scheme using techniques from diophantine approximation such as continued fractions and the short lattice vector algorithm. Lindsay Peters describes the tactical frequency management problem for assigning frequencies to groups of radios to minimise interference and explains the progress towards solving the problem. Finally, the paper by Mark Rudolph is a brief report on an analytic approach to the Reed-Solomon code.

In the third section, the main focus switches from cryptography to number theory. The paper by Richard Mollin and Hugh Williams takes up some of the problems raised in the earlier papers by Hugh Williams and reviews the recent advances concerning the class numbers of real quadratic fields. One result from their investigations has been a sequence of new records for prime-producing polynomials. The papers by Teturo Kamae, Nakai and Iekato Shiokawa, and Gerry Myerson and Andrew Pollington deal with various aspects of uniform distribution and randomness. Underlying these papers is the old and difficult problem of proving the randomness in some sense of decimal expansions such as the one for $\sqrt{2}$ given above. Bela Brindza gives a new result on the number of solutions of the Thue equation

$$X^n + c_1 X^{n-1} Y + \ldots + c_{n-1} X Y^{n-1} + c_n Y^n = 1,$$

a result based on the theory of linear forms in the logarithms of algebraic numbers. Frank Garvan describes a new 'crank' for 'coloured' partitions and a number of new congruences in the style of Ramanujan. Finally, Alice Silverberg discusses techniques for calculating the Mordell-Weil groups of abelian varieties. Although apparently disparate, these papers are related to each other and to the previous ones because abelian varieties are the higher dimensional analogues of elliptic curves and the analytic treatment of partitions, elliptic curves and abelian varieties rests heavily on the theory of modular functions. It seems quite possible that these more abstract diophantine problems will also contain useful ideas for cryptography, just as elliptic curves have already done.

I wish to record my grateful thanks to Macquarie University, The Australian Mathematical Society and the Australian Telecommunications and Electronics Research Board for their support of the Conference and the Workshop which was the basis for this book. I also thank the contributors for their lectures and the papers reproduced here, and I thank the organising committee of the conference for making the whole enterprise possible.

*John Loxton*
*November, 1989.*

# 1

# SOME MATHEMATICAL ASPECTS OF RECENT ADVANCES IN CRYPTOLOGY

**Rudolf Lidl**

We report on some recent achievements in parts of cryptology: Wiener's attack on short secret *RSA* exponents, McCurley's composite Diffie and Hellman key distribution scheme, and Niederreiter's continued fraction tests for pseudorandom sequences. The topics are selected because of their emphasis on mathematical concepts such as continued fractions and discrete logarithms.

## 1. Introduction.

Cryptology is a very active and expanding field that brings forward new developments in cryptography and cryptanalysis every year. A comprehensive survey and up-to-date report (as of 1988) on the advances in designing ciphers and cryptographic schemes and on recent successes in breaking cryptosystems has been published in [18] as a collection of important papers on cryptology. This special section of the Proceedings of the IEEE, from May 1988, may serve the interested reader as a first introduction to a variety of topics in contemporary cryptology, ranging from conventional cyptosystems and the Data Encryption Standard, over public-key cryptography and recent advances in cryptanalysis, to a survey of information authentification. Brassard [1] gives a brief survey of cryptography, which is kept at the nontechnical level. Mathematical concepts in cryptography are emphasised in the books by van Tilborg [19], Patterson [13] and Lidl and Niederreiter [6]. For those interested in details of the design and analysis of ciphers, we refer to the proceedings volumes of the annual meetings EUROCRYPT and CRYPTO, see for example [3] and [14].

The emphasis in this paper will be on some mathematical aspects of cryptosystems that have been considered very recently. In Section 2, we describe the recent work of Wiener [20] on the cryptanalysis of an RSA cipher with a short private exponent. This analysis makes use of the continued fraction algorithm for representing rational numbers as finite simple continued fractions. Section 3 is devoted to a brief description of work by McCurley [7] that modifies the Diffie and Hellman key distribution scheme so that it will still be secure if the cryptanalyst knows a very fast algorithm for either factoring or computing discrete logarithms, but not both. The final Section 4 discusses recent work by Niederreiter [10], [11] on the linear complexity profile of keystream sequences. A crucial fact here is the connection between the linear complexity profile of a sequence of elements of the finite field $\mathbf{F}_q$ and the continued fraction expansion of the generating function of the sequence.

## 2. Wiener's cryptanalysis of short private RSA exponents.

The *RSA* cipher is arguably the most important and widely investigated public-key cryptosystem. Since its publication in 1978, efforts to break the cipher have resulted in increased activity on developing faster and better factorisation algorithms for integers. Some of the recent advances in factoring are presented by R. Brent in these Proceedings. In this context, a major breakthrough has been achieved by the Department of Computer Science of the University of Chicago in 1989: the factorisation of 100 digit numbers. The use of the multiple polynomial quadratic sieve method together with the utilisation of cheap distributed computing power through the use of electronic mail as the communication mechanism made it possible to factor integers such as $2^{353}+1$, a 106 digit integer, over a period of four months. However, the *RSA* cipher remains secure from factorisation attacks if the modulus $n$ is of size approximately $10^{200}$, as suggested originally.

At the recent EUROCRYPT '89 meeting, M. J. Wiener [20] presented a cryptanalytic attack on the use of short secret exponents in the *RSA* cipher which can be completed in time polynomial in the length of the modulus. This attack is successful if the secret exponent $d$ has up to approximately a quarter as many significant bits as the modulus. It will fail if $d$ is approximately the same size as $n$. The use of short exponents may be desirable in order to reduce the execution time of *RSA* where there is a large difference in computing power between two communication devices, such as in communications between smart cards and a large host computer.

Wiener's attack is based upon properties of continued fractions. We recall that any rational number can be written as a finite simple continued fraction $[a_0, a_1, \ldots, a_m]$; the $i$-th convergent is $n_i/d_i = [a_0, a_1, \ldots, a_i]$ $(i = 0, 1, \ldots, m)$. Let $f'$ be an underestimate of the fraction $f = n_m/d_m$, of the form $f' = f(1-\epsilon)$ for some $\epsilon \geq 0$. Let $a_i$ and $a'_i$ be the $i$-th quotients of $f$ and $f'$, respectively. If $\epsilon$ is sufficiently small, then $n_m$ and $d_m$ can be found using the following continued fraction algorithm:

repeat until $f$ is obtained:

generate the next quotient $a'_i$ of the continued fraction expansion of $f'$;

construct the fraction that equals

$$\begin{array}{ll} [a'_0, a'_1, \ldots, a'_{i-1}, a'_i + 1] & \text{for } i \text{ even} \\ [a'_0, a'_1, \ldots, a'_{i-1}, a'_i] & \text{for } i \text{ odd;} \end{array}$$

check whether this fraction equals $f$.

This algorithm will succeed if

$$\begin{array}{ll} [a_0, a_1, \ldots, a_{m-1}, a_m - 1] < f' \leq f & \text{for even } m \\ [a_0, a_1, \ldots, a_{m-1}, a_m + 1] < f' \leq f & \text{for odd } m. \end{array}$$

It can be shown that this implies

$$\epsilon < \frac{2}{3 n_m d_m} \tag{2.1}$$

which is sufficient to guarantee the success of the algorithm. Put $u = \max(n_m, d_m)$. Assuming we have a test of whether the guess of $f$ is correct which is polynomial in $\log u$, then the continued fraction algorithm can be executed in time polynomial in $\log u$.

Next, we apply the continued fraction algorithm to determine the secret exponent $d$ in the *RSA* cipher. Let $(a, b)$ denote the greatest common divisor of $a$ and $b$. From the relationship $ed \equiv 1 \pmod{\operatorname{lcm}(p-1, q-1)}$ between the public and the secret exponents, we obtain

$$ed = \frac{k}{g}(p-1)(q-1) + 1 \tag{2.2}$$

where

$$g = \frac{(p-1, q-1)}{(K, (p-1, q-1))}, \quad k = \frac{K}{(K, (p-1, q-1))},$$

with

$$K = \frac{ed-1}{\operatorname{lcm}(p-1, q-1)}$$

and $(g, k) = 1$. Then division of (2.2) by $dn$ gives

$$\frac{e}{n} = \frac{k}{dg}(1-\epsilon) \tag{2.3}$$

with

$$\epsilon = \frac{p+q-1-g/k}{n}. \tag{2.4}$$

Here, the fraction $f' = e/n$ is a close underestimate of the fraction $f = k/dg$. Also, $\epsilon$ is sufficiently small according to (2.1) if, on taking $n_m = k$ and $d_m = dg$ and $\epsilon$ as in (2.4), we have

$$kdg < \frac{2pq}{3(p+q-1-g/k)}.$$

This can be simplified to

$$kdg < \frac{2pq}{3(p+q)} \tag{2.5}$$

for $\epsilon = (p+q)/pq$, by dropping the term $-1 - g/k$ which is small in comparison to $p+q$.

We assume $ed > n$ and thus $k > g$. Then (2.2) implies that $edg$ divided by $k$ gives a quotient $(p-1)(q-1)$ and a remainder $g$, providing a guess of $(p-1)(q-1)$ and $g$. The quotient has to be nonzero, otherwise we have to start a new iteration of the continued fraction algorithm. Furthermore, the identity

$$\tfrac{1}{2}(n - (p-1)(q-1) + 1) = \tfrac{1}{2}(p+q)$$

must represent an integer, otherwise the guesses of $k$ and $dg$ are wrong. The identity

$$\left[\tfrac{1}{2}(p+q)\right]^2 - n = \left[\tfrac{1}{2}(p-q)\right]^2$$

shows that the guess of this (right-hand side) integer must be a perfect square, otherwise the iteration has to be stopped. The secret exponent $d$ can be obtained by dividing $d_m$ $(= dg)$ by $g$. We demonstrate this method by way of a small unrealistic example.

*Example.* Let $n = 10541, e = 4133$ be the public key in an RSA cipher. We summarise the computations in a tableaux, where $a_i'$ and $r_i'$ denote the quotients and remainders for the underestimate $f' = e/n$ of $f = k/dg$.

| Quantity | Iteration | | |
|---|---|---|---|
| | $i = 0$ | $i = 1$ | $i = 2$ |
| $a_i'$ | 0 | 2 | 1 |
| $r_i'$ | 4133/10541 | 2275/4133 | 1858/2275 |
| $n_i'/d_i'$ | 0/1 | 1/2 | 1/3 |
| guess of $k/dg$ | 1/1 | 1/2 | 2/5 |
| guess of $edg$ | 4133 | 8266 | 20665 |
| guess of $(p-1)(q-1)$ | 4133 | 8266 | 10332 |
| guess of $g$ | 0 | 0 | 1 |
| guess of $(p+q)/2$ | 3204·5 | 1138 | 105 |
| guess of $[(p-q)/2]^2$ | | 1133·36 | $484 = 22^2$ |
| $d$ | | | $5/1 = 5$ |

Therefore we have found $d = 5$, $p = 127$ and $q = 83$. Also $k = 2$ and $g = 1$.

Equation (2.5) indicates that this continued fraction attack is successful in polynomial time if the secret exponent $d$ has up to approximately a quarter as many significant bits as the modulus. Wiener suggests as a counter measure to combat this attack that the public key satisfy $e > n^{1.5}$, since then the algorithm is not guaranteed to work for any size of the secret exponent.

## 3. McCurley's variation of key distribution.

McCurley [7] proposes a variant of the Diffie and Hellman key distribution scheme that requires the cryptanalyst to solve two problems that are presumed difficult rather than just one. The scheme combines the security of the original scheme (which is based on the difficulty of computing discrete logarithms) with the difficulty of factoring large integers. McCurley works in $\mathbf{Z}_n^*$ with composite $n$ and proves that if the keys are chosen carefully, then any algorithm that will break the system can be used to factor the modulus $n$ and break the original Diffie and Hellman scheme modulo the factors of $n$.

Suppose two users $A$ and $B$ wish to establish a common key. Let $g$ be an element of a group $G$ and assume that an efficient algorithm for multiplying elements in $G$ is known. Then a general form of a Diffie and Hellman ($DH$) key distribution scheme works as follows.

*DH scheme*:

1. $A$ chooses a random integer $x$, kept secret, computes $g^x$ and sends it to $B$.
2. $B$ chooses a random integer $y$, kept secret, computes $g^y$ and sends it to $A$.
3. $A$ and $B$ compute $g^{xy}$ and use it as their common key.

The best known approach to break this scheme is to solve the discrete logarithm problem, that is, given $u$ and $g$, find the exponent $x$ in $u = g^x$, if it exists. It is conjectured that recovering the secret common key $g^{xy}$ is equivalent to the discrete logarithm problem. The original Diffie and Hellman scheme used the group $G = \mathbf{F}_p^*$, where $\mathbf{F}_p$ is the finite field of order $p$, $g$ a primitive root mod $p$ and $p$ a large prime $> 10^{100}$. For descriptions of the discrete logarithm problem and some algorithms for solving it over finite fields of small order, see van Tilborg [19], Lidl and Niederreiter [6]. The survey by Odlyzko [12] still describes the state of the art. An algorithm by Coppersmith has asymptotic running time $O(\exp(Cn^{1/3}\log^{2/3} n))$ for solving the discrete logarithm problem over $\mathbf{F}_q$ when $q = 2^n$. This can be extended to $q = p^n$. In the case of fields of prime order $p$, the running time is of order $O(\exp(C \log p \log\log p)^{1/2})$.

Instead of $\mathbf{F}_p^*$ or $\mathbf{F}_{2^n}^*$, McCurley proposes the use of other groups, since the discrete logarithm problem may be inherently harder for some of these groups than for others.

(1) The group of points on an elliptic curve over a finite field; this group has been used by Miller [8] and Koblitz [5] for a variety of cryptosystems. (See also Kit and Lidl [4].)

(2) The group of equivalence classes of binary quadratic forms of a given negative discriminant; this idea has been implemented by Buchman and Williams [2] for imaginary quadratic fields. (See also their paper in these Proceedings.)

(3) The group of invertible $m \times m$ matrices over a ring $R$.

McCurley's variation uses the published base $g = 16$ and works in $\mathbf{Z}_n^*$, where $n$ is the product of two primes satisfying certain specific requirements. (Shmuely [16] proposes a similar variation using a composite modulus and random bases $g$.) The Diffie and Hellman scheme with composite modulus has to satisfy the following specifications: choose positive integers $r$ and $s$ such that

$2r + 1$ has a large prime factor and $4r + 1$ and $8r + 3$ are both prime;

$s$ has a large prime factor and $4s - 1$ and $8s - 1$ are both prime.

Let $p = 8r + 3, q = 8s - 1$ and set $n = pq$, where $p$ and $q$ have at least 100 or even 150 digits. Regard $n$ as public. Then McCurley's variation of the *DH* scheme above is obtained by replacing $g$ by 16 and performing the computations $16^x$, $16^y$, $16^{xy}$ mod $n$. It is shown that breaking the *DH* scheme with composite modulus $n$ is equivalent to factoring $n$ as $n = pq$ if $p$ and $q$ satisfy the above specifications (see [7], pages 98–99).The paper [7] also contains modifications of the ElGamal cryptosystem (see [6], chapter 9) for composite modulus. The specifications of $n$ are as before, $n$ is supplied by a trusted centre and the factors $p$ and $q$ are kept secret. Let $A$'s public key be $y \equiv 16^a \bmod n$, where $a$ is an odd integer, kept secret. If $B$ wants to send an enciphered message $m$ to $A$, $B$ chooses a random integer $b$ with $1 < b < n$ and computes the number $t \equiv my^b \bmod n$ and $u \equiv 16^b \bmod n$. The enciphered message is the pair $(u, t)$. Deciphering with the secret key $a$ is performed by computing $m \equiv t(u^a)^{-1} \bmod n$.

## 4. Niederreiter's continued fraction tests.

Pseudorandom sequences have a number of applications, especially in stream ciphers, where they take the place of random key strings and, in this context, are called keystream sequences. It is essential to decide when a pseudorandom sequence has

satisfactory randomness properties from the cryptographic viewpoint. It should be difficult or computationally infeasible to infer the parameters or keys in the algorithm that generates the sequence. Stream ciphers have the advantage over block ciphers in that they do not propagate errors and that analytic measures of their quality are more easily formulated. This and their ease of implementation probably account for the high popularity of stream ciphers in cryptographic applications. Randomness of keystream sequences can be tested by a variety of standard statistical tests, such as the equidistribution test, correlation test, serial test, and so on. Recently, Niederreiter [9], [10], [11] suggested new tests, the continued fraction tests, for assessing the randomness of keystream sequences. These tests should be used in addition to the standard statistical tests to give a more stringent and therefore better guide for accepting pseudorandom keystream sequences.

Designers of stream ciphers are greatly concerned with the linear complexity (or linear span or linear equivalence) of the keystream sequence. Many methods of generating keystream sequences use linear recurring (or feedback shift register) sequences in finite fields as their building blocks. A sequence, $s_1, s_2, \ldots$, of elements of the finite field $\mathbf{F}_q$ is called a $k$-th order (linear feedback) shift register sequence if there exist constant coefficients $a_0, a_1, \ldots, a_{k-1}$ in $\mathbf{F}_q$ such that

$$s_{i+k} = a_{k-1}s_{i+k-1} + \ldots + a_0 s_i \quad \text{for} \quad i = 1, 2 \ldots .$$

Until relatively recently, the use of such sequences was suggested for generating pseudorandom binary sequences. But this idea is fundamentally flawed, since any $2k$ consecutive terms of the sequence determine the coefficients $a_0, \ldots, a_{k-1}$ and the initial values and hence the whole sequence (see [6]). Shift register sequences have not been discarded for cryptographic applications. They form the basis of many stream cipher generators by combining the outputs from several such sequences using non-linear logic.

A useful measure for complexity of a periodic sequence $S$ is the linear complexity $L(S)$, which is defined as the least $k$ such that $S$ is a $k$-th order shift register sequence. In view of the comments above, only sequences with a very large linear complexity are acceptable as keystream sequences. $L(S)$ can be calculated by the Berlekamp-Massey algorithm. If we combine the output sequences of several shift registers in a non-linear way, we can obtain a sequence with large linear complexity. There is the danger, however, that individual shift register sequences will be correlated with the keystream sequence. The designer will have to make a trade-off between correlation-immunity and linear complexity (see Siegenthaler [17] and the paper by E. Dawson in these proceedings).

Linear complexity by itself does not give enough information to judge the randomness of a periodic sequence $S$. Consider an arbitrary sequence $S$ of elements of $\mathbf{F}_q$. For $n > 0$, the local linear complexity $L_n(S)$ is the least $k$ such that $s_1, \ldots, s_n$ form the first $n$ terms of a $k$-th order shift register sequence. The sequence $L_1(S), L_2(S), \ldots$ is called the linear complexity profile ($LCP$) of $S$. The $LCP$ measures the extent to which the initial segments of the sequence $S$ can be simulated by shift register sequences. Rueppel [15] showed that for random binary sequences and fixed $n$, the expected value of $L_n(S)$ is $\frac{1}{2}n + c_n$, with $0 \leq c_n \leq \frac{5}{18}$. In general, the sequence $S$ of elements of $\mathbf{F}_q$ is said to have a perfect $LCP$ if $L_n(S) = [\frac{1}{2}(n+1)]$ for all $n \geq 1$. Niederreiter [10]

developed a probabilistic theory to study the behaviour of $L_n(S)$ as $n$ varies, when $S$ is randomly chosen and then fixed. This study relies on the connection between the *LCP* and continued fraction expansions.

A sequence $S$ of elements $s_1, s_2, \ldots$ in $\mathbf{F}_q$ can be identified with its generating function $S = \sum_{i=1}^{\infty} s_i x^i$. We then consider the continued fraction expansion

$$S = [A_1, A_2, \ldots] = \cfrac{1}{A_1 + \cfrac{1}{A_2 + \cdots}},$$

where the $A_j$ are polynomials in $x$ over $\mathbf{F}_q$ and $d_j = \deg(A_j) \geq 1$ for $j \geq 1$. Niederreiter [9] showed that $S$ has a perfect *LCP* if and only if $S$ is nonperiodic and $\deg(A_j) = 1$ for all $j \geq 1$.

The *LCP* of the sequence $S$ can be read off from the continued fraction expansion of the generating function, which can be calculated by means of the Berlekamp-Massey algorithm (see [6]). It has the form

$$0, \ldots, 0, d_1, \ldots, d_1, d_1 + d_2, \ldots, d_1 + d_2, \ldots$$

with 0 repeated $d_1 - 1$ times and $\sum_{i=1}^{j} d_i$ repeated $d_j + d_{j+1}$ times for $j \geq 1$. The probabilistic theory developed in [10] implies that the *LCP* of a random sequence $S$ has this form, where the $d_j = d_j(S)$ are independent and identically distributed random variables with the probability distribution $\mathrm{Prob}(d_j = r) = (q-1)q^{-r}$ for all positive integers $r$. If $P^+$ denotes the set of all polynomials over $\mathbf{F}_q$ of positive degree, then for a random sequence $S$, the polynomials $A_j = A_j(S)$ $(j = 1, 2, \ldots)$ are independent and identically distributed $P^+$-valued random variables with the probability distribution $\mathrm{Prob}(A_j = f) = q^{-2\deg(f)}$ for all $f$ in $P^+$. To use this information for Niederreiter's continued fraction tests, let $g$ be an arbitrary real-valued function on $P^+$ and let $X_j(S) = g(A_j(S))$ for $j \geq 1$. Then $X_1, X_2, \ldots$ are independent and identically distributed random variables with the probability distribution $\mathrm{Prob}(X_j = z) = \sum q^{-\deg(f)}$, where the summation is over all $f$ in $P^+$ with $g(f) = z$ and $z$ is in the countable range of $g$. For a specific sequence $S$, the sequence $X_1, X_2, \ldots$ is then tested by conventional methods such as the $\chi^2$-test, with the null hypothesis that this is a sample sequence of independent and identically distributed random variables. The special case of $g$ being the degree function on $P^+$ leads to a test based on the step heights $d_1, d_2, \ldots$ in the *LCP*. These continued fraction tests check the distribution and independence of step heights and of polynomials in the continued fraction expansion and should augment the standard statistical tests for randomness.

## References

1. G. Brassard, *Modern Cryptology, a Tutorial*, Lecture Notes in Computer Science **325** (1988). (Springer Verlag, New York.)
2. J. Buchmann and H. C. Williams, 'A key exchange system based on imaginary quadratic fields', *J. Cryptology* **1** (1988), 107–118.
3. C. G. Gunther (ed.), *Advances in Cryptology—EUROCRYPT '88*, Lecture Notes in Computer Science **330** (1988). (Springer Verlag, Berlin.)

4. C. Y. Kit and R. Lidl, 'On implementing elliptic curve cryptosystems', *Contributions to General Algebra* **6** (1988), 155–166. (Hölder-Pichler-Tempsky, Wien.)
5. N. Koblitz, 'Elliptic curve cryptosystems', *Math. Comp* **48** (1987), 203–209.
6. R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications.* (Cambridge University Press, Cambridge, 1986.)
7. K. S. McCurley, 'A key distribution system equivalent to factoring', *J. Cryptology* **1** (1988), 95–105.
8. V. S. Miller, 'Use of elliptic curves in cryptography', *Lecture Notes in Computer Science* **218** (1986), 417–426. (Springer Verlag, New York.)
9. H. Niederreiter, 'Cryptology—the mathematical theory of data security', *Proc. Prospects of Math. Science (Tokyo, 1986)* (1988), 189–209. (World Scientific Publishers, Singapore.)
10. H. Niederreiter, 'The probabilistic theory of linear complexity', *Advances in cryptology—EUROCRYPT '88*, Lecture Notes in Computer Science **330** (1988), 191–209. (Springer Verlag, Berlin.)
11. H. Niederreiter, 'The linear complexity profile of keystream sequences', Proc. Workshop on Stream Ciphers (Karlsruhe, 1989), to appear.
12. A. Odlyzko, 'Discrete logarithms in finite fields and their cryptographic significance', *Advances in Cryptology—Proceedings of EUROCRYPT '84*, Lecture Notes in Computer Science **209** (1985), 224–314. (Springer Verlag, New York.)
13. W. Patterson, *Mathematical Cryptology for Computer Scientists and Mathematicians.* (Rowman and Littlefield, Totowa NJ, 1987.)
14 C. Pomerance (ed.), *Advances in Cryptology—CRYPTO '87*, Lecture Notes in Computer Science **293** (1988). (Springer Verlag, New York.)
15. R. A. Rueppel, *Analysis and Design of Stream Ciphers.* (Springer Verlag, Berlin, 1986.)
16. Z. Shmuely, 'Composite Diffie-Hellman public-key generating systems are hard to break', Technical Report no. 356, Computer Science Department, Technion, Israel Institute of Technology, February 1985.
17. T. Siegenthaler, 'Correlation-immunity of nonlinear combining functions for cryptographic applications', *IEEE Trans. Informat. Theory* **IT–30** (1984), 776–780.
18. G. J. Simmons (ed.), 'Special section on cryptology', *Proceedings IEEE* **76** (1988), 533–627.
19. H. C. A. van Tilborg, *An Introduction to Cryptology.* (Kluwer Academic Publishers, Boston, 1988.)
20. M. J. Wiener, 'Cryptanalysis of short RSA secret exponents' (abstract), *EUROCRYPT'89.* (Houthalen, Belgium, 1989.)

*Department of Mathematics, University of Tasmania, Hobart, Tasmania, AUSTRALIA.*

# 2

# QUADRATIC FIELDS AND CRYPTOGRAPHY

**Johannes Buchmann and H. C. Williams**

## 1. Introduction.

Let $D$ be a square-free integer and let $\mathcal{K} = \mathbf{Q}(\sqrt{D})$ be the quadratic field formed by adjoining $\sqrt{D}$ to the rationals $\mathbf{Q}$. The purpose of this paper is to show how the properties of $\mathcal{K}$ can be applied to several cryptographic problems. We will also point out that this research, in turn, has led to new ideas in the development of algorithms for solving problems in $\mathcal{K}$, such as the determination of the class number, principal ideal testing, and (where appropriate) regulator calculation. We will conclude with a discussion of some questions arising from this research.

In order to achieve the objectives of this paper it will first be necessary to review some of the properties of $\mathcal{K}$. Those mentioned in this section can be found in any standard work, such as Cohn [10] or Hua [14]. We begin by putting

$$\sigma = \begin{cases} 1 & \text{when } D \equiv 2, \text{ or } 3 \pmod 4 \\ 2 & \text{when } D \equiv 1 \pmod 4. \end{cases}$$

The *discriminant* of $\mathcal{K}$ is then given by $\Delta = (2/\sigma)^2 D$. Also, if $\alpha, \beta \in \mathcal{K}$ we use $\overline{\alpha}$ to denote the *conjugate* of $\alpha$ in $\mathcal{K}$ and $[\alpha, \beta]$ to denote the module $\{x\alpha + y\beta \mid x, y \in \mathbf{Z}\}$. We further define the *trace* of $\alpha$ as $\mathrm{Tr}(\alpha) = \alpha + \overline{\alpha}$ and the *norm* of $\alpha$ as $\mathrm{N}(\alpha) = \alpha\overline{\alpha}$. The *integers of* $\mathcal{K}$ are those elements $\alpha$ of $\mathcal{K}$ such that both $\mathrm{Tr}(\alpha)$ and $\mathrm{N}(\alpha) \in \mathbf{Z}$; we denote the set of these integers by $\mathcal{O}_\mathcal{K}$. It is well known that

$$\mathcal{O}_\mathcal{K} = [1, \omega],$$

where

$$\omega = (\sigma - 1 + \sqrt{D})/\sigma.$$

If $\alpha, \beta \in \mathcal{O}_\mathcal{K}$, we say that $\alpha$ *divides* $\beta$ ($\alpha \mid \beta$) if there exists some $\gamma \in \mathcal{O}_\mathcal{K}$ such that $\beta = \alpha\gamma$. If $\eta \in \mathcal{O}_\mathcal{K}$ and $\eta \mid 1$, we say that $\eta$ is a *unit* of $\mathcal{K}$. It is well known that when $D < 0$, the only possible values of $\eta$ are $\pm 1$ when $D = -2$ or $|D| > 3$. Also if $D = -1$, then the units of $\mathcal{K}$ are $\pm 1$, $\pm\sqrt{-1}$; if $D = -3$, the units are $\pm 1$, $(\pm 1 \pm \sqrt{-3})/2$. However, when $D > 0$ the situation changes and there are infinitely many units in $\mathcal{O}_\mathcal{K}$. If $\eta$ is one of them, then $\eta$ can be given as $\eta = \pm\epsilon^n$, where $n \in \mathbf{Z}$ and $\epsilon$ ($> 1$) is the *fundamental* unit of $\mathcal{K}$.

As $\mathcal{O}_\mathcal{K}$ is an integral domain, it has ideals. $I$ will be an (integral) ideal of $\mathcal{O}_\mathcal{K}$ if

(1) $\alpha + \beta \in I$ whenever $\alpha, \beta \in I$, and

(2) $\alpha\omega \in I$ whenever $\alpha \in I$.

From these properties it is easy to prove (see Ince [15])

**Theorem 1.1.** *If $I$ is any ideal of $\mathcal{O}_K$, then there exist $a, b, c \in \mathbf{Z}$ such that $a, c > 0$, $0 \leq b < a$, $c \mid a$, $c \mid b$, $ac \mid \mathrm{N}(b + c\omega)$, and*

$$I = [a, b + c\omega].$$

In this case we have

$$a = \min\{I \cap \mathbf{Z}^+\}.$$

We also point out that if $I = [a, b + c\omega]$ with $a, b, c \in \mathbf{Z}$, $c \mid a$, $c \mid b$ and $ac \mid \mathrm{N}(b + c\omega)$, then $I$ is an ideal of $\mathcal{O}_K$. The *norm* of $I$ is given by $\mathrm{N}(I) = ac$. $I$ is said to be *primitive* if $c = 1$.

If $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m \in \mathcal{O}_K$ and we define

$$I = \left\{ \sum_{i=1}^{m} \gamma_i \alpha_i \;\middle|\; \gamma_i \in \mathcal{O}_K, \quad i = 1, 2, \ldots, m \right\},$$

then $I$ is clearly an ideal of $\mathcal{O}_K$. We say that $I$ is the ideal *generated* by $\alpha_1, \alpha_2, \ldots, \alpha_m$ and denote it by $(\alpha_1, \alpha_2, \ldots, \alpha_m)$. If $I = (\alpha)$, we say that $I$ is a *principal* ideal with generator $\alpha$. Notice that if $I = [\alpha_1, \alpha_2]$, then $I = (\alpha_1, \alpha_2)$; thus, from Theorem 1.1 we see that no more than two generators are ever needed to characterize any ideal of $\mathcal{O}_K$. If $I = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ and $J = (\beta_1, \beta_2, \ldots, \beta_n)$ are ideals of $\mathcal{O}_K$, we define the *product* of $I$ and $J$ to be the ideal

$$IJ = (\alpha_1\beta_1, \alpha_1\beta_2, \ldots, \alpha_1\beta_n, \alpha_2\beta_1, \ldots, \alpha_2\beta_n, \ldots, \alpha_m\beta_n)$$

generated by the $mn$ generators $\alpha_i\beta_j$ $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$. We also have $\mathrm{N}(IJ) = \mathrm{N}(I)\mathrm{N}(J)$. Given ideals $I_1 = [a_1, b_1 + c_1\omega]$, $I_2 = [a_2, b_2 + c_2\omega]$ with $0 \leq b_1 < a_1$, $0 \leq b_2 < a_2$, there is a simple $O(\log(a_1 a_2 D))$ algorithm for finding $a_3$, $b_3$, $c_3$ such that

$$I_3 = [a_3, b_3 + c_3\omega] = I_1 I_2 \quad (0 \leq b_3 < a_3).$$

This algorithm, expressed in the language of quadratic forms, goes back to Gauss. In more recent works it can be found in Lenstra [21] and Shanks [34] or in terms of ideals in Buchmann and Williams [4]; see also Shanks [36]. Note also that if $I = [a, b + c\omega]$ is an ideal of $\mathcal{O}_K$, then so is $\overline{I}$, where $\overline{I}$ is the *conjugate* ideal of $I$ and is given by $\overline{I} = [a, b + c\overline{\omega}] = [a, -b - c\,\mathrm{Tr}(\omega) + c\omega]$.

The concept of multiplication of ideals leads us immediately to that of division. We say that an ideal $I$ of $\mathcal{O}_K$ divides an ideal $J$ of $\mathcal{O}_K$ if there exists some ideal $H$ of $\mathcal{O}_K$ such that

$$J = IH.$$

This, it turns out, is equivalent to saying that $I \supseteq J$. We can now define a *prime ideal* $P$ as one which is not $(1) = \mathcal{O}_K$ but is divisible by only itself and (1). With

this definition it is possible to prove that any ideal of $\mathcal{O}_\mathcal{K}$ can be uniquely (up to order) expressed as a product of distinct prime ideal powers. That is, the use of ideals restores to us a unique factorization property in $\mathcal{K}$, something we lose in general for the algebraic integers of $\mathcal{K}$.

For a given rational prime $p$, the various prime ideals of $\mathcal{O}_\mathcal{K}$ fall into three categories, depending upon the value of the Legendre symbol $(\Delta/p)$. When $(\Delta/p) = 1$, we find that $(p) = P\overline{P}$, where $P$ and $\overline{P}$ are prime ideals of $\mathcal{O}_\mathcal{K}$; when $(\Delta/p) = -1$, we find that $(p)$ is a prime ideal of $\mathcal{O}_\mathcal{K}$; and when $p|\Delta$, then $(p) = P^2$, where $P$ is a prime ideal of $\mathcal{O}_\mathcal{K}$. Thus, since it is known that any prime ideal $P$ must divide $(p)$ for some rational prime $p$, we have completely described all the prime ideals which can occur in $\mathcal{O}_\mathcal{K}$.

If $I$ and $J$ are ideals of $\mathcal{O}_\mathcal{K}$ we say that $I$ is *equivalent* to $J$ $(I \sim J)$ if there exist principal ideals $(\alpha)$, $(\beta)$ $(\alpha, \beta \neq 0)$ such that

$$(\alpha)I = (\beta)J.$$

This relationship is a true equivalence relation which partitions the set of ideals of $\mathcal{O}_\mathcal{K}$ into distinct equivalence classes. Indeed, we have

**Theorem 1.2.** *There are only a finite number $h$ of equivalence classes of ideals of $\mathcal{O}_\mathcal{K}$. We call $h$ the class number of $\mathcal{K}$.*

In fact, if $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_h$ represent these distinct classes, and we define

$$\mathbf{C}_i\mathbf{C}_j = \{JH \mid J \in \mathbf{C}_i, H \in \mathbf{C}_j\},$$

the set

$$G = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_h\}$$

is a group called the *class group* of $\mathcal{K}$. The class of principal ideals is the identity element of $G$. It is clear that there are infinitely many ideals in any given ideal class; in the next section we will show how to select a finite number of specific ideals from any of these classes. We use these special ideals to develop algorithms for solving problems in $\mathcal{K}$.

## 2. Reduced ideals and the infrastructure.

If $I$ is an ideal of $\mathcal{O}_\mathcal{K}$, we say that $I$ is a *reduced* ideal of $\mathcal{O}_\mathcal{K}$ if

(1) $I$ is primitive, and

(2) there does not exist a non-zero element $\alpha$ of $I$ such that

$$|\alpha| < \mathrm{N}(I) \text{ and } |\overline{\alpha}| < \mathrm{N}(I).$$

Let $I = [\mathrm{N}(I), b + \omega]$ be a primitive ideal and put $Q_0 = \sigma\mathrm{N}(I)$, $P_0 = \sigma b + \sigma - 1$. When $D < 0$ define

$$\begin{aligned} q_i &= N_e(P_i/Q_i), \\ P_{i+1} &= q_iQ_i - P_i, \\ Q_{i+1} &= (P_{i+1}^2 - D)/Q_i \quad (i = 0, 1, 2, \ldots), \end{aligned}$$

where by $N_e(\alpha)$ we denote an integer such that $|\alpha - N_e(\alpha)| \leq 1/2$ (unique unless $\alpha = \pm 1/2$). When $D > 0$ define

$$\begin{aligned} q_i &= [(P_i + \sqrt{D})/Q_i], \\ P_{i+1} &= q_i Q_i - P_i, \\ Q_{i+1} &= (D - P_{i+1}^2)/Q_i \quad (i = 0, 1, 2, \ldots), \end{aligned}$$

where by $[\alpha]$ we denote the integer part of $\alpha$. The reader will recognize this as the algorithm for the regular continued fraction expansion of $(P_0 + \sqrt{D})/Q_0$.

In each of these cases it is easy to show that

$$I_{j+1} = [Q_j/\sigma, (P_j + \sqrt{D})/\sigma]$$

is an ideal of $\mathcal{O}_\mathcal{K}$. Furthermore, we have

$$(P_i - \sqrt{D})\ I_{i+1} = (Q_i)\ I_i. \tag{2.1}$$

In [4] and Williams and Wunderlich [44] it is shown that if $j > j_0 = O(\log N(I_1))$, then $I_j$ is a reduced ideal of $\mathcal{O}_\mathcal{K}$. In view of (2.1), we see that we have a fast polynomial time algorithm for finding a reduced ideal in any given ideal class. Also, if $D < 0$, it can be shown (see [4]) that there are at most two reduced ideals $I$ and $J$ in any given equivalence class and that $N(I) = N(J) < \sqrt{|D|/3}$. In the case of $D > 0$, the problem of counting the number of reduced ideals in any given ideal class is somewhat more complicated and will be discussed later. We can say here that if $I$ is a reduced ideal of $\mathcal{O}_\mathcal{K}$ when $D > 0$, then $N(I) < \sqrt{\Delta}$, so the number of such ideals is finite (see Corollary 3.5.1 of [44]).

For the remainder of this section we will assume that $D > 0$. If $I\ (= I_1)$ is a reduced ideal, then it can be shown (see [44]) that the algorithm given above will produce a sequence

$$I_1, I_2, I_3, \ldots, I_p, I_{p+1}, \ldots$$

of equivalent, reduced ideals. Further, the sequence is purely periodic with $I_{p+1} = I_1$ for some minimal positive $p$. In addition to this it can also be shown that if $J$ is any reduced ideal equivalent to $I$, then $J = I_m$ for some $m$ with $1 \leq m \leq p$. That is, our algorithm produces *all* of the reduced ideals in any given ideal class.

If we put

$$\psi_i = (P_i + \sqrt{D})/Q_{i-1}$$

and

$$\theta_1 = 1, \quad \theta_n = \prod_{i=1}^{n-1} \psi_i \quad (n \geq 2),$$

then from standard results concerning continued fractions (see again [44]) we get

$$\theta_n = (A_{n-2} + B_{n-2}\sqrt{D})/Q_0, \tag{2.2}$$

where $A_{-1} = Q_0$, $B_{-1} = 0$, $A_0 = P_1$, $B_0 = 1$ and

$$\begin{aligned} A_{k+1} &= q_{k+1}A_k + A_{k-1}, \\ B_{k+1} &= q_{k+1}B_k + B_{k-1}. \end{aligned} \tag{2.3}$$

Also, $|N(\theta_n)| = |Q_{n-1}/Q_0|$ and by (2.1) we have $(N(I_1))I_n = (N(I_1)\theta_n)\ I_1$ with $N(I_1)\theta_n \in \mathcal{O}_\mathcal{K}$. If $n = p+1$, then $\theta_{p+1}$ must be a unit of $\mathcal{K}$; in fact, it is the fundamental unit $\epsilon$ of $\mathcal{K}$.

Let $\delta_n = \delta(I_n)$ denote the *distance* between $I_1$ and $I_n$, which is defined to be the value of $\log\theta_n$. Notice that $\delta_{p+1} = \log\theta_{p+1} = \log\epsilon = R$, the *regulator* of $\mathcal{K}$. If $F_n$ denotes the $n$-th Fibonacci number, then $Q_0\theta_n > F_{n+1} > \tau^{n-1}$ ($\tau = (1+\sqrt{5})/2$) by (2.2), (2.3) and the Binet formula. It follows that $\log Q_0 + \delta_n > (n-1)\log\tau$. Putting $n = p+1$, we find that $p < (\log Q_0 + R)/\log\tau$. Thus, in order to get some idea of the number of reduced ideals in any given ideal class, we must find an upper bound on $R$. This can be done by examining the analytic class number formula

$$2hR = \sqrt{\Delta}L(1,\chi_\Delta),$$

where

$$L(1,\chi_\Delta) = \sum_{n=1}^{\infty}\left(\frac{\Delta}{n}\right)\frac{1}{n}. \tag{2.4}$$

By a result of Hua [14], we can show that

$$R < \sqrt{\Delta}\left(\tfrac{1}{2}\log\Delta + 1\right);$$

thus,

$$p = O(\sqrt{\Delta}\log\Delta).$$

If $I_1 = (1) = [1,\omega]$ and $I_m$, $I_n$ are any two reduced ideals ($\sim I_1$) with $m,n \leq p$, then $I_m = (\theta_m)$, $I_n = (\theta_n)$ and if $H = I_mI_n \sim I_1$ and $J$ is the ideal found by applying our reduction algorithm to $H$, then $J \sim I_1$, and $J$ is reduced; hence, $J = I_k$ for some $k$. Also $\delta_k = \delta_m + \delta_n + \eta$, where it can be proved (see [44]) that $-\log\Delta < \eta < \log 2$ and, furthermore, $\eta$ can be explicitly evaluated by making use of the continued fraction algorithm. Since $\eta$ tends to be small compared to $\delta_m$ and $\delta_n$ for large $m$ and $n$, we see that the multiplication and reduction of two reduced principal ideals provides us with an ideal which is at distance roughly the sum of the distances of the two given ideals. This idea, discovered by Shanks [35], allows us to pass through the reduced ideals in the principal class much more rapidly then we could by using the single step continued fraction algorithm. This structure within the principal ideal class was termed by Shanks the *infrastructure* of this class. It should be noted that the other ideal classes also have infrastructures.

## 3. A one-way function.

Now that we have mentioned most of the results concerning quadratic fields that we will require throughout the remainder of this work, we can begin to discuss some of their cryptographic applications. In modern cryptography the trick is to find a

hard problem that can be used as a basis of the security of the scheme. That is, your opponent should, in order to break the system, be forced to solve a problem that is widely thought to be computationally difficult. As the presentation of a mathematically rigorous proof of the difficulty of any of these problems (computationally) seems not to be forthcoming in the near future, we can only certify the difficulty of these problems by using a somewhat dubious measure: An admissible problem is one that has resisted solution over many years of concerted attack by very knowledgeable and skilled practitioners.

A problem thought to lie in this category is the integer factoring problem: given $N$ $(> 0)$ which is not a prime, find $a, b > 1$ such that $ab = N$. This problem has a long history and, in spite of many new results and insights, still stubbornly resists any computationally rapid solution. We quote H. W. Lenstra [22],

> 'Suppose, for example, that two 80-digit numbers $p$ and $q$ have been proved prime; this is easily within reach of the modern techniques .... Suppose further, that the cleaning lady gives $p$ and $q$ by mistake to the garbage collector, but that the product $pq$ is saved. How to recover $p$ and $q$? It must be felt as a defeat for mathematics that, in these circumstances, the most promising approaches are searching the garbage dump and applying mnemo-hypnotic techniques.'

The best complexity measure that we have for the various algorithms that have been devised for solving this problem is

$$L(N)^{1+o(1)},$$

where by $L(N)$ we denote

$$L(N) = \exp(\sqrt{\log N \log\log N}).$$

It should be noted that this is not even rigorously proved, but is very probably correct (see Pomerance [29] and [30]).

Another very difficult problem seems to be the principal ideal problem, which, while not so famous as the factoring problem, is certainly deserving of some attention. For our purposes we will be content to state it within the narrow confines of quadratic fields, but a more general statement can be easily formulated.

*Given $\mathcal{K}$ and some ideal $I$ of $\mathcal{O}_\mathcal{K}$, determine whether or not $I$ is principal.*

The problem is easy when $D < 0$. We need only reduce the ideal $I$, a process we have already seen can be accomplished quickly, and then check whether or not the reduced ideal is $\mathcal{O}_\mathcal{K} = (1)$, the only reduced principal ideal of $\mathcal{O}_\mathcal{K}$. However, when $D > 0$, the problem is much more difficult, the best rigorously proved complexity measure for it being

$$O(\log(\mathrm{N}(I)\Delta)) + O(\Delta^{o(1)}R^{1/2})$$

of Buchmann and Williams [3].

To get an idea of the speed of this algorithm we must have some estimate of how large $R$ can get. If we refer once again to (2.4), we see that we need a lower bound on

$L(1, \chi_\Delta)$. Unfortunately, this function is very difficult to bound from below. Tatuzawa [41] has shown that with one possible exception we have

$$L(1, \chi_\Delta) > 0 \cdot 655\Delta^{-\eta}$$

where $0 < \eta < 1/2$ and $\Delta > \max\{e^{1/\eta}, e^{11.2}\}$. Also, Littlewood [23] has shown that under the extended Riemann Hypothesis on $L(s, \chi_\Delta)$, we would have

$$L(1, \chi_\Delta) > \frac{\pi(1 + o(1))}{12e^\gamma \log\log\Delta}.$$

Thus, it seems reasonable to assert that

$$hR \gg \Delta^{1/2-o(1)}.$$

If $D$ is a prime, or $2p$ or $pq$, where $p$, $q$ are primes and $p \equiv q \equiv -1 \pmod 4$, then $2 \nmid h$. Other results on the exact power of 2 which can divide $h$ can be found in Kaplan [17]. Further, heuristics of Cohen and Lenstra [9] predict that the odd part of $h$ will be 1 with probability about .754, a prediction which has been tested extensively in the case of $D$ a prime by Stephens and Williams [40]. Thus, it is reasonable to expect (and the calculations bear this out) that $R \gg \Delta^{1/2-o(1)}$ fairly frequently. It follows, then, that the best algorithm for determining the principality of $I$ is likely, in the worst case, to be of exponential complexity. (There may, however, be a sub-exponential algorithm for doing this—see section 7.)

Of particular interest to those interested in producing secure computer login techniques is the concept of a "one-way function". A *one-way function* is defined (see Massey [24]) as a function $f$ such that for every $x$ in the domain of $f$, $f(x)$ is easy to compute; but for virtually all $y$ in the range of $f$ it is computationally infeasible to find an $x$ such that $y = f(x)$. It should be noted that this is not a mathematically precise definition, but it leaves little doubt as to what is intended and can in any context be made quite precise (see Goldwasser [12]). Many possible candidates for such functions have been put forward; in this section we suggest another possibility.

Suppose $x \in \mathbf{Z}$ and $0 < x < R$. For a given $\mathcal{K}$, let $Y$ be the reduced principal ideal such that $|\delta(Y) - x|$ is minimal. (This is well defined because the distance between adjacent ideals is not arbitrarily small but must exceed $(\sqrt{\Delta}+1)^{-1}$.) We define $f$ such that $Y = f(x)$. Since $f$ here is not an injective map, its inverse is not strictly defined. We overcome this difficulty by denoting by $f^*(Y)$ that $x \in \mathbf{Z}$ such that $|x - \delta(Y)|$ is least. By using the infrastructure ideas discussed above it is easy to see that, given $x$, $Y$ can be found by an algorithm of complexity $O(\log x(\log\Delta)^2)$. We simply find some $J_1 = I_s$ such that $\delta_s \approx x/2^k$ $(k \approx \log x)$ and then compute $J_{k+1} \sim J_k^2$ until we get $J_k = Y$. However, given $Y$, the best algorithm [3] currently available for finding $f^*(Y)$ is of complexity $O(R^{1/2}\Delta^{o(1)})$.

Of course, we really do not know how hard it is to determine $f^*$, but we can say that if we have an algorithm that will find $f^*$ quickly, then we can use this same algorithm to factor $\Delta$ quickly. The reason for this is given below. We first point out that in the notation of Section 2, $Q_{p-1} = Q_1$ and $P_p = P_1$ (see, for example, Stephens and Williams [39], Theorem 2.1). Thus, since $R = \log\theta_{p+1}$, we get

$$R \approx f^*([Q_1/\sigma, (P_2 + \sqrt{D})/\sigma]) + \log(P_1 + \sqrt{D})/Q_0).$$

From this we can easily obtain an approximate value for $R$ which is very close to the actual value. We next compute $J = f(R/2)$. There is a very high probability that there is a reduced ideal $H$ very near to $J$ such that $H^2 = (\mathrm{N}(H))$. For such an ideal we have $\mathrm{N}(H) \mid \Delta$. By a result of Schoof [32], we can guarantee (assuming certain Riemann Hypotheses) to find a non-trivial factor of $\Delta$ by applying this idea (with $D$ replaced by small prime multiples of $D$) no more than $O((\log \Delta)^2)$ times.

We also see that if we have any ideal $L$ of $\mathcal{O}_\mathcal{K}$, then we can quickly find a reduced ideal $I \sim L$. Further, $L$ is a principal ideal if and only if $f^*(I)$ can be computed. Thus, any algorithm that inverts $f$ must solve two problems that are considered to be very difficult, an indication that $f$ could be regarded as a one-way function.

## 4. A key exchange system.

In this section we will describe a scheme by which two individuals, $A$ and $B$, who never meet can still develop a secret key for communication over a public channel. The basic idea, which is essentially that of Diffie and Hellman [11], is that $A$ and $B$ agree on some group $G$, and some element $g \in G$ ($g$ should have large order in $G$), both of which can be made public. $A$ selects some positive integer $x$ ($< |G|$) at random and transmits $b = g^x$ to $B$. $B$ selects some positive integer $y$ ($< |G|$) at random and transmits $c = g^y$ to $A$. $A$ determines $k = c^x$ and $B$ determines $k = b^y$; $k$ is used as the secret communication key. An individual tapping the line between $A$ and $B$ would know $G$, $g$, $b$, $c$, but would not know $x$ or $y$. If we could determine $x$ or $y$ from this information, we could compute $k$. We call the problem of determining $x$, given $G$, $g$ and $b$, the *discrete logarithm problem* in $G$. When $G = F_q^\times$, where $F_q = GF(q)$, and $g$ is a primitive element of $F_q$, the discrete logarithm problem is considered, as is the factoring problem, to be very difficult. Indeed, the best algorithm known for solving it (see Odlyzko [27]) is of somewhat similar complexity to the best algorithm known for solving the factoring problem. In this section, we will present a key exchange system of Buchmann and Williams [4] in which $G$ is the class group of a quadratic field $\mathbf{Q}(\sqrt{D})$ with $D < 0$.

In this scheme $A$ and $B$ agree on some $D < 0$ such that $|D|$ is sufficiently large (say $|D| \approx 10^{200}$) and an ideal $I$ of $\mathcal{O}_\mathcal{K}$. $D$ and $I$ can be made public. The following steps are then performed:

(1) $A$ selects at random an integer $x$ ($0 < x < \sqrt{|D|}$) and computes a reduced ideal $J$ such that $J \sim I^x$. $J$ is sent to $B$. ($J$ can be computed quickly by using a power algorithm technique like that discussed in Knuth [18] with a reduction being performed after each multiplication step.)

(2) $B$ selects at random an integer $y$ ($0 < y < \sqrt{|D|}$) and computes a reduced ideal $H$ such that $H \sim I^y$. $H$ is sent to $A$.

(3) $A$ computes a reduced ideal $L_1 \sim H^x$. $B$ computes a reduced ideal $L_2 \sim J^y$. Since $L_1 \sim H^x \sim (I^y)^x = (I^x)^y \sim J^y \sim L_2$, we have $\mathrm{N}(L_1) = \mathrm{N}(L_2)$; thus, this can be used as the secret key.

A computer implementation of this technique has been described by Buchmann, Düllmann, and Williams [8]. In this work a detailed complexity analysis of the methods involved is presented together with extensive running time statistics.

Let $A$ be any algorithm that on being given $I$, $J$, $H$ produces $L = L_1 = L_2$. In [4] it is shown that if $D$ is composite, then it is very likely that algorithm $A$ can be used to find a non-trivial factor of $D$. The method of proof of this is based on an idea employed by McCurley [25] in the case of a key exchange system which used the group of reduced residues modulo a product of two distinct primes. Thus, it seems that this technique is at least as secure as it is difficult to factor $D$. Naturally, if we could solve the discrete logarithm problem in $G$, then we could also break this system. This problem can be solved in sub-exponential time by index calculus methods if $h$ is known. However, the best algorithm known for determining $h$ at the time the paper [4] was written was the algorithm of Shanks [34] and Lenstra [21] of complexity $O(|\Delta|^{1/5+o(1)})$, assuming the extended Riemann Hypothesis on $L(s, \chi_\Delta)$; thus, it seemed that this technique was quite secure. In the next section we will describe some recent results concerning the difficulty of solving the discrete logarithm problem in $G$.

## 5. Complexity results.

As mentioned above, at the time of writing the paper [4] the best available algorithm for determining $h$ for $\mathbf{Q}(\sqrt{D})$ was of exponential complexity. Since that time Hafner and McCurley [13] and Buchmann and Williams [7a] have shown that, under certain Riemann Hypotheses, the value of $h$ for $\mathbf{Q}(\sqrt{D})$ can be computed with sub-exponential complexity

$$L(|\Delta|)^{\sqrt{2}+o(1)}.$$

While the details of the scheme are beyond the scope of this work, we will point out some of its important features.

The basis of the idea is the following observation of Pohst and Zassenhaus [28], Seysen [33], and Lenstra and Lenstra [20]. Suppose $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n\}$ is a set of ideal classes which generates the class group $G$. Consider the homomorphism

$$\theta: \mathbf{Z}^n \to G \quad \text{given by} \quad \mathbf{x} = (x_1, x_2, \ldots, x_n) \mapsto \prod_{i=1}^{n} \mathbf{C}_i^{x_i},$$

The kernel $\mathbf{K}$ of this map is a sub-lattice of $\mathbf{Z}^n$ and since the map is surjective we have by the homomorphism theorem $\mathbf{Z}^n/\mathbf{K} \cong G$; hence,

$$|\mathbf{Z}^n/\mathbf{K}| = h.$$

Now, as noted by Lenstra [21], we can use a result of Lagarias, Montgomery, and Odlyzko [19] to assert (under certain Riemann Hypotheses) that there exists a computable constant $c_1$ such that $G$ is generated by $\mathbf{C}_0 P_i$ $(i = 1, 2, \ldots, n)$, where $\mathbf{C}_0$ is the principal ideal class, $P_i$ is either of the prime ideals of $O_\mathcal{K}$ which divide $p_i$, the $i$-th rational prime such that the Legendre symbol $(\Delta/p_i) = 1$, and $p_i < c_1 (\log |\Delta|)^2$. (From the work of Bach [1], it follows that we can take $c_1 = 8$.) If $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ is a basis of $\mathbf{K}$ formed from vectors in $\mathbf{Z}^n$, then the class number $h$ is the absolute value of the determinant of the integer matrix $(\mathbf{a}_1^T\ \mathbf{a}_2^T\ \ldots\ \mathbf{a}_n^T)$. Moreover, if a system $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_l\} \subseteq \mathbf{K}$ contains $h$ linearly independent vectors then it generates a sub-lattice $\Lambda$ in $\mathbf{K}$ of finite index and its determinant $h'$, which can be found by means of Hermite reduction, is an integer multiple of the class number $h$:

$$h' = h\,|\mathbf{K}/\Lambda|.$$

We are now able to present the main steps of the algorithm of Hafner and McCurley for determining $h$ when $D < 0$.

(1) Find an approximation $h^*$ to the class number $h$ with the property $h \leq h^* < 2h$. This can be done easily under the Riemann Hypothesis on $L(1, \chi_\Delta)$ by using the analytic class number formula.

(2) Initialize $h'$ to 0 and $B$ to $\emptyset$.

(3) Generate a relation $\mathbf{b}$ among the chosen generators and set $B = B \cup \{\mathbf{b}\}$ (see below).

(4) If the index in $\mathbf{K}$ of the sub-lattice $\Lambda$ generated by $B$ over $\mathbf{Z}$ is not finite or if the determinant $h'$ of $\Lambda$ exceeds $h^*$ then return to (3). Otherwise $h = h'$ is the class number.

We remark that at the end of this algorithm a Smith normal form computation will yield a basis and the structure for $G$.

Since we know $h < \frac{1}{3}\sqrt{|\Delta|}\log|\Delta|$ for $|\Delta| > 4$ (Slavutskii [38]), the box

$$\mathbf{R}^n_{|\Delta|} = \{\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbf{R}^n \mid 0 \leq x_i \leq |\Delta|, 1 \leq i \leq h\}$$

contains a basis of $\mathbf{K}$. Hence, step 3 can be effected by first selecting $(x_1, x_2, \ldots, x_n)$ randomly from $\mathbf{Z}^n_{|\Delta|} = \mathbf{Z}^n \cap \mathbf{R}^n_{|\Delta|}$. Then a reduced ideal $I'$ is computed such that

$$I' \sim I = \prod_{i=1}^{n} P_i^{x_i}.$$

If $N(I')$ completely factors over the rational primes $p_1, p_2, \ldots, p_n$, then we can factor $I'$ completely over $P_1, P_2, \ldots, P_n$ (and $\overline{P}_1, \overline{P}_2, \ldots, \overline{P}_n$ where $P_i\overline{P}_i = (p_i)$). This allows us to find a relation among the generators of $G$. For, if

$$I' = \prod_{i=1}^{n} P_i^{y_i}\overline{P}_i^{z_i} \quad (y_i z_i = 0),$$

then we get

$$\prod_{i=1}^{n} P_i^{x_i + z_i - y_i} \sim (1),$$

because $\overline{P}_i \sim P_i^{-1}$ $(1 \leq i \leq n)$ and, therefore, $(x_1 + z_1 - y_1, \ldots, x_n + z_n - y_n) \in \mathbf{K}$.

If the number of elements in the factor base is $L(|\Delta|)^\beta$ for some $\beta \in \mathbf{R}^+$ then it can be shown that the probability of finding such a relation is $L[\frac{1}{4\beta}]$, where this $L$ function satisfies $L[\alpha] = L(|\Delta|)^{\alpha + o(1)}$ as $|\Delta| \to \infty$. Hafner and McCurley also show that the calculation of each reduced $I'$ and the attempt at factorising it takes time $L[\beta]$ and that the generation of $L[\beta]$ relations is sufficient to find a system of generators for $\mathbf{K}$. Hence, finding such a system requires $L[2\beta + \frac{1}{4\beta}]$ bit operations. Furthermore, finding the determinant by Hermite reduction takes $L[4\beta]$ operations and, thus, $\beta = \frac{1}{4}\sqrt{2}$ is optimal.

For $D > 0$ we must change the strategy. An approximation $h^*$ to $h$ as in step 1 of the above algorithm can no longer be computed in polynomial time. Instead one can quickly find an approximation $H^*$ to $hR$ where $R$ is the regulator of $\mathcal{K}$. Therefore, we introduce a lattice $\mathbf{K}'$ of determinant $hR$:

$$\mathbf{K}' = \left\{ (x_1, \ldots, x_n, \log|\alpha|) \in \mathbf{Z}^k \times \mathbf{R} \;\middle|\; (x_1, \ldots, x_n) \in \mathbf{K},\ \alpha \in \mathcal{K}^\times,\ \alpha\mathcal{O}_\mathcal{K} = \prod_{i=1}^{n} P_i^{x_i} \right\}.$$

This means that in the case $D > 0$ there is one more degree of freedom, the generator of the principal ideal representing the relation, which may be altered by multiplication by a unit of $\mathcal{O}_\mathcal{K}$. The above algorithm can then be used to find a generating system $\{\mathbf{b}'_1, \ldots, \mathbf{b}'_l\}$ for the new lattice $\mathbf{K}'$. However, we cannot simply adopt the method of Hafner and McCurley for finding the random relations, because reduction is no longer unique in this case. So the instruction "find the reduced ideal $I'$ in the class of $I$" is no longer well-defined. On the contrary, there are, in general, many reduced ideals in this class and therefore, the random vector $\mathbf{x}$ which determines the ideal $I$ will now have a further coordinate $x_{n+1}$ to randomize the choice of the reduced ideal $I' = \alpha^{-1}I$ with $\alpha \in \mathcal{K}^\times$ which is picked out from the class of $I$. The algorithm will also produce $\log|\alpha|$ which will be the last coordinate in the vector in $\mathbf{K}'$ if $I'$ can be decomposed over the factor base. Once, this modification has been made, it can be shown that $L[2\beta + \frac{1}{4\beta}]$ bit operations suffice to find a generating system $\mathbf{b}'_1, \ldots, \mathbf{b}'_l$ for $\mathbf{K}'$. The class number $h$ can then be found by means of Hermite reduction applied to the matrix whose columns are the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_l$, where $\mathbf{b}_i$ is obtained from $\mathbf{b}'_i$ by deleting the last coordinate. Moreover, using the diophantine approximation properties of the continued fraction algorithm one can also find the regulator $R$.

This means that we have

**Theorem 5.1.** *Under suitable Riemann Hypotheses, the class number of the quadratic number field of discriminant $\Delta$ can be determined in expected running time $L(|\Delta|)^{\sqrt{2}+o(1)}$ as $|\Delta| \to \infty$.*

This, together with the $L(|\Delta|)^{1+o(1)}$ algorithm given in [26] for determining the discrete logarithm of an element in $G$ (see Section 6 for some details), suggests that the problem of breaking the system in Section 4 might be much easier than was originally thought.

Another result obtained by McCurley in his remarkable paper [26] is

**Theorem 5.2.** *Under suitable Riemann Hypotheses, the problem of determining $h$ when $D < 0$ is in* **NP**. *That is, there is a short proof for the value of $h$ for any $\mathcal{K}$ with $D < 0$, but it may take a very long time to find this short proof.*

Although McCurley did not deal with the somewhat more difficult case of $D > 0$, by using his ideas and the infrastructure ideas of Shanks, Buchmann and Williams [5], [7], were able to show that under certain generalized Riemann Hypotheses, the following problems are in $\mathbf{NP} \cap \mathbf{co\text{–}NP}$.

(1) Is a given ideal $I$ of $\mathcal{O}_\mathcal{K}$ principal?

(2) Given ideals $I_1, I_2, \ldots, I_k$ of $\mathcal{O}_\mathcal{K}$, do their equivalence classes generate the class group of $\mathcal{O}_\mathcal{K}$?

(3) Given ideals $I_1, I_2, \ldots, I_k$ of $\mathcal{O}_\mathcal{K}$, do their equivalence classes form a basis for the class group of $\mathcal{O}_\mathcal{K}$?

Thus, the cryptographic scheme presented in Section 4 has led to the discovery of many new results on the complexity of solving problems in any quadratic field. One of the more interesting developments here is the apparent unlikelihood that any of these problems is **NP**-complete. This provides us with some (slight) hope that there may exist even faster algorithms for solving them.

## 6. A public-key system.

It is well known that the *RSA* public-key cryptosystem depends for its security on the supposed difficulty of factoring some large $N$ which is the product of two distinct rational primes $p$, $q$. What we do not know is whether the problem of breaking this *RSA* system is *equivalent* in difficulty to factoring $N$. In 1980, Williams [42], using an idea of Rabin [31], produced a public-key cryptosystem which is provably as secure as it is difficult to factor $N = pq$, where $p$, $q$ are primes such that $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$. This scheme has since found application in work of Simmons and Purdy [37] and Brassard [2].

In reporting on [42] one of the referees properly pointed out that breaking the system is not really equivalent in difficulty to factoring a general $N$, but rather a value of $N$ of the above special type. The question that arises from this is: does there exist a technique for public-key encryption that is as difficult to break as it is to factor a value of $N = pq$ with no further restriction on the primes $p$, $q$? This question was answered in the affirmative by Williams [43]. In order to do this, it was necessary to perform arithmetic in $\mathcal{K} = \mathbf{Q}(\sqrt{D})$ $(D > 0)$.

In this scheme, the designer, as in the *RSA* case, first selects two distinct large primes $p$, $q$ and multiplies them together to produce $N$. He also finds a positive value of $D \in \mathbf{Z}$ such that

$$\eta_p = (D/p) \equiv -p \pmod 4,$$
$$\eta_q = (D/q) \equiv -q \pmod 4,$$

where $(\cdot/p)$ denotes the Legendre symbol. According to certain Riemann Hypotheses such a value of $D$ exists with $D = O((\log N)^2)$ (see [32]). In practice, there is little difficulty in finding a value of $D$ by trial. The designer then evaluates

$$w = \tfrac{1}{4}(p - \eta_p)(q - \eta_q)$$

and finds a value for $A$ such that the Jacobi symbol $((A^2 - D)/N)$ is equal to 1 and $\gcd(A, D) = 1$. This latter task is also easy to accomplish in practice as about one-half of all values of $X$ mod $N$ such that $\gcd(X, D) = 1$ are such that $((X^2 - D)/N) = 1$. After selecting a random value for $e$ such that $1 < e < N$ and $\gcd(e, w) = 1$, he also determines a value of $d$ such that

$$ed \equiv \tfrac{1}{2}(\omega + 1) \pmod w.$$

The public key is $\{N, e, A, D\}$ which does not occupy much more space than that occupied by just $\{N, e\}$ as in the *RSA* case and the private (secret) key is $\{d\}$.

If someone wishes to communicate with the designer, he computes for a given message $M$ (coded numerically) an element $\alpha(M) \in \mathcal{O}_\mathcal{K}$ (the value of $A$ may be used in this step) and then, in effect, raises this element to the public exponent $e$ and reduces it modulo $N$. Part of this result is sent to the designer, who with the knowledge of $d$ can recover $M$. In order to ensure that the transmission band width is not greatly expanded by this scheme, the implementation details are somewhat more complicated than this simple explanation would suggest, but this is the basic idea. The main point here is that this is an *RSA*-like scheme which can be proved to be as difficult to break as it is to factor $N$. Further, it was necessary to utilize the elements of $\mathcal{K} = \mathbf{Q}(\sqrt{D})$ in order to develop such a scheme.

## 7. Questions and conclusions.

In this closing section we will discuss a number of questions which arise from the work described above.

1. How practical are the ideas of Hafner and McCurley for finding $h$ in an imaginary quadratic field?

At present we do not know the answer to this question. At the time of writing, Buchmann and Düllmann are working on this problem. They report that they have experienced some difficulty in working with the matrices that occur when this method is used as the numbers involved (even for fairly small discriminants) do get rather large. However, they believe that this problem can be handled. It should be most interesting to hear of further developments in this work.

2. How practical are the ideas of Buchmann and Williams for finding $h$ and $R$ for a real quadratic field?

At the moment we have very little knowledge concerning this question. A computer implementation of the techniques would be an important first step in evaluating their effectiveness.

3. When $D > 0$ is there a short proof that a given non-principal ideal $I$ of $\mathcal{O}_\mathcal{K}$ is not principal?

We do know [5] that there is an unconditional proof that the problem of the principality of an ideal of $\mathcal{O}_\mathcal{K}$ is in **NP**; however, in order to prove that principality of an ideal is a problem in **co–NP**, we need some Riemann Hypotheses. Is it possible to eliminate these hypotheses from this result?

4. Can we embed a trap-door in the function $f$ described in Section 3?

If it were possible to do this, we could probably develop a public-key cryptosystem based on this function. Recall that a trap-door one-way function is one that can be easily inverted by its designer if he has some special information (obtained through the design process) that is not available to any one else who might want to invert the function. At the moment we know nothing about this question.

5. Can the key exchange idea in Section 4 be used in a real quadratic field?

The answer is "yes" when $R$ is small as is the case, for example, if $D = M^2 + 1$ (there are many other such examples). The reason for this is that there cannot be many reduced ideals in any given ideal class of $\mathcal{O}_\mathcal{K}$ when $R$ is small; hence, the communication

partners could agree (for example) to select as the key ideal that one which has the largest norm value of all the reduced ideals in the class that they both would find.

When $R$ is large the problem is somewhat more difficult, but here again the answer is "yes". Buchmann and Williams [6] have shown how this can be done by using a sort of distance function $\delta(I, x)$ which is defined on the principal ideals of $\mathcal{O}_\mathcal{K}$ and the reals as follows: let $I = (\alpha)$ where $\alpha$ is that element of $\mathcal{O}_\mathcal{K}$ such that $\alpha > 0$ and $|x + \log \alpha|$ $(x \in \mathbf{R})$ is minimal; then $\delta(I, x) = x + \log \alpha$. Also, by $I(x)$ we denote that reduced principal ideal such that $|\delta(I(x), x)|$ is least; that is, $I(x)$ is the closest reduced ideal to $x$. We denote by $\delta(x)$ the quantity $\delta(I(x), x)$.

The communication protocol can now be roughly described as follows. The communication partners are $A$ and $B$ who exchange $I(c)$ and $\delta(c)$ for some $c \in \mathbf{R}^+$. $A$ secretly selects $a \in \{1, 2, \ldots, [\sqrt{D}]\}$ and $B$ secretly selects $b \in \{1, 2, \ldots, [\sqrt{D}]\}$. $A$ computes $I(ac)$ and $\delta(ac)$ and sends this information to $B$ who computes $I(bc)$ and $\delta(bc)$ and sends this to $A$. From this information, and an additional simple protocol, each partner determines the same ideal near to $I(abc)$. This is used as the secret key.

The main problem with this idea is that $A$ and $B$ cannot evaluate $\delta(ac)$, $\delta(bc)$, $I(ac)$, $I(bc)$ exactly. Instead they must work with approximations to these quantities. In [6] it is shown how this difficulty can be overcome. It is of some interest to note that this is the first example known of a Diffie-Hellman type of key exchange which is not based on a structure which is a group. The set of reduced principal ideals of $\mathcal{O}_\mathcal{K}$ does not form a group.

6. Is there a faster than $O(R^{1/2}\Delta^{o(1)})$ method for finding the distance between two principal reduced ideals in a real quadratic field?

The answer to this question also seems to be "yes". Once we know a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ for $\mathbf{K}$ $(D < 0)$ or $\mathbf{b}'_1, \ldots, \mathbf{b}'_{n+1}$ for $\mathbf{K}'$ $(D > 0)$ we can also solve the "discrete logarithm" problem in sub-exponential time. The discrete logarithm problem is the following: given a reduced ideal $I$ of $\mathcal{K}$, find the order of $I\mathbf{C}_0$ in the class group and, in the case of $D > 0$, find $\log |\alpha|$ for some $\alpha \in \mathcal{K}^\times$ with $\alpha^{-1}\mathcal{O}_\mathcal{K} = I$. If we are able to solve the latter problem then we are able to find the distance between any two reduced ideals in the same class.

The algorithm is very similar to the index calculus method. Again, we choose at random an exponent vector $\mathbf{x} \in \mathbf{Z}^n_{|\Delta|}$ (for $D < 0$) or $\mathbf{x} \in \mathbf{Z}^{n+1}_{|\Delta|}$ (for $D > 0$). Then we determine the reduced ideal $I'$ in the class of

$$\widetilde{I} = I \prod_{i=1}^{n} P_i^{x_i}$$

(where for $D > 0$ this reduced ideal is determined by the random coordinate $x_{n+1}$). As above, we attempt to factor $I' = \alpha^{-1}\widetilde{I}$, $\alpha \in \mathcal{K}^\times$, over the factor base. In case of success we find

$$I' = \prod_{i=1}^{n} P_i^{y_i}\overline{P}_i^{z_i} = \prod_{i=1}^{n} p_i^{z_i} P_i^{y_i - z_i}$$

which means that

$$I = \prod_{i=1}^{n} p_i^{z_i} P_i^{y_i - z_i - x_i}. \tag{6.1}$$

We can represent the exponent vector $\mathbf{e} = (y_1 - z_1 - x_1, \ldots, y_n - z_n - x_n)$ in terms of the basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$, by solving the appropriate linear system, say

$$\mathbf{e} = \frac{1}{d} \sum_{i=1}^{n} k_i \mathbf{b}_i, \tag{6.2}$$

with $d \in \mathbf{Z}^+$, $k_i \in \mathbf{Z}$ $(1 \leq i \leq n)$ and $\gcd(d, k_1, \ldots, k_n) = 1$. Then $d$ is the order of $\mathbf{C}_0 I$ in $G$. Note that for $D > 0$ we again obtain $\mathbf{b}_i$ from $\mathbf{b}'_i$ by deleting the last coordinate. Hence, $I$ is principal if and only if $d = 1$. In this case, it follows from (6.1) and (6.2) that

$$v = \sum_{i=1}^{n} k_i b'_{n+1,i}$$

is the logarithm of the inverse of some generator of the principal ideal $I \prod_{i=1}^{n} p_i^{-z_i}$. Hence, we can easily determine such a logarithm for $I$.

As in the preceding cases there are many difficulties which remain to be investigated, but in principle it appears that the problem of determining the distance between two principal reduced ideals is of sub-exponential complexity, assuming, as usual, certain Riemann Hypotheses.

By now it is hoped that the reader has appreciated that even a subject as esoteric as algebraic number theory could be a very useful tool in the development of differing types of cryptosystems. Also, these cryptographic concerns often lead to the development of new results in the invention and implementation of algorithms for solving problems arising in algebraic number theory. Further, as this area has really only just begun to be investigated, there are many interesting and unsolved problems yet to be examined.

## References

1. E. Bach, 'What to do until the witness comes: explicit bounds for primality testing and related problems', *Math. Comp.* (to appear).
2. G. Brassard, 'How to improve signature schemes'. *Proc. EUROCRYPT 89* (to appear).
3. J. Buchmann and H. C. Williams, 'On the infrastructure of the principal ideal class of an algebraic number field of unit rank one', *Math. Comp.* **50** (1988), 569–579.
4. J. Buchmann and H. C. Williams, 'A key exchange system based on imaginary quadratic fields', *Journal of Cryptology* **1** (1988), 107–118.
5. J. Buchmann and H. C. Williams, 'On the existence of a short proof for the value of the class number and regulator of a real quadratic field', *Number Theory and Applications* **265** (1989), 327–345. NATO ASI Series C (Dortrecht).
6. J. Buchmann and H. C. Williams, 'A key exchange system based on real quadratic fields'. Extended abstract (unpublished MS).
7. J. Buchmann and H. C. Williams, 'Remarks concerning the complexity of computing class groups of quadratic fields', unpublished MS (1989).

7a. J. Buchmann and H. C. Williams, 'A sub-exponential class group and regulator algorithm for quadratic fields', to appear.

8. J. Buchmann, S. Düllmann and H. C. Williams, 'On the complexity of a new key exchange system'. *Proc. EUROCRYPT 89* (to appear).

9. H. Cohen and H. W. Lenstra, Jr., 'Heuristics on class groups of number fields', *Number Theory* (Noordwijkerhout, 1983). *Lecture Notes in Math.* **1068** (1984), 33–62. (Springer-Verlag, Berlin and New York.)

10. H. Cohn, *A Second Course in Number Theory* (Wiley, New York, 1962).

11. W. Diffie and M. Hellman, 'New directions in cryptography', *IEEE Trans. Informat. Theory* **IT-22** (1976), 644–654.

12. S. Goldwasser, 'The search for provably secure cryptosystems', *Lecture Notes on Cryptology and Computational Number Theory*, AMS Short Course Series (1989), to appear.

13. J. Hafner and K. M. McCurley, 'A rigorous sub-exponential algorithm for computation of class groups', Research Report, IBM Research Division (San Jose, CA), 1989.

14. L. K. Hua, *Introduction to Number Theory.* (Springer-Verlag, New York, 1982.)

15. E. L. Ince. 'Cycles of reduced ideals in quadratic fields', *Mathematical Tables* **IV** (1934), British Association for the Advancement of Science (London).

16. R. Kannan and A. Bachem, 'Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix', *SIAM Journal of Computing* **8** (1979), 499–507.

17. P. Kaplan, 'Sur le 2-groupe des classes d'ideaux des corps quadratiques', *J. reine angew. Math.* **283/284** (1976), 313–363.

18. D. E. Knuth, *The Art of Computer Programming, Vol. II: Seminumerical Algorithms.* (2nd ed., Addison-Wesley (Reading, Mass), 1981).

19. J. C. Lagarias, H. L. Montgomery and A. M. Odlyzko, 'A bound for the least prime ideal in the Chebotarev density theorem', *Invent. Math.* **58** (1979), 271–296.

20. A. K. Lenstra and H. W. Lenstra, Jr., 'Algorithms in number theory', Technical Report 87-008 (May, 1987), Department of Computer Science, University of Chicago.

21. H. W. Lenstra, Jr., 'On the calculation of regulators and class numbers of quadratic fields', *London Math. Soc. Lecture Note Series* **56** (1982), 123–150.

22. H. W. Lenstra, Jr., 'Primality testing', *Computational Methods in Number Theory* **154** (1982), 55–87. Mathematical Centre Tracts (Amsterdam).

23. J. E. Littlewood, 'On the class number of the corpus $P(\sqrt{-k})$', *Proc. London Math. Soc.* **27** (1928), 358–372.

24. J. L. Massey, 'An introduction to contemporary cryptography', *Proc. IEEE* **76** (1988), 533–549.

25. K. McCurley, 'A key distribution system equivalent to factoring', *J. of Cryptology* **1** (1988), 95–105.

26. K. McCurley, 'Cryptographic key distribution and computation in class groups', *Number Theory and Applications* **265** (1989), 459–479. NATO ASI Series C.

27. A. Odlyzko, 'Discrete logarithms in finite fields and their cryptographic significance'. *Proc. EUROCRYPT 84* (1984), 224–314. (Springer-Verlag New York.)
28. M. Pohst and H. Zassenhaus, 'Über die Berechnung von Klassenzahlen und Klassengruppen algebraischer Zahlkörper', *J. reine angew. Math.* **361** (1985), 50–72.
29. C. Pomerance, 'Analysis and comparison of some integer factoring algorithms', *Computational Methods in Number Theory* **154** (1982), 89–140. (Mathematical Centre Tracts, Amsterdam.)
30. C. Pomerance, 'Factoring', *Lecture Notes on Cryptology and Computational Number Theory*, AMS Short Course Series (1989), to appear.
31. M. O. Rabin, 'Digitized signatures and public-key functions as intractable as factorization', *M.I.T. Lab. for Computer Science*, Tech. Rep. LCS/TR212 (1979).
32. R. Schoof 'Quadratic fields and factorization', *Computational Methods in Number Theory* (H. W. Lenstra, Jr., and R. Tijdeman, eds.) **155** (1983), 235–286. Math. Centrum Tracts (Amsterdam).
33. M. Seysen, 'A probabilistic factorization algorithm with quadratic forms of negative discriminant', *Math. Comp.* **48** (1987), 757–780.
34. D. Shanks, 'Class number, a theory of factorization and genera', *Proc. Symp. Pure Math.* **20** (1971), 415–440. (Amer. Math. Soc.)
35. D. Shanks, 'The infrastructure of real quadratic fields and its application', *Proc. 1972 Number Theory Conf.* (1973), 217–224. (Boulder, Colorado).
36. D. Shanks, 'On Gauss and composition I and II', *Number Theory and Applications* **265** (1989). NATO ASI Series C (Kluwer, Dordrecht).
37. G. J. Simmons and G. B. Purdy, 'Zero-knowledge proofs of identity and veracity of transactions receipts'. *Proc. EUROCRYPT 88* (1988), 35–49. (Springer-Verlag, New York, N.Y.)
38. I. S. Slavutskii, 'Upper bounds and numerical calculation of the number of ideal classes of real quadratic fields', *Amer. Math. Soc. Transl. (2)* **82** (1969), 67–71.
39. A. J. Stephens and H. C. Williams, 'Some computational results on a problem concerning powerful numbers', *Math. Comp.* **50** (1988), 619–632.
40. A. J. Stephens and H. C. Williams, 'Computation of real quadratic fields with class number one', *Math. Comp.* **51** (1988), 809–824.
41. T. Tatuzawa, 'On a theorem of Siegel', *Japan J. Math.* **21** (1951), 163–178.
42. H. C. Williams, 'A modification of the RSA public-key encryption procedure', *IEEE Transactions on Information Theory* **IT-26** (1980), 726–729.
43. H. C. Williams, 'Some public-key crypto-functions as intractable as factorization', *Cryptologia* **9** (1985), 223–237.
44. H. C. Williams and M. C. Wunderlich, 'On the parallel generation of the residues for the continued fraction factoring algorithm', *Math. Comp.* **48** (1987), 405–423.

*FB-10 Informatik, Universität des Saarlandes, D-6600 Saarbrücken, WEST GERMANY.*

*Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, CANADA R3T 2N2*

3

# PARALLEL ALGORITHMS FOR INTEGER FACTORISATION

**Richard P. Brent**

The problem of finding the prime factors of large composite numbers has always been of mathematical interest. With the advent of public key cryptosystems it is also of practical importance, because the security of some of these cryptosystems, such as the Rivest-Shamir-Adelman system, depends on the difficulty of factoring the public keys.

In recent years the best known integer factorisation algorithms have improved greatly, to the point where it is now easy to factor a number with 60 decimal digits, and possible to factor numbers larger than 120 decimal digits, given the availability of enough computing power.

We describe several algorithms, including the *elliptic curve method*, and the *multiple polynomial quadratic sieve* algorithm, and discuss their parallel implementation. It turns out that some of the algorithms are very well suited to parallel implementation. Doubling the degree of parallelism (i.e. the amount of hardware devoted to the problem) roughly increases the size of a number which can be factored in a fixed time by 3 decimal digits.

Some recent computational results are mentioned—for example, the complete factorisation of the eleventh Fermat number, a number of 617 decimal digits, which was accomplished using the elliptic curve method.

## 1. Introduction.

It has been known since Euclid's time (though first clearly stated and proved by Gauss in 1801) that any natural number $N$ has a unique *prime power decomposition*

$$N = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$$

($p_1 < p_2 < \cdots < p_k$ rational primes, $\alpha_j > 0$), and for many purposes we would like an efficient algorithm for computing this decomposition. Note that it is sufficient to have an algorithm for finding a nontrivial factor $f$ of $N$, because this can be applied recursively to $f$ and $N/f$ to obtain the complete prime power decomposition of $N$.

### 1.1. *Serial algorithms.*

A *polynomial time* algorithm is one which runs in time $O((\log N)^c)$ for some constant $c$. However, no such algorithm is known. The algorithms described in sections 5 and 6 run in time $O(N^\epsilon)$ for any positive $\epsilon$. In fact, they are conjectured to run in time $O(\exp(c(\log N \log\log N)^{1/2}))$, where $c$ is a certain constant.

Most useful factorisation algorithms fall into one of two classes.

A. The run time depends mainly on the size of $N$, the number being factored, and is not strongly dependent on the size of the factor found. Examples are:

Lehman's algorithm [18] which has a rigorous worst-case run time bound $O(N^{1/3})$;

Shanks's *SQUFOF* algorithm [38], which has expected run time $O(N^{1/4})$;

Shanks's Class Group algorithm [34, 35] which has run time $O(N^{1/5+\epsilon})$ on the assumption of the Generalised Riemann Hypothesis.

The Continued Fraction algorithm [25] and the Multiple Polynomial Quadratic Sieve algorithm [29]. Under plausible assumptions these algorithms have expected run time $O(\exp(c(\log N \log\log N)^{1/2}))$, where $c$ is a constant (depending on details of the algorithm).

B. The run time depends mainly on the size of $f$, the factor found. (We can assume that $f \leq N^{1/2}$.) Examples are:

The trial division algorithm, which has run time $O(f \cdot (\log N)^2)$;

The Pollard "rho" algorithm [1, 28] which under plausible assumptions has expected run time $O(f^{1/2} \cdot (\log N)^2)$;

Lenstra's "Elliptic Curve Method" (*ECM*) [4, 22] which under plausible assumptions has expected run time $O(\exp(c(\log f \log\log f)^{1/2}) \cdot (\log N)^2)$, where $c$ is a constant.

In these examples, the term $(\log N)^2$ is a generous allowance for the cost of performing arithmetic operations on numbers of size $O(N)$ or $O(N^2)$, and could theoretically be replaced by $(\log N)^{1+\epsilon}$ for any $\epsilon > 0$.

Algorithms in class B are useful for "naturally occurring" numbers $N$ which are quite likely to have small factors. Note that the difficulty of factoring a number $N$ by an algorithm in class B depends on the size of the *second-largest* prime factor $p_{k-1}$ of $N$ rather than on the size of the *largest* prime factor $p_k$. For randomly chosen $N$ there is a 50 percent chance that $p_{k-1} < N^{0.212}$. Thus, if we have an algorithm in class B which can find factors of size $10^{22}$ in a reasonable time, there is a 50 percent chance that the algorithm will be able to completely factor a random number $N$ of size about $10^{100}$. (See [7] for an example, and [12, 16] for the theory.)

In cryptographic applications [33], the numbers $N$ to be factored are not random. More likely they have been constructed with the intention of being difficult to factor. For such numbers, algorithms in class A are preferable. However, it is generally worth attempting to find small factors with an algorithm in class B before embarking on a long computation with an algorithm in class A.

1.2. *Parallel algorithms.*

The time bounds mentioned above assume that only one arithmetic operation is performed at a time. A practical way of speeding up computations is by the use of parallelism. We would hope that an algorithm which required time $T_1$ on a computer with one processor could be implemented to run in time $T_P \sim T_1/P$ on a computer with $P$ independent processors. This is not always the case, since it may be impossible to use all $P$ processors effectively. However, it is often true for integer factorisation algorithms, provided that $P$ is not too large.

The *speedup* of a parallel algorithm is $S = T_1/T_P$ and the *efficiency* is $E = S/P$. We aim for a linear speedup, i.e. $S = \Omega(P)$. If the speedup is linear in the number

of processors $P$, then each processor is being used with efficiency bounded below by a positive constant.

There are several recent surveys of integer factorisation algorithms [8, 9, 13, 24, 29, 32]. In this paper we concentrate on the efficient parallel implementation of the algorithms.

## 2. Trial division.

Trial division is a straightforward factorisation algorithm. We just try potential divisors $d = 2, 3, \ldots$ until one of the following occurs:

(1) $d > N^{1/2}$, in which case $N$ is prime; or

(2) $d < N$ and $d \mid N$, in which case $d$ is a nontrivial prime divisor of $N$; or

(3) $d$ exceeds some preassigned bound $B < N^{1/2}$, in which case all we can say is that any prime factor $p$ of $N$ satisfies $p > B$.

Naturally, refinements are possible. If $N$ is odd, only odd $d$ need be considered. Multiples of 3 may also be excluded if $N \not\equiv 0 \bmod 3$. Carrying this process to its logical conclusion, we need only consider prime divisors $d$, but it is necessary to consider how the set of primes less than $B$ is computed and whether there is any overall saving.

The simplest version of trial division takes time $O(p \cdot \log N)$ to find the smallest prime factor $p$ of $N$. Here the factor "$\log N$" allows for division of the multiple-precision number $N$ by single-precision trial divisors $d \leq p$. The run time might be reduced by a factor of $\log N$ or $\log p$ with various refinements.

The parallel implementation of trial division is extremely straightforward. With $P$ processors we can perform up to $P$ trials in parallel. Thus, provided $P \ll p$, a linear speedup is obtained.

In sections 3 to 6 we assume that $N$ is composite, since in practice this is easily checked using a probabilistic primality test [16, 32] which runs in time $O(\log N)^3$. It is also convenient to assume that all "small" factors of $N$ have been removed by trial division.

## 3. The Pollard "rho" algorithm.

Pollard's "rho" algorithm [28] uses an iteration of the form

$$x_{i+1} = f(x_i) \bmod N, \quad i \geq 0,$$

where $N$ is the number to be factored, $x_0$ is a random starting value, and $f$ is a polynomial with integer coefficients. In practice a quadratic polynomial, $f(x) = x^2 + a$, is used ($a \not\equiv 0, -2 \bmod N$).

Let $p$ be the smallest prime factor of $N$, and $j$ the smallest positive index such that $x_{2j} = x_j \bmod p$. Making some plausible assumptions, it is easy to show that the expected value of $j$ is $E(j) = O(p^{1/2})$. The argument is related to the well-known "birthday" paradox. The probability that $x_0, x_1, \ldots, x_k$ are all distinct mod $p$ is approximately

$$(1 - 1/p) \cdot (1 - 2/p) \cdots (1 - k/p) \sim \exp(-k^2/(2p)),$$

and if $x_0, x_1, \ldots, x_k$ are not all distinct mod $p$ then $j \leq k$.

In practice we do not know $p$ in advance, but we can detect $x_j$ by taking greatest common divisors. We simply compute $\gcd(x_{2i} - x_i, N)$ for $i = 1, 2, \ldots$ and stop when a nontrivial gcd (necessarily a factor of $N$) is found.

Various refinements are possible. Because gcd's are more expensive than multiplications (mod $N$), it is preferable to avoid the computation of most of the gcd's by accumulating the product $\prod(x_{2i} - x_i) \bmod N$. Also, the choice of subscripts $2i$ and $i$ here is not optimal [1].

The "rho" algorithm is an improvement over trial division in that it has (conjectured) expected run time $O(p^{1/2}(\log N)^2)$ to find a prime factor $p$ of $N$. A disadvantage is that the run time is now only a (conjectured) expected value, not a rigorous bound.

An example of the success of a variation on the Pollard "rho" algorithm is the complete factorisation of the Fermat number $F_8 = 2^{2^8} + 1$ by Brent and Pollard [7].

Unfortunately, parallel implementation of the "rho" algorithm does not give linear speedup. Because the degree of $x_i$, regarded as a polynomial in $x_0$, is $2^i$, it does not seem possible to use parallelism to speed up the computation of the sequence $(x_i)$ by a significant amount. A plausible use of parallelism is to try several different pseudo-random sequences (generated by different polynomials $f$). If we have $P$ processors and use $P$ different sequences in parallel, the probability that the first $k$ values in each sequence are distinct mod $p$ is approximately $\exp(-k^2P/2p)$, so the speedup is $O(P^{1/2})$ and the efficiency is only $O(P^{-1/2})$.

## 4. The Pollard "p - 1" algorithm.

Pollard's "$p - 1$" algorithm [24, 27] is based on Fermat's theorem

$$a^{p-1} = 1 \bmod p$$

for $0 < a < p$, $p$ prime. Suppose that $p$ is a prime factor of $N$ and that $E$ is a multiple of $p - 1$. Then, from Fermat's theorem,

$$p \mid \gcd(a^E - 1, N)$$

and $\gcd(a^E - 1, N)$ gives us a factor (possibly trivial) of $N$.

Since $p$ is not known in advance, the algorithm supposes that all prime power factors of $p-1$ are bounded above by some arbitrarily chosen number $m$. Then, taking $E$ as a product of prime powers $q^e$, $q^e \leq m$, we obtain a multiple of $p-1$. If $p-1$ has a prime factor greater than $m$, then the algorithm will generally fail to give a nontrivial factor of $N$.

Because $E$ is very large, it is not actually computed. Instead, $a^E \bmod N$ is computed via a sequence of computations

$$a \leftarrow a^{q^e} \bmod N.$$

The work involved in such an exponentiation is $O(\log(q^e))$ multiplications mod $N$, so the total work involved is $O(m)$ multiplications mod $N$, and the total time required is $O(m \cdot (\log N)^2)$.

The "$p-1$" algorithm is very effective in the fortunate case that $p-1$ has all "small" prime factors. For example, Baillie found the factor

$$p_{25} = 1155685395246619182673033$$

of the Mersenne number $M_{257} = 2^{257}-1$ (claimed to be prime by Mersenne) using the "$p-1$" algorithm. In this case

$$p_{25} - 1 = 2^3 \cdot 3^2 \cdot 19^2 \cdot 47 \cdot 67 \cdot 257 \cdot 439 \cdot 119173 \cdot 1050151,$$

and $m \geq 1050151$ is sufficient.

A more extreme example: using the "$p-1$" algorithm we found the factor

$$p_{32} = 4985899058078884305401269007884 1$$

of the Mersenne number $M_{977} = 2^{977}-1$. Here

$$p_{32} - 1 = 2^3 \cdot 5 \cdot 13 \cdot 19 \cdot 977 \cdot 1231 \cdot 4643 \cdot 74941 \cdot 1045397 \cdot 11535449,$$

but because we used a two-phase algorithm $m \geq 1045397$ was sufficient (see Section 5.4 for the idea of the second phase).

The examples just given are not typical. If we are unlucky, $(p-1)/2$ may be prime, so the worst case time bound for the "$p-1$" algorithm is no better than for trial division.

Parallel implementation of the "$p-1$" algorithm is difficult, because the inner loop seems inherently serial. At best, parallelism can speed up the multiple precision operations by a small factor (depending on $\log N$ but not on $p$).

In the next section we show that it is possible to overcome the main handicaps of the "$p-1$" algorithm, and obtain an algorithm which is easy to implement in parallel and does not depend on a lucky factorisation of $p-1$.

## 5. Lenstra's elliptic curve algorithm.

Pollard's "$p-1$" algorithm may be regarded as an attempt to generate the identity in the multiplicative group of $F_p = GF(p)$. The motivation for H. W. Lenstra's elliptic curve algorithm (usually denoted "*ECM*") is as follows. If we can choose a "random" group $G$ with order $g$ close to $p$, we may be able to perform a computation similar to that involved in Pollard's "$p-1$" algorithm, working in $G$ rather than in $F_p$. If all prime factors of $g$ are less than the bound $m$ then we find a factor of $N$. Otherwise, repeat with a different $G$ (and hence, usually, a different $g$) until a factor is found.

A curve of the form

$$y^2 = x^3 + ax + b \tag{5.1}$$

over some field $F$ is known as an *elliptic curve*. A more general cubic in $x$ and $y$ can be reduced to the form (5.1), which is known as the Weierstrass normal form, by rational transformations.

There is a well-known way of defining an Abelian group $(G, *)$ on an elliptic curve over a field. Formally, if $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are points on the curve, then the point $P_3 = (x_3, y_3) = P_1 * P_2$ is defined by

$$(x_3, y_3) = (\lambda^2 - x_1 - x_2,\ \lambda(x_1 - x_3) - y_1), \tag{5.2}$$

where

$$\lambda = \begin{cases} (3x_1^2 + a)/(2y_1) & \text{if } P_1 = P_2 \\ (y_1 - y_2)/(x_1 - x_2) & \text{otherwise.} \end{cases}$$

The identity element $I$ in $G$ is the "point at infinity".

The geometric interpretation is straightforward. We refer the reader to [14, 17] for an introduction to the theory of elliptic curves.

In Lenstra's algorithm [22] the field $F$ is the finite field $F_p$ of $p$ elements, where $p$ is a prime factor of $N$. The multiplicative group of $F_p$, used in Pollard's "$p-1$" algorithm, is replaced by the group $G$ defined by (5.1) and (5.2). Since $p$ is not known in advance, computation is performed in the *ring* of integers modulo $N$ rather than in $F_p$. We can regard this as using a redundant representation for elements of $F_p$.

5.1. *Computing inverses mod N.*

In order to implement (5.2) we need to compute inverses mod $N$. Suppose that $x$ is given and we want to compute $z$ such that $xz = 1 \bmod N$. This is easily done via the extended Euclidean algorithm applied to $x$ and $N$, which gives $u$ and $v$ such that

$$ux + vN = \gcd(x, N).$$

If $\gcd(x, N) = 1$ then $ux = 1 \bmod N$, so $z = u$. If $\gcd(x, N) > 1$ then $\gcd(x, N)$ is a nontrivial factor of $N$, so we stop. It is curious that Lenstra's algorithm finds a factor of $N$ precisely when an inverse computation breaks down (formally, when the identity element of $G$ arises in a nontrivial way).

The cost of an extended gcd computation is about the same as that of 10 to 12 multiplications mod $N$ (see [4, 19]).

5.2. *One trial of Lenstra's algorithm.*

A *trial* is the computation involving one random group $G$. The steps involved are

(1) Choose $x_0, y_0$ and $a$ randomly in $[0, N)$. This defines $b = y_0^2 - (x_0^3 + ax_0) \bmod N$. Set $P \leftarrow P_0 = (x_0, y_0)$.

(2) For prime $q = 2, \ldots, m$ set $P \leftarrow P^{q^e}$ in the group $G$ defined by $a$ and $b$, where $e$ is an exponent chosen as in Pollard's "$p-1$" algorithm. If $P = I$ then a factor of $N$ will have been found during an attempt to compute an inverse mod $N$.

The work involved is $O(m)$ group operations. Note that several trials can be performed in parallel.

5.3. *The choice of m.*

Given $x \in F_p$, there are at most two values of $y \in F_p$ satisfying (5.1). Thus, allowing for the identity element, we have $g = |G| \leq 2p + 1$. Although this would be

sufficient for an approximate analysis of *ECM*, a much stronger result, the *Riemann hypothesis for finite fields* [15], is known, namely

$$|g - p - 1| < 2p^{1/2}. \tag{5.3}$$

Making the (incorrect, but close enough) assumption that $g$ behaves like a random integer distributed uniformly in $(p - 2p^{1/2}, p + 2p^{1/2})$, we may show that the optimal choice of $m$ is $m = p^{1/\alpha}$, where

$$\alpha \sim (2 \ln p / \ln \ln p)^{1/2}. \tag{5.4}$$

The expected run time is

$$T = p^{2/\alpha + o(1/\alpha)}. \tag{5.5}$$

For details, see [4, 22]. From (5.5), we see that the exponent $2/\alpha$ should be compared with 1 (for trial division) or 1/2 (for Pollard's "rho" method). For $10^{10} < p < 10^{30}$, we have $\alpha \in (3{\cdot}2, 5{\cdot}0)$. Because of the overheads involved with *ECM*, a simpler algorithm such as Pollard's "rho" is preferable for finding factors of size up to about $10^{10}$ (see Figure 1 in [4]), but for larger factors the asymptotic advantage of *ECM* becomes apparent.

5.4. *A second phase.*

Both the Pollard "$p - 1$" and Lenstra elliptic curve algorithms can be speeded up by the addition of a second phase. The idea of the second phase is to find a factor in the case that the first phase terminates with a group element $P \neq I$, such that $|\langle P \rangle|$ is reasonably small (say $O(m^2)$). (Here $\langle P \rangle$ is the cyclic group generated by $P$.) There are several possible implementations of the second phase. One of the simplest uses a pseudorandom walk in $\langle P \rangle$. By the birthday paradox argument, there is a good chance that two points in the random walk will coincide after $O(|\langle P \rangle|)^{1/2}$ steps, and when this occurs a nontrivial factor of $N$ can usually be found. Details may be found in [4, 24].

The use of a second phase provides a significant speedup in practice, but does not change the asymptotic time bound (5.5). Similar comments apply to other implementation details, such as ways of avoiding most divisions and speeding up group operations [11, 23, 24], ways of choosing good initial points [24, 37], and ways of using preconditioned polynomial evaluation [4, 26, 40].

5.5. *Parallel implementation of ECM.*

So long as the expected number of trials is much larger than the number $P$ of processors available, linear speedup is possible by performing $P$ trials in parallel. In fact, if $T_1$ is the expected run time on one processor, then the expected run time on a parallel machine with $P$ processors is

$$T_P = T_1/P + O(T_1^{1/2+\epsilon}) \tag{5.6}$$

The bound (5.6) applies on single-instruction multiple-data (*SIMD*) machines if we use the Montgomery-Chudnovsky form [11, 24]

$$by^2 = x^3 + ax^2 + x$$

instead of the Weierstrass normal form (5.1) in order to avoid divisions.

In practice, it may be difficult to perform $P$ trials in parallel because of storage limitations. The second phase requires much more storage than the first phase. Fortunately, there are several possibilities for making use of parallelism during the second phase of each trial. Our implementation performs the first phase of $P$ trials in parallel, but the second phase of each trial sequentially, using $P$ processors to speed up the evaluation of the polynomials

$$\prod_{i,j}(x_i - \overline{x}_j)$$

which constitute most of the work during the second phase.

## 6. Quadratic sieve algorithms.

Quadratic sieve algorithms belong to a wide class of algorithms which try to find two integers $x$ and $y$ such that

$$x^2 = y^2 \bmod N \tag{6.1}$$

Once such $x$ and $y$ are found, there is a good chance that $\gcd(x-y, N)$ is a nontrivial factor of $N$.

One way to find $x$ and $y$ is to find a set of relations of the form

$$u_i^2 = v_i^2 w_i \bmod N, \tag{6.2}$$

where the $w_i$ have all their prime factors in a moderately small set of primes (called the *factor base*). Each relation (6.1) gives a row in a matrix $M$ whose columns correspond to the primes in the factor base. Once enough rows have been generated, we can use Gaussian elimination in $F_2$ [39] to find a linear dependency (mod 2) between a set of rows of $M$. Multiplying the corresponding relations now gives a relation of the form (6.1).

In quadratic sieve algorithms the numbers $w_i$ are the values of one (or more) polynomials with integer coefficients. This makes it easy to factorise the $w_i$ by *sieving*. For details of the process, we refer to the recent papers [10, 20, 29, 30, 31, 36]. The conclusion is that the best quadratic sieve algorithms (such as the *multiple polynomial quadratic sieve algorithm MPQS* [29, 36]) can, under plausible assumptions, factor a number $N$ in time $O(\exp(c(\log N \log\log N)^{1/2}))$, where $c \sim 1$. The constants involved are such that *MPQS* is usually faster than *ECM* if $N$ is the product of two primes which both exceed $N^{1/3}$. (The exponent "1/3" is empirical, based on experience with $N < 10^{100}$.)

Although *MPQS* is a probabilistic algorithm, depending on the factorisation of the numbers $w_i$ over the factor base, it is much more predictable than *ECM*. This is because *MPQS* depends on obtaining a large number of factorisations of $w_i$, so the law of large numbers applies and we can predict with confidence how much work will be required, as a function of $N$. *ECM*, on the other hand, depends on *one* unlikely event occurring, so the run time behaves like an exponentially distributed random variable

whose expectation is a function of $p$, the factor eventually found. In practice, we know $N$ but not $p$ in advance.

The reader may be surprised that algorithms as different as *MPQS* and *ECM* have similar expected time bounds. However, this is not really so surprising. *MPQS* requires $O(B)$ factorisations of numbers $w_i$ of size $O(N^{1/2+\epsilon})$ over the factor base of size $B$, and the work per trial is small (because of the sieving process). On the other hand, *ECM* requires only one number (the order of the group $G$) to factor completely over primes not exceeding $m$, but the work per trial is $O(m)$. Use of "partial relations", i.e. incompletely factored $w_i$, in *MPQS* is analogous to the second phase of *ECM*.

6.1. *Parallel implementation of MPQS.*

Like *ECM*, *MPQS* is ideally suited to parallel implementation. Different processors may use different polynomials, or sieve over different intervals with the same polynomial. Thus, there is a linear speedup so long as the number of processors is not much larger than the size of the factor base. The process requires very little communication between processors. Each processor can generate relations and forward them to some central collection point. This has been demonstrated most clearly by A. K. Lenstra and M. S. Manasse [20] who distribute their program and collect relations via electronic mail. The processors are scattered around the world – anyone with access to electronic mail and a C compiler can volunteer to contribute. (The final stage – Gaussian elimination to combine the relations – is not so easily distributed. However, in practice it is only a small fraction of the computation.)

## 7. Some recent computational results.

In the process of proving the non-existence of an odd perfect number less than $10^{300}$ [5, 6], we needed many factorisations of numbers of the form $p^n - 1$, where $p$ and $n$ are prime. For example, the factorisation

$$c_{101} = (467^{41} - 1)/(466 \cdot 1022869) = 4089568263561830388113662969166474269 \cdot p_{65}$$

was found by *ECM*.

We recently [2] completed the factorisation of the 617-decimal digit Fermat number $F_{11} = 2^{2^{11}} + 1$. In fact

$$F_{11} = 319489 \cdot 974849 \cdot 167988556341760475137 \cdot 3560841906445833920513 \cdot p_{564}$$

where the 21-digit and 22-digit prime factors were found using *ECM*, and $p_{564}$ is a 564-decimal digit prime. The factorisation required about 360 million multiplications mod $N$, which took less than 2 hours on a Fujitsu VP 100 vector processor.

Using the *MPQS* algorithm and their worldwide distributed network [20], Lenstra and Manasse (with many assistants, including the present author) have factorised several numbers larger than $10^{100}$, the largest (at the time of writing) having 106 decimal digits. For example, the most recently completed was the 103-decimal digit number

$$(2^{361} + 1)/(3 \cdot 174763) = 687430161753482750935057576845435624502540 3 \cdot p_{61}$$

Such factorisations require many years of CPU time, but an "elapsed time" of only a month or so because of the number of different processors which are working in parallel, using machine cycles which would otherwise be idle.

Lenstra and Manasse [21] recently announced the factorisation of the 122-decimal digit number $c_{122} = (7^{149} + 1)/(2^3 \cdot 10133)$, in fact

$$c_{122} = 47338433355189929279110650931837806119829008573928501623 \cdot p_{66}$$

This impresssive factorisation was obtained using an unpublished algorithm, the *Number Field Sieve* (*NFS*) due to J. M. Pollard, A. K. Lenstra and H. W. Lenstra, Jr. (Unlike *ECM* or *MPQS*, the *NFS* algorithm took advantage of the special form of $c_{122}$, so it is not clear whether a 122-digit number intended for use in a public key cryptosystem could be factorised in a comparable time.) Since the *NFS* algorithm uses similar ideas to the *MPQS* algorithm, it should be possible to implement it equally well on a parallel machine.

## Remark

We take this opportunity to announce the availability of an integer factorisation program written in Turbo Pascal for the IBM PC [3].

## References


1. R. P. Brent, 'An improved Monte Carlo factorization algorithm', *BIT* **20** (1980), 176–184.
2. R. P. Brent, 'Factorization of the eleventh Fermat number' (preliminary report), *AMS Abstracts* **10** (1989), 89T-11-73.
3. R. P. Brent, *Factor: an integer factorization program for the IBM PC*, Computer Sciences Laboratory, Australian National University, Sept. 1989. (Available from the author.)
4. R. P. Brent, 'Some integer factorization algorithms using elliptic curves', *Australian Computer Science Communications* **8** (1986), 149–163.
5. R. P. Brent and G. L. Cohen, 'A new lower bound for odd perfect numbers', *Mathematics of Computation*, July 1989.
6. R. P. Brent, G. L. Cohen and H. J. J. te Riele, *Improved techniques for lower bounds for odd perfect numbers.*Technical Report, Computer Sciences Laboratory, Australian National University, August 1989 (to appear).
7. R. P. Brent and J. M. Pollard, 'Factorization of the eighth Fermat number', *Mathematics of Computation* 36 (1981), 627–630.
8. J. Brillhart, D. H. Lehmer, J. L. Selfridge, B. Tuckerman and S. S. Wagstaff, Jr., *Factorizations of* $b^n \pm 1, b = 2, 3, 5, 6, 7, 10, 11, 12$ *up to high powers*, American Mathematical Society, Providence, Rhode Island, second edition, 1985.
9. D. A. Buell, 'Factoring: algorithms, computations, and computers', *J. Supercomputing* **1** (1987), 191–216.
10. T. R. Caron and R. D. Silverman, 'Parallel implementation of the quadratic sieve', *J. Supercomputing* **1** (1988), 273–290.

11. D. V. Chudnovsky and G. V. Chudnovsky, *Sequences of numbers generated by addition in formal groups and new primality and factorization tests*, Dept. of Mathematics, Columbia University, July 1985.

12. K. Dickman, 'On the frequency of numbers containing prime factors of a certain relative magnitude', *Ark. Mat., Astronomi och Fysik*, 22A, 10 (1930), 1–14.

13. R. K. Guy, 'How to factor a number', *Congressus Numerantium XVI*, Proc. Fifth Manitoba Conference on Numerical Mathematics, Winnipeg, 1976, 49–89.

14. K. F. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, Springer-Verlag, 1982, Ch. 18.

15. J-R. Joly, 'Equations et variétés algébriques sur un corps fini', *L'Enseignement Mathématique* **19** (1973), 1–117.

16. D. E. Knuth, *The Art of Computer Programming*, Vol. 2, Addison Wesley, 2nd edition, 1982.

17. S. Lang, *Elliptic Curves—Diophantine Analysis*, Springer-Verlag, 1978.

18. R. S. Lehman, 'Factoring large integers', *Mathematics of Computation* **28** (1974), 637–646.

19. D. H. Lehmer, 'Euclid's algorithm for large numbers', *Amer. Math. Monthly* **45** (1938), 227–233.

20. A. K. Lenstra and M. S. Manasse, *Factoring by electronic mail*, preprint, 10 June 1989.

21. A. K. Lenstra and M. S. Manasse, personal communication, 28 August 1989.

22. H. W. Lenstra, Jr., 'Factoring integers with elliptic curves', *Ann. of Math.* (2) **126** (1987), 649–673.

23. P. L. Montgomery, 'Modular multiplication without trial division', *Mathematics of Computation* **44** (1985), 519–521.

24. P. L. Montgomery, 'Speeding the Pollard and elliptic curve methods of factorization', *Mathematics of Computation* **48** (1987), 243–264.

25. M. A. Morrison and J. Brillhart, 'A method of factorization and the factorization of $F_7$', *Mathematics of Computation* **29** (1975), 183–205.

26. M. Paterson and L. Stockmeyer, 'On the number of nonscalar multiplications necessary to evaluate polynomials', *SIAM J. on Computing* **2** (1973), 60–66.

27. J. M. Pollard, 'Theorems in factorization and primality testing', *Proc. Cambridge Philos. Soc.* **76** (1974), 521–528.

28. J. M. Pollard, 'A Monte Carlo method for factorization', *BIT* **15** (1975), 331-334.

29. C. Pomerance, 'Analysis and comparison of some integer factoring algorithms'. *Computational Methods in Number Theory* (edited by H. W. Lenstra, Jr. and R. Tijdeman), Math. Centrum Tract 154, Amsterdam, 1982, 89–139.

30. C. Pomerance, J. W. Smith and R. Tuler, 'A pipeline architecture for factoring large integers with the quadratic sieve algorithm', *SIAM J. on Computing* **17** (1988), 387-403.

31. H. J. J. te Riele, W. Lioen and D. Winter, *Factoring with the quadratic sieve on large vector computers*, Report NM-R8805, Centre for Mathematics and Computer

Science, Amsterdam, 1988.

32. H. Riesel, *Prime Numbers and Computer Methods for Factorization*, Birkhäuser, Boston, 1985.
33. R. L. Rivest, A. Shamir and L. Adelman, 'A method for obtaining digital signatures and public-key cryptosystems', *Communications of the ACM* **21** (1978), 120–126.
34. R. J. Schoof, 'Quadratic fields and factorization'. *Studieweek Getaltheorie en Computers* (edited by J. van de Lune), Math. Centrum, Amsterdam, 1980, 165-206.
35. D. S. Shanks, 'Class number, a theory of factorization, and genera', *Proc. Symp. Pure Math.* **20**, American Math. Soc., 1971, 415–440.
36. R. D. Silverman, 'The multiple polynomial quadratic sieve', *Mathematics of Computation* **48** (1987), 329–339.
37. H. Suyama, *Informal preliminary report* (8), personal communication, October 1985.
38. M. Voorhoeve, 'Factorization'. *Studieweek Getaltheorie en Computers* (edited by J. van de Lune), Math. Centrum, Amsterdam, 1980, 61–68.
39. D. Wiedemann, 'Solving sparse linear equations over finite fields', *IEEE Trans. Inform. Theory* **32** (1986), 54–62.
40. S. Winograd, 'Evaluating polynomials using rational auxiliary functions', *IBM Technical Disclosure Bulletin* **13** (1970), 1133–1135.

*Computer Sciences Laboratory, Australian National University,*
*GPO Box 4, Canberra, ACT 2601, AUSTRALIA.*

4

# AN OPEN ARCHITECTURE NUMBER SIEVE

A. J. Stephens and H. C. Williams*

The technique of 'sieving' has been known since the time of Eratosthenes, and has been used to solve problems involving systems of linear congruences since the end of the eighteenth century. As a wide range of number theoretic problems can be converted into sieving problems, considerable effort has been expended in constructing machines which are fast enough to solve problems that are otherwise intractable. Most notable is the work of D. H. Lehmer, who pioneered the development of automatic sieving hardware in the 1920's. The latest development in automated sieving is the 'Open Architecture Sieve System' (*OASiS*). The system features a specially-designed computer capable of testing possible solutions to a system of linear congruences at a rate of over 200 million numbers per second. Unlike current sieve devices, the *OASiS* hardware is capable of assuming a variety of configurations, allowing the user to alter the number of congruences being tested in hardware and their size. It also increases the speed at which many sieving problems are solved by automatically optimizing sieving whenever one or more congruences with a single acceptable residue class are present. In this paper, we trace the development of the automatic sieving system from its beginnings to the present day. We also present some results of problems that have been run using *OASiS*, which includes extending the tables of known pseudo-squares and pseudo-cubes, finding periodic continued fractions with long periods, and finding polynomials which have a high density of prime values.

## 1. Introduction.

The technique of sieving has been used to solve mathematical problems since the time of Eratosthenes (fl. 230 B.C.). Although it is a general method that can be applied to a variety of problems in number theory, sieving is also computationally intensive; as a result, its applications are bounded by the rate at which the required operations can be performed. Since the late eighteenth century, a variety of devices have been constructed to increase the speed of sieving, ranging from simple mechanical aids that improve the efficiency of sieving by hand to high speed computer systems that do sieving automatically. In the next few sections, we describe the generalized sieving problem and some of its applications, and trace the development of mechanized sieving over the last 200 years.

Any mathematical problem that requires finding solutions to an arbitrary system of linear congruences involves an instance of the *generalized sieving problem* (*GSP*). In general, a sieving problem $\mathcal{P}$ defines $k$ linear congruences,

$$x \equiv r_{i1}, r_{i2}, \ldots, r_{in_i} \pmod{m_i} \quad (i = 1, 2, \ldots, k,\ 1 \leq n_i \leq m_i),$$

* Research supported by NSERC of Canada Grant #A7649.

where the moduli $m_1, m_2, \ldots, m_k$ are positive integers. It may be assumed that the $m_i$ are relatively prime in pairs and that each set of admissible residues

$$R_i = \{r_{i1}, r_{i2}, \ldots, r_{in_i}\}$$

contains distinct, non-negative integers less than $m_i$. The solution set $S(\mathcal{P})$ for $\mathcal{P}$ is defined to be all $x \in \mathbf{Z}$ that lie within an interval specified by $\mathcal{P}$, say $A \leq x < B$, satisfy all $k$ congruences, and satisfy any additional restrictions placed on $x$ by $\mathcal{P}$. Depending on the precise requirements of the problem, solving $\mathcal{P}$ may entail generating $S(\mathcal{P})$, finding a subset of $S(\mathcal{P})$, or simply determining the number of elements in $S(\mathcal{P})$.

As an illustration of a sieving problem, consider a $\mathcal{P}$ that utilizes the five congruences

$$x \equiv 1 \pmod 8, \quad x \equiv 1 \pmod 3, \quad x \equiv 1, 4 \pmod 5,$$
$$x \equiv 1, 2, 4 \pmod 7, \quad x \equiv 1, 3, 4, 5, 9 \pmod{11},$$

the restriction $x \neq n^2$, and the range $0 \leq x < 3000$. This problem arises naturally from the study of quadratic residues (see Section 7). In this instance, there are two solutions in $S(\mathcal{P})$: 2641 and 2689.

The entries of $S(\mathcal{P})$ are the result of an exceedingly complex function of the numbers $m_i, r_{ij}, A, B$, and the additional restrictions. The simplest algorithm for finding them is to treat each congruence as a 'screen' or 'sieve' which lets through only those integers which lie in the acceptable residue classes. Finding all of the $x$ values that solve the $k$ congruences then involves taking each integer in $[A, B)$, applying it to each of the $k$ sieves in turn, and putting it in $S(\mathcal{P})$ if it passes through all the sieves. Any additional restrictions specified by $\mathcal{P}$ can be imposed by using a further sieve that rejects any value lacking the required properties. This method of solving the problem is what gives it its name.

The sieving method can be implemented in a variety of ways, ranging from a strictly sequential approach to a massively parallel one. Opportunities for parallel processing arise because the final acceptance or rejection of a given $x$ does not depend on the order in which the sieves are applied or on the results of sieving any other value in the search interval. It is theoretically possible to test a given integer against all of the sieves simultaneously, to test all of the integers in $[A, B)$ against a given sieve simultaneously, or both. In practice, the amount of parallelism that can be employed is limited by implementation factors, which means that some form of serial processing must be performed when these limits are exceeded. A typical sieving implementation tests elements of the specified interval in sets of $t$, starting with $A, A+1, \ldots, A+t-1$. Each set of $t$ values is applied simultaneously to as many sieves as possible, and then to the remaining sieves in a sequential manner. If a point is reached where all $t$ in a set of entries have been rejected, the remaining sieves need not be applied. This approach limits the number of values tested in parallel to a feasible number, and also allows the system to handle 'search' problems, in which the upper bound of the interval $[A, B)$ is unknown. Such problems arise frequently in number theory, and usually involve finding the $n$ smallest elements of $S(\mathcal{P})$. Sieving terminates either when the search interval has been completely processed or when $n$ solutions have been found.

The simple, deterministic nature of the sieving process makes it an attractive method for solving the *GSP*. Unfortunately, a comparison of the running time of the algorithm against the size of the problem shows that sieving is not a polynomial time algorithm. The specification of an arbitrary sieving problem that has no additional restrictions on its solutions defines $k$ congruences and a search interval $[A, B)$, and can be represented using

$$\sum_{i=1}^{k} m_i + \log_2 A + \log_2 B$$

bits. If testing a given $x$ value to see if it satisfies a single congruence is considered to be a basic operation, then finding all entries of $S(\mathcal{P})$ may require up to $k(B - A)$ operations to test all of the values in $[A, B)$ against all of the problem's congruences. This is an exponential function of the problem's size. The same is true for search problems, which use an open-ended interval ($B = \infty$). Here, determining $S(\mathcal{P})$ can require up to

$$k \prod_{i=1}^{k} m_i$$

operations. After this point, the pattern of solutions repeats with a period equal to the product of the moduli of the congruences.

The exponential nature of the sieving algorithm means that it is easy to construct problem instances that are too large to be solved within a reasonable amount of time, so it would be nice to find a better way to solve an arbitrary system of congruences. Unfortunately, not only have mathematicians been unable to find a polynomial-time, deterministic algorithm that does this, but not even a polynomial-time non-deterministic algorithm is known! For example, if the claim is made that a particular sieving problem has

$$S(\mathcal{P}) = \{x_1, x_2, \ldots, x_s\},$$

it is easy to verify that the $x_i$ are solutions. However, the only method known for showing that there are no others is to test the remaining values in $[A, B)$ to show that they are not solutions. Thus, the task of verifying a proposed $S(\mathcal{P})$ is just as difficult as generating it by sieving. Similarly, a claim that $y$ is the smallest solution in $S(\mathcal{P})$ can only be verified by testing all of the values less than $y$.

The failure to find a computationally efficient method for solving the *GSP* is a result of its intrinsic difficulty. Patterson [Pat89] has recently shown that the NP-complete problem of 'quadratic congruences' [MA78] can be transformed into a sieving problem, demonstrating that the *GSP* is NP-hard—that is, at least as hard as the problems in NP. Thus, it will probably be some time before the 'brute force' technique of sieving will be replaced by a better algorithm for solving arbitrary sieving problems. Since there is currently no good method for verifying the solutions to a sieving problem, there is no proof that the *GSP* even lies in NP at all. It may lie in a class of problems that are even more difficult than NP, making the task of finding an improved algorithm even harder.

Even though the sieving method is an exponential algorithm, it does not mean that sieving cannot be used to solve non-trivial sieving problems. From a human standpoint,

a sieving problem with the interval $[0, 2B)$ can be considered twice as difficult as the same problem covering $[0, B)$ because there are twice as many possible solutions; the same is true if $2k$ congruences are involved instead of $k$ congruences, since there are twice the number of restrictions on its solutions. If this 'human' notion of problem size is used, rather than the normal 'bit counting' approach, then the time required to solve a problem by sieving is directly proportional to its size. This, along with the fact that the simple, deterministic nature of sieving allows it to be implemented efficiently, means that sieving can be utilized to solve many significant problems in number theory.

## 2. Applications of sieving.

The sorts of sieving problems that arise in number theory are quite varied in nature, and several date back many centuries. One is related to the well-known *Sieve of Eratosthenes* (see [Dic19]) and uses the first $k$ primes $p_1, p_2, \ldots, p_k$ as moduli, while its residue sets, $R_i$, exclude only zero. Such a system of congruences removes only multiples of the primes used, so sieving the interval $[p_k + 1, p_{k+1}^2)$ generates all primes lying in that range. For example, the system of congruences obtained using $k = 4$ is

$$x \equiv 1 \pmod 2,\ x \equiv 1,2 \pmod 3,\ x \equiv 1,2,3,4 \pmod 5,\ x \equiv 1,2,3,4,5,6 \pmod 7,$$

and produces the primes from 11 to 113 when the interval [8,121) is sieved.

At the opposite end of the spectrum is the *Chinese Remainder Problem*, in which only one residue class is acceptable for each congruence. Such problems have exactly one solution modulo $m_1 m_2 \cdots m_k$. For example, the system

$$x \equiv 1 \pmod 2,\ x \equiv 2 \pmod 3,\ x \equiv 3 \pmod 5,$$

has the single solution $x \equiv 23 \pmod{30}$. The *CRP* is notable because it is one of the few classes of sieving problem which can be solved by a means other than sieving. One simple approach repeatedly merges pairs of congruences by means of the Euclidean Algorithm until only one congruence remains (see Section 4.3.2 of [Knu81]).

A much larger class of sieving problems, and one which is of considerable interest to mathematicians, involves finding integers $x$ and $y$ which solve $f(x, y) = 0$, where $f$ is a polynomial of arbitrary degree with integer coefficients of arbitrary size. The 'method of exclusion' of Gauss allows us to convert this Diophantine equation to a set of necessary (but not sufficient) sieve conditions which can be solved for $x$ by sieving. The solutions to the original equation are then found among the few $x$ values that remain instead of the entire solution space of possible $x$ and $y$ values. The lack of restrictions on $f$ allows this technique to be applied to a wide range of problems in number theory. A brief list of such problems has been given by Lehmer [Leh66] and includes:

- finding the representations of a large number by a given binary quadratic form, for example $N = x^2 - y^2$,
- finding the binomial units of a given algebraic number field,
- finding the solutions (or the number of solutions) for a given polynomial that lie between given limits.

As an illustration of this technique, a method of factoring an arbitrary large integer is given below. Although based on Fermat's idea of writing the integer as the difference of two squares [Dic19], the actual algorithm was developed by Gauss ([GC66], articles 319–325). A more complete discussion of both approaches is given by Brillhart [Bri81].

Suppose that we wish to factor an odd integer, $n > 0$. If $n$ is composite, then it must be the product of two odd factors, say $n = UV$, where $U \geq V$. If we define

$$x = \tfrac{1}{2}(U + V), \quad y = \tfrac{1}{2}(U - V)$$

and solve for $U$ and $V$, we find

$$U = x + y, \quad V = x - y$$

and thus,

$$n = (x + y)(x - y) = x^2 - y^2 \quad \text{or} \quad y^2 = x^2 - n.$$

Therefore, if we can find an $x$ value for which $x^2 - n$ is a perfect square, we can calculate the factors $U$ and $V$ of $n$.

Suppose we have an $x$ such that $y^2 = x^2 - n$. Then $y^2 \equiv x^2 - n \pmod{p}$ for any $p \in \mathbf{Z}$, meaning $x^2 - n$ must be a quadratic residue modulo $p$. If $p$ is an odd prime which does not divide $n$, only $\frac{1}{2}(p + 1)$ of the $p$ residue classes can contain $x^2 - n$. Consequently, roughly half of the residue classes modulo $p$ cannot possibly contain $x$. We therefore construct

$$x \equiv r_1, r_2, \ldots, r_k \pmod{p}, \quad k = \tfrac{1}{2}(p + 1),$$

which specifies the residue classes in which $x$ might be found, and start sieving. The sieving does not guarantee that each $x$ found will lead to an $x^2 - n$ that is a perfect square, but helps to exclude many values that do not.

The chances of finding a suitable $x$ is improved greatly by applying this approach to a number of different $p$ values. This results in a set of linear congruences, each of which excludes roughly half of its residue classes. If $k$ congruences are used, only about one in $2^k$ numbers will make it through the sieving process to require the test to see if $x^2 - n$ is a perfect square. Since finding an $x$ whose corresponding $x^2 - n$ is a quadratic residue of all $k$ primes but is not a perfect square is relatively unlikely, most $x$ values that solve the congruences allow us to factor $n$ once $k$ becomes large. The effectiveness of the sieving can be improved significantly by forming congruences modulo $p^2$ rather than modulo $p$ since this results in a lower proportion of acceptable residue classes.

The interval to be searched can be bounded easily. We can arbitrarily restrict $x > 0$, since if $x^2 - y^2 = n$, then $(-x)^2 - y^2 = n$ as well. Thus, $x \geq \lceil \sqrt{n} \rceil$. As $n$ is odd, we know that the second term in the factorization $(x + y)(x - y) = n$ must be at least 3, meaning the first term (and therefore $x$) must be $\leq \lfloor n/3 \rfloor$. If this range is exhausted without finding a factorization of $n$, then $n$ is prime.

As an example of this procedure, let us factor $n = 5917$ using the primes 3, 5 and 7. We first construct a table to determine the residue classes to be searched.

| $p$ | $x \pmod{p}$ | $x^2 \pmod{p}$ | $n \pmod{p}$ | $x^2 - n \pmod{p}$ |
|---|---|---|---|---|
| 3 | 0,1,2 | 0,1,1 | 1 | 2,0,0 |
| 5 | 0,1,2,3,4 | 0,1,4,4,1 | 2 | 3,4,2,2,4 |
| 7 | 0,1,2,3,4,5,6 | 0,1,4,2,2,4,1 | 2 | 5,6,2,0,0,2,6 |

The underlined values in the final column of the table signify the residue classes which cannot contain perfect squares. The set of congruences to be used is therefore

$$x \equiv 1, 2 \pmod 3,\ x \equiv 1, 4 \pmod 5,\ x \equiv 2, 3, 4, 5 \pmod 7.$$

We sieve with $A = \lceil\sqrt{5917}\rceil = 77$ and $B = \lfloor 5917/3 \rfloor + 1 = 1973$. The smallest solution to the congruences is found at $x = 79$. Since $79^2 - 5917 = 324$ and $\sqrt{324} = 18$, this immediately gives us the factorization $5917 = 61 \times 97$ since $U = 79 + 18 = 97$ and $V = 79 - 18 = 61$. Note that the next highest solution to the congruences, $x = 86$, does not lead to a factorization since $86^2 - 5917 = 1479$, which is not a perfect square.

The factoring algorithm is typical of applications of sieving—its usefulness is entirely dependent on the ability of the user to find solutions to a set of congruences quickly. When an efficient method exists, 15 digit numbers can be factored in less than a day even though the algorithm's running time is $O(n)$. To factor values of up to 25 digits, a modified version of the factoring algorithm can be used whose running time is $O(\sqrt{n})$ [LL74]. Although much better factoring algorithms now exist that under suitable heuristics are of complexity $O(\exp(\sqrt{\log n \log\log n} + o(1))$ [Pom89], until 1970 the most efficient means known of factoring an arbitrary large integer was to use a sieving approach [MB75].

## 3. The development of sieve automation.

Mathematicians have used the technique of sieving as a tool for solving problems in number theory for over 200 years. During that time, a variety of mechanisms have been developed to increase the speed and accuracy of the sieving process. Unfortunately, those who utilized sieving often published the results of their work with little or no description of the methods they used to obtain them. Some may have felt that little could (or should) be said about a technique that amounted to little more than a brute force search; others may have found the effort of publishing their methods too great to bother with given the limited interest in sieving among mathematicians and computing specialists. Whatever the reason, the result is that much is unknown about many of the sieving systems used, and what is known has been pieced together from a variety of sources, some of which give conflicting accounts. The next few sections amalgamate this material into a single account that highlights the most significant developments in automated sieving.

The earliest attempt to apply a mechanical technique to solving the *GSP* was made by Legendre in 1794 [Rub83]. While his *strip method* increased the speed and accuracy of sieving, it was still a rather slow and inconvenient manual technique. Even so, it remained the best method for solving a *GSP* until well into the 20th century.

It is possible that the inspiration for Legendre's invention came from an earlier application of mechanical devices to sieving: the construction of *factor tables*. Such tables list the prime factors, or in some cases just the smallest prime factor, for every integer in a specified range, and were widely used by mathematicians in the pre-computer age. Not only did such a table provide a rapid test for properties directly related to the factorization of a number, such as primality or the presence of square factors, but it was often helpful in ordinary mathematical calculations, say, allowing the user to obtain a more accurate logarithm for a number by summing the logarithms

of its factors. Numerous factor tables were constructed between the 17th and 20th centuries, starting in 1659 with a table by Rahn (or Rhonius) extending up to 24,000; a list of early factor tables and their formation is described by J. W. L. Glaisher [Gla78].

The creation of a factor table is a relatively simple process that is based on the sieve of Eratosthenes [Leh18]. All of the integers in the desired interval are written down in a line, then a '2' is written underneath every even number, a '3' under every value divisible by 3, and so on; the table is complete when all primes less than the square root of the upper bound of the table have been processed in this manner. The table can be reduced in size by about 75% by arranging the numbers in rows of 30 and deleting the 22 columns containing the values divisible by 2, 3, or 5; no significant information is lost since any number divisible by these factors can be spotted easily without using the table.

Although a number of early factor tables were created entirely by hand, the fact that multiples of a prime $p$ are separated by exactly $p$ units means that the process of sieving out multiples is very regular. The first to take advantage of this was C. F. Hindenburg, who in the mid-1770s utilized a strip of thick paper with a regular pattern of holes (patrone) to aid in identifying the multiples of $p$. A machine containing a series of rods was also used, and is described by Bernoulli [Ber85]. Hindenburg's table was never published. At roughly the same time, Anton Felkel independently utilized the same approach in beginning the construction of a table that was to extend up to ten million. The Austrian government financed the publication of the portion up to 408,000 in 1776, but when it sold poorly (probably due to its unorthodox layout) the entire edition was scrapped to make cartridges for use in the war against the Turks. Only a few copies were saved. Later mathematicians used mechanical aids to produce tables that were more successful. Burckhardt used paper stencils to generate a table of factors for the integers up to three million, which he published in 1817. By 1883, similar tables by Glaisher[1] and Dase had continued the work up to nine million [Leh18]. Finally, in 1909, D. N. Lehmer produced a factor table (and a smaller table of primes) that extended up to ten million, reaching Felkel's goal over 130 years after it was first proposed.

Beginning in 1924, Lehmer used a different form of sieving to factor large integers: the *factor stencil* [LE39]. The technique is based on the principle that if a given integer $R$ is a quadratic residue of $N$, then it is also a quadratic residue of all of the factors of $N$. Each stencil represented a different $R$, and had 5000 cells corresponding to the first 5000 primes. A hole was punched in a given cell only if $R$ was a quadratic residue for the associated prime. Factoring was accomplished by finding a small set of quadratic residues for $N$ and superimposing the stencils for those $R$; only if all cells in a given position contained a hole could the corresponding prime possibly divide $N$. Trial division could then be used to show whether any of these primes were factors. For a 10 digit number, less than a dozen quadratic residues were usually sufficient to cut the number of possible factors down to 2 or 3. Although more complicated than referring to a factor table, the factor stencil method had an important advantage: it still gave useful information even if the number to be factored was larger than the stencils were designed to handle. Lehmer's first set of stencils were constructed largely

[1] The father of J. W. L. Glaisher.

by hand and only a few sets were made, but in 1939 a revised and extended version were issued on Hollerith cards by Elder; these could be mechanically reproduced and were widely distributed by the Carnegie Institution.

Despite the increases in speed and accuracy that these approaches gave, sieving remained a cumbersome technique. One problem was that the work was still done by hand and remained a relatively slow and error-prone process. As well, it was necessary to decide on the limits of the interval to be processed before sieving began. This was no drawback for problems such as the creation of factor tables since the entire interval was always processed regardless of the results obtained, but it presented a major handicap when applied to the *GSP*. If, for example, the smallest value satisfying a system of congruences lay near the start of the search interval, the entire interval was sieved using all but the last congruence before the solution was discovered; all of the effort involved in sieving the high end of the interval was wasted. It was not until the advent of the first automatic sieving machine that these difficulties were eliminated and sieving became practical on a large scale.

The periodic behaviour of the sieving process naturally lends itself to the construction of a machine that can solve the *GSP*. Consider the task of finding the non-negative solutions to an arbitrary set of $k$ linear congruences

$$x \equiv r_{ij} \pmod{m_i} \quad (i = 1, 2, \ldots, k,\ 1 \leq j \leq n_i < m_i).$$

The solutions can be found by constructing a table of $m_1 m_2 \ldots m_k$ columns numbered from 0 to $m_1 m_2 \ldots m_k - 1$. For each congruence, add a row to the table which indicates which values lie in the set of acceptable residue classes, $R_i$. Any column of the table which satisfies all of the congruences represents a solution to the congruences.

| | | | $x$ | 0 1 2 3 4 5 6 7 8 ... |
|---|---|---|---|---|
| (a) | $x \equiv 0$ | (mod 2) | mod 2 | $\checkmark_0 \times_1 \checkmark_0 \times_1 \checkmark_0\ \times_1 \checkmark_0 \times_1 \checkmark_0 \ldots$ |
| | $x \equiv 1, 2$ | (mod 3) | mod 3 | $\times_0 \checkmark_1 \checkmark_2 \times_0 \checkmark_1\ \checkmark_2 \times_0 \checkmark_1 \checkmark_2 \ldots$ |
| | $x \equiv 0, 2, 3$ | (mod 5) | mod 5 | $\checkmark_0 \times_1 \checkmark_2 \checkmark_3 \times_4\ \checkmark_0 \times_1 \checkmark_2 \checkmark_3 \ldots$ |
| | | | $x$ | 0 |
| (b) | $x \equiv 0$ | (mod 2) | mod 2 | $\checkmark_0 \times_1$ |
| | $x \equiv 1, 2$ | (mod 3) | mod 3 | $\times_0 \checkmark_1 \checkmark_2$ |
| | $x \equiv 0, 2, 3$ | (mod 5) | mod 5 | $\checkmark_0 \times_1 \checkmark_2 \checkmark_3\ \times_4$ |
| | | | $x$ | 3 |
| (c) | $x \equiv 0$ | (mod 2) | mod 2 | $\times_1 \checkmark_0$ |
| | $x \equiv 1, 2$ | (mod 3) | mod 3 | $\times_0 \checkmark_1 \checkmark_2$ |
| | $x \equiv 0, 2, 3$ | (mod 5) | mod 5 | $\checkmark_3 \times_4 \checkmark_0 \times_1 \checkmark_2$ |

Figure 1. Principles of Sieve Automation.

Figure 1 (a) shows a portion of a table for an example involving three congruences; the solutions at 2 and 8 are readily apparent. In practice, this sort of table is too large to construct, but since each row of the table exhibits a cyclic pattern corresponding to the list of acceptable and forbidden residue classes for its associated congruence, the information in the table can be compressed to a small fraction of its original size by recording the pattern of residues only once (Figure 1 (b)). Any column of the

original table can be generated by shifting each of the rows to the left to advance the appropriate set of residues to the first column; the leftmost residue wraps around to the rightmost end. For example, three shifts generates column 3 of the original table (Figure 1 (c)). In general, column $t$ can be obtained by performing $t$ shift operations.

From here, it is easy to see how a machine that performs sieving can be constructed. Each congruence used in the problem is represented by a loop, or *ring*, whose size corresponds to its modulus. The ring for congruence $i$ is built with $m_i$ positions, and tags are placed in those slots representing acceptable residue classes. The machine examines one position from each ring at a fixed location called the *tap*, or *window*, and simply advances the rings in unison until all of the rings present an acceptable residue simultaneously. A *trial counter* records the number of shifts performed by the machine, and is used to determine the value of each solution found. The machine can be set up to start searching from any point $A$ by setting ring $i$ so that position $A \pmod{m_i}$ is in the window when the machine is turned on; after $s$ shifts, the value being tested is $A + s$. Any machine of this type is known as a *sieve*.

An *automatic sieving system* consists of a sieve, a means of loading problems into the sieve, and a means of recording the solutions found by the sieve. The recording stage does not necessarily have to note the exact value of each solution, since some problems only require that the number of solutions be determined; in such cases, the recording stage can simply increment a counter each time a solution appears. The loading and recording parts of the system may be implemented by machines or by a human operator; the only portion of the system that must be entirely automatic is the sieving process itself. The construction of a completely automatic sieving system that can run problems without human intervention is a relatively recent development in the history of automated sieving.

The sieve design provides the user with a fast and accurate means of solving the *GSP*. Since the search is entirely automatic, it can be performed at high speed for long periods without error. By altering the placement of tags on the rings, the sieve can be loaded with any system of congruences which corresponds to its collection of rings. The finite size of the device does mean that some sets of congruences cannot be implemented by the sieve completely, but this apparent drawback can be overcome with little difficulty in most common problems. The application of such a machine to sieving also overcomes the flaws of hand sieving mentioned previously: it is efficient and, since it simultaneously tests each solution candidate against all of the congruences rather than each congruence against all of the solution candidates, solutions that occur near the beginning of the search interval are discovered quickly.

The performance of the basic sieve model can be increased significantly by testing multiple solution candidates in parallel. There are two ways of increasing the sieving rate by a factor of $t$: by using $t$ sieves or by constructing a single sieve with a $t$ position solution window that tests $t$ consecutive positions from each ring and then advances the rings by $t$ positions. This multi-tap approach is relatively cheap to implement since it is basically a single tap sieve with additional solution detection hardware; in contrast, a set of single tap machines involves the duplication of the complete sieve.

A dramatic increase in speed can also be realized at absolutely no cost by simply relabeling ring positions so that the sieve tests only values that satisfy any congruences

with a single acceptable residue class. For example, if $x \equiv a \pmod{m}$ is specified by the problem, then ring $i$ can be labelled so that consecutive positions differ by $m \pmod{m_i}$ rather than one; the sieve then skips over the $m-1$ values not congruent to $a \pmod{m}$ between tests, increasing performance by a factor of $m$. Figure 2 shows an example in which this technique increases the rate of sieving by a factor of three. For each ring where $m_i \neq m$, the optimization simply permutes the standard arrangement of residue classes within the ring. However, for a ring in which $m_i = m$, the optimization fills the ring with the single acceptable residue class, $a$, meaning the ring does not have to be tested at all.

| | | | $x$ | 0 1 2 3 4 5 6 7 8 ... |
|---|---|---|---|---|
| (a) | $x \equiv 1$ | (mod 3) | mod 3 | $\times_0\checkmark_1\times_2\times_0\checkmark_1\ \times_2\times_0\checkmark_1\times_2$ ... |
| | $x \equiv 2,3$ | (mod 5) | mod 5 | $\times_0\times_1\checkmark_2\checkmark_3\times_4\ \times_0\times_1\checkmark_2\checkmark_3$ ... |
| | $x \equiv 0,2,3$ | (mod 7) | mod 7 | $\checkmark_0\times_1\checkmark_2\checkmark_3\times_4\ \times_5\times_6\checkmark_0\times_1$ ... |
| | | | | standard ordering of residues |
| | | | $x$ | 1 4 7 10 13 16 19 22 25 ... |
| (b) | $x \equiv 1$ | (mod 3) | mod 3 | $\checkmark_1\checkmark_1\checkmark_1\checkmark_1\checkmark_1\checkmark_1\checkmark_1\checkmark_1\checkmark_1$ ... |
| | $x \equiv 2,3$ | (mod 5) | mod 5 | $\times_1\times_4\checkmark_2\times_0\checkmark_3\ \times_1\times_4\checkmark_2\times_0$ ... |
| | $x \equiv 0,2,3$ | (mod 7) | mod 7 | $\times_1\times_4\checkmark_0\checkmark_3\times_6\ \checkmark_2\times_5\times_1\times_4$ ... |
| | | | | optimized ordering of residues |

Figure 2. Single Residue Optimization.

For problems containing more than one single residue congruence, the optimization technique can be repeated using each congruence to increase the sieving rate by a factor equal to the product of the moduli involved. The easiest method of implementing such an approach is to combine the single residue congruences and use the resulting congruence as the basis for reordering the remaining congruences. Optimization can also be performed using a congruence of the form $x \equiv a_1, a_2, \ldots, a_r \pmod{m}$ by partitioning the original problem into $r$ sub-problems involving a single residue classes modulo $m$. This produces an increase in overall sieving speed of $m/r$, but incurs the overhead of running $r$ problems.

Automatic sieving systems are frequently required to run sieving problems that cannot be implemented by the system hardware, either because the sieve's rings cannot hold all of the problem's congruences or because the problem specifies restrictions on its solutions in addition to those imposed by the congruences. Any such problem $\mathcal{P}$ can still be solved by partitioning it into two subproblems: $\mathcal{P}_r$, the problem that defines only the congruences of $\mathcal{P}$ that can be loaded into the sieve's rings, and $\mathcal{P}_s$, the problem that defines the remaining congruences and any additional restrictions. Since all solutions to $\mathcal{P}$ are also solutions to any problem $\mathcal{P}'$ formed by removing one or more congruences or additional restrictions from $\mathcal{P}$, $S(\mathcal{P})$ can be found by taking the intersection of $S(\mathcal{P}_r)$ and $S(\mathcal{P}_s)$. In practice, it is not necessary to determine $S(\mathcal{P}_s)$; instead, one can generate $S(\mathcal{P}_r)$ using the sieve hardware and then test each solution individually to see if it also lies in $S(\mathcal{P}_s)$. This latter step is termed *solution filtering*.

A sieving system can implement filtering in two ways. The simplest method is to have the system solve $\mathcal{P}_r$ in the normal fashion and then test the elements of $S(\mathcal{P}_r)$ afterwards; the advantage of such *off-line* filtering is that it can be done on any automatic sieving system. Alternatively, the system can test each element of $S(\mathcal{P}_r)$ as

it is produced by the sieve hardware. While it requires more effort to implement, *on-line* filtering is essential if $\mathcal{P}_r$ generates a large number of solutions or if $\mathcal{P}$ is bounded by the number of solutions to be found rather than the end of a search interval.

## 4. Automatic sieving systems.

Since the early part of the 20th century, a variety of automatic sieving systems have been developed, ranging from relatively slow and unwieldly mechanical devices to electronic computer systems that are over a million times faster (see Table 1). At the same time, significant (although less spectacular) gains have also been made in terms of system flexibility and the amount of human intervention required to solve a problem. All of these systems have been based on the model just described, and differ only in the hardware used to implement the sieve. This evolution has been fueled largely by developments in technology, with a new system being built whenever the hardware of the day could be adapted to produce a better sieve. Unfortunately, the relatively esoteric nature of sieving has meant that the evolutionary process has tended to proceed in an erratic fashion; the few systems that have been constructed were usually the product of a few dedicated individuals working on a part-time basis with little or no support. Consequently, this account relates many failures and few total successes.

| Machine | Year | Rings | Trials/sec |
|---|---|---|---|
| Bicycle chains | 1926 | 19 | 50 |
| Photoelectric gears | 1932 | 30 | 5,000 |
| 16mm movie film | 1936 | 18 | 50 |
| SWAC* | 1950's | ? | 1,450 |
| IBM 7094* | 1960's | 21 or 22 | 150,000 |
| DLS-127 | 1965 | 31 | 1,000,000 |
| DLS-157 | 1969? | 37 | 1,000,000 |
| ILLIAC IV* | 1970's | 64 | 15,000,000 |
| SRS-181 | 1975 | 42 | 20,000,000 |
| UMSU | 1983 | 32 | 133,000,000 |
| OASiS | 1989 | 16 | 215,000,000 |
| Sun 4/280* | 1989 | 32 | 2,000,000 |

* Denotes a general purpose computer running special software.

Table 1. Automatic sieving systems.

The first proposal for an automatic sieving system seems to have been made by Maurice Kraitchik [Kra22], who suggested building a sieve made up of gears in 1922. Fifteen gears representing congruences for the first 15 primes (or small multiples) would rotate freely along a common axle; each gear would have one tooth per residue class. A second set of 15 gears, each containing 64 teeth, would be fixed to a second axle mounted in parallel with the first. By connecting this axle to a motor, the latter gears would cause the former to advance at a uniform rate of $n$ teeth per second. Kraitchik suggested placing electrical contacts at the gear positions corresponding to acceptable residues, allowing the machine to detect when all of the gears presented an acceptable residue simultaneously and stop automatically. Alternatively, holes could be drilled in

the gears and a light source placed at one end of the machine; if the light arrived at the other end, a solution was present. In either case, the user would inspect a counter that recorded the number of positions the gears had advanced and manually calculate the solution found.

Kraitchik did not speculate on how fast his machine would operate, but it would undoubtably have been much faster than the best stencil or strip methods then available. Although his proposal discussed a number of the important details, it was only an outline, not a full-fledged design, and no such machine was ever constructed. D. H. Lehmer has called the design 'impractical', questioning whether the available technology would have been capable of manufacturing the required gears and whether in fact the two sets of gears would have meshed properly [Leh89]. Nevertheless, it remains the first significant attempt to automate the sieve process.

The first workable sieve was constructed by Derrick Henry Lehmer in 1926 at the Berkeley campus of the University of California [Leh28], [Leh80]. Lehmer was a freshman at the time, and decided to build a machine after becoming frustrated when the series of 20 foot paper strips he was using to do sieving kept becoming entangled.

The sieve's rings consisted of 19 bicycle chains, each forming a loop that was draped over a 10 tooth sprocket. The sprockets were attached to a common axle which was turned by an electric motor. A small pin (actually a chicken wire staple) was placed on each link corresponding to an acceptable residue. Whenever a pin reached the top of the loop, it lifted a small spring and broke an electrical contact. The contacts for the rings were wired in series and connected to a relay that cut the power to the motor if all of the contacts were lifted simultaneously; the machine then coasted to a stop. A 12 ring version of the device is outlined in Figure 3.

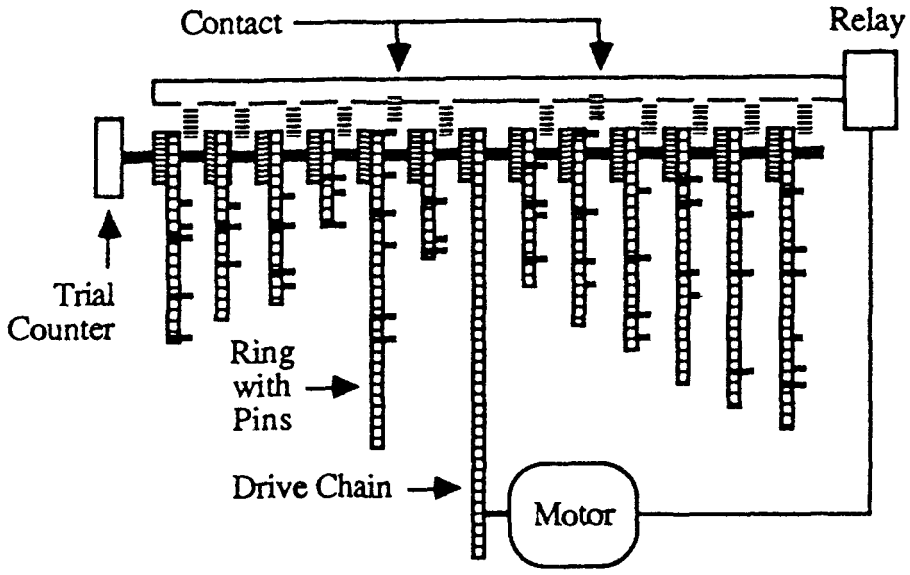



Figure 3. The bicycle chain sieve.

Lehmer personally constructed the sieve in a few weeks, spending about $20 of his own money. The final product was capable of testing about 50 values per second. Although it was possible to go faster, the free hanging chains would start slipping once the sieve reached 60 trials per second. The soft wooden bearings used in the device also caused problems: they wore out rapidly and required constant tinkering. Even

when the hardware was working, the sieve was not easy to use; setting up the rings was a cumbersome task requiring one to two hours, and when the machine found a solution and stopped, it remained stopped until its operator returned, backed the rings up to locate the solution, and restarted the search. Consequently, only one or two problems could be run in a week.

Despite these difficulties, the sieve was used on 50 to 100 problems over a period of about 8 months. Results included the factoring of previously intractible large numbers and the determination of some pseudo-squares [Leh28]. The machine was later disassembled by Lehmer and transported to Providence, R.I., but was stolen before it could be used again.

The success of the bicycle chain sieve inspired Lehmer to attempt a more ambitious device in 1932 [Leh33b], [Leh34], [Leh80]. Drawing from Kraitchik's 1922 proposal, Lehmer made several important modifications to the design to make the concept workable. As before, the congruences were represented by $n$ gears corresponding to the first $n$ primes, with each gear having a different size and containing one tooth per residue class. However, the gears were mounted on individual axles and were advanced at a uniform rate by a set of identical gears mounted on a motorized axle. A set of holes corresponding to each tooth was drilled in each gear at a fixed distance from its circumference; each hole was plugged up with a piece of a toothpick if it represented a residue that was not acceptable for the problem being run. A beam of light shone through the holes at a fixed position; if it was detected by a photocell at the other end, a solution had been found and the machine stopped. An outline of a 12 ring version of the photoelectric sieve is shown in Figure 4; the actual device contained 30 rings. Note how a pair of prisms were used to fold the path of light beam; this made the machine more compact and allowed each section of the drive train to turn two rings instead of one.

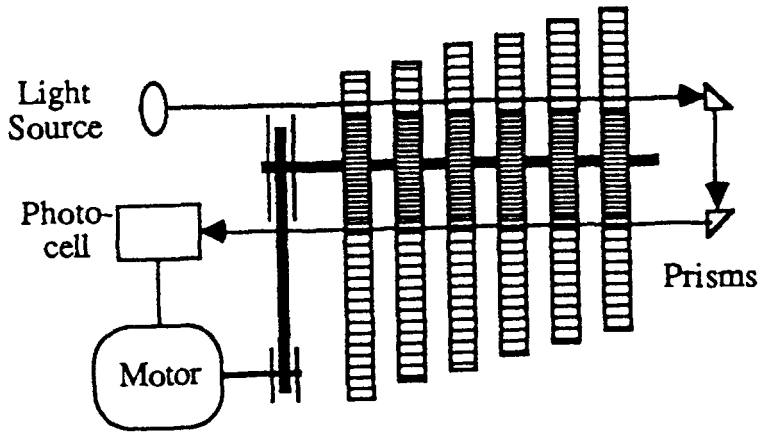



Figure 4. The photoelectric sieve.

Designed to run at 5000 trials per second, the new sieve was two orders of magnitude faster than the old one and considerably more complex. Its construction was a major endeavour lasting about 7 months and involving a variety of groups. Through the aid of his father, Lehmer was able to obtain a $1000 grant from the Carnegie Institution to build the machine. Lehmer had the Johnson Gearworks in West Berkeley put teeth onto the gear blanks, then drilled the residue holes himself. The other mechanical

parts were made at Stanford University with the help of T. J. Palmateer. A three-stage photoelectric amplifier was built at Burt Scientific Laboratories in Pasadena; it amplifed the light signal by a factor of over 700 million, and was extremely difficult to construct. The complete sieve was assembled at the student lab at the University of California, but failed to work; a complete overhaul at Burt Labs was necessary before the bugs were finally worked out.

Like its predecessor, the photoelectric sieve was difficult to work with. Setting up a single congruence required the user to fill all the holes in each gear and then knock out the unneeded pieces of toothpick using dental instruments; creating a complete problem took over two hours. The rebuilt amplifier was so sensitive that it was set up in an adjacent room and heavily shielded with copper to prevent electrical interference from triggering the sieve to stop. Such precautions were not always enough: a nearby ham radio operator who normally broadcast between 10 and 11 p.m. often prevented the sieve from being used during that period.

During a two month period at Burt Labs, the sieve was used to aid in factoring numbers of the form $b^n \pm 1$; some results can be found in [Leh33c] and [BLSTW88]. The device was something of a sensation at the time, with glowing accounts of its design and implications printed in newspapers [Kae33], [Oak33] and periodicals [Car33], [Leh33a]. Lehmer was soon asked to exhibit the sieve at the Century of Progress Exposition in Chicago; unemployed at the time, his initial reservations were overcome when the fair agreed to hire him as a demonstrator. Due to the nature of the demonstrations, no significant work was done while the machine was on display. Nor was it used in the years that followed. The formidable task of setting it up and using it apparently outweighed its great potential. An attempt to build a better amplifier was made at one point, but it was never finished. The machine languished in storage for a number of years before finding a permanent resting place at the ACM Computer Museum in Boston.

The fate of the photoelectric sieve clearly demonstrated that the world's fastest sieve was useless if no one could be persuaded to use it. Consequently, ease of use became as important a part of sieve design as raw speed, and greatly influenced the design of Lehmer's next machine in 1936 [Leh80].

The new sieve was a simple variation of the first design, with bicycle chains replaced by loops of 16mm movie film leader. As before, the loops were draped over a common shaft, but a roller lubricated with talcum powder was placed at the bottom of each loop to ensure that the film holes remained firmly seated on the drive sprockets, since the weight of the film itself was insufficient for the purpose. A quarter inch hole was punched in a loop to indicate an unacceptable residue class. Solutions were detected by resting a metal screw on top of each film loop and applying an electric current; if the circuit to the axle was blocked by all 18 film loops simultaneously, a relay was tripped and the machine stopped.

Like the bicycle chain sieve, the movie film sieve was simple and cheap: Lehmer built the machine at his home in Pennsylvania over a two month period, spending about $50 for parts. Although it ran at only 50 trials per second, the sieve was quiet and required only about 30 minutes to set up a problem; consequently, it was frequently used. Since the film loops wore out after about ten hours of use, the sieve was normally

put to work on small problems that required only one or two hours of running time.

When Lehmer returned to Berkeley he took the machine with him, where it was used until 1941. One of D. N. Lehmer's students, D. Swift, obtained a Ph.D. by using the sieve to run problems involving binary quadratic forms. It also examined values congruent to 1 and 3 (mod 4) to determine which class contained more primes. Unlike the two earlier machines, the movie film sieve was never used for factoring or primality testing purposes. It is now on exhibit at the Computer Museum.

The advent of digital computers in the latter half of the 1940's led Lehmer to experiment with the idea of programming a general purpose computer to do sieving. Not only was it far cheaper and easier to use existing hardware than to construct a specialized 'one off' device, but the programmable nature of the machines made the task of switching from one problem to another a relatively simple task.

Lehmer's first attempt at sieving on a computer utilized the world's first all-electronic digital computer, ENIAC, over the July 4th weekend in 1946 ([Leh49], [Leh74]). Although the results of this work were used to factor certain numbers of the form $2^n \pm 1$ [Leh47], it was not really a software equivalent of the earlier sieves. Instead of solving the *GSP*, it found the smallest positive $n$ that solved the single congruence $2^n \equiv 1 \pmod{p}$ by calculating $2^n \pmod{p}$ for $n = 1, 2, \ldots, 2000$. Sieving was utilized to generate the values of the primes $p$, and eliminated any candidate prime which had a factor $\leq 47$.

In the early 1950's, Lehmer programmed the Standards Western Automatic Computer (SWAC) to solve the *GSP* [Leh53]. Strings of bits were used to represent the congruences, with a '0' bit indicating an acceptable residue. The computer would merge sets of 36 residues from each ring and check for the occurrence of solutions, making the SWAC the first sieve to use multiple solution taps. A single test cycle required the machine to merge $k$ 36-bit string segments in a serial fashion, check for solutions, and perform a circular shift on the $k$ bit strings one by one; as a result, a large number of machine instructions were executed for every 36 values tested. However, since each instruction took very little time, the sieve program ran at about 1450 trials per second—about 30 times faster than the movie film sieve and almost 30% of the speed of the cumbersome photoelectric sieve. Lehmer used this speed to extend the pseudo-square search done using his first sieve [Leh54], as well as working on a variety of other problems.

In later years, sieving programs were implemented on other computers by a number of people. As available memory increased, sieving speed could be increased by combining sets of two or more congruences into a single larger congruence, thereby decreasing the number of bit strings to be merged. This technique was used by John Brillhart to compress 21 or 22 congruences into 10 or 11 bit strings, allowing an IBM 7094 to sieve at a rate of 150 000 trials per second [BS67]. A variety of new factorizations were discovered using this program [BLSTW88]. More recently, Michael Hermann and Cam Patterson bench-marked a 32 ring sieve on a Sun 4/280 computer at 2 000 000 trials per second [HP89].

In an effort to reach higher sieving rates than were possible with conventional computers, Lehmer returned to the technique of building special purpose sieving machines. With the assistance of Paul Morton of the Electrical Engineering department

at the University of California, Berkeley, he first attemped to construct a sieve out of switches and counters [Leh80]. Each counter cycled through a row of switches representing the residue classes for a congruence, routing the setting of each in turn to the solution detection circuitry. Thirty rows of switches provided congruences based on the first 30 primes, and utilized about 2000 switches in all. One of the major benefits of this approach was the ease with which a problem could be loaded. Unfortunately, the counter sieve never functioned reliably and was eventually abandoned after two years of effort.

Lehmer and Morton next built the Delay Line Sieve, DLS-127, in 1965. Although no formal description of the machine was ever given, aspects of its construction and operation are mentioned in [Leh66], [Leh68], and [Leh80]. The sieve had 31 rings, representing the primes from 2 to 127—hence its name. Each congruence was implemented by a segment of delay line, a form of wire that propogates an electrical pulse at a fixed speed. By forming a precisely measured loop and initializing it with a sequence of pulses representing residue classes, the pulses would circulate past a single solution tap and feed into a solution detection circuit. The delay line's high speed allowed the sieve to test one million values every second.

In addition to its record breaking speed, the DLS-127 displayed a number of important design innovations that greatly increased its versatility. Unlike the earlier hardware sieves, the task of setting up a problem on the DLS-127 was highly automated. A program running on an IBM 7094 allowed the sieve user to create a sieve problem based on any Diophantine equation $f(x, y) = 0$ by simply entering the coefficients of the polynomial, $f$; the resulting problem specification was punched into cards which were then transferred to paper tape. When one problem terminated, the next was read into the sieve from tape in a matter of seconds. Solution processing was also automated. Each time a solution was detected, the sieve shifted into 'idle mode', printed the solution value, and resumed sieving automatically; thus, the sieve could find multiple solutions to a problem without having to waste any time waiting for operator intervention. The problem of idling the sieve without losing the data it contained was accomplished by reconfiguring the delay lines into a single loop and continuing to circulate the pulses; to resume sieving, the device waited until the residues had returned to their original locations and then split the rings back into separate loops. The DLS-127 could also be run in 'solution counting mode', in which it would simply count each solution found instead of stopping and printing its value. This mode was used when solving a problem with a high proportion of solutions to trial values, such as finding pseudo-squares.

The delay line sieve was undoubtably Lehmer's most successful hardware sieve. The hardware was built in less than a year at a cost of roughly $2000 as an unsponsored educational project at the Berkeley campus of the University of California [2]. Since the basic sieving hardware contained no moving parts, the cost of operation and maintenance was negligible. The sieve was used on a wide variety of problems, including factoring ([BS67], [LL74], [BLS75], [LM78]) and a study of various properties of integer sequences ([LLS70], [Sha73]). After several years, the sieve was outfitted with

[2] The delay lines were obtained for about $150 after being rejected by the Navy for use in the tropics because they were not protected against fungus!

six more rings (made out of shift registers rather than delay lines) and renamed the DLS-157. It was eventually retired in 1975 and can now be found in the Computer Museum.

## 5. Further developments.

In the early 1970s, Lehmer again utilized an existing computer to perform sieving [Leh76]. This time the basis was the ILLIAC IV, or I4, an experimental 'one of a kind' parallel processor designed at the University of Illinois.

The I4 contained 64 processors operating in a single instruction, multiple data (*SIMD*) manner. As with the SWAC, the sieve operated by merging and shifting bits strings, but each congruence was implemented by a separate processor, allowing the sieve to shift its 'pseudo-rings' in parallel as in the earlier hardware sieves. Testing for solutions was also performed in parallel by logically OR-ing sets of 64 residues from all 64 rings in a single instruction; a '0' bit in the result indicated a solution. This switch from serial to parallel operation, along with the speed of the I4 instruction set, allowed the sieve to test 15 million values per second—over two orders of magnitude faster than previous software sieve implementations.

Since the experimental nature of the ILLIAC IV prevented Lehmer from utilizing it for sieving on a long term basis, the I4 sieve was only intended to be a demonstration of the feasibility of adapting a parallel processor to sieving. Despite its success, the lack of widespread availability of parallel processing systems hampered further developments along these lines. Instead, researchers experimented with hardware sieves that offered the potential for even higher levels of performance.

In 1962, Gerry Estrin and others suggested constructing a sieving system capable of testing in excess of 100 million numbers per second by using readily available integrated circuits [CEFT62]. Each ring would consist of one or more high speed linear shift registers connected together to form a single circular shift register of the desired size. Using '1' bits to represent acceptable residue classes, the device would sieve by logically AND-ing one bit from each ring and then shifting the bit strings. Utilizing chips capable of shifting every 200 nanoseconds, a $t$ tap sieve would be capable of $5t \times 10^6$ trials per second. The proposal cleverly demonstrated that a shift register capable of shifting by one bit each clock cycle could be made to shift by $t$ bits per cycle by relabelling the bit positions. It also recommended connecting the sieve to a dedicated general purpose computer that would implement in software any congruences that were not implemented in hardware, eliminating the need for operator intervention in such problems.

Curiously, Estrin's proposal was not acted on for over a decade, either by his group, or by Lehmer's, who opted to build the much slower delay line sieve instead. This may have been due in part to the radical nature of the project, which was far more ambitious than any previous hardware sieve, or it may have been too complicated and expensive to build using the integrated circuit technology of the time. By the early 1970's, however, the potential benefits of the shift register design eventually convinced Lehmer and Morton to build one [Leh80], [Leh89]. About the size of a breadbox, their sieve had only a single solution tap, but ran at 20 million trials per second; more taps were planned for later installation but were never added. A reference to the device as

the 'SRS-181' [LL74] indicates that it contained 42 rings, and, like the delay line sieve, it provided both solution counting and regular search modes. Instead of combining the sieve with a conventional computer, a special board was constructed that would act as *host*: loading the problem into the sieve, monitoring its execution, and processing the answers that it generated. Unfortunately, before the board was completed, the sieve itself was mistakenly removed from its lab and sold as scrap while its owners were absent. A second model was never built. An even more ambitious shift register sieve was later planned by a group at the University of Illinois for use as a factoring machine [Don74]. This device was to have had 32 rings, achieving a sieving rate of 320 million trials per second using 32 taps. Although the project progressed at least as far as a partial set of schematics, it was apparently never built.

A complete system based on Estrin's model was finally constructed by Cam Patterson at the University of Manitoba in 1983 [Pat83], [PW83]. The University of Manitoba Sieve Unit, or *UMSU*[3], contained 32 rings representing congruences for the first 32 primes. Eight sets of residues were tested every 60 nanoseconds, resulting in a sieving rate of just over 133 million trials per second. Unlike the SRS-181, it did not provide a solution counting mode, its designer deeming that the small proportion of problems that would utilize it could not justify the extra hardware required. Altogether, the sieve required about 500 integrated circuits and three wire-wrap circuit boards, and cost approximately $7000 (Canadian).

Instead of having a dedicated computer act as its host, *UMSU* ran as a peripheral to an existing PDP-11/45 minicomputer. This not only saved money, but also allowed users to access the sieve from a number of terminals both on and off campus, rather than from a single location. Special software running on the PDP provided commands to create problems, queue them for execution, and inspect the results of completed problems. A permanent background process processed the entries in the problem queue in a serial fashion, automatically translating each into a sequence of commands that could be understood by *UMSU*'s microprogrammed control sequencer, and processing the solutions it generated. Since most problems generated solutions infrequently, UMSU required little attention from the background process and had little impact on the PDP's other users.

The host software written for *UMSU* made the system far easier to use than earlier hardware sieves and was an important step forward in sieve design. Extending Estrin's suggestion, it allowed the user to include tests for any special solution requirements, as well as automatically simulating congruences that were not provided in hardware, making it the first hardware-based system to provide on-line solution filtering. The software also handled the problem of hardware faults and power failures by verifying and recording the state of the current problem every hour, automatically restarting the problem after an error without losing more than the previous hour's work. On the other hand, the system did not optimize sieving when the problem contained one or more congruences with a single acceptable residue, nor could the user create and inspect sieving problems without temporarily suspending the execution of the current problem.

---

[3] Pronounced 'ŭm so͞o'. The acronym was stolen from a more widely known UMSU on campus, the University of Manitoba Students Union.

The *UMSU* system was used between 1983 and 1984 to find periodic continued fractions with long periods [PW85]. When the PDP minicomputer was upgraded to a Vax 750 in 1985, re-installing its UNIX-oriented software under the VMS operating system proved to be difficult; the different bus architectures used by the two minicomputers also forced modifications to be made to the parallel interface between UMSU and its host. A subset of the original software was eventually established, but the sieve developed a hardware fault shortly thereafter. Efforts to diagnose the fault were hindered by the departure of its designer and a lack of documentation, so the system was temporarily abandoned. In 1987, the arrival of a MicroVax II minicomputer running UNIX and utilizing a PDP-compatible bus encouraged a second effort to re-install *UMSU*. Patterson returned for a short period of time and was able to successfully repair the sieve and modify its software, so the complete system is again in operation. Subsequent research efforts have concentrated on finding polynomials that generate prime values [FW89], [MW89].

Despite *UMSU*'s unprecedented speed, filtering ability, and ease of use, experience with the system gradually exposed three significant short-comings in its design that limit its usefulness.

(1) The sieve hardware supports only 32 congruence moduli, forcing the host system to implement all congruences of other sizes through software.

(2) The host software does not optimize sieving when congruences with a small number of acceptable residue classes are present. Although the user can do the optimization manually, this is seldom done.

(3) The system is non-portable. The host software requires the host to run the UNIX operating system, and the interface between UMSU and its host is incompatible with most general-purpose computers.

By May 1985, a desire to tackle problems that were beyond *UMSU*'s capabilites, coupled with the difficulties encountered in installing the *UMSU* system on its new host, led the second author, its main user, to ask the first author to build a totally new system as its successor. The result, the Open Architecture Sieve System (*OASiS*), contains many significant improvements that make it the fastest and most flexible tool ever built for solving the generalized sieving problem. These improvements include a higher basic sieving rate, programmable size rings, automatic optimization of sieving, and increased portability. The system's 'open architecture' also provides for the introduction of additional sieving hardware, and even multiple independent sieves which will enable sieving to be done in parallel.

## 6. A brief description of OASiS.

As with the UMSU system, *OASiS* utilizes a conventional computer system and a specially constructed sieving device working in tandem: in this case, a MicroVaxII minicomputer system and the Open Architecture Sieve (*OAS*). Each part takes upon itself the task(s) to which it is best suited. The sieve's role in solving a sieving problem is rather limited: it performs high speed sieving using some or all of the congruences of the problem. In contrast, the host handles many things, including controlling the operation of the sieve, supplying any sieving capabilities that the *OAS* cannot, and handling the various aspects of problem management (problem creation and deletion,

scheduling problems for execution, and the processing of problem results). This division of responsibility breaks the system up into two relatively independent pieces and puts the majority of the system's complexity into software running on the host, where it can be readily modified to enhance the system's capabilities.

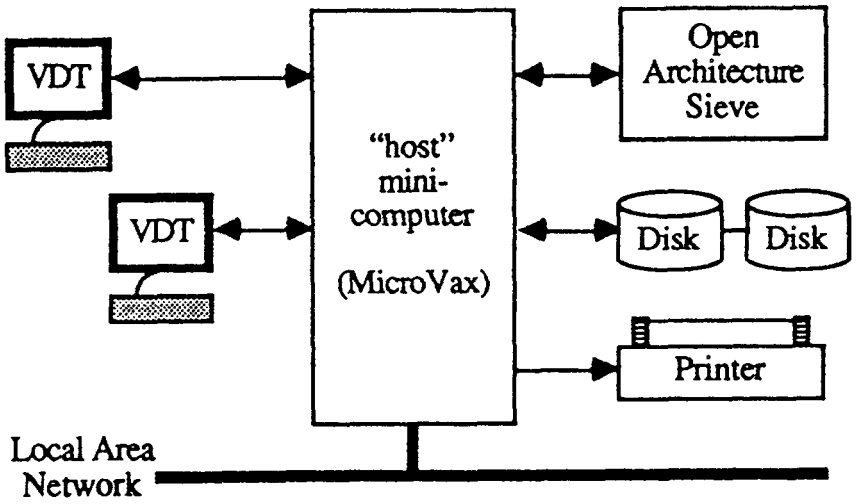



Figure 5. The *OAS* and its host.

The current arrangement of the system hardware is shown in Figure 5. Physically, the MicroVax and the Open Architecture Sieve are connected by a standard serial communication line. A user can utilize the sieve in much the same way as any other peripheral, such as a printer, by simply signing on to the MicroVax timesharing system and typing the appropriate commands. The MicroVax itself can be accessed from a number of nearby video display terminals or from a remote computer through a local area network.

Access to *OASiS* is restricted to a special account with the username OASIS. The MicroVax's standard command set is supplemented with a small number of new commands that allow the user to create new sieving problems, maintain a queue of problems awaiting execution, monitor or terminate the progress of the currently executing problem, and process the results of any completed problem. The other resources normally provided by the host (compilers, printers, mail system, etc.) are not affected and remain available for use at all times.

The user defines a sieving problem by creating from one to three files. A *problem file* is an ordinary text file that specifies the congruences used by the problem and the points at which sieving begins and ends. A problem can be defined to stop using any combination of the following conditions:

(i) when the search for solutions has reached a specified upper bound,

(ii) when a specified number of solutions have been found, or

(iii) after a specified time interval has elapsed.

Although the *OAS* provides a solution counting mode, the *OASiS* software currently supports only problems in which each solution is recorded as it is found; support for solution counting problems may be added at a later date. Any additional restrictions defined by the problem require the creation of an executable file called a *filter program*,

which accepts solution candidates and indicates whether or not each displays certain required properties. The problem file is a mandatory part of every *OASiS* problem, however the filter program can be omitted if the problem imposes no additional restrictions. A second mandatory text file, called a *configuration file*, specifies the number and size of the rings that are available for use during sieving. Since the *OAS* hardware is seldom altered at the present time, *OASiS* provides a default configuration file that is used if the user does not supply one explicitly. The user creates problem and configuration files using the standard editors provided by the host's system software. Filter programs can be written in any language supported by the host, although C is preferable. The two text files can be tested prior to execution by issuing a command that detects syntactic and semantic errors in the problem's definition. The filter program can also be tested to ensure that it functions properly. The complete problem is then added to a priority queue of problems awaiting execution, after which the user may proceed with other work or sign off.

Once a problem reaches the top of the problem queue, it is automatically executed by the *OASiS* software. The software reads in the problem definition, isolates any congruences containing a single acceptable residue class, and loads as many of the remaining congruences into the sieve as it can. If any single residue congruences are found, the residues for the congruences being loaded into the *OAS* are reordered as described in Section 3 to increase the effective sieving rate. The *OAS* then begins sieving. Each time a value that satisfies its congruences is found, the sieve notifies the host. The host reads the value from the sieve and tests it against any congruences that were not loaded into the sieve and against the filter program (if any). If the value satisfies these additional constraints, it is considered to be a solution to the problem. The *OAS* sieves at about 215 million trials per second, making it about 60% faster than UMSU. However, the optimization done by the *OASiS* software frequently allows the system to solve a problem as much as 40 times faster than *UMSU*.

The *OASiS* software performs extensive error checking on itself and on the *OAS* hardware during the execution of a sieving problem; if a fault is detected, the software either restarts the problem or terminates it with an error message. A problem may also be restarted after a system crash (handled automatically) or after being prematurely terminated by the user (the user is required to resubmit the problem). To prevent large quantities of work from being lost in such cases, the software periodically takes a *checkpoint*; that is, it verifies the contents of the sieve hardware and records what point in the search interval has been reached. This allows a partially executed problem to be resumed from its most recent checkpoint, rather than starting over from the very beginning.

All results generated during the execution of a problem (solutions, checkpoints, error messages, etc.) are appended to the problem file. For convenience, *OASiS* permits the user to view the problem file during execution to determine how sieving is progressing. Normally, the problem executes until it reaches one of the termination conditions defined in the problem file, however it can also be terminated by the user using a special command. In either case, the user is notified through the host's mail system whenever the problem ends. The solutions found by *OASiS* can be extracted from a problem file using another command and either printed or used as input to user written programs that process them.

The capabilities of the *OASiS* environment allow the user to solve sieving problems quickly and easily, and are similar in most respects to those provided by *UMSU*. However, in terms of implementation, the *OASiS* software is very different. *OASiS* keeps almost all of the problem information in a single place—the problem file—where it can be readily inspected and processed; *UMSU*, on the other hand, stores the results of a problem separately from its definition and uses a machine readable format that is difficult for humans to comprehend. Similarly, the *OASiS* software is a collection of relatively independent, single function commands, many of which are short, simple programs that can be easily written and maintained; as a result, the capabilities of *OASiS* can be enhanced easily by modifying existing commands or adding new ones. In contrast, *UMSU* uses a pair of large, multi-function programs that are not easily changed. The *OASiS* software consists of a several new commands which control the execution of a single problem and existing MicroVax commands that handle the problem before and after execution. Incorporating standard host commands into the design of *OASiS* allows users who are familiar with the MicroVax environment to learn to use *OASiS* quickly, and reduces the burden of writing and maintaining the *OASiS* software. Although this reliance on the host's system software might appear to make *OASiS* non-portable, it actually increases portability by allowing the software to exploit the existing resources of a host computer rather than insisting on a fixed set of requirements that must be provided. As a result, some form of *OASiS* could be established fairly easily on almost any minicomputer or microcomputer system that supports the C language.

## 7. The open architecture sieve.

The Open Architecture Sieve is a high speed sieving device based on an extremely flexible architecture. The sieve hardware can be configured to match the needs of the problem(s) at hand by altering the number of sieve processors used, the number of rings in each processor, and the size of each ring. At the present time, the sieve consists of a single processor with 16 rings, each of which can be loaded with any congruence whose modulus lies in the range 1 to 8192. The *OAS* provides a sieving rate of nearly 215 million trials per second, and can be directed either to stop and report each solution it finds during sieving or to simply count solutions without stopping.

The most revolutionary feature of the *OAS* is its programmable size rings, which allow it to sieve any set of 16 or fewer congruences involving moduli no larger than 8K. Sets of more than 16 congruences can often be handled by combining two or more congruences into a single larger one; consequently, the host can load the *OAS* with any set of congruences whose moduli involve the first 37 primes. If necessary, the sieve can be upgraded to a maximum of 32 rings, and each ring increased to a capacity of 32K (32 768), with little difficulty.

The *OAS* implements the basic ring concept in a new way. Rather than shifting the residues past a fixed solution tap, it moves the tap past the residues (see Figure 6). Residues are stored in a $t$ bit wide random access memory (RAM), using '1' bits to indicate acceptable residue classes. Each time a clock pulse is applied to the ring, the $t$ residues at the memory location specified by the *current address register* (*CAR*) are captured by a residue latch and placed onto the solution bus, where they are logically AND-ed with the residues from the other rings to generate $t$ solution flags. The same

clock pulse also increments the *CAR*, causing the residues in the next RAM location to be captured and tested during the following clock cycle. The *CAR* does not revert to zero once it reaches its maximum value, $M$; instead, it loads the value given in the *start address register* (*SAR*), $S$. Thus, a sequence of clock pulses causes the ring to cycle through the residues located in the RAM locations from $S$ to $M$, inclusive, simulating a shift register with $(M - S + 1)t$ positions.

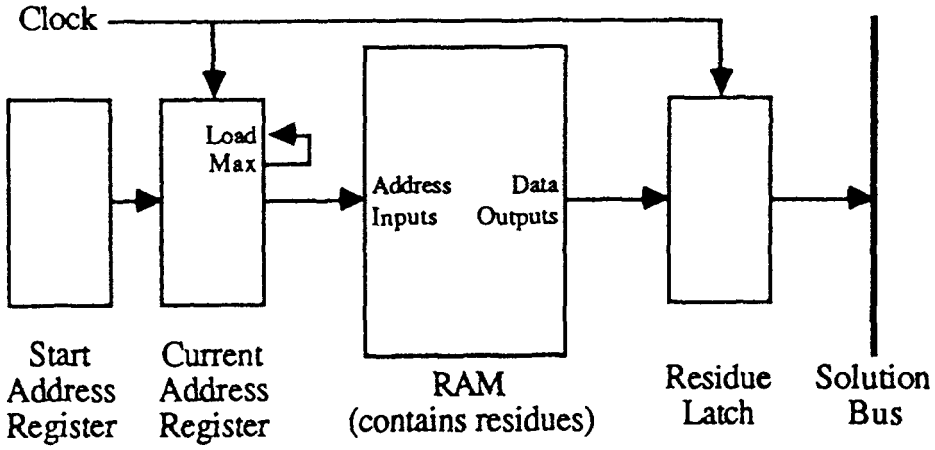



Figure 6. *OAS* ring design.

The host adjusts the size of the ring to match a given congruence by loading the appropriate starting value into a ring's *SAR*. Since the congruence's residue classes must occupy an integral number of RAM locations, it is often necessary to load the RAM with multiple copies of the residues to ensure that the total number of residues involved is a multiple of $t$. For a congruence with modulus $m$, altogether $t/\gcd(m,t)$ copies are required (see Figure 7). As a result, a ring with an $n$ word by $t$ bit RAM can implement any congruence where $m/\gcd(m,t)$ is less than or equal to $n$; this includes all congruences with a modulus not exceeding $n$ and some with a modulus as large as $nt$.

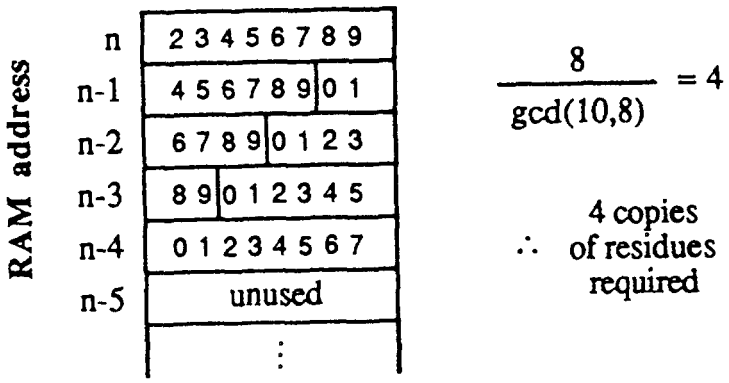



Figure 7. Residue replication in an *OAS*-type ring ($m = 10$, $t = 8$).

The *OAS* rings use a 16 bit *SAR* and *CAR*, giving access to 65 536 RAM locations. The number that can actually be used during sieving depends on the ring's construction. Currently, all rings have an $n$ of 8192 (corresponding to RAM locations $E000_{16} - FFFF_{16}$), but this can be incresed to 32 768 ($8000_{16} - FFFF_{16}$) by using larger memory chips. The RAMs and residue latch are also 16 bits wide, providing a $t$

of 16 for each ring. When driven by a clock with a frequency of about 13.3 MHz, the rings are able to test nearly 215 million values per second.

The rings are only a portion of a complete sieving device called a *sieve unit.* A sieve unit is organized much like a conventional microcomputer system (see Figure 8) and contains a microcontroller (8 bit microprocessor, 2K EPROM, 128 byte RAM), up to 32 programmable size rings, a 16 bit solution window, a 16 bit solution mask, a 48 bit trial counter, an 8 bit clock control register, and a 64 bit solution counter.

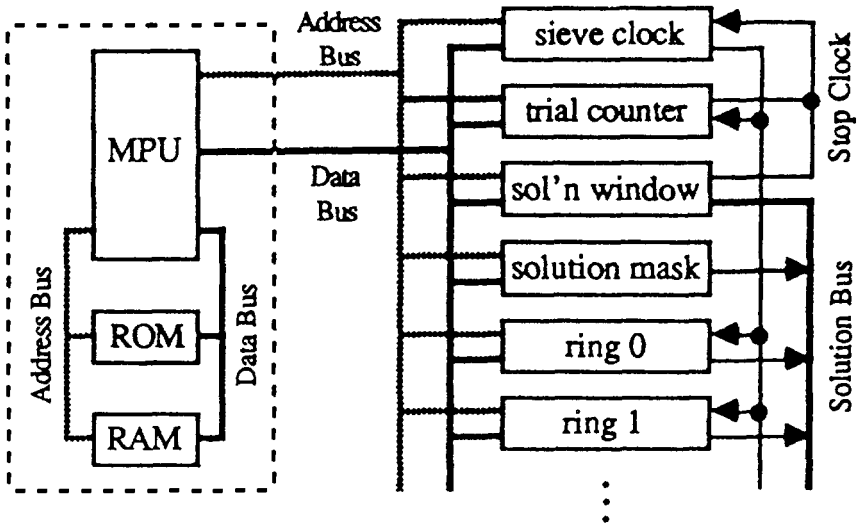



Figure 8. *OAS* sieve unit design.

The 'brain' of the sieve unit is its 8 bit microprocessor, which acts as an interface between the sieve's host and the actual sieving hardware. The device runs a program known as its *firmware* that is permanently resident in EPROM. The *OAS* firmware is a simple command interpreter that accepts commands from the host and manipulates the remaining components appropriately in order to carry them out. The combination of an erasable medium for the firmware and the ability to program the microprocessor in assembly language allows the low level operation of the sieve hardware to be modified easily to correct bugs or add new features. The slow speed of the microprocessor relative to the rest of the sieve hardware is seldom important, since the microprocessor and the rings operate independently during sieving.

The commands provided by the sieve firmware carry out actions at a relatively low level that requires some knowledge of the sieve unit's components. Thus, the host must translate the definition of a sieving problem into the appropriate sequence of commands and translate the sieve's responses into meaningful solution information. The commands available to the host allow it to load the sieve with a set of congruences, set the range of values to be searched, start and stop sieving, and inspect the solutions the sieve finds. Both host commands and sieve responses are simple ASCII character strings, allowing the user to monitor transmissions by tapping into the communication line between the two machines. The user can also use the command set to diagnose system malfunctions by connecting the sieve to a dumb terminal and entering commands manually; this is a simpler and more flexible means of troubleshooting than using pre-written diagnostic software, and prevents a bug in the host software from masquerading as a problem with the sieve.

The *OAS* normally sieves in *solution recording mode*, in which case it automatically halts and notifies the host whenever a solution is detected. As mentioned, the sieve uses the residues presented by its rings to generate 16 solution flags during each clock cycle. These are loaded into the solution window, which automatically stops the sieve clock if a '1' bit is present. The trial counter keeps track of the number of clock pulses that have occurred. When it is notified of a solution, the host can read these two components and calculate the value of the solution(s) that have been found. The trial counter also blocks the sieve clock when it reaches its maximum count of $2^{48} - 1$, so by initializing the counter correctly the host can program the sieve to search a specified interval and then stop. If a problem requires more than $2^{48}$ clock cycles to complete (about 250 days) the host must manually reset the counter in order to continue sieving.

Although it is not currently used by the *OASiS* software, the *OAS* can also be run in *solution counting mode*. In this mode, the sieve increments the solution counter each time solutions appear in the solution window and continues sieving without notifying the host. Unlike the other sieve components, the solution counter is not implemented in hardware but is maintained in the microcontroller's RAM memory. Thus, the sieve must pause briefly whenever the counter is incremented to enable the controller to read the solution window. Although a hardware solution counter would have enabled the sieve to run at full speed continuously, the firmware counter is almost as fast and does not require any additional hardware.

The remaining pair of sieve components are provided to aid in debugging hardware faults. The solution mask can be programmed to selectively disable any combination of solution taps. The clock control register can be used to alter the frequency of the sieve clock between 13.3 MHz and 4 MHz to enable timing problems to be detected more easily.

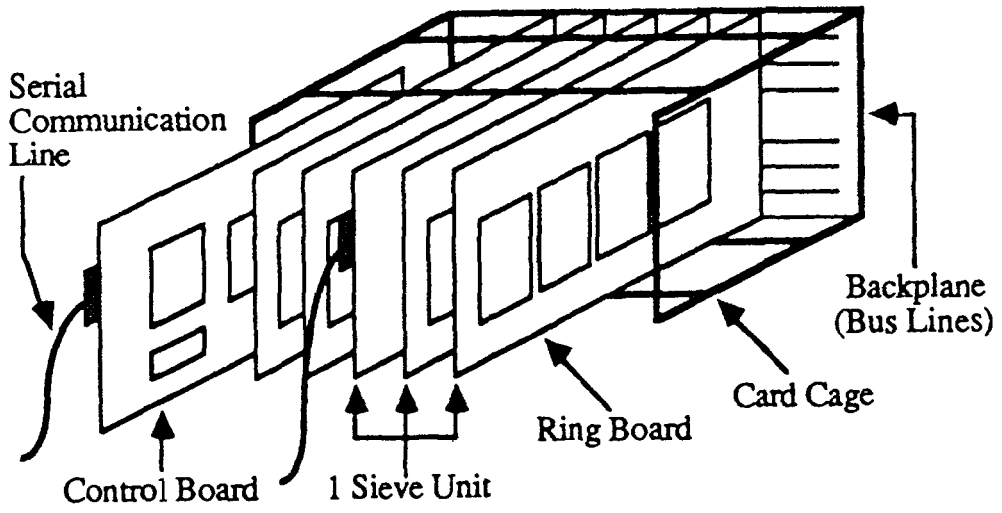



Figure 9. *OAS* construction.

Physically, the Open Architecture Sieve consists of a card cage containing a number of custom two layer printed circuit boards populated with commercially available high speed digital electronic components (see Figure 9). The boards slide into a common backplane, and can be added or removed easily. The boards are grouped into one or more sieve units, each of which operates as an independent sieve. A sieve unit

consists of a single *control board* (containing the microcontroller and other non-ring components) and all *ring boards* in adjacent slots up to the next active control board or empty slot. Each ring board contains four rings, so a sieve unit can use up to eight ring boards before its limit of 32 rings is reached. A sieve unit communicates with its host over a separate serial communication line attached to its control board, thereby avoiding the portability problems that plagued *UMSU*. Serial transmission has the drawback of being relatively slow, but since most sieving problems involve little interaction between host and sieve, this is seldom important.

The design of the *OAS* allows the device to be configured in a variety of ways by altering the number and arrangement of boards within the card cage. This lets the user alter the number of sieve units and the distribution of rings between them to match the available boards to a specific sieving problem in the best possible way. Ring boards can be added to a sieve unit to allow it to handle larger sets of congruences, or removed if they became defective or are needed elsewhere. Similarly, using multiple control boards allows the *OAS* to run several problems at once or to solve a single problem faster by dividing the work among several processors; rings can then be distributed according to the needs of each processor. To simplify the task of reconfiguring the sieve, the architecture of the *OAS* allows the host to put each control board into *sleeping mode*; the board then logically disconnects itself from the backplane and its rings become part of the adjacent sieve unit. Any number of sieve units can be merged in this fashion, as long as the resulting sieve unit contains no more than 32 rings. While putting control boards to sleep is easier on the hardware (and the user) than physically moving boards, it may not always be possible to achieve the desired configuration in this manner.

Currently, the *OAS* consists of a single control board and four ring boards, providing a single sieve unit with 16 rings. Ths sieve communicates with its host MicroVax II mini-computer over a 9600 baud terminal line. A second control board has been built and tested, but is of limited use until enough ring boards have been constructed to complete a second sieve unit. In all, the hardware cost $5600 (Canadian); an additional $5000 was spent on equipment and software for designing and building printed circuit boards.

The Open Architecture Sieve System became operational in January, 1989. Since then, it has been put to work on a variety of problems in number theory, including pseudo-powers, periodic continued fractions with long periods, and quadratic polynomials with a high density of prime values. The following sections describe these problems and the results obtained.

## 8. Pseudo-squares and negative pseudo-squares.

Let $p_1, p_2, \ldots, p_m$ be a set of odd primes and $\epsilon_1, \epsilon_2, \ldots, \epsilon_m$ be a sequence with $\epsilon_i^2 = 1$. We are interested in integers whose quadratic character with respect to each $p_i$ is specified by the corresponding $\epsilon_i$; that is, we wish to find $N$ such that

$$(N/p_i) = \epsilon_i, \quad i = 1, 2, \ldots, m,$$

where $(a/b)$ is the Legendre symbol. This is a natural sieving problem in which each equation is represented by a congruence that specifies either the quadratic residues or non-residues modulo $p_i$ for $\epsilon_i = 1$ and $-1$, respectively. Several instances of this

problem are of special interest because they have applications to other branches of number theory. Previous work in this area includes contributions by Lehmer, Lehmer, and Shanks [LLS70] and Shanks [Sha73].

In sections 8 through 10, $p_i$ is the $i$-th odd prime (i.e. $p_1 = 3, p_2 = 5, \ldots$) and all $\epsilon_i$ are equal. Following Shanks, $aR_p$ ($aN_p$) denotes the set of positive $N \neq n^2$ of the form $8k + a$ which are non-zero quadratic residues (non-residues) of all odd primes $q \leq p$. Similarly, $-aR_p$ and $-aN_p$ denote the sets for which $-N$ has the specified quadratic character, rather than $N$ itself. In each problem, *OASiS* searched for $N$ values belonging to one or more of these classes. Because the congruences modulo 3 and 8 contain only a single acceptable residue, *OASiS* was able to increase its hardware sieving rate by a factor of 24 (to approximately $5{\cdot}12 \times 10^9$ trials per second) by combining them into one congruence and optimizing its search.

The class $1R_p$ represents non-square integers, $N$, where $N \equiv 1 \pmod 8$ and $(N/q) = 1$ for all primes $q \leq p$. Since these values appear to have the quadratic character of a perfect square for primes up to $p$, but are not perfect squares, they are known as pseudo-squares. Pseudo-squares can be used in a variety of applications, including a test for primality [Hal33] and an inexpensive test for a perfect square [Cob66].

| $p$ | $N_p$ | Source | $p$ | $N_p$ | Source |
|---|---|---|---|---|---|
| 2 | 17 | Kraitchik | 97 | 2 805 544 681 | Shanks |
| 3 | 73 | (1924) | 101 | 10 310 263 441 | (1970) |
| 5 | 241 | moveable | 103 | 23 616 331 489 | DLS-127 |
| 7 | 1 009 | strips | 107 | 85 157 610 409 | |
| 11 | 2 641 | | 109 | 85 157 610 409 | |
| 13 | 8 089 | | 113 | 196 265 095 009 | |
| 17 | 18 001 | | 127 | 196 265 095 009 | |
| 19 | 53 881 | | 131 | 2 871 842 842 801 | Lehmer |
| 23 | 87 481 | | 137 | 2 871 842 842 801 | (1973) |
| 29 | 117 049 | | 139 | 2 871 842 842 801 | DLS-157 |
| 31 | 515 761 | | 149 | 26 250 887 023 729 | [unpublished] |
| 37 | 1 083 289 | | 151 | 26 250 887 023 729 | |
| 41 | 3 206 641 | | 157 | 112 434 732 901 969 | Williams |
| 43 | 3 818 929 | | 163 | 112 434 732 901 969 | (1988) |
| 47 | 9 257 329 | | 167 | 112 434 732 901 969 | *UMSU* |
| 53 | 22 000 801 | Lehmer (1928) | 173 | 178 936 222 537 081 | [unpublished] |
| 59 | 48 473 881 | bicycle chains | 179 | 178 936 222 537 081 | |
| 61 | 48 473 881 | | 181 | 696 161 110 209 049 | |
| 67 | 175 244 281 | Lehmer (1954) | 191 | 696 161 110 209 049 | |
| 71 | 427 733 329 | SWAC | 193 | 2 854 909 648 103 881 | Stephens) |
| 73 | 427 733 329 | | 197 | 6 450 045 516 630 769 | (1989 |
| 79 | 898 716 289 | | 199 | 6 450 045 516 630 769 | *OASiS* |
| 83 | 2 805 544 681 | Lehmer, | 211 | 11 641 399 247 947 921 | |
| 89 | 2 805 544 681 | Lehmer, | 223 | 11 641 399 247 947 921 | |

Table 2. Least pseudo-squares.

We denote the smallest member of the class $1R_p$ as $N_p$. The $N_p$ values from $N_2$ to

$N_{127}$ were found by Kraitchik [Kra24], Lehmer [Leh28], [Leh54], and Lehmer, Lehmer, and Shanks [LLS70], while unpublished calculations by Lehmer and others by Williams have found $N_{131}$ through $N_{191}$. *OASiS* has now been used to find $N_{193}$ through $N_{223}$. A complete list of $N_p$ values and their sources is given in Table 2.

*OASiS* searched for members of the class $1R_{157}$ from $10^{14}$ to $1{\cdot}25 \times 10^{16}$; these were later tested to determine whether each value was also an element of $1R_p$ for larger values of $p$. The class $1R_{157}$ was selected since it loaded the sieve hardware with as many congruences as possible, and provided a reasonable balance between generating too few solutions to be of interest and generating too many for the sieve's host minicomputer to process easily.

Since many odd perfect squares are also solutions to the problem's congruences, a simple filter program was used to eliminate them as they were generated by the sieve. Although the work involved in testing each solution candidate was not great, the numerous squares encountered meant that up to 80% of the host's CPU time was spent filtering, which greatly slowed the progress of sieving. The sieve usually found its next solution candidate long before the filter program had finished testing the previous one, forcing the machine to spend up to 92% of its time idle. As well, the low priority level given to the *OASiS* software (designed to prevent it from degrading performance for the host's other users) reduced the effective sieving rate even further during peak periods. As the search progressed, the decreasing density of squares improved the filtering situation considerably, but by the time it had reached $10^{16}$, *OASiS* was still only running at 65% of its full speed. In all, the interval from $10^{14}$ to $1{\cdot}25 \times 10^{16}$ required 1933 hours to search, for an average sieving rate of $1{\cdot}78 \times 10^9$ trials per second (35% of its theoretical limit). Had the *OASiS* software been extended to take advantage of the sieve's firmware solution counting capability, it might have been possible to use the technique described by Lehmer, Lehmer, and Shanks [LLS70] to eliminate the filter program entirely and achieve the maximum sieving rate much earlier.

We can make two observations about the entries in Table 2. First, Bach [Bac89] has shown under the generalized Reimann Hypothesis (GRH) that if $G$ is a proper multiplicative subgroup of the integers modulo $m$, then there exists some $n > 0$ such that $n \notin G$ and $n < 2(\log m)^2$. Thus, if $N_p$ is a prime and $r$ is the least prime such that $(N_p/r) = -1$, then we should have $r < 2(\log N_p)^2$, or $N_p > e^{\sqrt{r/2}}$. In fact, all of the entries in Table 2 greatly exceed this lower bound. Second, the numbers in the first part of the table only work for a single $p$, while later values tend to work for two or more consecutive primes. Since there is as yet no explanation for this distribution, it will be interesting to see if this tendency continues for values beyond $N_{223}$.

The class $-7R_p$ contains integers, $N$, where $-N \equiv 1 \pmod 8$ and $(-N/q) = 1$ for all primes $q \leq p$. Since the negatives of these $N$ have the same quadratic character as the values in the previous section, they can be thought of as negative pseudo-squares.

As before, the smallest member of the class is denoted by $N_p$. A table of $N_p$ values for $p \leq 131$ is given by Lehmer, Lehmer, and Shanks in [LLS70]. Shanks [Sha73] later determined the values up to $N_{163}$, but only one value was published. *OASiS* has now been used to extend the table up to $N_{211}$. A complete list of the $N_p$ values is given in Table 3.

*OASiS* searched for members of the class $-7R_{157}$ from 0 to $5 \times 10^{15}$; the values

generated were then tested to determine $N_p$ for $157 \leq p \leq 211$. A second, smaller search generated the entries from the class $-7R_{137}$ between 0 and $5 \times 10^{13}$, filling in $N_{137}$ through $N_{151}$. Since 7 is not a quadratic residue modulo 8, perfect squares were not generated by these searches; thus, the filter program used in finding positive pseudo-squares was not required and *OASiS* was able to sieve at full speed throughout. Consequently, the first search took about 12 days to perform and the second about three hours.

| $p$ | $N_p$ | Source | $p$ | $N_p$ | Source |
|---|---|---|---|---|---|
| 2 | 7 | Lehmer, | 97 | 2 570 169 839 | |
| 3 | 23 | Lehmer, | 101 | 2 570 169 839 | |
| 5 | 71 | Shanks | 103 | 2 570 169 839 | |
| 7 | 311 | (1970) | 107 | 2 570 169 839 | |
| 11 | 479 | DLS-127 | 109 | 2 570 169 839 | |
| 13 | 1 559 | | 113 | 328 878 692 999 | |
| 17 | 5 711 | | 127 | 328 878 692 999 | |
| 19 | 10 559 | | 131 | 513 928 659 191 | |
| 23 | 18 191 | | 137 | 844 276 851 239 | Stephens |
| 29 | 31 391 | | 139 | 1 043 702 750 999 | (1989) |
| 31 | 307 271 | | 149 | 4 306 732 833 311 | *OASiS* |
| 37 | 366 791 | | 151 | 8 402 847 753 431 | |
| 41 | 366 791 | | 157 | 47 375 970 146 951 | |
| 43 | 2 155 919 | | 163 | 52 717 232 543 951 | |
| 47 | 2 155 919 | | 167 | 100 535 431 791 791 | |
| 53 | 2 155 919 | | 173 | 251 109 340 045 079 | |
| 59 | 6 077 111 | | 179 | 493 092 541 684 679 | |
| 61 | 6 077 111 | | 181 | 493 092 541 684 679 | |
| 67 | 98 538 359 | | 191 | 493 092 541 684 679 | |
| 71 | 120 293 879 | | 193 | 1 088 144 332 169 831 | |
| 73 | 131 486 759 | | 197 | 1 088 144 332 169 831 | |
| 79 | 131 486 759 | | 199 | 1 088 144 332 169 831 | |
| 83 | 508 095 719 | | 211 | 1 088 144 332 169 831 | |
| 89 | 2 570 169 839 | | 223 | $> 5\,000\,000\,000\,000\,000$ | |

Table 3. Least negative pseudo-squares.

Negative pseudo-squares are used in the primality test of Selfridge and Weinberger (see Section 21 of [Wil78]). The result of Bach asserts that a value $N$ need only be tested against a relatively small number of primes (fewer than $2(\log N)^2$ in all) to ensure it is a prime, making this method an effective, polynomial-time primality test under the GRH. The fact that all of the values in Table 3 exceed Bach's bound provides evidence that supports this result and does not depend on the truth of the GRH.

## 9. Periodic continued fractions with long periods.

Williams [Wil81] has pointed out that if D is square-free, then the period length, $p(D)$, of the continued fraction expansion of $\sqrt{D}$ should be bounded above by an expression of the form $c\sqrt{D}\log\log D$. In particular, if

$$f(D) = \begin{cases} \sqrt{D}\log\log D & \text{for } D \equiv 1 \pmod 8, \\ \sqrt{D}\log\log 4D & \text{otherwise,} \end{cases}$$

then it should be true that

$$G(D) = \frac{p(D)}{f(D)} < k + o(1)$$

under the extended Riemann Hypothesis for $\zeta_K$ with $K = \mathbf{Q}(\sqrt{D})$. Here $k = 3{\cdot}7012$, but Lévy's Law indicates that $k$ could be as small as $12e^{\gamma} \log 2/\pi^2 \approx 1{\cdot}50103$. Patterson and Williams [PW85] examined many large $G(D)$ values to see how they approached this bound. The largest found was at $D = 13\,518\,648\,471\,574$ with $G(D) = 1{\cdot}081381$.

Earlier work by Williams [Wil81] suggests that $D$ values for which $G(D)$ is large are most likely integers of the form $q$ or $2q$, where $q$ is a prime and $q \equiv -1 \pmod 4$. It is also desirable to have $(D/r_i) = 1$ for the odd primes $r_1, r_2, \ldots, r_n$ where $n$ is as large as possible. Thus, candidate $D$ values can be found by solving a system of linear congruences.

Following the example set in [PW85], *OASiS* was used to find $D$ values of four types: (i) $D \equiv 3 \pmod 8$, $D$ prime, (ii) $D \equiv 7 \pmod 8$, $D$ prime, (iii) $D \equiv 6 \pmod 8$, $D/2$ prime, and (iv) $D \equiv 1 \pmod 8$, $D$ prime. The type (iv) $D$ values were examined to determine whether their associated $G$-values start to catch up with the larger values obtained by the other three classes, as predicted by Shanks. For the first three cases, *OASiS* searched the range 0 to $5 \times 10^{15}$ for members of the classes $3R_{157}, 7R_{157}$, and $6R_{157}$, respectively; each search took about 12 days. For the type (iv) case, we simply reused the results from the earlier pseudo-square search, examining members of the class $1R_{157}$ within the range from $10^{14}$ to $10^{16}$.

Calculating the exact value of $G(D)$ for all of the $D$ values generated by *OASiS* would have required many hours of computer time due to the repeated need to determine $p(D)$ using the continued fraction algorithm which has a time complexity of $O(D^{1/2+\epsilon})$. Instead, the $O(D^{1/4+\epsilon})$ 'large step' algorithm utilized by Stephens and Williams [SW88] was used to calculate the regulator, $R(D)$, of $\mathbf{Q}(\sqrt{D})$, from which $G'(D) = R(D)/f(D)$ was determined. Since $R(D)$ is expected by Lévy's Law to be proportional to $p(D)$, a large $G'(D)$ value is a likely indicator of a large $G(D)$. Using this as a guide, we then calculated $p(D)$ and $G(D)$ for a few of the $D$ values generated. Each regulator calculation averaged less than a second of computer time on an Amdahl 5870 computer.

| $D$ | $R(D)$ | $p(D)$ | $G(D)$ |
|---|---|---|---|
| 25 969 254 121 099 | 11 325 477.6 | 9 551 418 | 0.539479 |
| 35 474 768 258 491 | 26 025 216.6 | 21 942 306 | 1.057447 |
| 121 521 362 355 331 | 48 737 550.2 | 41 084 814 | 1.058503 |
| 152 290 419 440 611 | 54 891 577.2 | 46 274 886 | 1.062983 |
| 206 546 921 647 291 | 64 474 167.2 | 54 350 198 | 1.069334 |
| 820 362 746 906 299 | 131 207 980.2 | 110 604 710 | 1.079902 |
| 924 401 140 322 059 | 140 451 569.1 | 118 399 282 | 1.087996 |
| 4 976 709 946 053 091 | 330 581 613.0 | 278 655 786 | 1.089614 |

Table 4. $D$–type (i).

| $D$ | $R(D)$ | $p(D)$ | $G(D)$ |
|---|---|---|---|
| 963 864 514 519 | 2 107 959.8 | 1 778 716 | 0.538151 |
| 46 257 585 588 439 | 30 459 726.7 | 25 679 652 | 1.081244 |
| 289 358 196 053 551 | 77 740 401.4 | 65 539 148 | 1.086442 |
| 1 135 360 188 709 399 | 156 220 850.7 | 131 687 940 | 1.090171 |

Table 5. $D$–type (ii).

| $D$ | $R(D)$ | $p(D)$ | $G(D)$ |
|---|---|---|---|
| 27 266 351 212 006 | 1 448 975.4 | 1 219 624 | 0.067199 |
| 42 940 991 222 614 | 14 450 735.1 | 12 182 504 | 0.532732 |
| 48 888 369 417 694 | 30 359 479.5 | 25 594 764 | 1.047768 |
| 129 143 129 979 406 | 50 279 000.9 | 42 391 356 | 1.058905 |
| 193 289 509 403 566 | 62 162 428.3 | 52 405 772 | 1.066434 |
| 256 397 742 215 806 | 71 809 801.1 | 60 536 004 | 1.067108 |
| 285 278 695 393 246 | 76 061 142.8 | 64 119 584 | 1.070606 |
| 477 747 574 223 494 | 99 581 194.9 | 83 943 714 | 1.078595 |
| 600 206 879 107 606 | 112 267 896.3 | 94 642 190 | 1.082969 |
| 4 518 102 473 256 934 | 313 418 266.5 | 264 191 526 | 1.084990 |
| 4 826 678 427 841 846 | 326 631 901.1 | 275 319 106 | 1.093416 |

Table 6. $D$–type (iii).

| $D$ | $R(D)$ | $p(D)$ | $G(D)$ |
|---|---|---|---|
| 112 434 732 901 969 | 45 498 660.0 | 38 364 413 | 1.040660 |
| 162 516 480 029 401 | 54 895 119.1 | 46 286 149 | 1.040928 |
| 273 323 976 657 169 | 71 812 592.8 | 60 545 353 | 1.045206 |
| 457 165 855 430 761 | 94 198 806.8 | 79 417 945 | 1.055462 |
| 570 395 076 767 569 | 105 696 894.2 | 89 110 088 | 1.058265 |
| 732 376 785 497 449 | 120 433 435.9 | 101 538 843 | 1.061985 |
| 3 135 969 368 926 969 | 252 416 665.9 | 212 781 805 | 1.062959 |
| 4 113 071 509 075 489 | 290 678 761.5 | 245 039 927 | 1.066602 |

Table 7. $D$–type (iv).

Tables 4 through 7 list a subset of the $D$ values generated for each of the problems run by *OASiS*, along with their corresponding $p$- and $G$-values. Each $D$ is shown only if $G(D)$ exceeds the value of $G(d)$ for all computed values of $d$ of the same type with $d < D$. These results extend the work of [PW85] by about an order of magnitude, and give no indication that the projected limit of $k \approx 1{\cdot}50103$ will be exceeded. The growth of $G(D)$ for type (iv) $D$ values also appears to be slightly greater than for the other three types, indicating that it is indeed catching up to them.

## 10. Quadratic polynomials generating many primes.

Let $f_A(x) = x^2 + x + A$ ($A \in \mathbf{Z}$, $A > 0$) and let $P_A(n)$ be the number of prime values assumed by $f_A(x)$ for $x = 0, 1, 2, \ldots, n$. Polynomials for which $P_A(n)$ is relatively large have interested mathematicians since the time of Euler, who found

that $P_{41}(39) = 40$. Since that time, other examples have come to light which generate an even higher proportion of primes as $n$ approaches infinity.

The search for polynomials of this type is aided by considering Hardy and Littlewood's Conjecture $F$ [HL23] which asserts that $P_A(n) \sim C(D)L_A(n)$ where

$$D = 1 - 4A, \quad L_A(n) = 2\int_0^n \frac{dx}{\log f_A(x)}, \quad \text{and} \quad C(D) = \prod_{p \geq 3} \left(1 - \frac{(D/p)}{p-1}\right),$$

the latter product being taken over all odd primes $p$. If the conjecture holds, then $f_A(x)$ has a high asymptotic density of prime values whenever $C(D)$ is large. Considerable evidence in support of this has been given by Shanks [Sha59], [Sha60], [Sha63] and Fung and Williams [FW89]).

| $D$ | $C(D)$ | $q(D)$ |
|---|---|---|
| −1 936 187 977 599 283 | 4.6499685 | 191 |
| −2 075 334 218 440 387 | 4.4565543 | 191 |
| −2 080 538 415 662 947 | 4.8777963 | 191 |
| −2 280 563 801 157 163 | 4.8505941 | 191 |
| −2 443 638 146 871 667 | 4.5308361 | 191 |
| −2 572 394 636 095 243 | 4.5732258 | 191 |
| −2 621 536 114 644 283 | 4.5566321 | 191 |
| −2 667 258 747 189 523 | 4.5473160 | 191 |
| −2 692 385 915 777 083 | 4.7797430 | 193 |
| −2 773 901 630 544 907 | 4.5808188 | 191 |
| −2 980 855 020 126 547 | 4.4013186 | 191 |
| −3 126 717 241 727 227 | 4.5685162 | 193 |
| −3 127 258 244 646 187 | 4.6359081 | 193 |
| −3 149 000 695 013 947 | 4.7666727 | 191 |
| −3 306 963 252 448 003 | 4.5539217 | 193 |
| −3 349 571 159 430 787 | 4.8028039 | 193 |
| −3 361 682 073 539 827 | 4.7758725 | 191 |
| −3 426 547 415 712 283 | 4.6784043 | 191 |
| −3 501 931 983 774 547 | 4.6213901 | 191 |
| −3 567 911 491 464 643 | 4.4899719 | 193 |
| −3 728 839 196 991 283 | 4.5471659 | 193 |
| −3 794 312 952 243 643 | 4.6797966 | 191 |
| −3 820 909 447 274 203 | 4.6906107 | 193 |
| −4 087 860 124 363 723 | 4.9285914 | 191 |
| −4 234 760 613 525 187 | 5.0181916 | 197 |
| −4 311 527 414 591 923 | 4.5293043 | 211 |
| −4 499 600 282 582 827 | 4.6037822 | 197 |
| −4 692 944 772 737 323 | 4.7691045 | 193 |
| −4 700 459 864 595 763 | 4.6313214 | 191 |
| −4 867 291 923 996 643 | 4.6462439 | 193 |

Table 8. $D$–values where $q(D) \geq 191$.

Assuming Conjecture $F$, the task of locating an $A$ for which $C(D)$ is likely to be large is not hard. For $f_A(x)$ to assume prime values, $A$ must be odd; so we must have $-D = 4A - 1 \equiv 3 \pmod 8$. To maximize $C(D)$, it is necessary to have $(D/p) = -1$ for as many small primes $p$ as possible. As pointed out by Lehmer [Leh36], this restriction ensures $p$ does not divide $f_A(x)$ for any $x$; so if it applies for many primes then $f_A(x)$ is likely to be a prime. Hence, if $N$ is a member of the class $-3N_p$ then $D = -N$ is likely to have a large $C(D)$ value.

A set of candidate $N$ values was generated by having *OASiS* find the members of the class $-3N_{173}$ within the range 0 to $5 \times 10^{15}$, a task which took about 12 days. Since the infinite product method of calculating $C(D)$ converges quite slowly, Fung and Williams' modified 'giant step-baby step' method was used to evaluate $C(D)$ for each $D$. This approach is very fast but is contingent on the extended Riemann Hypothesis.

Table 8 lists a subset of the $D$ values generated by *OASiS*, along with their corresponding $C(D)$ and $q(D)$ values. The latter value denotes the least prime such that $(D/q(D)) \neq -1$. Only $D$ values greater than $10^{15}$ for which $q(D) \geq 191$ are shown. These results can be compared with a similar table given in [FW89] which extends up to $10^{15}$. The data in the table can be used to locate $N_p$, the smallest member of the class $-3N_p$. The values of $N_2$ through $N_{197}$ are known to be less than $10^{15}$ and are listed in [FW89]. A quick glance at Table 8 shows that $N_{199} = 4\ 311\ 527\ 414\ 591\ 923$ and that the next missing entry, $N_{211}$, must exceed $5 \times 10^{15}$.

Table 8 also illustrates that a large $q(D)$ value does not necessarily guarantee a large $C(D)$ value. In fact, $-N_{199}$ has one of the lowest $C(D)$ values found, indicating that it is a non-residue for relatively few primes beyond $p = 199$ when compared with the other entries in the table. Nor is the converse true: of the ten $D$ values found by *OASiS* whose $C(D)$ exceeds 4·9 (see Table 9), only two appear in Table 8 and a full half of them are members of $-3N_{173}$ but not of the more restrictive classes from $-3N_{179}$ on. Their large $C(D)$ values indicate that they have a relatively high proportion of non-residues for primes beyond 173.

| $D$ | $C(D)$ | $q(D)$ |
|---|---|---|
| −2 185 525 654 774 603 | 4.9186918 | 179 |
| −1 841 263 497 634 507 | 4.9220531 | 181 |
| −4 087 860 124 363 723 | 4.9285914 | 191 |
| −2 685 474 212 955 307 | 4.9375363 | 179 |
| −2 206 536 270 988 603 | 4.9451552 | 179 |
| −1 557 482 259 009 307 | 4.9774707 | 179 |
| −3 411 659 915 168 563 | 5.0031973 | 179 |
| −4 234 760 613 525 187 | 5.0181916 | 197 |
| −4 342 220 938 996 627 | 5.0302734 | 181 |
| −2 068 660 612 674 307 | 5.0978921 | 181 |

Table 9. $D$–values with $C(D) \geq 4{\cdot}9$.

The $D$ value −2 068 660 612 674 307 is particularly noteworthy because its $C(D)$ value of 5·0978921 is the highest ever found, surpassing the record of 5·0894316 discovered by Fung and Williams for $D = -531\ 497\ 118\ 115\ 723$. There may be other values $< 5 \times 10^{15}$ which have a larger $C(D)$, but they will be members of classes more general

than $-3N_{173}$ and correspondingly difficult to isolate. For the present, if Conjecture $F$ holds,

$$x^2 + x + 517165153168577$$

can be said to have the highest known asymptotic density of primes of any polynomial of that form. Additional support for the truth of the conjecture comes from the fact that

$$\frac{P_{517165153168577}(10^6)}{L_{517165153168577}(10^6)} = \frac{300923}{59031 \cdot 829} = 5{\cdot}0976398$$

is quite close to $C(-2068660612674307)$, as predicted.

## 11. Pseudo-cubes.

Another sieving problem, and one that is particularly well suited to the capabilities of *OASiS*, is the search for *pseudo-cubes*. These are analogous to the pseudo-squares seen earlier, and are defined to be those integers of the form $9k \pm 1$ which are (non-zero) cubic residues of all primes less than or equal to some prime $p$, but are not perfect cubes. Although a natural sieving problem, it has received far less attention than pseudo-squares; the only previous work is an unpublished table of least pseudo-cubes up to $N_{139}$ generated by Cobham using a conventional computer search [Cob68]. *OASiS* has now been used to duplicate and extend this work up to $N_{313}$. The results are summarized in Table 10. Since every number is a cubic residue modulo $p$ if $p \equiv 2 \pmod 3$, $N_p$ always equals the preceding table entry, $N_q$, unless $p \mid N_q$, so to save space the table omits entries for any $p \equiv 2 \pmod 3$ unless $N_p$ differs from $N_q$. The only instance of this phenomenon for $p \leq 313$ occurs when $N_{83} \neq N_{79}$; it apparently went unnoticed by Cobham, who set $N_{83} = 7\ 235\ 857$.

| $p$ | $N_p$ | Source | $p$ | $N_p$ | Source |
|---|---|---|---|---|---|
| *2 | 17 | Cobham | 151 | 1 612 383 137 | Stephens |
| 7 | 71 | (1968) | 157 | 1 612 383 137 | (1989) |
| 13 | 181 | computer | 163 | 7 991 083 927 | *OASiS* |
| 19 | 2 393 | search | 181 | 7 991 083 927 | |
| 31 | 3 457 | | 193 | 7 991 083 927 | |
| 37 | 5 669 | | 199 | 20 365 764 119 | |
| 43 | 74 339 | | 211 | 2 515 598 768 717 | |
| 61 | 74 339 | | 223 | 6 440 555 721 601 | |
| 67 | 166 249 | | 229 | 29 135 874 901 141 | |
| 73 | 2 275 181 | | 241 | 29 135 874 901 141 | |
| 79 | 7 235 857 | | 271 | 29 135 874 901 141 | |
| *83 | 7 298 927 | | 277 | 406 540 676 672 677 | |
| 97 | 8 721 539 | | 283 | 406 540 676 672 677 | |
| 103 | 8 721 539 | | 307 | 406 540 676 672 677 | |
| 109 | 91 246 121 | | 313 | 406 540 676 672 677 | |
| 127 | 91 246 121 | | 331 | $\geq$ 1 006 698 672 458 725 | |
| 139 | 98 018 803 | | | | |

All primes except those marked with * are $\equiv 1 \pmod 3$.

Table 10. Least pseudo-cubes.

*OASiS* searched for $N_p$ values by running a series of problems, each of which began at the point where the previous one left off and included an additional congruence. Since the congruences with $p \equiv 1 \pmod 3$ exclude about two-thirds of their residue classes, while the congruences with $p \equiv 2 \pmod 3$ exclude only one, only the former were actually loaded into the sieve; the remainder were simulated in software by the host. This greatly increased the time between solutions, and both increased the average sieving rate and reduced the number of solution candidates tested by the host. It is worth noting that a sieve with a fixed set of ring sizes would not have been able to perform this optimization. For example, *UMSU* would only have loaded 14 'desirable' congruences into hardware, whereas *OASiS* was able to load up to 28. *OASiS* also doubled its normal sieving rate to $4{\cdot}25 \times 10^8$ trials per second by taking advantage of the single residue congruence modulo 2 to optimize its search. Since some perfect cubes are also solutions to the congruences used in the problems, a simple filter program was used to reject them. However, unlike the pseudo-square case, relatively few perfect cubes were encountered, and they did not significantly affect the progress of sieving. Hence, *OASiS* was able to search the interval from 0 to $10^{15}$ in about 700 hours.

## References

[Bac89] E. Bach, 'What To Do Until the Witness Comes: Explicit Bounds for Primality Testing and Related Problems', *Mathematics of Computation*, to appear.

[Ber85] Joh. Bernoulli, 'Joh. Heinrich Lamberts ... deutscher gelehrter Briefwechsel', Vol.5. (Berlin: n.p., 1785.)

[BLS75] John Brillhart D. H. Lehmer and J. L. Selfridge, 'New Primality Criteria and Factorizations of $2^m \pm 1$', *Mathematics of Computation* **29** (1975), 620–647.

[BLSTW88] John Brillhart, D. H. Lehmer, J. L. Selfridge, Bryant Tuckerman, and S. S. Wagstaff, Jr., *Factorizations of* $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ *up to high powers'*. Contemporary Mathematics, vol. 22, 2nd ed., American Mathematical Society, 1988.)

[Bri81] John Brillhart, 'Fermat's Factoring Method and Its Variants', *Congressus Numerantium* **32** (1981), 29–48.

[BS67] John Brillhart and J. L. Selfridge, 'Some Factorizations of $2^n \pm 1$ and Related Results', *Mathematics of Computation* **21** (1967), 87–96.

[Car33] 'Machine Performs Difficult Mathematical Calculations', *Carnegie Institution of Washington—News Service Bulletin* **3**, no. 3 (March 12, 1933), 19–22.

[CEFT62] D. G. Cantor, G. Estrin, A. S. Fraenkel and R. Turn, 'A Very High-Speed Digital Number Sieve', *Mathematics of Computation* **16** (1962), 141–154.

[Cob66] Alan Cobham, 'The Recognition Problem for the Set of Perfect Squares', IBM Research Paper, R.C. 1704, 26 April 1966.

[Cob68] Alan Cobham, Letter to Drs. D. H. and E. Lehmer, 4 March 1968.

[Dic19] L. E. Dickson, *History of the Theory of Numbers*, Vol. 1, 1919. Reprint: Chelsea Publishing, New York, 1966.

[Don74] Mario Donzelli, *Theory and Design of a High Speed Electronic Sieve*. M.Sc. diss., University of Illinois, 1974.

[FW89] G. W. Fung and H. C. Williams, 'Quadratic Polynomials Which Have a High Density of Prime Values'. To appear.

[GC66] C. F. Gauss, *Disquisitiones Arithmeticae.* Translated by Arthur A. Clarke, S.J., Yale University Press, New Haven, 1966.

[Gla78] J. W. L. Glaisher, 'On factor tables, with an account of the mode of formation of the factor table for the fourth million', *Proceedings of the Cambridge Philosophical Society* **3** (1878), 99–138.

[Hal33] Marshall Hall, 'Quadratic Residues in Factorization', *Bulletin of the American Mathematical Society* **39** (1933), 758–763.

[HL23] G. H. Hardy and J. E. Littlewood, 'Partitio numerorum III: On the Expression of a Number as a Sum of Primes', *Acta Mathematica* **44** (1923), 1–70.

[HP89] Mike Hermann and Cameron Patterson,'A High Performance Mathematical Sieve', *Proceedings of the 1989 IEEE Canadian Conference on Electrical and Computer Engineering*, to appear.

[Kae33] Waldemar Kaempffert, '"Congruence Machine" Divides Numbers Quickly', *New York Times*, 12 March, 1933.

[Knu81] Donald E. Knuth, *The Art of Computer Programming. Vol. 2: Seminumerical Algorithms.* (2nd ed., Addison-Wesley, Reading, 1981.)

[Kra22] Maurice Kraitchik, *Théorie des Nombres*, Vol. 1. (Gauthier-Villars et Cie, Paris, 1922.)

[Kra24] Maurice Kraitchik, *Recherches sur la Théorie des Nombres*, Vol. 1. (Gauthier-Villars et Cie, Paris, 1924.)

[LE39] Derrick Norman Lehmer, *Factor Stencils.* Revised and extended by John D. Elder. (Carnegie Institution of Washington, September 1939.)

[Leh18] D. N. Lehmer, 'On the History of the Problem of Separating a Number into its Prime Factors', *The Scientific Monthly*, September 1918, 227–234.

[Leh28] D. H. Lehmer, 'The Mechanical Combination of Linear Forms', *American Mathematical Monthly* **35** (1928), 114–121.

[Leh33a] Derrick N. Lehmer, 'Hunting Big Game in the Theory of Numbers', *Scripta Mathematica*, March 1933, 229–235.

[Leh33b] D. H. Lehmer, 'A Photo-Electric Number Sieve', *American Mathematical Monthly* **40** (1933), 401–406.

[Leh33c] D. H. Lehmer, 'Some New Factorizations of $2^n \pm 1$', *Bulletin of the American Mathematical Society* **39** (1933), 105–108.

[Leh34] D. H. Lehmer, 'A Machine for Combining Sets of Linear Congruences', *Mathematische Annalen* **109** (1934), 661–667.

[Leh36] D. H. Lehmer, 'On the Function of $x^2 + x + A$', *Sphinx* **6** (1936), 212–214. Also *Sphinx* **7** (1937), 40.

[Leh47] D. H. Lehmer, 'On the Factors of $2^n \pm 1$', *Bulletin of the American Mathematical Society* **53** (1947), 164–167.

[Leh49] D. H. Lehmer, 'On the Converse of Fermat's Theorem II', *American Mathematical Monthly* **56** (1949), 300–309.

[Leh53] D. H. Lehmer, 'The Sieve Problem for All-Purpose Computers', *Mathematical Tables and Other Aids to Computation* **7**, no. 41 (1953), 6–14.

[Leh54] D. H. Lehmer, 'A Sieve Problem on "Pseudo-squares"', *Mathematical Tables and Other Aids to Computation* **8**, no. 48 (1954), 241–242.

[Leh66] D. H. Lehmer, 'An Announcement Concerning the Delay Line Sieve DLS-127', *Mathematics of Computation* **20** (1966), 645–646.

[Leh68] D. H. Lehmer, 'Machines and Pure Mathematics', *Computers in Mathematical Research*, ed. R. F. Churchhouse and J.-C. Herz. (North-Holland, Amsterdam, 1968.)

[Leh74] D. H. Lehmer, 'The Influence of Computing on Research in Number Theory', *The Influence of Computing on Mathematical Research and Education*, ed. Joseph P. LaSalle. Proceedings of Symposia in Applied Mathematics **20** (American Mathematical Society, Providence, 1974), 3–12.

[Leh76] D. H. Lehmer, 'Exploitation of Parallelism in Number Theoretic and Combinatorial Computation', Proceedings of 6th Manitoba Conference on Numerical Mathematics, *Congressus Numerantium* **18** (1976), 95–111.

[Leh80] D. H. Lehmer, 'A History of the Sieve Process', *A History of Computing in the Twentieth Century*, (Academic Press, 1980), 445–456.

[Leh89] D. H. Lehmer, personal interview, 16 June 1989.

[LL74] D. H. Lehmer and Emma Lehmer, 'A New Factorization Technique Using Quadratic Forms', *Mathematics of Computation* **28** (1974), 625–635.

[LLS70] D. H. Lehmer, Emma Lehmer and Daniel Shanks, 'Integer Sequences Having Prescribed Quadratic Character', *Mathematics of Computation* **24** (1970), 433–451.

[LM78] D. H. Lehmer and J. M. Masley, 'Table of Cyclotomic Class Numbers $h^*(p)$ and Their Factors for $200 < p < 521$', *Mathematics of Computation* **32** (1978), 577–582.

[MA78] K. Manders and L. Adleman, 'NP-Complete Decision Problems for Binary Quadratics', *Journal of Computer and System Sciences* **16** (1978), 168–184.

[MB75] Michael A. Morrison and John Brillhart, 'A Method of Factoring and the Factorization of $F_7$', *Mathematics of Computation* **29** (1975), 183–205.

[MW89] R. A. Mollin and H. C. Williams, 'Quadratic Non-Residues and Prime-Producing Polynomials', *Canadian Mathematical Bulletin*, to appear.

[Oak33] 'Wizard Machine Solves Mysteries in Mathematics', *Oakland Tribune*, 15 March, 1933.

[Pat83] Cameron Patterson, *Design and Use of an Electronic Sieve*, M.Sc. diss., University of Manitoba, 1983.

[Pat89] C. D. Patterson, 'The Complexity of Sieving', unpublished manuscript, 1989.

[Pom89] Carl Pomerance, 'Factoring', *Introductory Survey Lectures on Cryptology and Computational Number Theory.* (American Mathematical Society Short Course Series, American Mathematical Society, Boulder, 1989.)

[PW83] C. D. Patterson and H. C. Williams, 'A Report on the University of Manitoba Sieve Unit', *Congressus Numerantium* **37** (1983), 85–98.

[PW85] C. D. Patterson and H. C. Williams, 'Some Periodic Continued Fractions With Long Periods', *Mathematics of Computation* **44** (1985), 523–532.

[Rub83] Richard Rubinstein, 'D. H. Lehmer's Number Sieves', *The Computer Museum Report* (Spring 1983), 2–4.

[Sha59] Daniel Shanks, 'A Sieve Method for Factoring Numbers of the Form $n^2 + 1$', *Mathematical Tables and Other Aids to Computation* **13** (1959): 78–86.

[Sha60] Daniel Shanks, 'On the Conjecture of Hardy and Littlewood Concerning the Number of Primes of the Form $n^2 + a$', *Mathematics of Computation* **14** (1960), 320–332.

[Sha63] Daniel Shanks, 'Supplementary Data and Remarks Concerning a Hardy-Littlewood Conjecture', *Mathematics of Computation* **17** (1963), 188–193.

[Sha73] Daniel Shanks, 'Systematic Examination of Littlewood's Bounds on $L(1, \chi)$', *Analytic Number Theory*, ed. Harold C. Diamond, *Proceedings of Symposia in Pure Mathematics* **24** (American Mathematical Society, Providence, 1973), 267–283.

[SW88] A. J. Stephens and H. C. Williams, 'Computation of Real Quadratic Fields with Class Number One', *Mathematics of Computation* **51** (1989), 809–824.

[Wil78] H. C. Williams, 'Primality Testing on a Computer', *Ars Combinatoria* **5** (1978), 127–185.

[Wil81] H. C. Williams, 'A Numerical Investigation into the Length of the Period of the Continued Fraction Expansion of $\sqrt{D}$', *Mathematics of Computation* **36** (1981), 593–601.

*Department of Computer Science, University of Manitoba, Manitoba, CANADA R3T 2N2.*

5

# ALGORITHMS FOR FINITE FIELDS

H. W. Lenstra, Jr.*

In this paper, we survey the complexity status of some fundamental algorithmic problems concerning finite fields. In particular, we consider the following two questions: given a prime number $p$ and a positive integer $n$, construct explicitly a field that is of degree $n$ over the prime field of $p$ elements; and given two such fields, construct an explicit field isomorphism between them. For both problems there exist good probabilistic algorithms. The situation is more complicated if deterministic algorithms are required.

## 1. Introduction.

Every finite field has cardinality $p^n$ for some prime number $p$ and some positive integer $n$. Conversely, if $p$ is a prime number and $n$ a positive integer, then there exists a field of cardinality $p^n$, and any two fields of cardinality $p^n$ are isomorphic. These results are due to E. H. Moore (1893) [15]. In this paper, we discuss the complexity aspects of two algorithmic problems that are suggested by this theorem, and of two related problems.

*Constructing finite fields.* We say that a finite field is *explicitly given* if, for some basis of the field over its prime field, we know the product of any two basis elements, expressed in the same basis. Let, more precisely, $p$ be a prime number and $n$ a positive integer. Then by *explicit data* for a finite field of cardinality $p^n$ we mean a system of $n^3$ elements $(a_{ijk})_{i,j,k=1}^n$ of the prime field $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$, such that $\mathbf{F}_p^n$ becomes a field with the ordinary addition and multiplication by elements of $\mathbf{F}_p$, and the multiplication determined by

$$e_i e_j = \sum_{k=1}^{n} a_{ijk} e_k,$$

where $e_1, e_2, \ldots, e_n$ denotes the standard basis of $\mathbf{F}_p^n$ over $\mathbf{F}_p$.

The problem of constructing finite fields is the following: given a prime number $p$ and a positive integer $n$, find explicit data for a finite field of cardinality $p^n$. The elements of $\mathbf{F}_p$ are to be represented in the conventional way as integers modulo $p$. The current complexity status of this problem is discussed in Section 2.

*Finding isomorphisms between finite fields.* Given a prime number $p$, a positive integer $n$, and two sets of explicit data $(a_{ijk})_{i,j,k=1}^n$, $(a'_{ijk})_{i,j,k=1}^n$ for finite fields of cardinality $p^n$, find an isomorphism between these fields. The isomorphism is to be represented by means of its matrix on the given bases of the fields over the prime

* The author was supported by NSF contract DMS 87-06176.

field; more precisely, we ask for an invertible $n \times n$ matrix $(b_{ij})_{i,j=1}^{n}$ over $\mathbf{F}_p$ with the property that

$$\sum_{k=1}^{n} a_{ijk} b_{km} = \sum_{k,l=1}^{n} b_{ik} b_{jl} a'_{klm}$$

for all $i, j, m = 1, 2, \ldots, n$. This problem is discussed in Section 3.

*Irreducibility testing.* Given explicit data for a finite field $E$, and a non-zero polynomial $f \in E[X]$, the problem is to decide whether $f$ is irreducible in $E[X]$. This problem is the analogue of the *primality testing* problem, with the ring of integers $\mathbf{Z}$ replaced by the ring $E[X]$. We refer to Section 4 for a brief discussion of the results that are known.

*Factoring polynomials over finite fields.* Given explicit data for a finite field $E$, and a non-zero polynomial $f \in E[X]$, the problem is to determine the decomposition of $f$ into irreducible factors in $E[X]$. This is the analogue of the problem of factoring integers, with $\mathbf{Z}$ replaced by $E[X]$. In Section 5, we survey the main results that have been obtained on this problem.

Our main interest will be in the *running times* of the algorithms that have been proposed for the above four problems. In particular, we are interested in whether these algorithms run in *polynomial time*, i. e. in time $(n + \log p)^{O(1)}$ for the first two problems and in time $(\deg f + \log \#E)^{O(1)}$ for the last two. We will not be concerned with the problem of obtaining good values for the $O$-constants, and they will be left unspecified.

To do justice to the results that have been obtained, it is appropriate to distinguish three types of algorithms. The first are the deterministic algorithms for which the running time bounds have been proved rigorously. From a mathematical point of view, these are the most satisfactory algorithms.

Secondly, we will consider deterministic algorithms for which the running time bounds have only been established on the assumption of the *generalized Riemann hypothesis*, which asserts that all non-trivial zeros of the zeta function of any algebraic number field (see [10], Chapter VIII) have real part $\frac{1}{2}$. In other contexts, such as primality testing, one also encounters algorithms of which the *correctness* depends on the truth of the generalized Riemann hypothesis. We shall not encounter such algorithms in this paper.

The third type of algorithms are the *probabilistic* ones. These algorithms employ a random number generator and the random numbers that are drawn influence the course of the algorithm. Both the outcome of the algorithm and its running time have therefore, for each given input, a *distribution*. When we speak about the running time of a probabilistic algorithm, we shall mean the time that is needed to let the algorithm terminate successfully with probability at least $\frac{1}{2}$; to increase this success probability one can perform the algorithm several times. The time that the random number generator may need is not counted. We shall not be concerned with the problem of minimizing the number of calls that are made to the random number generator (see [4]).

Again, in primality testing one also encounters algorithms that are probabilistic in a different sense; namely, not the running time of the algorithm but the correctness of the answer is subject to uncertainty. We shall not have occasion to deal with such

algorithms in this paper: each algorithm that gives an answer gives provably a *correct* answer.

In practical circumstances, one is usually willing to use probabilistic algorithms instead of deterministic ones. As we shall see, there exists for each of our four problems a probabilistic algorithm that runs in polynomial time. This suggests that from a practical point of view all four problems may be considered to be well-solved, and this appears indeed to be the case.

Turning to deterministic algorithms, we shall see that there exist fully proved polynomial time algorithms for the second problem (*finding isomorphisms*) and the third problem (*irreducibility testing*). For the first problem (*constructing finite fields*) and the fourth problem (*factoring polynomials*) fully proved polynomial time algorithms are currently only known in special cases; e. g., the case that the characteristic $p$ is fixed, or more generally the case that $p$ is bounded by a fixed power of $n$ or $\deg f$. If we accept the truth of the generalized Riemann hypothesis, then also for the first problem (*constructing finite fields*) there is a deterministic polynomial time algorithm. This is not the case for the fourth problem (*factoring polynomials*), although many special cases have been dealt with.

## 2. Constructing finite fields.

Let $p$ be a prime number and $n$ a positive integer. If we know an irreducible polynomial $f \in \mathbf{F}_p[X]$ of degree $n$, then explicit data for a field of cardinality $p^n$ are readily calculated, since $\mathbf{F}_p[X]/f\mathbf{F}_p[X]$ is such a field. Conversely, given explicit data for a field of cardinality $p^n$, one can, in deterministic polynomial time, exhibit an element $\alpha$ in this field that has degree $n$ over $\mathbf{F}_p$, and calculate its irreducible polynomial $f$ over $\mathbf{F}_p$, which has degree $n$ (see [12], Section 2). Thus the problem of constructing explicit data for a finite field of cardinality $p^n$ is equivalent to the problem of constructing an irreducible polynomial of degree $n$ in $\mathbf{F}_p[X]$.

For $n = 2$ and $p$ odd, a clearly equivalent problem is to find, given $p$, an element $a$ of $\mathbf{F}_p$ that is not a square. This can easily be done in polynomial time if a probabilistic algorithm is allowed: draw $a$ at random, until one that satisfies $a^{(p-1)/2} = -1$ is found. The same applies if the generalized Riemann hypothesis is assumed: try all $a$ up to $2(\log p)^2$ (see [3]). However, even for this special case, there is apparently at present no hope of finding a fully proved deterministic polynomial time algorithm.

The general case can be reduced to the case that $n$ is prime in the following sense: if for each prime divisor $r$ of $n$ an irreducible polynomial of degree $r$ in $\mathbf{F}_p[X]$ is given, then an irreducible polynomial of degree $n$ can be constructed in deterministic polynomial time ([12], Theorem (1.1)).

The best fully proved deterministic algorithm for the problem of constructing finite fields is due to V. Shoup [22]. He proved that the problem can be reduced to the problem of factoring polynomials over finite fields. This is a "Turing reduction" in the sense that the construction of a single finite field requires the factorization of several polynomials, which are computed in the course of the algorithm. Shoup's reduction and the results of Section 5 lead to the running time bound $O\big(\sqrt{p} \cdot (n + \log p)^{O(1)}\big)$. In particular, there exists a fully proved deterministic polynomial time algorithm if $p$ is fixed, e. g. $p = 2$, or if $p$ is bounded by a fixed power of $n$.

If the generalized Riemann hypothesis is assumed, then the problem of constructing finite fields can be solved in polynomial time (see [1] and [7] (independently)). We briefly sketch the method of [1] in the case that $n$ is prime, to which the general case can be reduced, as we just saw. To find an irreducible polynomial of prime degree $n$ in $\mathbf{F}_p[X]$, one picks a small prime $q$ with the properties

$$q \equiv 1 \bmod n, \qquad p^{(q-1)/n} \not\equiv 1 \bmod q;$$

the generalized Riemann hypothesis guarantees that the least such $q$ is $(n + \log p)^{O(1)}$. Now let $H \subset \mathbf{F}_q^*$ be the subgroup of $n$-th powers and $\zeta_q$ a primitive $q$-th root of unity. Then the polynomial

$$f = \prod_{C \in \mathbf{F}_q^*/H} \left(X - \sum_{x \in C} \zeta_q^x\right)$$

lies in $\mathbf{Z}[X]$, and it can be proved that $(f \bmod p)$ is irreducible in $\mathbf{F}_p[X]$.

If probabilistic algorithms are allowed, the construction of finite fields is also possible in polynomial time. This follows, of course, from Shoup's result just mentioned, but it is easier to proceed as follows: pick $f \in \mathbf{F}_p[X]$, $\deg f = n$, at random, test $f$ for irreducibility (see Section 4), and repeat until an irreducible $f$ is found. This is efficient, because a random polynomial of degree $n$ in $\mathbf{F}_p[X]$ is irreducible with probability $(1 + O(p^{-n/2}))/n$.

The problem of constructing finite fields has an interesting companion problem: given positive integers $p$ and $n$ with $p \geq 2$ and a system of $n^3$ elements $(a_{ijk})_{i,j,k=1}^n$ of $\mathbf{Z}/p\mathbf{Z}$, decide whether these form explicit data for a field of cardinality $p^n$. For $n = 1$ this problem is equivalent to the *primality testing* problem: given an integer $p \geq 2$, decide whether $p$ is prime. For this problem no fully proved deterministic polynomial time algorithm is known. Using the techniques of [12, Section 2] one can show that primality testing is the *only* obstacle: there is a deterministic polynomial time algorithm that, given $p$, $n$, $(a_{ijk})$ as above, either proves that they do not form explicit data for a field of cardinality $p^n$, or proves that if $p$ is prime they do.

## 3. Finding isomorphisms.

Although the problem of finding an isomorphism between two explicitly given finite fields of the same cardinality is from a theoretical point of view just as fundamental as the problem of constructing finite fields, I do not believe that the problem arises in many practical circumstances. If it ever would, one would probably solve it by means of the following probabilistic algorithm. Let $E$, $E'$ be two explicitly given finite fields of cardinality $p^n$. As we mentioned in the previous section, one can find $\alpha \in E$ with $E = \mathbf{F}_p(\alpha)$ and determine the irreducible polynomial $f$ of $\alpha$ over $\mathbf{F}_p$. Finding a field isomorphism $E \to E'$ is now equivalent to finding a zero of $f$ in $E'$. Since finding a zero is equivalent to finding a linear factor, this problem can be solved by means of one of the algorithms discussed in Section 5.

The procedure just sketched, combined with the results of Section 5, shows that the problem of finding isomorphisms can be solved by means of a probabilistic algorithm in polynomial time. It was shown by S. A. Evdokimov [7] that it can be done by means of a deterministic polynomial time algorithm if the truth of the generalized

Riemann hypothesis is assumed. These two results were superseded recently, when it was proved that there is a fully proved deterministic polynomial time algorithm for finding isomorphisms between finite fields [12].

To illustrate the idea we consider the case $n = 2$, $p > 2$. First, let explicit data for a finite field $E$ of cardinality $p^2$ be given. It is not difficult to construct an element $\alpha \in E$ with $E = \mathbf{F}_p(\alpha)$ and $\alpha^2 = a \in \mathbf{F}_p$. Then $a$ is not a square in $\mathbf{F}_p$, so $a^{(p-1)/2} = -1$. Writing $p - 1 = 2^t v$ with $v$ odd, and replacing $\alpha$, $a$ by $\alpha^v$, $a^v$, we may assume that $a^{2^{t-1}} = -1$.

Now let another finite field $E'$ of cardinality $p^2$ be explicitly given. In a similar way we can write $E' = \mathbf{F}_p(\beta)$, where $\beta^2 = b \in \mathbf{F}_p$ and $b^{2^{t-1}} = -1$. To find an isomorphism $E \to E'$ it suffices to find $c \in \mathbf{F}_p$ with $a = bc^2$, since then we can map $\alpha$ to $\beta c$.

The element $c$ can be found by an iterative procedure that is due to A. Tonelli (1891) ([6], page 215). The iteration starts with $c = 1$. In each iteration step, one first determines the least non-negative integer $i$ for which $a^{2^i} = (bc^2)^{2^i}$. If $i = 0$ then $a = bc^2$, and the algorithm terminates. If $i > 1$, then we replace $c$ by $cb^{2^{t-i-1}}$ and iterate.

To prove that the algorithm is correct and runs in polynomial time, it suffices to observe that at the beginning of the algorithm, when $c = 1$, one has $i \leq t - 1$, and that $i$ decreases by at least 1 in every iteration step. To prove the latter assertion, note that

$$a^{2^{i-1}} = -(bc^2)^{2^{i-1}} = \left(b(cb^{2^{t-i-1}})^2\right)^{2^{i-1}}.$$

The main obstacle in extending this algorithm to the case of general $n$ is the impossibility of writing a general $n$-th degree extension of $\mathbf{F}_p$ in the form $\mathbf{F}_p(a^{1/n})$, with $a \in \mathbf{F}_p$. It turns out that it is sufficient to consider the case that $n$ is prime, $n \neq p$. Evdokimov [7] deals with this problem by passing to the $n$-th cyclotomic extension of $\mathbf{F}_p$ and using Kummer theory. It is for the construction of this cyclotomic extension that the generalized Riemann hypothesis is needed. In [12], this problem is circumvented by using cyclotomic *ring* extensions, which can be obtained without any unproved hypotheses, and developing the required Kummer theory for ring extensions.

The problem of finding isomorphisms between finite fields can be generalized in several ways. For example, one may ask for an *embedding* of one explicitly given finite field into another; or for *all* such embeddings; and one may add the restriction that the embeddings are the identity on an explicitly given common subfield. All these variants can be dealt with in a straightforward way by means of the techniques of [12] (see in particular [12], Section 2).

## 4. Irreducibility testing.

Let $E$ be an explicitly given finite field, and let $q = \#E$. The fact that irreducibility testing in $E[X]$ can be done by means of a deterministic polynomial time algorithm is an immediate consequence of the well known formula

$$X^{q^m} - X = \prod_g g,$$

where $m$ is any positive integer and $g$ ranges over the set of all monic irreducible polynomials in $E[X]$ of degree dividing $m$. It follows from this formula that a polynomial $f \in E[X]$ is irreducible if and only if

$$\gcd(X^{q^i} - X, f) = 1 \quad \text{for} \quad 1 \leq i \leq [(\deg f)/2],$$

and if and only if we have

$$X^{q^{\deg f}} \equiv X \bmod f$$

and

$$\gcd(X^{q^{(\deg f)/r}} - X, f) = 1$$

for each prime $r$ dividing $\deg f$. To see that each of these irreducibility criteria gives rise to a deterministic polynomial time irreducibility test it suffices to show how to calculate $X^{q^i} \bmod f$ for $i \leq \deg f$; the necessary greatest common divisors can then be calculated by means of the Euclidean algorithm. To calculate $X^{q^i} \bmod f$, one can use the well-known algorithm that depends on the binary expansion of $q^i$ (see [9], Section 4.6.3). Alternatively, one does this only for $i = 1$ in order to obtain the $\deg f \times \deg f$ matrix that expresses the $E$-linear map

$$Q: E[X]/fE[X] \to E[X]/fE[X], \qquad Q(x) = x^q \quad \text{for all } x,$$

on the basis $1, X, \ldots, X^{(\deg f)-1}$ of $E[X]/fE[X]$ over $E$. The coefficients of the remainder of $X^{q^i}$ modulo $f$ can be read from the $i$-th power of this matrix.

Once the matrix that describes the map $Q$ has been calculated, one can use it in a different way to test $f$ for irreducibility. Namely, $f$ is irreducible if and only if

$$\gcd(f, f') = 1 \quad \text{and} \quad \operatorname{rank}(Q - \mathrm{id}) = (\deg f) - 1,$$

where id denotes the identity function from $E[X]/fE[X]$ to itself. For a proof of this fact, and a comparison of the different irreducibility tests that we discussed, see [11], Sections 4 and 5.

The generalized Riemann hypothesis or random number generators do not enter into any efficient algorithm for irreducibility testing in $E[X]$ that I am aware of.

## 5. Factoring polynomials.

Let $E$ be an explicitly given finite field, $p$ its characteristic, and $f \in E[X]$ a non-zero polynomial. In this section, we discuss algorithms for factoring $f$ into irreducible factors. Although our interest is mainly theoretical, some of the ideas that will be discussed do have practical value. For a discussion of these aspects we refer to [11].

It is a fundamental consequence of Berlekamp's factoring algorithm ([11], Section 4), that there is a deterministic polynomial time algorithm that reduces the problem of factoring $f$ in $E[X]$ to the problem of factoring a polynomial $g \in \mathbf{F}_p[X]$ into irreducible factors in $\mathbf{F}_p[X]$, in the special case that it is known that all those factors are *linear* and *distinct*; i. e., $g$ divides $X^p - X$. This reduction is of a simpler sort than the "Turing" reduction of Shoup that we mentioned in Section 2: given $E$ and $f$, the reduction

produces in polynomial time a single polynomial $g$ as above, such that knowledge of the linear factors of $g$ in $\mathbf{F}_p[X]$ enables one to find the full factorization of $f$ in $E[X]$ in polynomial time. For a description of this reduction we refer to [11], Sections 3 and 4.

For the remainder of this section, we assume that $g \in \mathbf{F}_p[X]$ is a polynomial with $X^p \equiv X \bmod g$, and we put $n = \deg g$. We are interested in algorithms to find all linear factors of $g$. Equivalently, we may ask for all zeros of $g$ in $\mathbf{F}_p$. The problem is trivial if $n = 1$. We will often tacitly assume that $n > 1$, and in that case we may also be satisfied with obtaining a *splitting* of $g$, i.e. a decomposition $g = g_1 g_2$ into polynomials of lower degrees; one can then proceed recursively with $g_1$ and $g_2$.

For small $p$, an obvious approach is to try all elements of $\mathbf{F}_p$ as possible zeros of $g$. This works in time $O\big(p(n + \log p)^{O(1)}\big)$, and it shows that there is a deterministic algorithm that factors $f$ in $E[X]$ in time $O\big(p(\deg f + \log \#E)^{O(1)}\big)$.

In these results, one can replace the factor $p$ by $\sqrt{p}$ if one uses a faster method to check all elements of $\mathbf{F}_p$ as possible zeros of $g$. This can be done by means of a device that was used by Strassen in the context of factoring integers ([23], Section 6). Let $s$ be the least integer $> \sqrt{p}$, and let $h \in \mathbf{F}_p[X]$ be the polynomial

$$h = \prod_{i=1}^{s} (X - i).$$

For each integer $j$ we have

$$h(X - js) = \prod_{i=js+1}^{(j+1)s} (X - i).$$

Therefore the zeros of $\gcd(h(X - js), g)$ are precisely the zeros of $g$ among $js + 1$, $js + 2$, ..., $js + s$. Hence, if we could calculate all $h(X - js) \bmod g$ for $0 \leq j < s$ in time $\sqrt{p} \cdot (n + \log p)^{O(1)}$, then we could calculate all greatest common divisors $\gcd(h(X - js), g)$ and, for those values of $j$ for which the gcd is non-trivial, check the elements $js + 1$, $js + 2$, ..., $js + s$ one by one. This would give us all zeros of $g$ in time $\sqrt{p} \cdot (n + \log p)^{O(1)}$.

To calculate all $h(X - js) \bmod g$ efficiently we make use of methods that depend on the fast Fourier transform. The coefficients of $h$ can be computed in time $\sqrt{p} \cdot (\log p)^{O(1)}$ (see [2]). If $x$ denotes the image of $X$ in $\mathbf{F}_p[X]/g\mathbf{F}_p[X]$, then

$$h(x - js) = \big(h(X - js) \bmod g\big),$$

so it suffices to calculate the $s$ values

$$h(x), \quad h(x - s), \quad h(x - 2s), \quad \ldots, \quad h(x - (s - 1)s) \in \mathbf{F}_p[X]/g\mathbf{F}_p[X]$$

of the $s$-th degree polynomial $h$. This can be done in time $\sqrt{p} \cdot (n + \log p)^{O(1)}$, as required, by the results in [2].

The storage requirement of the algorithm just sketched is at least of the order $\sqrt{p}$. Below we shall mention a deterministic algorithm that achieves the same time bound $O(\sqrt{p} \cdot (n + \log p)^{O(1)})$, but with storage requirement only $(n + \log p)^{O(1)}$.

If $p$ is large, and in particular odd, then the following probabilistic algorithm splits $g$ in polynomial time. Pick $a \in \mathbf{F}_p$ at random. The polynomial $g$ divides

$$X^p - X = (X + a)^p - (X + a) = (X + a) \cdot \big((X + a)^{(p-1)/2} - 1\big) \cdot \big((X + a)^{(p-1)/2} + 1\big),$$

so we can hope to split $g$ by

$$g = \gcd(X + a,\, g) \cdot \gcd\big((X + a)^{(p-1)/2} - 1,\, g\big) \cdot \gcd\big((X + a)^{(p-1)/2} + 1,\, g\big).$$

The first gcd is usually trivial, unless $g(-a) = 0$, which can be checked directly. Before calculating the other gcd's, one should replace $(X + a)^{(p-1)/2}$ by its remainder upon division by $g$, which can be calculated by repeated squarings and multiplications modulo $g$.

If $n > 1$ then the above splitting is non-trivial for at least half of all $a \in \mathbf{F}_p$ (see [11], Lemma 3.3). This implies that the probabilistic algorithm just described runs in polynomial time.

This probabilistic algorithm can be turned into a deterministic one that runs in time $O(\sqrt{p}) \cdot (n + \log p)^{O(1)}$ by trying $a = 0, 1, 2, \ldots$, in succession, and showing that the least successful $a$ is at most $\sqrt{p} \cdot \log p$. This observation is due to Shoup [19, 20].

If we allow the generalized Riemann hypothesis in the running time analysis then the results do not become much better. Shoup [21] proved that in this case the factor $O(\sqrt{p})$ can be replaced by $\sqrt{S(p-1)}$, where $S(p-1)$ denotes the largest prime divisor of $p - 1$. Similar results involving $S(p - 1)$ have been proved by various authors [13, 14, 17, 24]. The number $p - 1$ occurs in these results because it is the order of the multiplicative group $\mathbf{F}_p^*$. Other groups can also be used [5, 18]. As a byproduct, Bach and Von zur Gathen [5] obtain, without any unproved hypotheses, the amusing result that $g$ can be factored by a deterministic algorithm in time $(n + \log p)^{O(1)}$ if $p$ is a Mersenne prime.

Rónyai [16] proved the following result, assuming the truth of the generalized Riemann hypothesis: if $r$ is a prime number dividing $n$, then a non-trivial factor of $g$ can be found in time $(n^r + \log p)^{O(1)}$. The same result can be proved without assuming the generalized Riemann hypothesis if an irreducible polynomial of degree $r$ over $\mathbf{F}_p$ is known; as we saw in Section 2, such a polynomial can be found in polynomial time if the generalized Riemann hypothesis is true.

It follows from Rónyai's result, under assumption of the generalized Riemann hypothesis, that polynomials $f \in E[X]$ with a bounded number of irreducible factors can be factored by a deterministic algorithm in polynomial time.

Several authors obtained results about factoring $(h \bmod p)$ in $\mathbf{F}_p[X]$ for polynomials $h \in \mathbf{Z}[X]$ of which the Galois group of $h$ over $\mathbf{Q}$ is subjected to various restrictions [7, 8, 18].

## References


1. L. M. Adleman, H. W. Lenstra, Jr., 'Finding irreducible polynomials over finite fields', *Proc. 18th Annual ACM Symp. on Theory of Computing (STOC)*, (1986), 350–355.
2. A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The design and analysis of computer algorithms*. (Addison-Wesley, Reading, 1974.)
3. E. Bach, 'Explicit bounds for primality testing and related problems', *Math. Comp.*, to appear.
4. E. Bach, V. Shoup, 'Factoring polynomials using fewer random bits', Computer Sciences Department, University of Wisconsin, Madison, 1988.
5. E. Bach, J. von zur Gathen, 'Deterministic factorization of polynomials over special finite fields', Technical Report #799, Computer Sciences Department, University of Wisconsin, Madison, 1988.
6. L. E. Dickson, *History of the theory of numbers*, vol. I. (Carnegie Institute, Washington, 1919; Chelsea, New York, 1971.)
7. S. A. Evdokimov, 'Efficient factorization of polynomials over finite fields and generalized Riemann hypothesis', preprint, Leningrad Institute for Informatics and Automatization, 1986.
8. M.-D. Huang, 'Riemann hypothesis and finding roots over finite fields', *Proc. 17th Annual ACM Symp. on Theory of Computing (STOC)*, (1985), 121–130.
9. D. E. Knuth, *The art of computer programming*, vol. 2. (Second edition, Addison-Wesley, Reading, 1981.)
10. S. Lang, *Algebraic number theory*. (Addison-Wesley, Reading, 1970.)
11. A. K. Lenstra, 'Factorization of polynomials', *Computational methods in number theory*, H. W. Lenstra, Jr., and R. Tijdeman (eds), Mathematical Centre Tracts **154/155** (Mathematisch Centrum, Amsterdam, 1982), 169–198.
12. H. W. Lenstra, Jr., 'Finding isomorphisms between finite fields', to appear.
13. M. Mignotte, C. P. Schnorr, 'Calcul déterministe des racines des polynômes dans un corps fini', *C. R. Acad. Sci. Paris Sér. I Math.* **306** (1988), 467–472.
14. R. T. Moenck, 'On the efficiency of algorithms for polynomial factoring', *Math. Comp.* **31** (1977), 235–250.
15. E. H. Moore, 'A doubly-infinite system of simple groups', *Bull. New York Math. Soc.* **3** (1893), 73–78; Mathematical Papers read at the Congress of Mathematics (Chicago, 1893), 208–242, Chicago, 1896.
16. L. Rónyai, 'Factoring polynomials over finite fields', *Proc. 28th IEEE Symp. on Foundations of Computer Science (FOCS)*, (1987), 132–137.
17. L. Rónyai, 'Factoring polynomials modulo special primes', *Combinatorica*, to appear.
18. L. Rónyai, 'Galois groups and factoring polynomials over finite fields', to appear.
19. V. Shoup, 'On the deterministic complexity of factoring polynomials over finite fields', Computer Sciences Technical Report #782, University of Wisconsin, Madison, 1988.

20. V. Shoup, *Removing randomness from computational number theory*, Ph. D. thesis, Computer Sciences Technical Report #865, University of Wisconsin, Madison, 1989.

21. V. Shoup, 'A theorem on factoring polynomials over finite fields', Computer Sciences Technical Report #866, University of Wisconsin, Madison, 1989.

22. V. Shoup, 'New algorithms for finding irreducible polynomials over finite fields', *Math. Comp.*, to appear.

23. V. Strassen, 'Einige Resultate über Berechnungskomplexität', *Jahresber. Deutsch. Math.-Verein.* **78** (1976), 1–8.

24. J. von zur Gathen, 'Factoring polynomials and primitive elements for special primes', *Theoret. Comput. Sci.* **52** (1987), 77–89.

*Department of Mathematics, University of California, Berkeley, California 94720, USA.*

6

# NOTES ON CONTINUED FRACTIONS AND RECURRENCE SEQUENCES

**A. J. van der Poorten***

## Introduction.

The purpose of these notes is to support the papers included in this volume by providing self-contained summaries of related mathematics emphasising those aspects actually used or required. I have selected two topics that are easily described from first principles yet for which it is peculiarly difficult to find congenial introductions that warrant citing.

The theory of continued fractions is less widely known than it should be; yet one can readily retrieve its fundamental results with little more than the ability to multiply $2 \times 2$ matrices.

A linear feedback shift register (*LFSR*) is just a recurrence sequence defined over the field $\mathbf{F}_2$ of 2 elements; or, better, its generating function is a rational function defined over $\mathbf{F}_2$. It seems useful to interpret familiar activities in the creation of stream ciphers in classical terms if only to establish a dictionary translating the jargon of stream ciphers into the language of mathematics.

## 1. Continued Fractions.

### 1.1. *An introduction to continued fractions.*

A continued fraction is an object of the shape

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots}}},$$

which we denote in a space-saving flat notation by

$$[a_0, a_1, a_2, a_3, \ldots].$$

Virtually all principles of the subject are revealed by the following correspondence:

* Research partially supported by the Australian Research Council.

**Proposition 1** (Fundamental Correspondence). *Given a sequence* $a_0, a_1, a_2, \ldots,$

$$\begin{pmatrix} a_0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_1 & 1 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_n & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{pmatrix} \quad \textit{for } n = 0,\ 1,\ 2,\ \ldots \tag{1}$$

*if and only if*

$$\frac{p_n}{q_n} = [a_0, a_1, \ldots, a_n] \quad \textit{for } n = 0,\ 1,\ 2,\ \ldots.$$

*Proof.* This correspondence is readily established by a thoughtful inductive argument. Notice firstly that the sequence of *partial quotients* $(a_h)$ defines the sequences $(p_h)$ and $(q_h)$ appearing in the first column of the matrix product. Since the empty product of $2 \times 2$ matrices is the identity matrix, we are committed to

$$\begin{pmatrix} p_{-1} & p_{-2} \\ q_{-1} & q_{-2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2}$$

We may then readily verify by induction on $n$ that the second column of the product indeed has the alleged entries. Thus we have the recursive formulae

$$\begin{aligned} p_{n+1} &= a_{n+1}p_n + p_{n-1} \\ q_{n+1} &= a_{n+1}q_n + q_{n-1}\,. \end{aligned} \tag{3}$$

We verify the principal claim by induction on the *number* $n+1$ of matrices appearing on the left in the product. The claim is easily seen to be true for $n = 0$ since, indeed $p_0 = a_0$ and $q_0 = 1$. Accordingly, we suppose that

$$\begin{pmatrix} a_1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_2 & 1 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_n & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} x_n & x_{n-1} \\ y_n & y_{n-1} \end{pmatrix}$$

if and only if

$$\frac{x_n}{y_n} = [a_1, a_2, \ldots, a_n],$$

noting that this is a case of just $n$ matrices.

But

$$\begin{pmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{pmatrix} = \begin{pmatrix} a_0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x_n & x_{n-1} \\ y_n & y_{n-1} \end{pmatrix} = \begin{pmatrix} a_0x_n + y_n & a_0x_{n-1} + y_{n-1} \\ x_n & x_{n-1} \end{pmatrix}$$

entails

$$\frac{p_n}{q_n} = a_0 + \frac{y_n}{x_n} = a_0 + \frac{1}{[a_1, \ldots, a_n]} = [a_0, a_1, \ldots, a_n], \tag{4}$$

verifying the claim by induction.

Taking determinants in the correspondence immediately yields the fundamental formula

$$p_nq_{n-1} - p_{n-1}q_n = (-1)^{n+1} \quad \text{or} \quad \frac{p_n}{q_n} = \frac{p_{n-1}}{q_{n-1}} + \frac{(-1)^{n-1}}{q_{n-1}q_n}. \tag{5}$$

It is then immediate that

$$\frac{p_n}{q_n} = a_0 + \frac{1}{q_0 q_1} - \frac{1}{q_1 q_2} + \cdots + \frac{(-1)^{n-1}}{q_{n-1} q_n}. \tag{6}$$

Almost invariably, but not always, in the sequel the $a_i$ are positive integers, excepting $a_0$ which may have any sign.

It follows that we can make sense of nonterminating continued fractions

$$\alpha = [a_0, a_1, \ldots],$$

for evidently,

$$\alpha = a_0 + \frac{1}{q_0 q_1} - \frac{1}{q_1 q_2} + \cdots = a_0 + \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{q_{n-1} q_n} \tag{7}$$

and, this being an alternating series of terms with decreasing size, the series converges to some real number $\alpha$.

In this context the terminating continued fractions

$$\frac{p_n}{q_n} = [a_0, a_1, \ldots, a_n] \qquad (n = 0, 1, 2, \ldots)$$

are called *convergents* of $\alpha$ and the tails

$$\alpha_{n+1} = [a_{n+1}, a_{n+2}, \ldots] \tag{8}$$

are its *complete quotients*. Note that we have, formally,

$$\alpha = [a_0, a_1, \ldots, a_n, \alpha_{n+1}] \qquad (n = 0, 1, 2, \ldots). \tag{9}$$

These remarks immediately yield the approximation properties of the convergents. For we have

$$\alpha - \frac{p_n}{q_n} = (-1)^n \left( \frac{1}{q_n q_{n+1}} - \frac{1}{q_{n+1} q_{n+2}} + \cdots \right). \tag{10}$$

This shows that the sequence $(q_n \alpha - p_n)$ alternates in sign and that, in absolute value, it converges monotonically to zero. Less precisely, we see that

$$\left| \alpha - \frac{p_n}{q_n} \right| < \frac{1}{q_n q_{n+1}}$$

and, recalling from (3) that $q_{n+1} = a_{n+1} q_n + q_{n-1}$, we have yet less accurately that

$$\left| \alpha - \frac{p_n}{q_n} \right| < \frac{1}{a_{n+1} q_n^2}.$$

Thus a convergent yields an exceptionally sharp approximation when the *next* partial quotient is exceptionally large. For example, we remember that

$$\pi = [3, 7, 15, 1, 292, 1, \ldots]$$

and noting

$$[3,7] = 22/7\,, \qquad [3,7,15,1] = 355/113\,,$$

we have

$$\left|\pi - \frac{22}{7}\right| < \frac{1}{15.7^2}\,, \qquad \left|\pi - \frac{355}{113}\right| < \frac{1}{292.113^2}\,,$$

making appropriate the popularity of these rational approximations to $\pi$.

Because the sequence $(q_n\alpha - p_n)$ alternates in sign, it is clear that one need only consider every second convergent if one is interested in just those approximations below (underestimating), respectively above (overestimating) $\alpha$. It is an interesting exercise to confirm that if, say, $[a_0, a_1, \ldots, a_n]$ is a convergent overestimating $\alpha$, and $a_n > 1$, then the *intermediate convergent* $[a_0, a_1, \ldots, a_n - 1]$ is a quite good underestimate for $\alpha$.

We now return to the beginning. Noting that

$$\alpha = [a_0, a_1, \ldots] = a_0 + \frac{1}{[a_1, a_2, \ldots]}$$

we see that

$$a_0 = \lfloor \alpha \rfloor$$

and

$$\alpha_1 = [a_1, a_2, \ldots] = (\alpha - a_0)^{-1}\,.$$

The general step in the continued fraction algorithm is

$$a_n = \lfloor \alpha_n \rfloor \text{ and } \alpha_{n+1} = (\alpha_n - a_n)^{-1} \qquad (n = 0,\ 1,\ 2,\ \ldots)$$

An infinite partial quotient terminates the expansion. Since

$$[a_0, a_1, \ldots, a_n]$$

is rational it is evident that if the continued fraction of some $\alpha$ terminates then that $\alpha$ is rational. Conversely, since $p_n$ and $q_n$ are relatively prime from (5) and the sequences $(|p_n|)$ and $(q_n)$ are both monotonic increasing from (3), it follows that if $\alpha$ is rational then its continued fraction does terminate. Indeed, for a rational $\alpha = b/c$, the continued fraction algorithm is just the Euclidean algorithm. Thus

$$\begin{aligned}
b &= a_0 c + c_1 && 0 \le c_1 < c \\
c &= a_1 c_1 + c_2 && 0 \le c_2 < c_1 \\
c_1 &= a_2 c_2 + c_3 && 0 \le c_3 < c_2 \\
&\vdots \\
c_{n-1} &= a_n c_n
\end{aligned}$$

corresponds to

$$\frac{b}{c} = [a_0, a_1, \ldots, a_n] \text{ with } \gcd(b, c) = d = c_n$$

and explains the term 'partial quotient'. Since $b/c = p_n/q_n$ and $\gcd(p_n, q_n) = 1$, we must have $dp_n = b$ and $dq_n = c$. Moreover, by (5)

$$p_n q_{n-1} - p_{n-1} q_n = (-1)^{n+1} \text{ so } bq_{n-1} - cp_{n-1} = (-1)^{n-1} d,$$

and this displays the greatest common divisor as a $\mathbf{Z}$-linear combination of $b$ and $c$. Since $|p_{n-1}| < |p_n|$ and $q_{n-1} < q_n$ it follows that this combination is minimal.

This is an appropriate point at which to remark that it will be an easy matter to generalise the continued fraction algorithm to completions of the quotient field of any Euclidean domain — for $\mathbf{R}$ is just the completion with respect to the usual absolute value $|\ |$ of the quotient field $\mathbf{Q}$ of the rational integers $\mathbf{Z}$. An evident example replaces $\mathbf{Z}$ by the ring of polynomials over some field and $\mathbf{R}$ by the field of Laurent series over that field.

The entire matter of continued fractions of real numbers could have been introduced using the following

**Proposition 2.** *A rational $p'/q'$ (with $\gcd(p', q') = 1$) is a convergent of $\alpha$ if and only if*

$$|q'\alpha - p'| < |q\alpha - p| \text{ for all integers } q < q' \text{ and } p.$$

*Proof.* Suppose, as we may, that $q_{n-1} < q < q_n$. Then, by the unimodularity of the matrix

$$\begin{pmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{pmatrix}$$

there are integers $a$ and $b$ so that

$$\begin{aligned} ap_{n-1} + bp_n &= p \\ aq_{n-1} + bq_n &= q \end{aligned}$$

and, necessarily, $ab < 0$. Multiplying by $\alpha$ and subtracting yields

$$q\alpha - p = a(q_{n-1}\alpha - p_{n-1}) + b(q_n\alpha - p_n).$$

But, by (10), we have $(q_{n-1}\alpha - p_{n-1})(q_n\alpha - p_n) < 0$. Hence

$$|q\alpha - p| = |a||q_{n-1}\alpha - p_{n-1}| + |b||q_n\alpha - p_n|,$$

and plainly $|q\alpha - p| > |q_n\alpha - p_n|$ as asserted.

The proposition asserts that the convergents of $\alpha$ are exactly those quantities yielding the *locally best approximations* to $\alpha$. It is an interesting exercise to develop the entire theory (working backwards in the present program) from the notion of locally best approximation; once again, the formula (5) plays the fundamental role.

Moreover, we have the following useful criterion:

**Proposition 3.** *If $|q\alpha - p| < 1/2q$, then $p/q$ is a convergent of $\alpha$.*

Note that this condition is sufficient but not nececessary.

*Proof.* By proposition 2, it suffices to show that $|q\alpha - p|$ is a locally best approximation. To see that is so take integers $r$, $s$ with $0 < s < q$ and notice that

$$1 \leq |qr - ps| = |s(q\alpha - p) - q(s\alpha - r)| \leq s|q\alpha - p| + q|s\alpha - r| \\ \leq \frac{s}{2q} + q|s\alpha - r|\,.$$

So certainly $q|s\alpha - r| \geq 1 - s/2q > 1/2$ and it follows that $|q\alpha - p| < |s\alpha - r|$ as claimed.

Notice that it is just this criterion that is applied by Worley [7] in Section 8 of his paper in these Proceedings.

I conclude by applying the matrix correspondence to develop a formulaire: From

$$\alpha = [a_0, a_1, \ldots, a_n, \alpha_{n+1}] \longleftrightarrow \begin{pmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{pmatrix} \begin{pmatrix} \alpha_{n+1} & 1 \\ 1 & 0 \end{pmatrix}$$

we have

$$\alpha = \frac{\alpha_{n+1}p_n + p_{n-1}}{\alpha_{n+1}q_n + q_{n-1}} \text{ and } \alpha_{n+1} = -\frac{q_{n-1}\alpha - p_{n-1}}{q_n\alpha - p_n}\,.$$

Transposition of

$$\begin{pmatrix} a_0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a_1 & 1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} a_n & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{pmatrix}$$

yields

$$\frac{p_n}{p_{n-1}} = [a_n, a_{n-1}, \ldots, a_0] \text{ and } \frac{q_n}{q_{n-1}} = [a_n, a_{n-1}, \ldots, a_1]\,.$$

Hence

$$-\alpha_{n+1} = \frac{-\alpha q_{n-1} + p_{n-1}}{\alpha q_n + p_n} \longleftrightarrow -\alpha_{n+1} = [a_n, a_{n-1}, \ldots, a_0, -\alpha]\,.$$

1.2. *Applying the theory of continued fractions.*

Suppose one suspects that some computed number is actually some nice neat vulgar fraction. For example, one's calculator has produced the number

$$\alpha = 2.117647059\ldots$$

Expanding $\alpha$ as a continued fraction yields

$$\alpha \approx [2, 8, 2] = 36/17$$

up to the accuracy of the data, essentially verifying the suspicion that $\alpha = 36/17$. Thus the continued fraction algorithm provides an efficient method of converting a decimal to a vulgar fraction reversing the more usual algorithm that converts a fraction to a decimal. The point is, of course, that instead of having to test all possible rational approximations one only meets very good rational approximations.

This is the spirit of Wiener's cryptanalytic attack on the use of short secret exponents in the *RSA* cipher mentioned by Lidl [4] in these Proceedings. Recall that $n = uv$ (I use $u$ and $v$ for the unknown primes since I wish to mind my $p$'s and $q$'s for convergents) and one hopes to guess $(u-1)(v-1) = n - (u+v) + 1$. More precisely, one wishes, given the public key $e$ to find a secret key $d$ so that

$$ed \equiv 1 \bmod \operatorname{lcm}(u-1, v-1).$$

Now this is just

$$ed = 1 + k(u-1)(v-1) = 1 + k(n - (u+v) + 1),$$

so certainly

$$\frac{k}{d} - \frac{e}{n} < \frac{k(u+v)}{nd}.$$

One expects that $u \simeq v \simeq n^{\frac{1}{2}}$. Then $k/d$ is necessarily a convergent of $e/n$ if $n^{\frac{1}{2}} \gg kd$. The convergents can all be found in polynomial time and, as Lidl points out, each is readily tested for correctness by parity check and detection of squares. Thus the encryption scheme is insecure if $d < n^{\frac{1}{4}}$ and $k$ is not large. However $kn \simeq ed$, so, indeed, choosing $e \gg n^{\frac{3}{2}}$ always protects against the present rather naïve attack. (Of course, $e$ is defined mod $\operatorname{lcm}(u-1, v-1)$, but may be given such an artificially large value if desired.)

To save clutter I have not emphasised the fact that $k$ is a rational with denominator $\gcd(u-1, v-1)$ rather than an integer as my notation suggests.

1.3. *Continued fraction expansion of Laurent series.*

Suppose now that the partial quotients $a_i = a_i(X)$ are polynomials each (other than perhaps $a_0(X)$ which may be constant) of degree at least 1. The formalism is unchanged but one needs to understand the sense in which a series

$$\alpha(X) = a_0(X) + \frac{1}{q_0(X)q_1(X)} - \frac{1}{q_1(X)q_2(X)} + \cdots = a_0(X) + \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{q_{n-1}(X)q_n(X)} \quad (11)$$

converges when $(q_n(X))$ is a sequence of polynomials with monotonically increasing degree. The essence is to so 'value' rational functions that the terms of the sum (11) have value decreasing to zero. It turns out that the order of vanishing at infinity yields the appropriate value. In effect one views the polynomials $q_i(X)$ as rational functions in $X^{-1}$ and then the sum (11) 'converges' as a formal power series in $X^{-1}$. The limit $\alpha$ of the continued fraction is a Laurent series, in fact, the sum of a polynomial $a_0(X)$ and a power series in $X^{-1}$. The continued fraction algorithm proceeds by taking the polynomial part (including the constant term) of the complete quotient and then inverting the remainder to yield the next complete quotient.

1.4. *Continued fraction expansion of algebraic numbers.*

It is not difficult to see that a periodic continued fraction represents a zero of a quadratic polynomial. The converse, Legendre's Theorem, is somewhat deeper but,

indeed, every real quadratic irrational has a periodic continued fraction expansion. In an important sense the continued fraction algorithm is tailored to quadratic quantities: that is manifested in the correspondence with products of $2 \times 2$ matrices. Williams [6] alludes in Section 2 of his paper in these Proceedings to the manner in which the continued fraction algorithm yields information on the ideal class group of a real quadratic field.

For algebraic numbers of higher degree, it is conjectured on deep theoretical grounds that the partial quotients are always unbounded but, in fact, no example displaying that property is known (nor, of course, is any counterexample). There is not all that much experimental data and more might prove instructive. On the other hand, surprisingly perhaps, the analogous situation for Laurent series over a finite field is different [1]. There are nonperiodic continued fractions all of whose partial quotients are polynomials of degree 1 which represent Laurent series algebraic over the rational functions. In the terms used by Lidl [4], Section 4, there are sequences $(s_n)$ with perfect linear complexity profile for which $\sum s_n X^{-n}$ is an algebraic function of degree greater than 2.

## 2. Recurrence Sequences.

### 2.1. *Generalised power sums, rational functions and recurrence sequences.*

A *generalised power sum* $a(h)$, $h = 0, 1, 2, \ldots$, is an expression of the shape

$$a(h) = \sum_{i=1}^{m} A_i(h)\alpha_i^h, \quad h = 0, 1, 2, \ldots \tag{12}$$

with *roots* $\alpha_i$, $1 \leq i \leq m$, distinct non-zero quantities, and *coefficients* $A_i(h)$ polynomials of respective degrees $n_i - 1$, for positive integers $n_i$, $1 \leq i \leq m$. The generalised power sum $a(h)$ is said to have *order*

$$n = \sum_{i=1}^{m} n_i .$$

Set

$$s(X) = \prod_{i=1}^{m}(1 - \alpha_i X)^{n_i} = 1 - s_1 X - \cdots - s_n X^n . \tag{13}$$

Then the sequence $(a_h)$ with $a_h = a(h)$, $h = 0, 1, 2, \ldots$, satisfies the linear homogeneous recurrence relation

$$a_{h+n} = s_1 a_{h+n-1} + \cdots + s_n a_h , \quad h = 0, 1, 2, \ldots . \tag{14}$$

To see this let $E : f(h) \mapsto f(h+1)$ be the shift operator and $\Delta = E - 1$ the difference operator. Then

$$(E - \alpha)\big(A_i(h)\alpha_i^h\big) = (\Delta A_i(h))\alpha_i^{h+1}$$

and since $\Delta A_i(h)$ has lower degree than does $A_i$, by linearity of $E$ and induction it is plain that

$$\prod_{i=1}^{m}(E-\alpha_i)^{n_i}$$

annihilates the sequence $(a_h)$ as asserted. Thus generalised power sums are interesting in that they coincide with the sequences satisfying the recurrence relations (14). It follows that there is a polynomial $r(x)$, of degree less than $n$, so that the power series

$$\sum_{h=0}^{\infty} a_h X^h = \frac{r(X)}{s(X)} \tag{15}$$

is a rational function; to see this multiply by $s(X)$ and note the recurrence relation.

Conversely given a rational function as above, with $\deg r < \deg s$, a partial fraction expansion yields

$$\frac{r(X)}{s(X)} = \sum_{i=1}^{m}\sum_{j=1}^{n_i} \frac{r_{ij}}{(1-\alpha_i X)^j} = \sum_{h=0}^{\infty}\left(\sum_{i=1}^{m}\sum_{j=1}^{n_i} r_{ij}\binom{h+j-1}{j-1}\alpha_i^h\right)X^h$$

and the coefficients of $X^h$, $h = 0, 1, 2, \ldots$, are indeed the values of a generalised power sum as described.

Accordingly, results on generalised power sums are equivalent to corresponding results for the Taylor coefficients of rational functions.

A sequence $(a_h)$ satisfying a relation (14) is often called a *recurrence sequence* (or *linearly recursive sequence*) of *order* $n$; the polynomial $X^n s(X^{-1})$ reciprocal to the polynomial (13) is called the *characteristic* or *companion polynomial* of the recurrence sequence. Our "roots" $\alpha_i$ are the distinct zeros of the companion polynomial. The archetypal example of a recurrence sequence is of course the celebrated Fibonacci sequence $(f_h)$ defined by

$$f_{h+2} = f_{h+1} + f_h\,,\ h = 0, 1, 2, \ldots\,,\ \text{with } f_0 = 0\,, f_1 = 1\,;$$

and generated by

$$\frac{X}{1-X-X^2} = \sum_{h=0}^{\infty} f_h X^h\,.$$

The expression (12) for the $a_h = a(h)$ as a generalised power sum provides a well known formula for the terms of a recurrence sequence. One obtains a less well known formula from directly expanding (15). In terms of the given *initial values* $a_0, a_1, \ldots, a_{n-1}$ of $(a_h)$ one has

$$r(X) = \sum_{j=0}^{n-1}\left(a_j - \sum_{i=1}^{j} s_i a_{j-i}\right)X^j\,,$$

and

$$s(X)^{-1} = \sum_{h=0}^{\infty} \sum_{j_1+2j_2\cdots+nj_n=h} \frac{(j_1+j_2\cdots+j_n)!}{j_1!\ldots j_n!} s_1^{j_1}\cdots s_n^{j_n} X^h .$$

For the Fibonacci numbers this yields (with the usual conventions for interpreting the combinatorial symbol)

$$f_{h+1} = \sum_j \binom{h-j}{j} .$$

2.2. *Hadamard operations.*

If the power series $\sum a_h X^h$ and $\sum b_h X^h$ represent rational functions, then so do their sum $\sum(a_h + b_h)X^h$ and their *Hadamard product*

$$\sum a_h b_h X^h .$$

This is not obvious as stated but is an immediate consequence of the fact that the sum and, respectively, the product of generalised power sums is again a generalised power sum. Incidentally, it turns out that the Hadamard product of a rational and of an algebraic power series is algebraic but over a field of characteristic zero the Hadamard product of algebraic functions is not necessarily algebraic. The most quoted example is

$$(1-4x_1)^{-1/2} = \sum \binom{2h}{h} x_1^h , \text{ but } \sum \binom{2h}{h}^2 x_1^h$$

is not algebraic. The first remark is the useful identity

$$\binom{2h}{h} = (-1)^h \binom{-\frac{1}{2}}{h}$$

and, with a little work and some elementary calculus one sees that the latter series is given by the integral

$$\frac{2}{\pi} \int_0^{\pi/2} \frac{dt}{\sqrt{(1-16x_1\sin^2 t)}} .$$

This is a complete elliptic integral well known not to represent an algebraic function.

Remarkably, the Hadamard product of algebraic power series defined over a field of *positive* characteristic is always again algebraic (see [3]); in particular this is so for the Hadamard product of algebraic power series over a finite field. It turns out the sequences of Taylor coefficients of algebraic functions over finite fields are generated by finite automata, suggesting a rather more subtle source for stream ciphers quite different from those generated by rational functions. An efficient introduction to the mathematical background may be found in [5].

2.3. *Recurrence sequences and LFSR's.*

Recall that $(a_h)$ is a recurrence sequence if and only if its terms are given by a generalised power sum

$$a(h) = a_h = \sum_{i=1}^{m} A_i(h)\alpha_i^h , \quad h = 0, 1, 2, \ldots ,$$

and that the recurence sequence has characteristic polynomial

$$\prod_{i=1}^{m}(X-\alpha_i)^{n_i}.$$

Obviously, for each positive integer $d$, $(a_{dh})$ again yields a generalised power sum, and it has characteristic polynomial

$$\prod_{i=1}^{m}(X-\alpha_i^d)^{n_i}.$$

A generalised power sum yields a periodic sequence if and only if each root $\alpha_i$ is a root of unity and each coefficient $A_i(h)$ is a periodic function of $h$. Over a finite field a nonzero element is a root of unity and in characteristic $p$ a polynomial in $h$ is trivially periodic with period $p$. Thus, over a finite field every recurrence sequence is periodic. Conversely, a periodic sequence with period $t$ is the sequence of Taylor coefficients of a rational function with denominator $1-X^t$.

Given the characteristic polynomial, the initial values $a_0, a_1 \ldots, a_{n-1}$ of the recurrence sequence, the coefficients $A_i$ of the generalised power sum and the numerator of the generating rational function determine one another. Different recurrence sequences with the same characteristic polynomial are the sequences of Taylor coefficients of rational functions with the one denominator but different numerators.

The product of $k$ generalised power sums each with roots $\alpha_i$, but possibly with different coefficients, is a generalised power sum with roots consisting of all monomials of weight $k$ in the $\alpha$'s. Hence the non-linear combination of recurrence sequences defined at Section 4.1 of [2] is a recurrence sequence with roots consisting of all monomials of weight at most $L$ in the roots of the original sequence. The Groth sequences mentioned at Section 4.3 of [2] are sums of different recurrence sequences each with characteristic polynomial having zeros consisting of all pairs $\alpha_i\alpha_j$.

More generally, each of the sequences mentioned in [2], no matter how apparently complex its formation rule, is itself a recurrence sequence which, in principle, can be generated by the one simple *LFSR*. That follows immediately over the finite field $\mathbf{F}_2$ by periodicity, but in fact, with the exception of the 'multiplexed sequence' mentioned at Section 7.2, would hold had the generating sequences been defined over any field. The formation rules are a trade-off between ease of generation and the endeavour to satisfy various 'randomness' criteria.

## References

1. L. E. Baum and M. M. Sweet, 'Continued fractions of algebraic power series in characteristic 2', *Annals of Math.* **103** (1976), 593–610.
2. Ed Dawson, 'Linear feedback shift registers and stream ciphers', this volume.
3. H. Furstenberg, 'Algebraic functions over finite fields', *J. Alg.* **7** (1967), 271–277.
4. R. Lidl, 'Some mathematical aspects of recent advances in cryptology', this volume.

5. Leonard Lipshitz and Alfred J. van der Poorten, 'Rational functions, diagonals, automata and arithmetic', *Number Theory*, ed. Richard A. Mollin (First Conference of the Canadian Number Theory Association, Banff 1988). (De Gruyter, 1989.)
6. H. C. Williams, 'Quadratic fields and cryptography' this volume.
7. R. T. Worley, 'Insecurity of the knapsack one-time pad', this volume.

*School of Mathematics, Physics, Computing and Electronics, Macquarie University, New South Wales 2109, AUSTRALIA.*

# SECURITY IN TELECOMMUNICATION SERVICES OVER THE NEXT DECADE

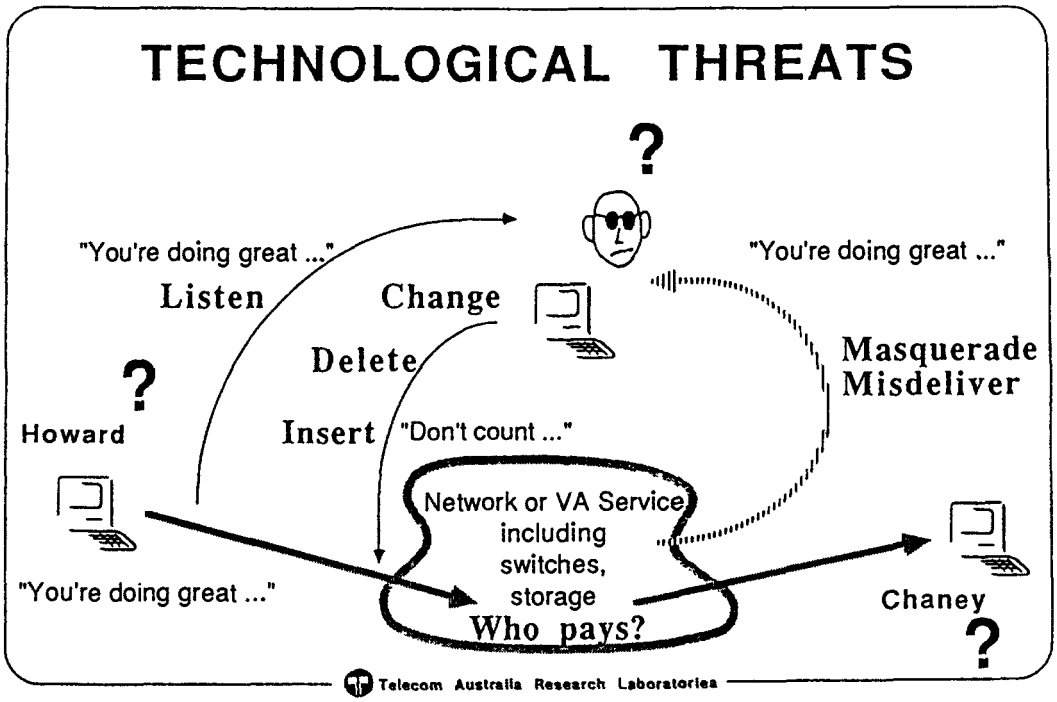
**John Snare**

The next decade will see implementation of telecommunication services to handle more and more critical business data. Appropriate features must be included to establish user confidence in the security of such services. This paper starts with a discussion of the security needs likely to be placed on advanced telecommunication services in the 1990's. There is much more to communication security than just encryption, and this paper explores the types of protection possible, and ways that users might perceive the benefits.

The paper then introduces the technologies relevant to satisfying these needs and discusses some of the cryptographic techniques relevant to practical solutions. Emphasis is placed on the role of public key ciphers, and their application in authentication, digital signature and key distribution services. Finally, an example based on electronic mail is presented showing possible synergy between terminal and network functions.

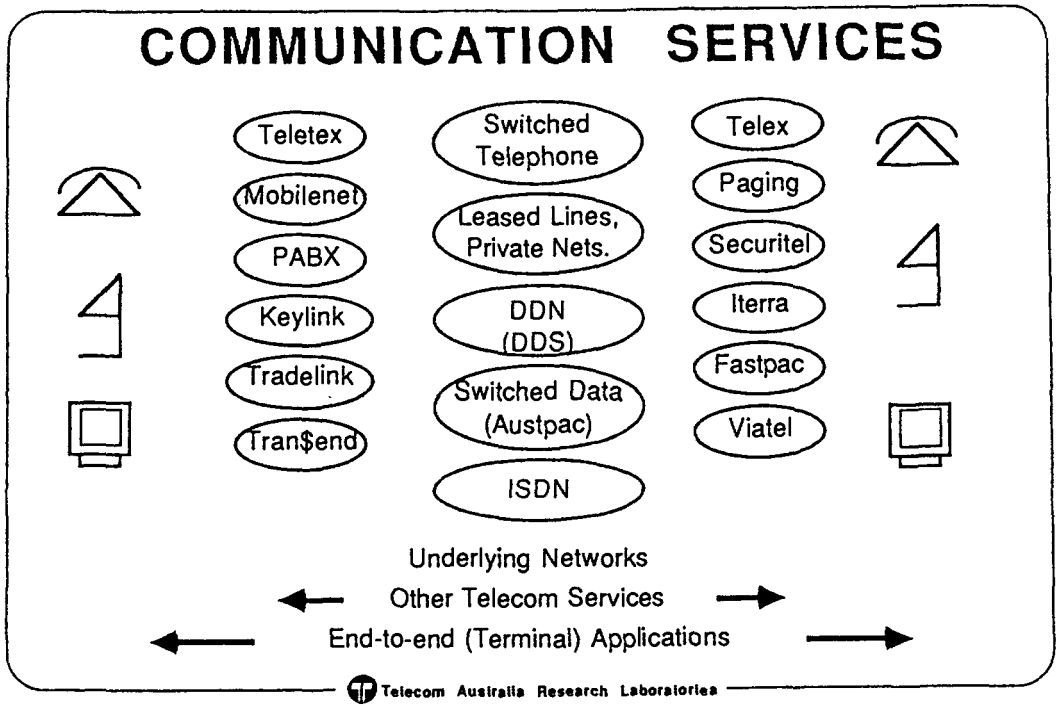
## 1. Application Security Requirements.

Information is valuable. Its value may depend on its currency, secrecy, accuracy, or relationships with other information. Information can be stolen without the owner noticing, and it may be changed or substituted transparently. There is a trend towards an increasing number of applications involving telecommunication services, to allow information systems to be distributed, and/or remotely accessed.



In such cases there are a large number of security related questions that a system designer needs to consider. The technological threats include:

- Who is out there?
- Are they allowed to see this information?
- Is the right person going to be charged?
- Will information end up with the right person?
- Has information been changed in transit?
- Is something missing?
- Has information been seen by unknown parties?
- Will a sender deny sending particular information?
- Can a receiver pretend not to have received a message?
- How can sending, delivery, receipt be proved?




Satisfactory answers to these questions will increasingly require the use of cryptography in the application design. The design will also need to take into account the increasingly diverse range of telecommunication services that are becoming available, and the security characteristics they offer. Communication services can be based on:

- the traditional public switched telephone network (PSTN);
- packet switched data networks (for example, Telecom Austpac);
- private networks built using dedicated digital (or analogue) services (for example, Telecom DDN);
- mobile network services (for example, Telecom Mobilenet);
- satellite communication links (for example, Telecom Iterra);
- integrated services digital networks (ISDN);

- optical fibre customer access networks (similar to Telecom prototype MACNET system);
- local area 'LAN' networks;
- high speed metropolitan area 'MAN' networks (for example, Telecom Fastpac);

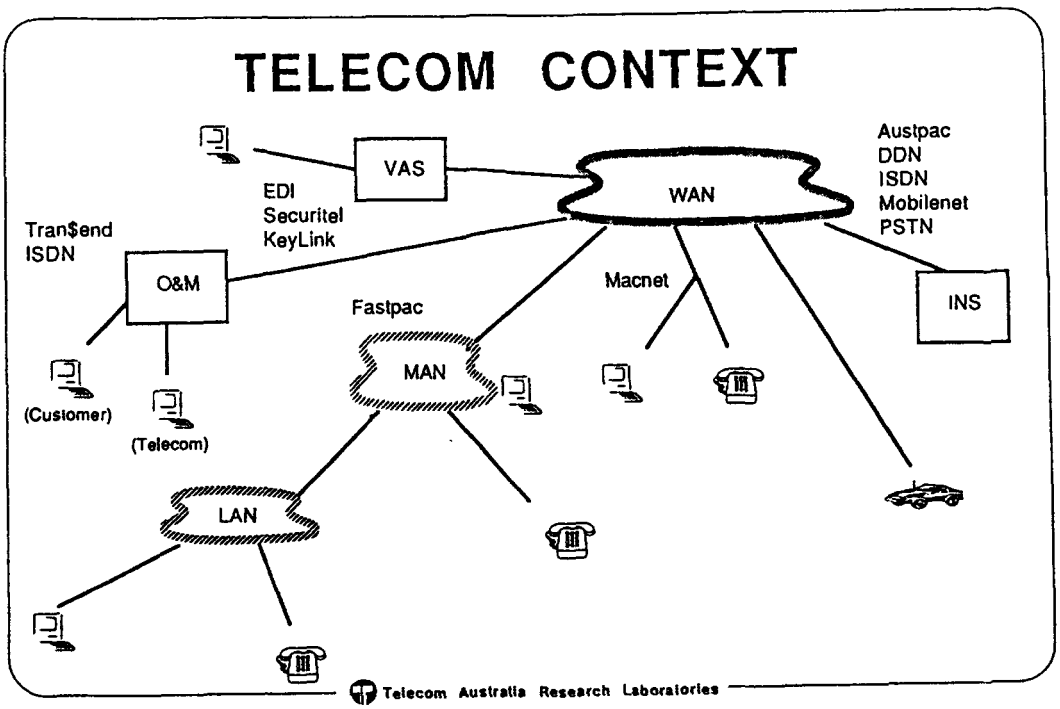
and the list will grow.

Furthermore, networks can provide a range of enhanced facilities such as:

- teletex for document transfer (for example, Telecom Teletex);
- electronic mail (for example, Telecom Keylink);
- electronic funds transfer (for example, Telecom Tran$end);
- electronic (business) data 'EDI' interchange (for example, Telecom Tradelink);
- facsimile (for example, Telecom Faxstream);
- telex;
- paging;
- alarm and telemetry services (for example, Telecom Securitel);
- videotex for database access (for example, Telecom Viatel);
- intelligent network 'IN' services that allow customer customization of basic services;

and again the list will grow.

As the 1990's develop, telecommunication networks will become more complex, as new types of network (with a variety of service capabilities) are implemented and interconnected. Security solutions will need to take into account the capabilities of this complex network environment if they are to achieve the flexibility users will demand.



Suitable selection of appropriate communication services can allow cost effective security by appropriate combinations of network-supported and end-to-end security functions.

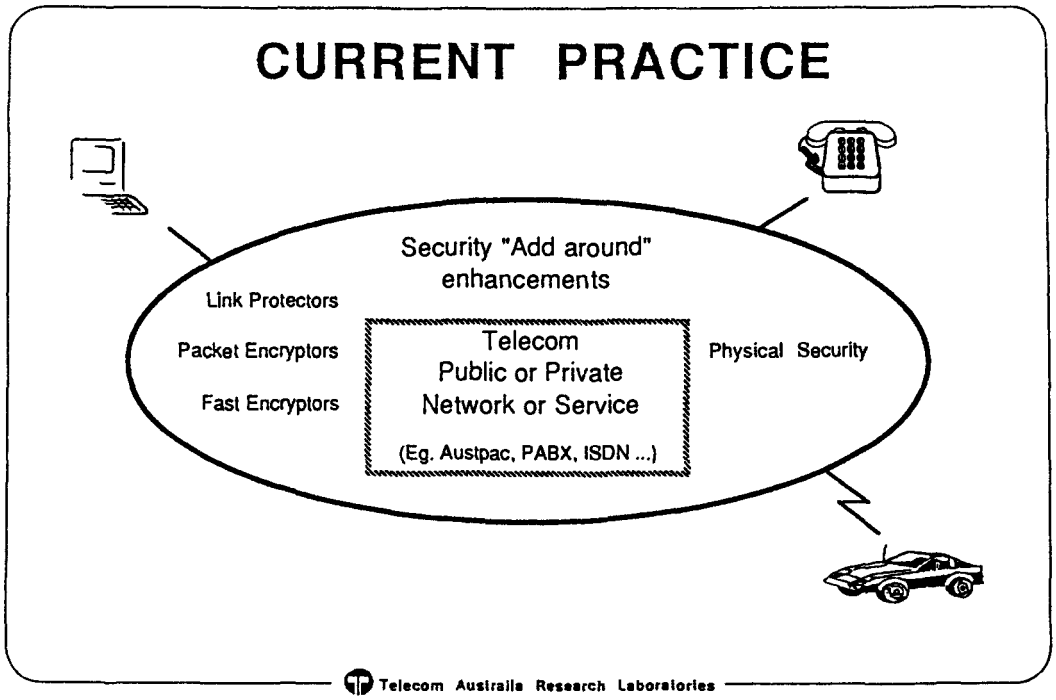
For example, networks can help identify end users by knowing where they are. They can also aid access control security by limiting call initiation or receipt with mechanisms including various forms of call barring and closed user groups. It is also possible for networks to permit connections only to be made at certain times of the day or week, and for networks to control the routing of information (perhaps to avoid transit networks of dubious security).

Beyond these capabilities, network provided enhanced services can be used to facilitate cryptographic key management and perform functions associated with resolution of repudiation disputes.

There are of course some cases where additional security functions must be performed in a way that is independent of network support. For example, electronic funds transfer messages require end-to-end integrity and PIN confidentiality protection.

## 2. Security Implementation Options.

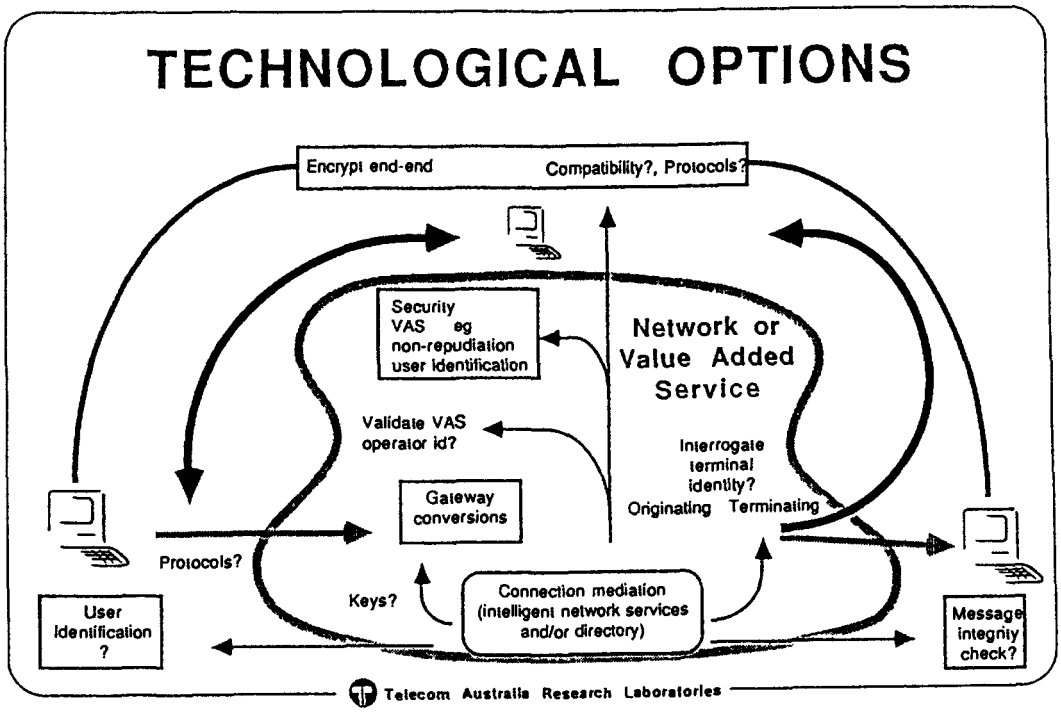
A major factor that will influence the rate of adoption of secure communication services is operational convenience. The objective in this area is to provide security features that do not significantly degrade performance, and which are simple to use. Both these criteria suggest a move away from the current practice of adding 'black boxes' to existing services.




While this approach yields expedient solutions for special cases, it does not lend itself to large scale applications. Lack of standardization means that solutions adopted for similar reasons in different organizations will not work together. Additionally, it

is likely that separate implementations will be required within organizations for each application. Both these considerations suggest the likelihood of high cost solutions if this approach is adopted. Furthermore, operation of such 'black boxes' is often not well integrated into the application and involves administratively cumbersome manual procedures.

Preferred solutions for the 1990's will therefore involve security that is integrated into the application design, with security levels appropriate to particular applications, and many security functions being performed in a manner transparent to the end user. Already the international telecommunications standards body CCITT is working on standards that will underpin this approach, with standards now available covering authentication, user identification over untrustworthy networks, and secure message transfer. Similarly, the International Organization for Standardization ISO is developing security standards for both banking applications as well as general computer based applications. The challenge for the 1990's will be to complete this work and implement applications based on them.




## 3. The Role of Public Key Ciphers.

Public key cryptosystems [1] will play an important role in achieving security objectives for secure commercial services with widespread applicability. Such services could include secure document transfer based on teletex, secure facsimile over ISDN, secure electronic mail, secure EDI, secure remote facilities management, secure voice, secure remote meter reading, secure financial services, and the list will grow.

The following properties of public key cryptosystems are attractive for implementation of secure telecommunication based applications:

- different keys are used to encrypt and decrypt data;
- it is not feasable to compute one key from knowledge of the other;

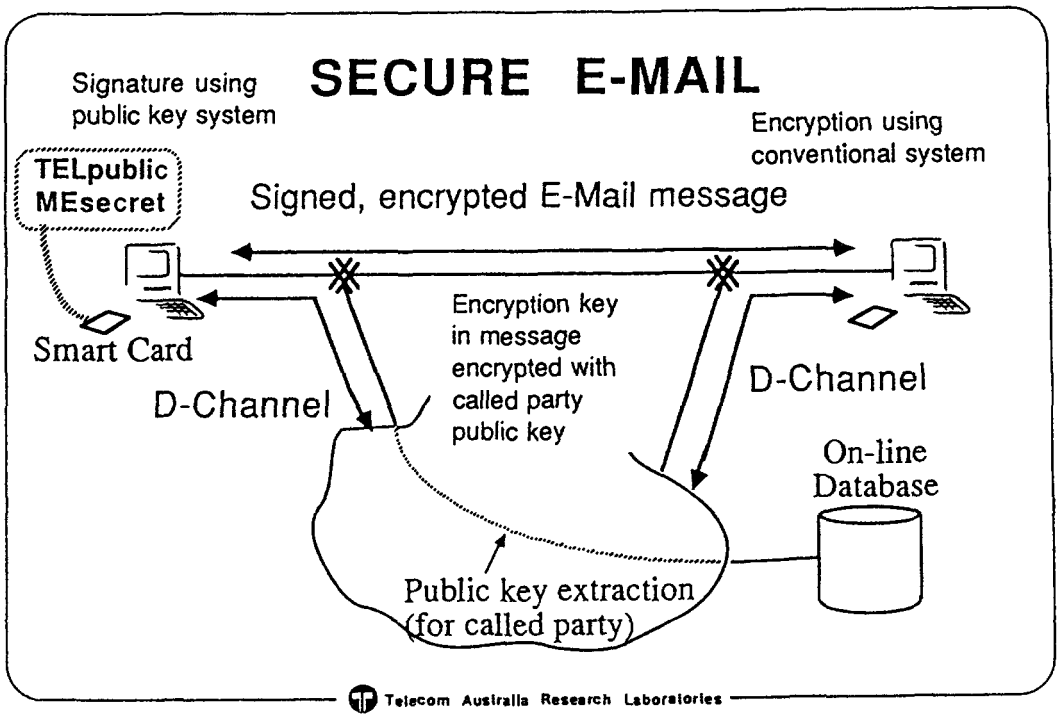
- one of the keys can be made publicly available without prejudicing system security; and
- it is possible to prove possession of secret data without revealing it.

It is likely that public key cryptosystems will be used to authenticate user identity, prove integrity and origin of data, and to secretly exchange security parameters such as encryption keys during an information transfer pre-amble. If secrecy of other information is an issue, it is likely that symmetric (single key) systems will be used.

A major issue that must be solved in systems based on public key cryptosystems is proving the true ownership of any keys used. A generally applicable means to achieve this is to rely on a trusted third party (called a certification centre) to certify owner-key relationships using a public key based digital signature. Users would then only have to securely obtain a single public key from the certification centre to be able to validate at any time other keys that have been certified by such a centre. Details of how such a scheme could work can be found in CCITT Recommendation X.509 [2].

## 4. An Example—Secure Electronic Mail.

The concepts developed in this paper could, for example be applied to an electronic mail service delivered over an ISDN as follows.




A user could have a public/secret key pair, certified by a Telecom certification centre and stored on a smart card. To send a secure electronic mail message they could compose a message on a personal computer (PC) using their favourite word processor, and when ready, call up an electronic mail service using an ISDN service integrated into their PC. They would be requested to identify the recipient (perhaps using an easily remembered name or other alias), to insert their smart card into an attached reader, and activate it by entering a suitable password. From the user perspective the

mail would then be sent (although at a later time they would receive a report of its delivery or otherwise).

Invisibly to the user the following types of action would be needed. Firstly, the user's terminal would need to find the public key of the recipient, if it was not on a local database. It could do this by calling up an on-line directory using the ISDN data and signalling channel. At the same time as requesting the recipient's public key, the user's computer could also ask for the correct electronic mail address for the named recipient. The database would possibly need to know who to charge for providing this information. It would therefore engage in a dialogue with the user's computer and smart card and using a cryptographically based 'one time password-like' procedure validate the originator, and return the requested information. The sender's computer would check the validity of the receiver's key information received from the network using the public key supplied by the Telecom certification centre.

The user's computer could then use the originator's secret key to calculate a digital signature for the message, to allow the recipient to be sure who it came from and that it hadn't been changed. The originator could also attach to the message a copy of their public key certificate, to allow the recipient to validate the message without needing to first engage in a procedure to find the originator's public key. (If the message was of major significance, the recipient may wish to interrogate an on-line database to check that the originator's public key had not been revoked or placed on a 'don't use hot list'.) The originator could generate a key for a conventional symmetric crypto-system and encrypt the entire message for privacy using it. The privacy key, encrypted under the recipient's public key, could be attached to the encrypted message.

When receiving such a message, the recipient would be asked to insert their smart card and activate it. The recipient's computer would then invisibly:

- decrypt the privacy key using the recipient's secret key stored on their smart card;
- decrypt the rest of the message using the recovered privacy key;
- identify the sender;
- check the sender's key certificate using the Telecom certification centre public key; and
- check the integrity of the message using the sender's public key.

The received message would then be presented to the receiver along with a simple security report. The receiver can then be certain who the message was from and that:

- it has not been tampered with;
- nobody else could have read it (if it was encrypted); and
- that they were the intended recipient.

## 5. Conclusions.

This paper has introduced the complexities of the emerging telecommunications environment, and raised the security issues that will become increasingly important in new applications that are developed. It is suggested that public key cryptography will play an important part in this new telecommunications world. Finally, it is observed that a practically useful approach must be based on integrating security into system designs in a way that exploits network service capabilities as well as terminal functions.

**Acknowledgement.**

The permission of the Executive General Manager, Research, of Telecom Australia to publish the above paper is hereby acknowledged.

## References

1. D. W. Davies and W. L. Price, *Security for Computer Networks* (Wiley, 1984), chapters 8 and 9.
2. CCITT Recommendation X.509, Geneva 1989.

*Secure Communication Systems Section, Telecom Australia Research Laboratories, 770 Blackburn Rd, Clayton, Victoria, AUSTRALIA.*

8

# LINEAR FEEDBACK SHIFT REGISTERS AND STREAM CIPHERS

**Ed Dawson**

A stream cipher is the process of encryption where a random binary sequence is combined modulo two to binary plaintext to produce ciphertext. In this paper, we give several recent methods for forming this random binary sequence by the nonlinear combination of sequences produced by linear feedback shift registers whose characteristic polynomials are primitive polynomials. It will be demonstrated how to form a pseudo random sequence which is secure from cryptanalytic attack in that the sequence has a large period, large linear complexity and is correlation immune.

## 1. Stream ciphers ([2], [14], [16]).

1.1. *Definition.*

Let $(u_t)$ be a binary sequence, where $u_t$ denotes the $t$-th term of the sequence. Let $p_t$ denote the $t$-th term of binary plaintext. A stream cipher is a process of encryption where binary ciphertext $c_t$ is formed by

$$c_t = u_t + p_t$$

and the + operation indicates modulo two addition. Decryption is by $p_t = u_t + c_t$, so that both receiver and sender need to generate $(u_t)$.

1.2. *Properties of stream ciphers.*

(a) *Period.* The sequence $(u_t)$ is said to be periodic if there exists a smallest positive integer $p$ such that $u_{t+p} = u_t$ for all $t$; $p$ is the period of sequence.

In this report all sequences will be assumed to be periodic unless otherwise stated.

(b) *Linear feedback shift register (LFSR).* Any infinite periodic sequence $(u_t)$ can be defined by a recurrence relation

$$u_{t+r} = \sum_{i=0}^{r-1} c_i u_{t+i} \quad \text{for all} \quad t \geq 0,$$

where the $c_i$ are binary constants such that $c_0 = 1$ and arithmetic is modulo two. The vector $(u_0, u_1, \ldots, u_{r-1})$ is called an initial state vector. Clearly, an initial state vector together with the above equation defines the other terms of sequence.

Associated with the above equation is a binary polynomial

$$f(x) = c_0 + c_1 x + \cdots + c_{r-1} x^{r-1} + x^r,$$

called the characteristic polynomial of the sequence. The coefficients $c_i$ are feedback constants. Such sequences can be mechanised by using a linear feedback shift register (*LFSR*) whose tap settings are defined by the feedback constants.

*Example 1.* The sequence $(u_t)$ defined by the recurrence relation

$$u_{t+6} = u_{t+5} + u_{t+4} + u_{t+1} + u_t$$

has characteristic polynomial

$$f(x) = x^6 + x^5 + x^4 + x + 1.$$

For this sequence

$$u_6 = u_5 + u_4 + u_1 + u_0$$
$$u_7 = u_6 + u_5 + u_2 + u_1$$
$$u_8 = u_7 + u_6 + u_3 + u_2$$

and so forth. An *LFSR* for the sequence is

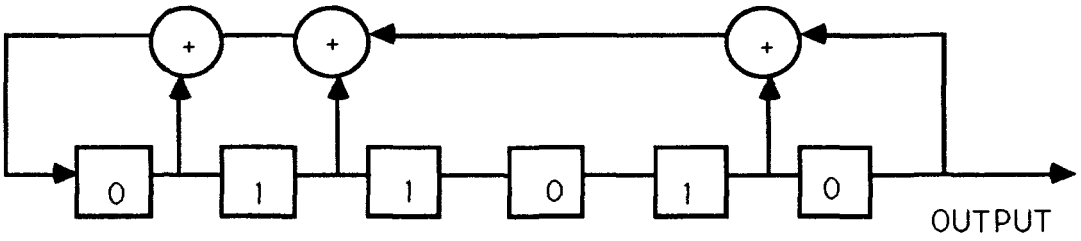



Figure 1.

If the initial state vector is (0, 1, 0, 1, 1, 0) then the first period of the sequence is

0101100101010010011110000011011100110001110101111110110100010000.

(c) *Linear Complexity.* There is a unique *LFSR* of shortest length which will produce $(u_t)$. The length of this *LFSR* is defined to be the linear complexity of the sequence. The characteristic polynomial of this *LFSR* is defined to be the minimal polynomial of the sequence. In Example 1 the linear complexity of the sequence is 6 since the *LFSR* used was of shortest length and $x^6 + x^5 + x^4 + x + 1$ is the minimal polynomial of the sequence. If the linear complexity of the sequence is $L$ then the minimal polynomial can be found by the Berlekamp-Massey algorithm provided $2L$ consecutive terms of the sequence are known. Given the minimal polynomial and $L$ consecutive terms, the entire sequence can be derived.

(d) *Noiselike characteristics.* In order to avoid statistical analysis by an attacker the sequence $(u_t)$ should have noiselike characteristics. There are three basic properties to measure the randomness of a binary sequence.

(i) Approximately one half the terms in a period should be one.

(ii) In a period one half the runs should have length one, a quarter length two and so forth.

(iii) The out-of-phase autocorrelation function should be constant.

If $(u_t)$ satisfies (i) to (iii), it is said to be $G$-random. Such a sequence is also said to be a pseudo-noise ($PN$) sequence.

1.3. *Summary of Basic Properties.*

If $(u_t)$ is used in a stream cipher for encrypting binary plaintext then

(i) period of $(u_t)$ should be large.

(ii) linear complexity of $(u_t)$ should be large.

(iii) $(u_t)$ should be approximately $G$-random.

A good stream cipher with these properties appears the same as a one-time pad to an attacker.

If the period $p$ is large an attacker only has available a small fraction of the total period for analysis. In order to be secure from statistical attack, the sequence should be locally random. There are five different standard statistical tests in reference [2] to measure the local randomness of a binary sequence, namely the frequency test, the serial test, the poker test, the autocorrelation test and the runs test.

## 2. Maximal length sequences ([2], [14], [16]).

2.1. *Definition.*

A binary polynomial $f(x)$ of degree $L$ is said to be primitive if

(i) $f(x)$ is irreducible, and

(ii) $f(x)$ is a factor of $x^{2^L-1}+1$ but is not a factor of $x^r+1$ for any $r < 2^L-1$.

A maximal length sequence or an $m$-sequence is a sequence whose minimal polynomial is primitive. For example, the sequence from Example 1 is an $m$-sequence since $f(x) = x^6 + x^5 + x^4 + x + 1$ is primitive.

2.2. *Properties of m-sequences.*

Let $(u_t)$ be an $m$-sequence defined by a primitive polynomial of degree $L$.

(i) Period of $(u_t)$ is $2^L - 1$.

(ii) Linear complexity of $(u_t)$ is $L$.

(iii) $(u_t)$ is $G$-random.

(iv) Let $\phi(n)$ denote the number of positive integers less than $n$ which are relatively prime to $n$. There exist $\lambda(L) = \phi(2^L-1)/L$ primitive polynomials of Degree $L$. Hence there are $\lambda(L)$ distinct $m$-sequences of period $2^L - 1$ such that none is a circular shift of another.

| $L$ | $\lambda(L)$ | $L$ | $\lambda(L)$ | $L$ | $\lambda(L)$ |
|---|---|---|---|---|---|
| 1 | 1 | 9 | 48 | 17 | 7710 |
| 2 | 1 | 10 | 60 | 18 | 8064 |
| 3 | 2 | 11 | 176 | 19 | 27594 |
| 4 | 2 | 12 | 144 | 20 | 24000 |
| 5 | 6 | 13 | 630 | 21 | 84672 |
| 6 | 6 | 14 | 756 | 22 | 120032 |
| 7 | 18 | 15 | 1800 | 23 | 356960 |
| 8 | 16 | 16 | 2048 | 24 | 276480 |

Table 1. Number of primitive polynomials of degree $L$.

The primitive polynomials of a given degree can be found by using the *GALOIS* Software Package from the University of Tasmania.

An $m$-sequence should not be used by itself in a stream cipher since the linear complexity is small in comparison to the period length.

2.3. *Decimation of sequences.*

One method of producing an $m$-sequence of length $L$ without having to alter or reprogram the feedback connections on the shift register from another $m$-sequence of length $L$ that has a different minimal polynomial is by decimation. Let $(u_t)$ be an $m$-sequence whose minimal polynomial $f(x)$ has degree $L$. Suppose that one selects a speed factor $d$ and defines a sequence $(s_t)$ by $s_t = u_{td}$.

The sequence $(s_t)$ is defined to be the $d$-th decimation of $(u_t)$; $(s_t)$ can be formed by a system clock which clocks at a rate $d$ times slower than the high speed clock which forms $(u_t)$.

*Example 2.* Let $d = 3$. The following table illustrates the calculation:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| High-speed clock times | $u_0$ | $u_1$ | $u_2$ | $u_3$ | $u_4$ | $u_5$ | $u_6$ | $u_7$ |
| System clock times | $s_0$ | | | $s_1$ | | | $s_2$ | |

Sequences formed by decimating $m$-sequences have the following properties.

**Property 1.** *If $d$ and $2^L - 1$ are relatively prime and $(u_t)$ is an $m$-sequence, then $(s_t)$ is an $m$-sequence of period $2^L - 1$.*

**Property 2.** *Let $f(x)$ and $g(x)$ be primitive polynomials of degree $L$. Suppose that $(u_t)$ is an $m$-sequence with minimal polynomial $f(x)$. There exists an integer $d$ which is relatively prime to $2^L - 1$ such that the sequence $(s_t)$ defined by $s_t = u_{td}$ has $g(x)$ as its minimal polynomial.*

## 3. Nonlinear theory of sequences ([16]).

3.1. *Method of combining.*

There are two basic methods of using an $m$-sequence to form a sequence $(z_t)$ for use in a stream cipher. The first method involves a nonlinearly filtered *LFSR*. In this

method, a nonlinear Boolean function is applied to the stages of an *LFSR* as shown in figure 2. This technique will be discussed in section 4. The second method employs the nonlinear combination of *LFSR*'s. In this method, the output sequences of several *LFSR*'s are combined using a nonlinear Boolean function as shown in figure 3. This technique will be discussed in section 5.

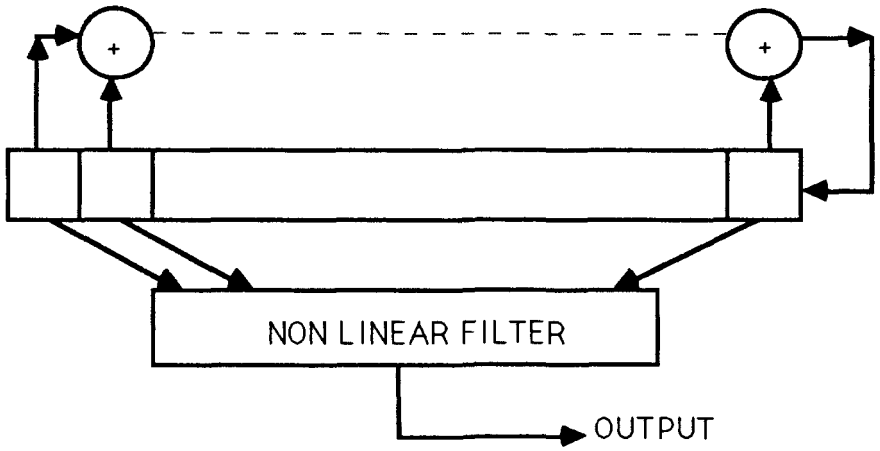



Figure 2. Non-linearly Filtered *LFSR*.

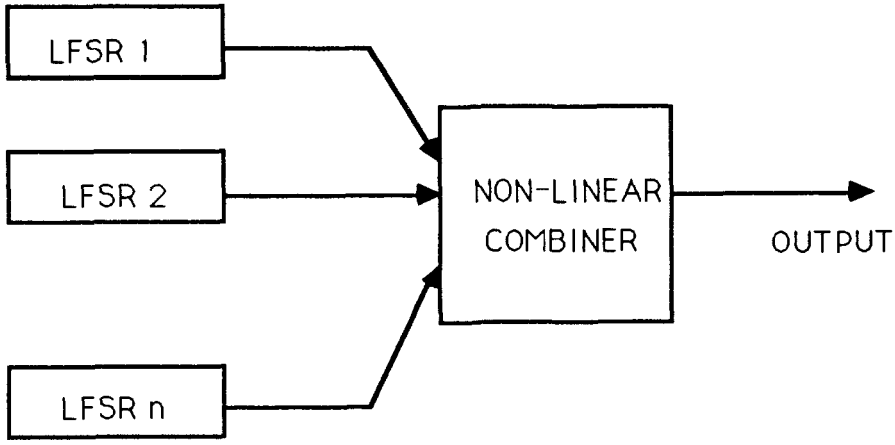



Figure 3. Non-linear Combination of *LFSR*'s.

3.2. *Key size.*

There are three different key sources for a sequence formed as in figures 2 and 3.

(i) Initial state vectors. Each *LFSR* used of length $L$ has $2^L - 1$ possible initial state vectors.

(ii) Tap settings. There are $\lambda(L)$ possible tap settings for each *LFSR* of length $L$ which results in an $m$-sequence.

(iii) Non linear combining function.

Provided sufficiently many large *LFSR*'s are used, one can have available a very large key space. The initial state vector needs to be changed at each session. Tap settings are often used for a customer key.

3.3. *Synchronization of Stream Ciphers.*

The stream ciphers discussed in this paper are synchronous ciphers. If one bit of ciphertext is lost or inserted in transmission then there is a loss of synchronization between the receiver and sender. The receiver is unable to decrypt ciphertext. However, such a system has an advantage in that a wiretapper cannot insert false messages or delete part of the message without detection.

A self synchronizing stream cipher can be designed by feeding ciphertext back as shown in Figure 4. An error in one bit will effect at most $n$ following ciphertext bits where $n$ is the length of the *LFSR*.

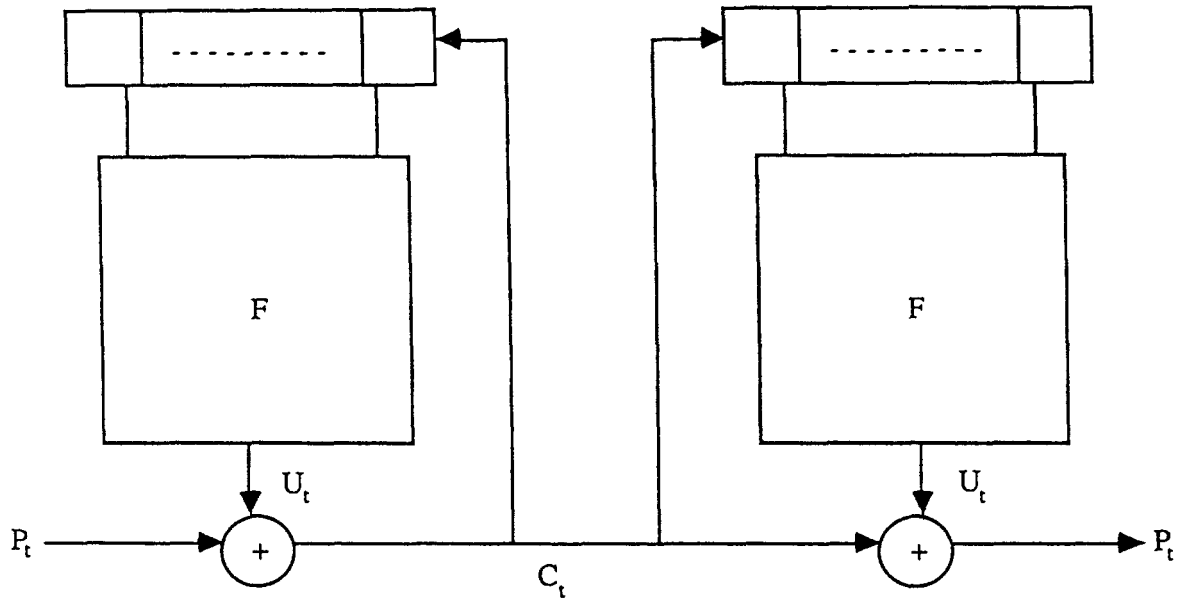



Figure 4. Principle of a self-synchronizing stream cipher.

Self-synchronous stream cipher have the following disadvantages.

(i) Error Propogation. One error in transmission may cause $n$ further errors.

(ii) Wiretapping Protection. Due to the self-synchronization property it is difficult to detect when a wiretapper is inserting or deleting messages.

(iii) Known State Vector. Seed values for the *LFSR* are known to the cryptanalyst since ciphertext is used in seeding the *LFSR*.

(iv) Mathematical Properties. In general the sequence is non-periodic since the message sequence in general is non-periodic. Hence it is difficult to determine mathematical properties of the sequence.

For the remainder of this report stream ciphers will be synchronous ciphers. For a synchronous cipher one method of maintaining synchronization is frame synchronization which is often used in digital communications.

## 4. Nonlinear filtering of an m-sequence ([1], [11], [16]).

4.1. *Algebraic Normal Form.*

Let $(u_t)$ be an $m$-sequence produced by an *LFSR* of length $L$. Let $\mathbf{s}_1$ be the binary vector of length $2^L - 1$ corresponding to the first period of $(u_t)$ and $\mathbf{s}_i$ for $i = 2, \ldots, L$

be binary vectors corresponding to circular shifts of $\mathbf{s}_1$ by $i-1$ places to the left. Thus, $\mathbf{s}_i$ corresponds to the first period of the sequence in stage $i$ of the *LFSR*.

Let $(z_t)$ be the sequence produced by nonlinearly combining the stages of the *LRSR* and $\mathbf{z}$ denote the binary vector corresponding to the first $2^L - 1$ terms of the sequence. Then $(z_t)$ can be defined by the Boolean function:

$$\mathbf{z} = a_1\mathbf{s}_1 + \cdots + a_L\mathbf{s}_L + a_{12}\mathbf{s}_1\mathbf{s}_2 + \cdots + a_{1\ldots L}\mathbf{s}_1 \cdots \mathbf{s}_L.$$

This is the algebraic normal form of $(z_t)$.

The $2^L - 1$ vectors $\mathbf{s}_1, \ldots, \mathbf{s}_L, \mathbf{s}_1\mathbf{s}_2, \ldots, \mathbf{s}_1 \cdots \mathbf{s}_L$ form a basis of binary $(2^L - 1)$-tuples. Hence any binary $(2^L-1)$-tuple can be written as was $\mathbf{z}$ above. Clearly, extreme care must be taken in the choice of the Boolean function if the sequence $(z_t)$ is to be used in a stream cipher since a sequence formed as above may have bad cryptographic properties as described in Section 1.2.

4.2. *Properties.*

(i) Period of $(z_t)$ is $2^L - 1$ or a divisor of $2^L - 1$.

(ii) Linear complexity of $(z_t)$ is at most

$$\sum_{i=1}^{k} \binom{L}{i},$$

where $k$ denotes the algebraic order of the Boolean function. (For example, the algebraic order of $\mathbf{s}_1 + \mathbf{s}_1\mathbf{s}_4 + \mathbf{s}_2\mathbf{s}_3 + \mathbf{s}_1\mathbf{s}_3\mathbf{s}_4$ is three.) If the linear complexity is strictly less than the above bound, the sequence is called degenerate.

(iii) Noiselike characteristics depend on the choice of the Boolean function. If, for example, the function has the form

$$\mathbf{s}_1 + f_1(\mathbf{s}_2, \ldots, \mathbf{s}_n),$$

then approximately one half of the terms are one.

4.3. *Groth sequences.*

A Groth sequence is formed by summing second order products of the $L$ stages of an $m$-sequence at one or several layers.

*Example 3.* The Boolean function $\mathbf{s}_1\mathbf{s}_5 + \mathbf{s}_2\mathbf{s}_4 + \mathbf{s}_3\mathbf{s}_6$ applied, as shown in Figure 5, to the length six *LFSR* from Example 1 gives the sequence.

101100111100101111000000110111110001000110100100001010010000001

Groth sequences seem to satisfy the properties of a sequence for use in a stream cipher. Provided enough layers of nonlinear logic are used a Groth sequence can be formed with linear complexity close to the period length. A description of Groth sequences can be found in references [1] and [11].

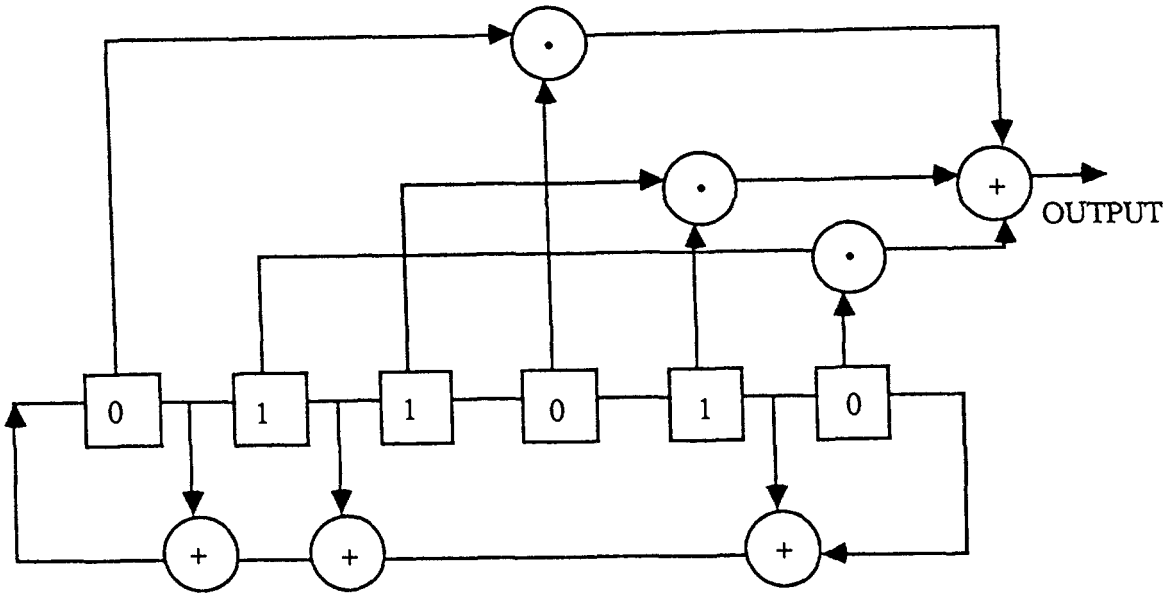



Figure 5.

## 5. Nonlinear combination of m-sequences ([16]).

### 5.1. *Introduction.*

Let $(a_{t1}), \ldots, (a_{tk})$ denote $m$-sequences formed by shift registers of respective lengths $L_1, \ldots, L_k$. Let $(z_t)$ be formed by a Boolean function applied to these sequences, say

$$z_t = f(a_{t1}, \ldots, a_{tk})$$

If the lengths $L_i$ and $L_j$ are pairwise relatively prime for all $i$ and $j$ then the period $p$ and linear complexity $L$ of $(z_t)$ are given by

$$p = \prod_{i=1}^{k} (2^{L_i} - 1)$$
$$L = f(L_1, \ldots, L_k),$$

where $f(L_1, \ldots, L_k)$ is evaluated over the integers.

*Example 4.* Let $(a_{t1}), (a_{t2}), (a_{t3})$ be $m$-sequences produced by $LFSR$'s of relatively prime lengths $L_1, L_2, L_3$. Define a sequence $(z_t)$ by $z_t = a_{t1}a_{t2} + a_{t2}a_{t3} + a_{t3}$. Then the period $p$ and the complexity $L$ of $(z_t)$ are given by

$$p = (2^{L_1} - 1)(2^{L_2} - 1)(2^{L_3} - 1)$$
$$L = L_1L_2 + L_2L_3 + L_3$$

and the algebraic order is two.

In order to increase the value of the linear complexity one or more of the input sequences could be chosen not to be $m$-sequences. For example, we could use a Groth sequence as one of the input sequences in the Geffe sequence in example 4 above. In certain cases it is possible to place a lower bound on the linear complexity as shown in [7].

5.2. *Correlation Immunity* ([16], [18], [19]).

Let $f$ be a binary-valued function of $n$ independent identically distributed binary random variables $x_1, \ldots, x_n$ where $z = f(x_1, \ldots, x_n)$. We say $f$ is $m$-th order correlation immune if and only if for each choice of indices $i_1, i_2, \ldots, i_m$ with $1 \leq i_1 < i_2 < \ldots < i_m \leq n$, the random variable $z$ is statistically independent of the random vector $(x_{i_1}, x_{i_2}, \ldots, x_{i_m})$.

*Example 5.* The Geffe sequence is defined by $z_t = a_{t1}a_{t2} + a_{t2}a_{t3} + a_{t3}$ (see table 2); $(z_t)$ is 0-th order correlation immune but not first order correlation immune since $\text{Prob}(z_t = a_{t1}) = \text{Prob}(z_t = a_{t3}) = \frac{3}{4} \neq \frac{1}{2}$.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $a_{t1}$ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $a_{t2}$ | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $a_{t3}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $z_t$ | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

Table 2. Logic table for the Geffe sequence.

Siegenthaler [19] has demonstrated how to attack a stream cipher formed by a nonlinear combination of *LFSR*'s in the case when the nonlinear function is 0-th order correlation immune. This attack can consist of ciphertext only. The method used involves correlating the ciphertext with the output of a generating *LFSR*. In order to avoid this attack we add the following property (iv) to the basic prerequisites for a good stream cipher in section 1.3.

**Property (iv).** *The nonlinear function should have a correlation immunity order greater than zero.*

Siegenthaler's attack can be extended to a stream cipher in the case where the function is $(m-1)$-th order correlation immune but not $m$-th order correlation immune by correlating the ciphertext with $m$ of the generating *LFSR*'s.

Let $z_t = f(a_{t1}, \ldots, a_{tk})$ as before. The order of correlation immunity, $m$, and the algebraic order, $r$, of the nonlinear combining function satisfy $m + r \leq k - 1$.

In general the linear complexity of $(z_t)$ depends on having a high algebraic order. Hence there is a trade off between linear complexity and correlation immunity. By increasing the order of correlation immunity one may need to decrease the algebraic order and hence the linear complexity. As will be shown by the next example one can solve this trade off between linear complexity and correlation immunity by using one bit of memory.

## 6. Nonlinear combining function with memory ([16]).

6.1. *Introduction.*

Rueppel has shown how to construct a correlation immune sequence by using one bit of memory.

Let $(a_{t1}), \ldots, (a_{tk})$ be sequences produced by shift registers of lengths $L_1, \ldots, L_k$ where the lengths are pairwise relatively prime. Let $(z_t)$ be a sequence of the form

$$z_t = \sum_{i=1}^{k} a_{ti} + s_{t-1} \quad s_t = f(a_{t1}, \ldots, a_{tk}, s_{t-1}),$$

starting with $s_{-1} = 0$. The sequence $(z_t)$ is $k-1$ order correlation immune. There are no restrictions on the choice of the function $f$ to define the memory bit $s_t$.

6.2. *Integer addition sequence.*

Let $(a_t)$ and $(b_t)$ be $m$-sequences, produced by $LFSR_1$ and $LFSR_2$ of relatively prime lengths $m$ and $n$. Define a sequence $(z_t)$ as shown in Figure 6 below:

$$z_t = f_1(a_t, b_t, s_{t-1}) = a_t + b_t + s_{t-1}$$
$$s_t = f_2(a_t, b_t, s_{t-1}) = a_t b_t + (a_t + b_t)s_{t-1}, \quad \text{with} \quad s_{-1} = 0.$$

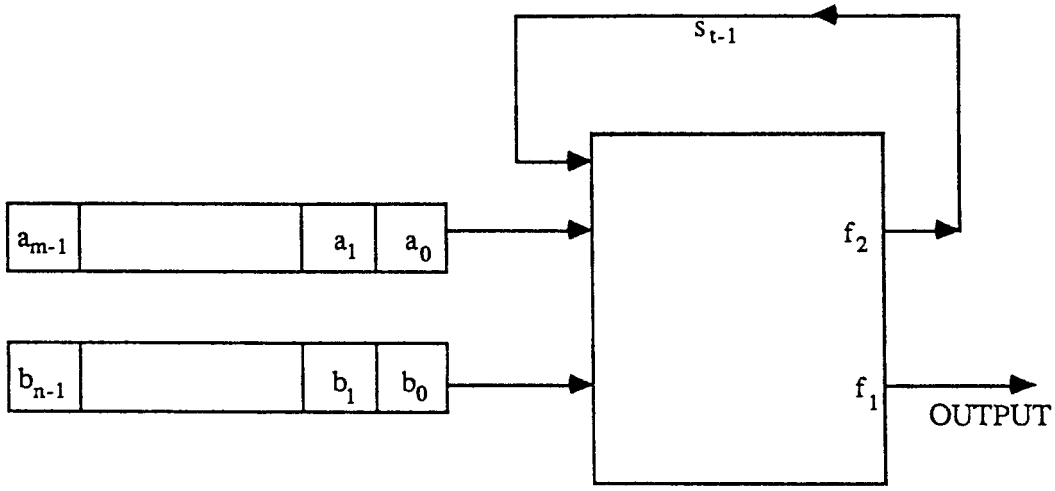



Figure 6.

Such a sequence has the following properties:

(i) Period is $(2^m - 1)(2^n - 1)$.

(ii) Linear Complexity is approximately $(2^m - 1)(2^n - 1)$.

(iii) Sequence is first order correlation immune.

(iv) All sequences tested so far have passed local randomness tests.

Rueppel defines such a sequence to be an integer addition sequence since

$$z_0 = a_0 + b_0$$
$$z_1 = a_1 + b_1 + a_0 b_0$$
$$z_2 = a_2 + b_2 + a_1 b_1 + a_1 a_0 b_0 + b_1 a_0 b_0.$$

Define integers by their binary expansions:

$$a = a_0 + a_1 2 + a_2 2^2 + \cdots, \quad b = b_0 + b_1 2 + b_2 2^2 + \cdots.$$

Then

$$a + b = z = z_0 + z_1 2 + z_2 2^2 + \cdots,$$

where

$$z_0 \equiv a_0 + b_0 \pmod 2$$
$$z_1 \equiv a_1 + b_1 + a_0 b_0 \pmod 2$$
$$z_2 \equiv a_2 + b_2 + a_1 b_1 + a_1 a_0 b_0 + b_1 a_0 b_0 \pmod 2.$$

As can be seen, this has the same form as the definition of the sequence.

We can extend the integer addition sequence to combine $k$ or more sequences. For example, let $(a_t), (b_t), (c_t)$ be $m$-sequences produced by *LFSR*'s of relatively prime lengths. Define $(z_t)$ by

$$z_t = a_t + b_t + c_t + s_{t-1}$$
$$s_t = a_t b_t + a_t c_t + b_t c_t + (a_t + b_t + c_t) s_{t-1},$$

where $s_{-1} = 0$.

In [8], it is shown how to use one bit of memory to design a new type of stream cipher referred to as a Universal Logic Sequence. This sequence seems to offer a similar cryptanalytic strength to Rueppel's integer addition sequence with a larger key size.

## 7. Control LFSR.

### 7.1. *Stop and Go generator* ([3], [20]).

One method of using one *LFSR* to control the clock of another *LFSR* is the Stop and Go generator as shown in Figure 7.

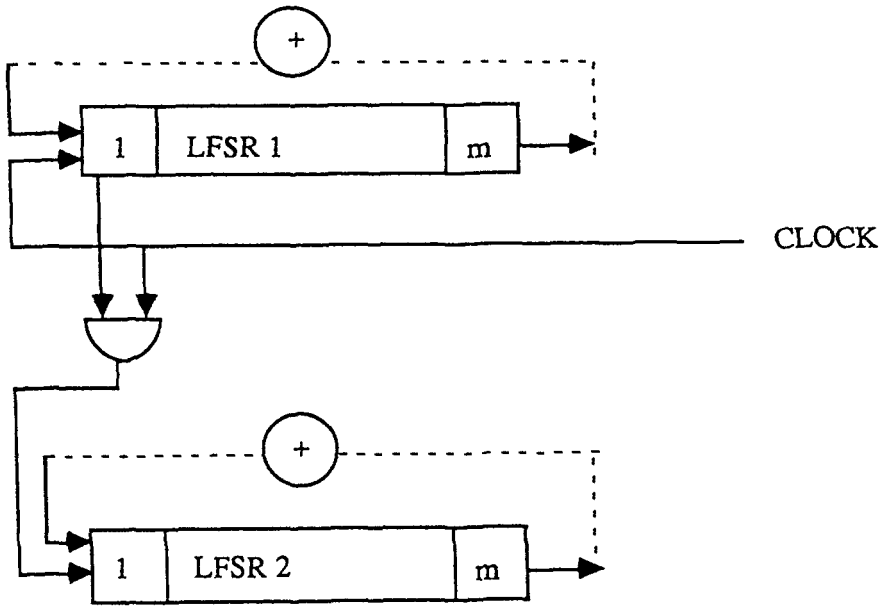



Figure 7.

Suppose we have $m$-sequences $(a_t)$ and $(b_t)$ produced by $LFSR_1$ and $LFSR_2$ respectively each of length $m$. Let the output sequence be $(z_t)$ where

(i) if $a_t = 1$, $LFSR_2$ is shifted one place and the output is $z_t$.

(ii) if $a_t = 0$, $LFSR_2$ is not shifted and the previous output of $LFSR_2$ is used for $z_t$.

These sequences have the following properties

(i) Period is $(2^m - 1)^2$.

(ii) Linear Complexity $m(2^m - 1)$.

(iii) Noiselike characteristics are bad in that asymptotic relative frequencies of the pairs 00 and 11 are $\frac{3}{8}$ and of 10 and 01 are $\frac{1}{8}$.

Other clock controlled generators are discussed in [4], [5], [6], [10], [12], [13], [17] including sequences which satisfy all the properties for a secure sequence for use in a stream cipher.

7.2. *Multiplexed sequences* ([2]).

One method of using the outcome of one *LFSR* to select from another *LFSR* is a multiplexed sequence. Let $LFSR_1$ be of length $m$ and $LFSR_2$ be of length $n$ where $(m, n) = 1$. Let $h \leq m$ where $2^h \leq n$. Select $h$ stages of $LFSR_1$. At each time period these stages define an integer $r_t \leq 2^h$. Define $(z_t)$ by $z_t = b_{t+r_t}$ as shown in Figure 8.

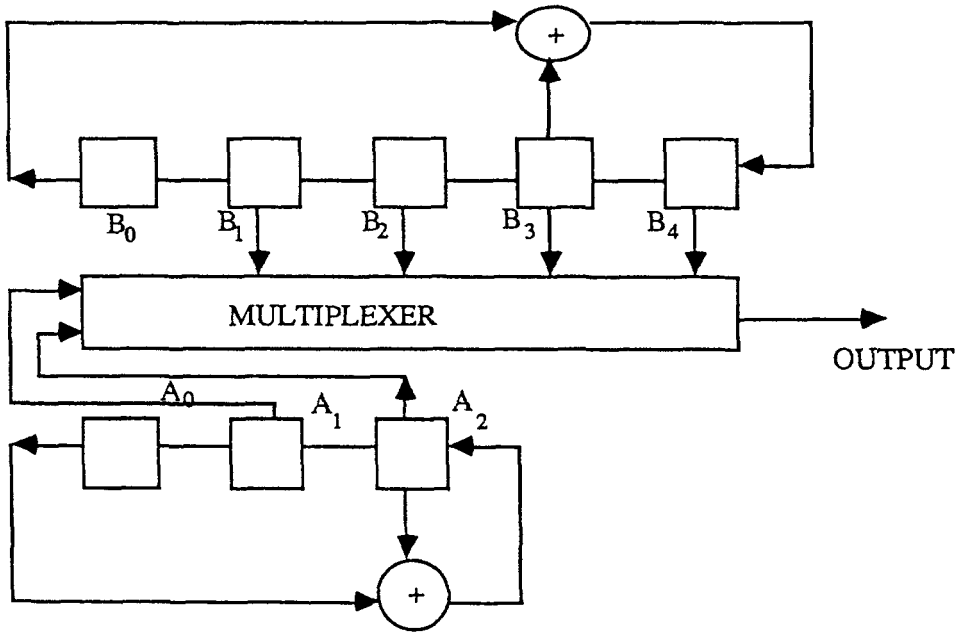



Figure 8.

This sequence has been recommended by the European Broadcasting Union as the standard for scrambling television [9].

These sequences have the following properties:

(i) Period is $(2^m - 1)(2^n - 1)$.

(ii) Linear complexity is $n \sum_{i=0}^{h} \binom{m}{i}$.

(iii) Noiselike characteristics are good.

However, in [15] it is shown that $(z_t)$ is 0-order correlation immune since $b_t$ can be correlated with $z_t$.

## 8. Conclusion.

As has been demonstrated, it is possible to design a sequence for use in a stream cipher secure from cryptanalytic attack by using *LFSR*'s provided a sufficient number of shift registers are used and the right choice is made for the nonlinear combining function. One advantage of using a stream cipher as opposed to a block cipher is

that it may be possible to determine the analytic properties of the sequence used in the stream cipher. For a block cipher it remains difficult to determine the analytic properties of the cipher. Other advantages reside in the simplicity of the design and the speed of the encryption process for a stream cipher in comparison to a block cipher.

## References


1. J. Asenstorfer, E. Dawson and P. Gray, 'Analysis of Groth Sequences', *Journal of Electrical and Electronics Engineering, Australia* **8**, No. 4 (December 1988), 211–221.
2. H. Becker and F. Piper, *Cipher Systems: The Protection of Communications'*. (Northwood Books, London, 1982.)
3. T. Beth and F. C. Piper, 'The Stop and Go Generator', *Proceedings of Eurocrypt '84* (1985), 88–92. (Springer Verlag.)
4. W. G. Chambers, 'Clock-controlled shift registers in binary sequence generators', *IEE Proceedings*, **135**, part E., No. 1 (January 1988), 17–24.
5. W. G. Chambers and D. Gollman, 'Generators for sequences with near-maximal linear equivalence', *IEE Proceedings*, **135**, part E, No. 1 (January 1988), 67–69.
6. W. G. Chambers and S. M. Jennings, 'Linear Equivalence of Certain BRM Shift-Register Sequences', *Electronic Letters*, **20** (November 1984), 1018–1019.
7. E. Dawson, 'Evaluating the Linear Complexity of Nonlinear Generated Binary Sequences', *Journal of Combinatorial Mathematics and Combinatorial Computing*, to appear.
8. E. Dawson and B. Goldburg, 'Universal Logic Sequence', *AUSCRYPT 90*, to appear.
9. European Broadcasting Union, 'Specifications of the systems of the MAC/packet Family', Tech 3258-E (Brussels: EBU technical centre), 1986.
10. D. Gollman, 'Pseudo Random Properties of Cascade Connections of Clock Controlled Shift Registers', *Proceedings of Eurocrypt '84*, (1985), 93–98. (Springer Verlag.)
11. E. J. Groth, 'Generation of binary sequences with controllable complexity', *IEEE Transactions on Information Theory*, **IT-17** (May 1971), 288–296.
12. C. G. Gunther, 'Alternating Step Generators Controlled by De Bruijn Sequences', *Proceedings of Eurocrypt 1987* (1988), 5–14. (Springer Verlag.)
13. K. Kjeldsen and E. Andresen, 'Some Randomness Properties of Cascaded Sequences', *IEEE Transactions on Information Theory*, **IT-26**, No. 2 (March 1980), 227–232.
14. R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications.* (Cambridge University Press, Great Britain, 1986.)
15. S. Mund, D. Gollman and T. Beth, 'Some Remarks on the Cross Correlation Analysis of Pseudo Random Generators', *Advances in Cryptology, Eurocrypt '87*, (1988), 25–33. (Springer Verlag.)
16. R. A. Rueppel, *Analysis and Design of Stream Ciphers.* (Springer-Verlag, Berlin, 1986.)

17. R. A. Rueppel, 'When shift registers clock themselves', *Proceedings of Eurocrypt '87* (1988), 53–64. (Springer Verlag.)
18. T. Siegenthaler, 'Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications', *IEEE Transactions on Information Theory*, **IT-31**, 776–780.
19. T. Siegenthaler, 'Decrypting a Class of Stream Ciphers Using Ciphertext Only', *IEEE Transactions on Computers*, **C-34**, No.1 (January 1985), 81–85.
20. R. Vogel, 'On the linear complexity of cascaded sequences', *Proceedings of Eurocrypt '84* (1985), 99–109. (Springer Verlag.)

*School of Mathematics, Queensland University of Technology,*
*GPO Box 2434, Brisbane, Queensland 4001, AUSTRALIA.*

# APPLYING RANDOMNESS TESTS TO COMMERCIAL LEVEL BLOCK CIPHERS

**Helen Gustafson, Ed Dawson and Bill Caelli**

## 1. Introduction.

In this paper, a report is given on the development of a package for analysis and comparison of block ciphers. This package is being designed in such a manner that block ciphers of the same block length can be compared.

Several *DES* (Data Encryption Standard) replacement block ciphers have been published including:

(i) *FEAL* 4 and 8 (Fast Data Encryption Algorithm),

(ii) Madryga algorithm.

A brief description of each of these ciphers will be given in Section 2. In Section 3, three measures of randomness for finite sequences will be discussed. These measures are:

(i) statistical tests,

(ii) sequence complexity,

(iii) the binary derivative.

In Section 4, it will be shown how the above measures of randomness can be used to examine plaintext–ciphertext independence and the avalanche effect. Several results on experiments in which these measures are applied to the *DES*, *FEAL* 4 and 8, and Madryga will be given.

## 2. Block ciphers.

### 2.1. *FEAL 4 and FEAL 8 algorithms.*

A description of *FEAL* 4 and 8 can be found in [9] and [10]. These ciphers were designed as *DES* replacement ciphers. The ciphers are faster in encryption and decryption speed than *DES* in both software and hardware implementation. A brief description of these ciphers is given below.

(i) The key length is 64 bits.

(ii) The plaintext and ciphertext block lengths are 64 bits.

(iii) Each cipher is a Feistel type cipher like *DES* in that there are several rounds of $F$ function encryption where

round $i : (L_i, R_i)$

round $i+1 : (L_{i+1} = R_i, R_{i+1} = L_i \oplus F(R_i))$

where $\oplus$ denotes addition modulo 2.

(iv) The $F$ function consists of:

(a) addition mod 256,

(b) two bit rotation of bytes,

(c) exclusive -or operation between bytes where *FEAL* 4 has four rounds of the $F$ function and *FEAL* 8 has eight rounds of the $F$ function.

(v) There is both a key processing component and a data processing component in the algorithm where:

(a) the key processing component generates an extended key (256 bits) from the 64 bit key;

(b) the data processing component mixes plaintext with the extended key to produce ciphertext.

In [4], Den Boer has a chosen plaintext cryptanalysis of *FEAL* 4. *FEAL* 4 was extended to *FEAL* 8 in order to avoid this attack. Recently it has been reported that Shamir has designed a successful attack on any eight round Feistel cipher and that a twelve round version of *FEAL* (*FEAL* 12) has been used in order to protect from this attack.

### 2.2. *Madryga algorithm.*

The Madryga algorithm was designed as a *DES* replacement block cipher by the Canadian Imperial Bank of Commerce. A description of this algorithm is given below. A complete description of this algorithm can be found in [8].

(i) The recommended key length is at least 64 bits. In the experiments reported in this paper a key length of 64 bits was used.

(ii) Plaintext and ciphertext block lengths are variable. In the experiments reported in this paper block lengths of 64 bits for both plaintext and ciphertext were used.

(iii) This is not a Feistel type cipher. The encryption algorithm consists of two nested cycles.

(a) The inner nested cycle contains a number of iterations equal to the number of bytes in the message.

(b) The outer nested cycle contains eight passes of the inner cycles through the entire message.

(iv) There is a key hash of the same length as the key which operates with the key to produce pseudorandom sequences for use in the algorithm.

## 3. Measures of randomness.

### 3.1. *Statistical tests.*

There are several statistical tests described in [1] to measure the randomness of a finite sequence. The tests which were used in experiments described in Section 4 are described below.

Suppose that sequences have length $n$ (in the case used $n$ is 64). Let $n_0$ and $n_1$ be the number of zeros and ones respectively in a sequence.

(i) *Frequency test.* Determines whether $n_0$ and $n_1$ are approximately equal. The $\chi^2$ statistic is used to examine the hypothesis that $n_0 = n_1$ where

$$\chi^2 = \frac{(n_0 - n_1)^2}{n}.$$

In the experiments described in Section 4 values of $n_1 > 40$ or $< 24$ were used for a level of significance of $\chi^2$ of approximately 5% and values of $n_1 > 42$ and $< 22$ were used for a level of significance of approximately 1%. Results are shown in Tables 2 and 3.

(ii) *Serial test.* Used to ensure that the probability for consecutive entries being equal or different is about the same. Let

$n_{00}$ be the number of 00 entries,
$n_{01}$ be the number of 01 entries,
$n_{10}$ be the number of 10 entries,
$n_{11}$ be the number of 11 entries.

For random sequences

$$n_{00} = n_{01} = n_{10} = n_{11} \approx \tfrac{1}{4}(n-1) \; (\approx 16 \text{ when } n = 64).$$

The $\chi^2$ statistic is used to test the above hypothesis where

$$\chi^2 = \frac{4}{n-1} \sum_{i=0}^{1} \sum_{j=0}^{1} n_{ij}^2 - \frac{2}{n} \sum_{i=0}^{1} n_i^2 + 1.$$

In the experiments described in Section 4 with two degrees of freedom, $\chi^2$ values at the 5% and 1% levels of significance of 5·991 and 9·210 respectively were used. Results are shown in Tables 4 and 5.

(iii) *Runs test.* The binary sequence is divided into blocks (runs of ones) and gaps (runs of zeros). The runs test examines the number of runs for random data. This test is only applied if the sequence has already passed the serial test in which case it is known that the number of blocks and gaps are in acceptable limits. The number of runs is normally distributed with

$$\text{Mean} = 1 + \frac{2n_0 n_1}{n},$$

$$\text{Variance} = \frac{(\text{Mean} - 1)(\text{Mean} - 2)}{n-1},$$

$$z = \frac{\text{Runs} - \text{Mean}}{\sqrt{\text{Variance}}}.$$

In the experiments described in Section 4 confidence intervals at the 95% and 99% levels were used giving $z$ scores $\pm 1{\cdot}96$ and $\pm 2{\cdot}58$ respectively as shown in Tables 6 and

7. Since the number of runs is a discrete variable a continuity correction of $\pm 0{\cdot}5$ was included in the numerator of the standard normal variable $z$ and the statistic used was in fact

$$|z| = \frac{|\,|\text{Runs} - \text{Mean}| - 0{\cdot}5|}{\sqrt{\text{Variance}}}.$$

3.2. *Sequence complexity.*

A measure for the complexity of a finite sequence $s$ is given in terms of the number of new patterns which appear as we move along the sequence. This number is $c(s)$ called the complexity of $s$ (see references [6] and [7]). For example, the sequence

$$\begin{aligned} s &= 1\,0\,0\,1\,1\,1\,1\,0\,1\,1\,0\,0\,0\,0\,1\,1\,1\,0 \\ &= 1/0/0\ 1/1\,1\,1\ 0/1\,1\,0\ 0/0\,0\,1\,1\,1\,0 \end{aligned}$$

has complexity $c(s) = 6$.

In [6] it is shown that almost all binary sequences of length $n$ have complexity exceeding $n/\log_2 n$. This value will be used as a threshold of complexity for random sequences. In the case where $n = 64$ this threshold value is $10\frac{2}{3}$. In the experiments described in Section 4, Table 8 lists the percentage of blocks in each cipher in each experiment having a complexity $\leq 10$.

Since the sequence complexity test counts new patterns the poker test and autocorrelation tests from [1] were not included in the statistical tests described in Section 3.1. The poker test counts the number of similar patterns of a chosen length in the block and the autocorrelation test checks for periodicity in the block.

3.3. *Binary derivative.*

Given a string of binary digits the first derivative is taken by considering each overlapping pair of digits and recording a zero if they are the same and a one if they are different (see references [2] and [3]). Every successive binary derivative drops one digit. A sequence of length 16 and its first four binary derivatives are given below:

$$\begin{array}{cccccccccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ & & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ & & & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}$$

Let $p(i)$ denote the fraction of ones in the $i$-th derivative where $p(0)$ denotes the fraction of ones in original sequence. Suppose that $k$ derivatives are evaluated for a string of length $n$. Let $r$ denote the range of the $p(i)$ and $p$ denote the average of the $p(i)$ where

$$\begin{aligned} p_{\max} &= \max p(i) \\ p_{\min} &= \min p(i) \\ r &= p_{\max} - p_{\min} \\ p &= \sum_{i=0}^{k} \frac{p(i)}{k+1}. \end{aligned}$$

For the above sequence of length 16, we find $p(0) = 0{\cdot}44$, $p(1) = 0{\cdot}67$, $p(2) = 0{\cdot}43$, $p(3) = 0{\cdot}54$, $p(4) = 0{\cdot}75$, $r = 0{\cdot}32$, $p = 0{\cdot}57$.

In [3] it is stated that one can use $p(0), p$ and $r$ to differentiate between patterned and random strings as shown in Table 1.

| Attribute | Patterned strings | Random strings |
|---|---|---|
| $p(0)$ | variable | close to 0·5 |
| $p$ | low | close to 0·5 |
| $r$ | high | low |

Table 1.

In order to determine how many derivatives to evaluate to differentiate between patterned and random sequences of length $n$ it is suggested in [2] that one use experimental results. To this end, the complexity test from Section 3.2 was used on pseudorandom binary data generated by a stream cipher to give 1000 low complexity binary strings of length 64 (Rand 1) and 1000 high complexity binary strings of length 64 (Rand 2) as shown in Figure 1. Binary derivatives were calculated for each of these sets of sequences as shown in Figure 2 where the horizontal axis represents the number of derivatives calculated, say $k$, and the vertical axis represents the average of the ranges for each set of sequences after $k$ derivatives have been taken. From the graph it appears that between the 5-th and 8-th derivatives there is the greatest difference between the average ranges for the low and high complexity sequences. It was decided to use seven derivatives to measure randomness.

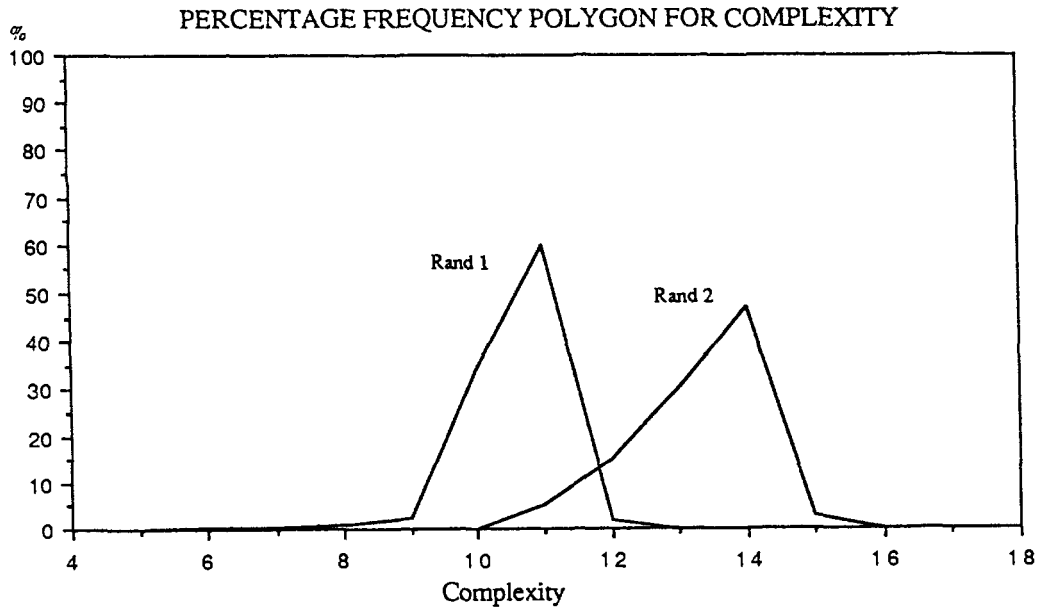



Figure 1.

In order to measure significance levels for $p$ and $r$, 10000 random blocks of length 64 were generated by a stream cipher. For these blocks 95% and 99% confidence intervals for $p$ were approximately $0{\cdot}45 \leq p \leq 0{\cdot}54$ and $0{\cdot}43 \leq p \leq 0{\cdot}55$ respectively. These

figures were used as 5% and 1% significant levels for the experiments in Section 4 for $p$ as shown in Tables 9 and 10. For the 10000 random blocks of length 64 generated, 95% and 99% had values of approximately $r$ less than 0·28 and 0·33 respectively. Tables 11 and 12 show the % of blocks in each of the experiments described in Section 4 which had $r > 0{\cdot}28$ and 0·33 respectively.

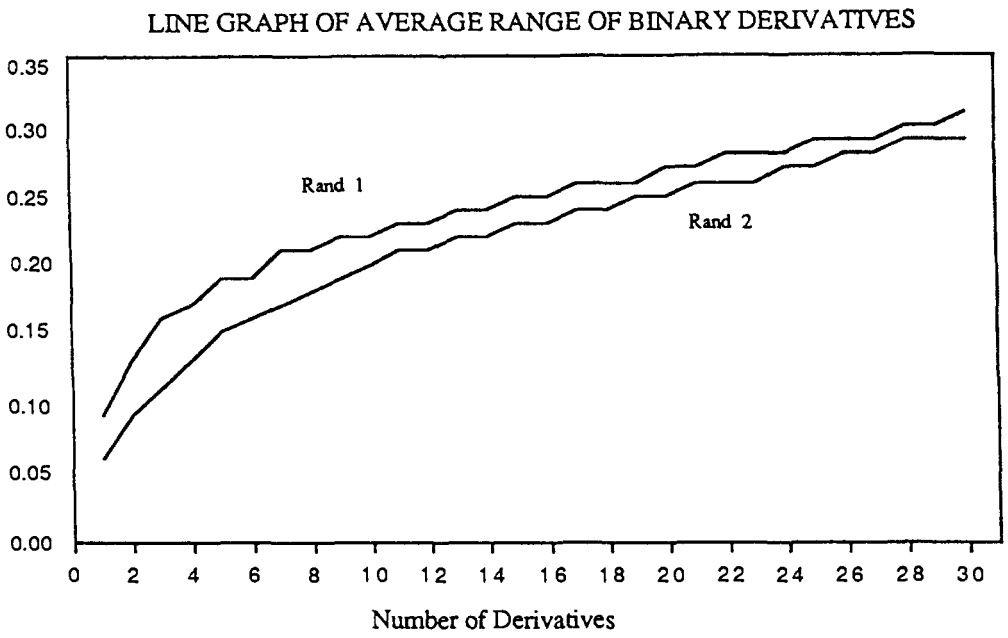



Figure 2.

## 4. Applying randomness tests.

### 4.1. *Plaintext and ciphertext independence.*

Several methods have been suggested in [7] to examine plaintext and ciphertext independence.

One method is to generate nonrandom sequences as binary plaintext. If the ciphertext is independent of the plaintext it should be random. In experiment 1 all vectors of length 64 containing 64, 63, 62 zeros or ones were used as plaintext. There are a total of 4162 such vectors. The results of experiment 1 are included in tables 2 to 12. By inspection of these tables, there is no noticeable difference between the ciphers. However, the Madryga cipher generated very few sequences with an odd number of ones which indicates a possible weakness in this cipher.

A second method is to generate a large number of random plaintext vectors $\mathbf{p}_1, \ldots, \mathbf{p}_r$. Let $\mathbf{c}_1, \ldots, \mathbf{c}_r$ be the resultant ciphertext vectors with a fixed key. Define $\mathbf{s}_i = \mathbf{p}_i \oplus \mathbf{c}_i$ for $i = 1, \ldots, r$. The vectors $\mathbf{s}_i$ should be random if ciphertext is independent of plaintext.

In experiment 2, 10000 random plaintext vectors were used. The random plaintext vectors were generated by a stream cipher. The results of experiment 2 are included in Tables 2 to 12. By inspection of this data, there are no noticeable differences between the ciphers. However, the Madryga cipher generated no vectors $\mathbf{s}_i$ with an odd number of ones which indicates a possible weakness in this cipher.

4.2. *Avalanche Effect.*

A block cipher satisfies the avalance effect if a small change in the plaintext gives rise to large change in the ciphertext (see [7], [5], [11]).

In order to measure the avalanche effect generate a large number of random plaintext vectors $\mathbf{p}_1, \ldots, \mathbf{p}_r$. Let $\mathbf{c}_1, \ldots, \mathbf{c}_r$ be the resulting ciphertext vectors. Let $\mathbf{p}'_1, \ldots, \mathbf{p}'_r$ be vectors resulting from changing the $i$-th bit of $\mathbf{p}_1, \ldots, \mathbf{p}_r$ (where $i$ is some fixed value selected at random). Let $\mathbf{c}'_1, \ldots, \mathbf{c}'_r$ be the resulting ciphertext. Let $\mathbf{u}_j = \mathbf{c}_j \oplus \mathbf{c}'_j$ for $j = 1, \ldots, r$. Then, the $\mathbf{u}_j$ should be random vectors if the cipher satisfies the avalanche effect in the $i$-th place. In order to completely examine the avalanche effect it would be necessary to select all possible values for $i$.

In experiment 3, 10000 random plaintext vectors generated by a stream cipher were used for each cipher and 20000 ciphertext vectors corresponding to these plaintext vectors and vectors resulting from one bit change in first position were generated for each cipher. These defined 10000 avalanche vectors $\mathbf{u}_j$ as defined above for each cipher. The results of the experiment are included in tables 2 to 12. The Madryga cipher generated no vectors $\mathbf{u}_j$ with an even number of ones. From the numerical results included in tables 2 to 12 the Madryga cipher has failed the avalanche effect in terms of the first bit position.

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 2·90 | 3·63 | 2·38 | 3·56 |
| Exp 2 | 3·31 | 3·02 | 1·64 | 2·12 |
| Exp 3 | 3·41 | 3·40 | 8·86 | 3·56 |

Table 2. Number of Ones (% of $n_1$ where $n_1 < 24$ or $n_1 > 40$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 0·72 | 0·72 | 0·70 | 1·01 |
| Exp 2 | 0·85 | 0·79 | 0·59 | 0·82 |
| Exp 3 | 0·92 | 0·76 | 5·16 | 0·90 |

Table 3. Number of Ones. (% of $n_1$ where $n_1 < 22$ or $n_1 > 42$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 4·52 | 4·48 | 5·33 | 4·88 |
| Exp 2 | 5·06 | 4·62 | 4·74 | 4·82 |
| Exp 3 | 5·05 | 4·72 | 9·46 | 4·97 |

Table 4. Serial Test. (% of $\chi^2 > 5{\cdot}991$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 0·84 | 0·84 | 1·13 | 1·20 |
| Exp 2 | 1·13 | 1·01 | 1·03 | 0·93 |
| Exp 3 | 0·91 | 1·02 | 5·21 | 1·20 |

Table 5. Serial Test. (% of $\chi^2 > 9{\cdot}210$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 3·41 | 3·48 | 3·89 | 3·44 |
| Exp 2 | 3·37 | 3·77 | 3·09 | 3·39 |
| Exp 3 | 3·63 | 3·28 | 6·47 | 3·39 |

Table 6. Runs Test. (% of $|z| > 1{\cdot}96$).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 0·67 | 0·67 | 0·60 | 0·60 |
| Exp 2 | 0·76 | 0·71 | 0·59 | 0·61 |
| Exp 3 | 0·60 | 0·68 | 3·00 | 0·72 |

Table 7. Runs test (% of $|z| > 2{\cdot}5758$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 3·33 | 4·08 | 4·08 | 4·11 |
| Exp 2 | 3·81 | 3·99 | 4·15 | 4·37 |
| Exp 3 | 4·00 | 4·08 | 7·67 | 4·01 |

Table 8. Complexity Data. (% $\leq 10$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 5·05 | 4·20 | 4·97 | 4·32 |
| Exp 2 | 4·59 | 5·19 | 4·67 | 4·93 |
| Exp 3 | 4·86 | 4·51 | 9·27 | 4·48 |

Table 9. Average of Binary Derivative. (% of $p$ where $p < 0{\cdot}45$ or $p > 0{\cdot}54$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 1·15 | 0·72 | 1·03 | 0·89 |
| Exp 2 | 0·91 | 1·09 | 0·92 | 1·02 |
| Exp 3 | 1·24 | 0·91 | 5·24 | 0·95 |

Table 10. Average of Binary Derivative (% of $p$ where $p < 0{\cdot}43$ or $p > 0{\cdot}55$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 4·88 | 5·07 | 4·81 | 5·21 |
| Exp 2 | 4·93 | 4·90 | 5·15 | 4·96 |
| Exp 3 | 4·94 | 5·10 | 5·30 | 4·84 |

Table 11. Range of Binary Derivative. (% of $r > 0{\cdot}28$.).

| | *FEAL* 4 | *FEAL* 8 | Madryga | *DES* |
|---|---|---|---|---|
| Exp 1 | 1·01 | 1·25 | 1·18 | 1·15 |
| Exp 2 | 1·33 | 1·19 | 1·06 | 1·00 |
| Exp 3 | 0·99 | 1·24 | 1·17 | 1·09 |

Table 12. Range of Binary Derivative. (% of $r > 0{\cdot}33$.).

## 5. Conclusion.

Based on the results of experiments 1 to 3, it is concluded that the three Feistel Ciphers *FEAL* 4 and 8, and *DES* pass all the tests for randomness. The Madryga cipher based on the results of the avalanche effect and the peculiarity with the number of ones in the ciphertext needs further investigation.

It is planned to add further tests to the package being designed including the key avalanche effect and the formation and analysis of an avalanche matrix.

The aim of this project is the formation of a package to investigate and compare block ciphers of the same length. A cryptographer could use such a package to help identify a weakness in a cipher system.

## References

1. H. Becker and F. Piper, *Cipher Systems: The Protection of Communications.* (John Wiley and Sons, 1982.)
2. J. M. Carroll, 'The binary derivative test for the appearance of randomness and its use as a noise filter'. Technical Report No. 221, Dept. of Computer Science, University of Western Ontario, November 1988.
3. J. M. Carroll and L. E. Robbins, 'Using Binary Derivatives to Test an Enhancement of *DES*', *Cryptologia* **XII**, Number 4 (Oct. 1988), 193–208.
4. B. Den Boer, 'Cryptanalysis of F.E.A.L.', *Advances in Cryptology: Proc. Eurocrypt '88*, 293–299.
5. A. G. Konheim, *Cryptography: A Primer.* (John Wiley and Sons, 1981.)
6. A. Lempel and J. Ziv, 'On the Complexity of Finite Sequences', *IEEE Trans. on Information Theory* **IT–22** (Jan. 1976), 75–81.
7. A. K. Leung and S. E. Tavares, 'Sequence Complexity as a Test for Cryptographic Systems', *Advances in Cryptology, Crypto '84*, 468–474.
8. W. E. Madryga, 'A High Performance Encryption Algorithm', *Computer Security: A Global Challenge.* (Elsevier Science Publishers B.V., 1984, 557–570.)
9. S. Miyaguchi, A. Shiraishi and A. Shimizu, 'Fast Data Encipherment Algorithm *FEAL*-8', *Review of the Electrical Communications Laboratories*, **36**, No. 4 (1988), 433–437.
10. A. Shimizu and S. Miyaguchi, 'Fast Data Encipherment Algorithm *FEAL*', *Advances in Cryptology: Proc. Eurocrypt '87* (1988), 267–278.
11. A. F. Webster and S. E. Tavares, 'On the Design of S-Boxes', *Advances in Cryptology: Crypto '85* (1986), 523-530.

*School of Mathematics, Queensland University of Technology,*
*GPO Box 2434, Brisbane, Queensland 4001, AUSTRALIA.*

*Information Security Research Centre, Queensland University of Technology,*
*GPO Box 2434, Brisbane, Queensland 4001, AUSTRALIA.*

10

# PSEUDO-RANDOM SEQUENCE GENERATORS USING STRUCTURED NOISE

R. S. Safavi-Naini and J. R. Seberry

Stream ciphers use the output of a Pseudo-Random ($PR$) generator to mask the information stream. The security of these cipher systems ultimately depends on the structure of the $PR$ generator. There are some minimum necessary criteria such as long period, flat statistical distribution and high linear complexity that the $PR$ generator of a stream cipher system should satisfy to resist the basic cryptanalytic attacks on such systems. We propose a class of $PR$ generators using the coset elements of a Reed-Muller code. The linear complexity of these generators is analysed and conditions that assure the highest possible linear complexity for them are specified. It is shown that the above mentioned criteria do not gurantee the security of a stream cipher system and the proposed $PR$ generator, although it satisfies all of them, is not secure.

## 1. Introduction.

Stream ciphers assimilate the one time pad, the only provably perfect secure system. However with the replacement of the random generator by a pseudo-random ($PR$) one, the perfect security of the system vanishes. It is easy to see that the assessment of the security of these systems is directly related to the properties of the $PR$ generator. There are some necessary criteria which must be satisfied by the $PR$ generator of a secure stream cipher. It is recognised that these generators should satisfy Golomb's criteria and have high linear complexity [1], [3]. Linear feedback shift register ($LFSR$) generated sequences satisfy Golomb's criteria but have a small linear complexity and hence fail to offer high security. There have been many attempts to devise algorithms based on $LFSR$'s that retain the good properties of these sequences and increase the linear complexity of them. In this paper we propose noise addition as a mechanism to achieve the above-mentioned goal. In the rest of this section we briefly review some relevant background material. In Section 2, the properties of some special subsets of the binary vector space of dimension $2^m - 1$ are studied and the results are used in Section 3 to develop a new class of stream ciphers. The paper is concluded by establishing the insufficiency of linear complexity and Golomb's criteria for the security of $PR$ generators.

*Linear Equivalence.* A linear feedback shift register ($LFSR$) is a finite state machine, the state of which at time $t$ is determined by its content at time $t$ and its next state is determined by its feedback function (we consider the binary case only).

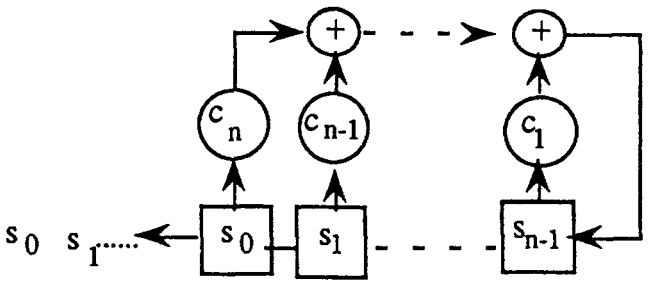



Figure 1.

The feedback polynomial of the *LFSR* is defined as

$$C(D) = 1 + c_1 D + c_2 D^2 + \ldots + c_m D^m,$$

where $c_i$'s are binary feedback coefficients. Let $s = s_0, s_1, s_2, \ldots$ denote the semi-infinite output sequence of the *LFSR*. The $D$-transform of $s$ is $S(D)$ defined by

$$S(D) = s_0 + s_1 D + s_2 D^2 + \ldots .$$

The output sequence $s$ is a periodic sequence, the period of which is determined by the properties of $C(D)$ [1]. An $m$-sequence has the maximum period, $2^m - 1$, and corresponds to the case when the feedback polynomial is primitive. (See the contribution by Dawson in these Proceedings.) The $D$-transform of a sequence $s$ generated by an *LFSR* can be written as:

$$S(D) = \frac{P(D)}{C(D)},$$

where $P$ is a polynomial with degree less than $m$. In fact there is a one-to-one correspondence between polynomials of degree less than $m$ and the set of $2^m - 1$ output sequences of the *LFSR* corresponding to the $2^m - 1$ non-zero possible initial conditions.

Every periodic sequence $s$ of period $T$ can be thought of as generated by a *LFSR* of length $T$. The $D$-transform of the sequence can be written as:

$$S(D) = \frac{S(D)^*}{1 + D^T}$$

where $S(D)^*$ denotes the $D$-transform of the first period. By cancelling the common factors from the numerator and the denominator one can find the *LFSR* of minimum length that can generate the sequence. The length of this *LFSR* is the linear equivalence of the sequence and is taken as a measure of its complexity.

The first periods of periodic sequences of period $T$ constitute a $T$-dimensional vector space $V_T$. The linear equivalence of $\mathbf{v}$ in $V_T$ is denoted by $L(\mathbf{v})$ and defined as the linear equivalence of the semi-infinite sequence with first period equal to $\mathbf{v}$. The $D$-transform of $\mathbf{v}$ is defined as the $D$-transform of the first period of this sequence. One set of basic vectors can be obtained by decomposing $1 + D^T$ into irreducible factors:

$$1 + D^T = \prod_i C_i(D),$$

where $C_i(D)$'s are irreducible polynomials over $GF(2)$. By applying a partial fraction expansion to $S(D)$, the $D$-transform of a sequence $s$ of period $T$, we have

$$S(D) = \sum_i \frac{P_i(D)^*}{C_i(D)},$$

which shows the sequence can be generated by adding (modulo 2) the output of some basic *LFSR*'s with irreducible feedback polynomials given by $C_i(D)$ and their initial content determined by $P_i(D)$. The total length of the basic *LFSR*'s is equal to $T$.

*Example 1*: Let the first period of a sequence $s$ be:

1 0 1 0 1 0 1 1 1 0 0 0 0 1 1

and $S(D)$ denotes its $D$-transform. Partial fraction expansion of $S(D)$ results in:

$$S(D) = \frac{1 + D^2 + D^4 + D^6 + D^7 + D^8 + D^{13} + D^{14}}{1 + D^{15}}$$
$$= \frac{D}{1 + D + D^2} + \frac{1}{1 + D + D^2 + D^3 + D^4}$$

So the sequence can be generated by a *LFSR* of length 6.

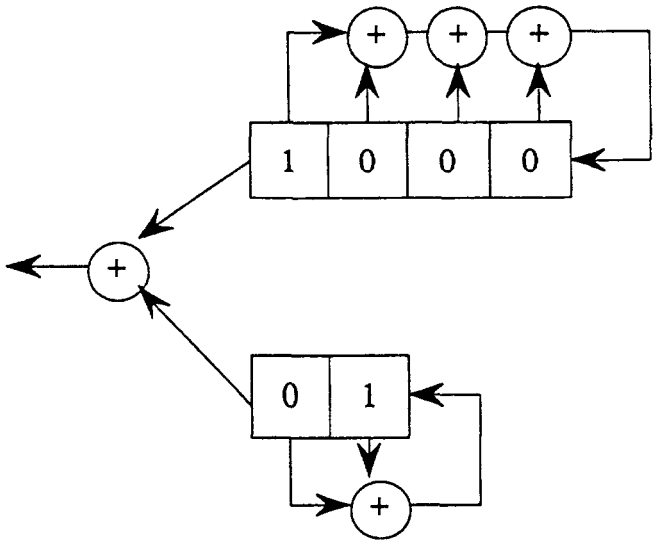



Figure 2.

*Reed-Muller Code.* Binary Reed-Muller code of order $r$ and length $2^m - 1$, denoted by $RM(r, m)$ is the set of all Boolean functions $f$ of $m$ Boolean variables $v_1, v_2, \ldots, v_m$ where $f(v_1, v_2, \ldots, v_m)$ is a polynomial of order at most $r$ when written in algebraic normal form [2]. The Reed-Muller code of order one comprises the set of linear Boolean functions of $m$ variables and can be partitioned into two subsets one of which is the set of complement vectors of the other:

$$RM(1, m) = O_m \cup (1 + O_m),$$

where $O_m$ is a subspace by itself and has an all-zero column. The linear code obtained by deleting this column, denoted by $O_m^*$, is in fact the set of $m$-sequences generated by a *LFSR* with the primitive feedback polynomial equal to the parity check polynomial of $O_m^*$. The minimum distance of $O_m^*$ is $2^{m-1}$ and can correct $2^{m-2}-1$ errors. Hence in the standard array correspnding to $O_m^*$, all vectors of weight $2^{m-2}-1$ appear as coset leaders and these cosets can be identified by their leaders [3].

*Example 2*: The generator polynomial $g(x)$ and the parity check polynomial $h(x)$ of $O_3^*$ are :

$$g(x) = 1 + x^2 + x^3 + x^4 = (1+x)(1+x+x^3)$$
$$h(x) = 1 + x^2 + x^3$$

The non-zero codewords (listed below) can be generated by a *LFSR* with feedback polynomial $C(D) = 1 + D^2 + D^3$:

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |

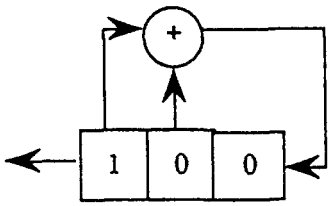



Figure 3.

## 2. PR Sequences from RM(1, m).

The following propositions give some classification of the elements of $V_T$ in terms of their linear equivalence.

**Proposition 1.** *The linear equivalence of the vectors of* $O_m^*$ *is* $m$.

*Proof.* Follows from the fact that the codewords are $m$-sequences.

**Proposition 2.** *Let* $\mathbf{e} \in V_T$ $(T = 2^m - 1)$, *be a coset leader of* $O_m^*$ *with* $L(\mathbf{e}) = h$. *The linear equivalence* $l_{\mathbf{e}}$ *of all but at most one of the elements of the coset with* $\mathbf{e}$ *as the leader is the same. The only possible values of* $l_{\mathbf{e}}$ *are* $h$ *and* $h \pm m$.

*Proof.* See [4].

So cosets of $O_m^*$ are subsets of constant (almost) linear equivalence and coset decomposition of $V_T$ can be regarded as decomposition in terms of linear complexity also. In fact, to specify the linear equivalence of a vector, it is almost always enough to determine the coset to which it belongs. This is the standard decoding problem of $O_m^*$ which is well studied [2].

**Proposition 3.** *Let* $\mathbf{e} \in V_T$ $(T = 2^m - 1)$ *and* $w(\mathbf{e}) = 1$. *Then* $L(\mathbf{e}) = 2^m - 1$.

*Proof.* See [4].

**Corollary.** *In every coset of weight one of* $O_m^*$ *there exists exactly one vector of linear equivalence* $2^m - 1 - m$. *All the other vectors have linear equivalence* $2^m - 1$.

This gives an explicit expression of the linear equivalence of all vectors of cosets of weight one and shows that almost all these vectors have the highest linear equivalence. Dependence of the linear equivalence of coset elements on the weight of the coset leaders cannot be extended to higher weights for which the specific error pattern affects the complexity of the vector. Let $w(\mathbf{e})$ denote the Hamming weight of $\mathbf{e}$.

**Proposition 4.** *The linear equivalence of a vector* $\mathbf{e} \in V_T$ *with* $w(\mathbf{e}) = 2$, *depends on the distance between the two non-zero components. If this distance is denoted by* $t$ *and* $\deg(\gcd(1 + D^t, 1 + D^T)) = u$, *then* $L(\mathbf{e}) = T - u$.

*Proof.* See [4].

**Corollary.** *The linear equivalence of vectors of cosets of weight two of* $O_m^*$ *depends on the distance* $t$ *between the two non-zero components of the coset leader. If we have* $T = 2^m - 1$ *and* $\deg(\gcd(1 + D^t, 1 + D^T)) = u$, *then the linear complexity of the coset element is at least* $2^m - 1 - u - m$.

*Example 3.* Consider the vectors of weight 2 in $V_{15}$. Let $t$ be the distance between the non-zero bits of a vector $\mathbf{e}$ where $w(\mathbf{e}) = 2$. The possible values of $t$ are $1, 2, \ldots, 14$ which result in the following values for the linear equivalence:

| $t$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $L(\mathbf{e})$ | 14 | 14 | 12 | 14 | 10 | 12 | 14 | 14 | 12 | 10 | 14 | 12 | 14 | 14 |

It should be noted that the distance between the two non-zero components and not their actual position is important. So all the vectors listed below have the same linear equivalence 10 corresponding to $t = 10$ in the above table:

| | |
|---|---|
| $\mathbf{e}_1$ | 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 |
| $\mathbf{e}_2$ | 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 |
| $\mathbf{e}_3$ | 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 |
| $\mathbf{e}_4$ | 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 |
| $\mathbf{e}_5$ | 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 |

For cosets of weight greater than two the linear complexity is independent of the shift in the error pattern.

**Proposition 5.** *Let* $E^*(D) = D^k E(D)$, $E(0) = 1$, *denote the* $D$*-transform of a coset leader* $\mathbf{e}$ *and* $\deg(\gcd(E(D), 1 + D^T)) = u$ $(T = 2^m - 1)$. *The linear complexity of a coset element is at least* $2^m - 1 - u - m$.

*Proof.* The $D$-transform of a sequence with first period given by $\mathbf{e}$ is

$$\frac{E(D)}{1 + D^T} \quad (T = 2^m - 1).$$

It follows that the linear equivalence of this sequence is $T - u$ and the linear equivalence of a vector of the coset with $\mathbf{e}$ as a leader is at least $T - u - m$. In fact there is exactly one vector of this complexity and the linear complexity of the rest is $T - u$.

**Corollary 1.** *Let* $\mathbf{e} \in V_T$ *and* $E(D)$ *be a power of a prime polynomial. Then* $L(\mathbf{e}) = 2^m - 1$.

**Corollary 2.** *Let* $\mathbf{e}$ *and* $\mathbf{e}'$ *be two vectors of length* $T = 2^m - 1$ *such that*

$$E_{\mathbf{e}}(D) = D^T E_{\mathbf{e}'}(D^{-1}).$$

*Then* $L(\mathbf{e}) = L(\mathbf{e}')$.

*Proof.* Follows from the fact that

$$\gcd(E_{\mathbf{e}}(D), 1 + D^T) = \gcd(E_{\mathbf{e}'}(D), 1 + D^T).$$

## 3. Key Generators Using Structured Noise.

One can use noise addition to increase the linear equivalence of a vector. A PR-generator based on this idea is shown in Figure 4.

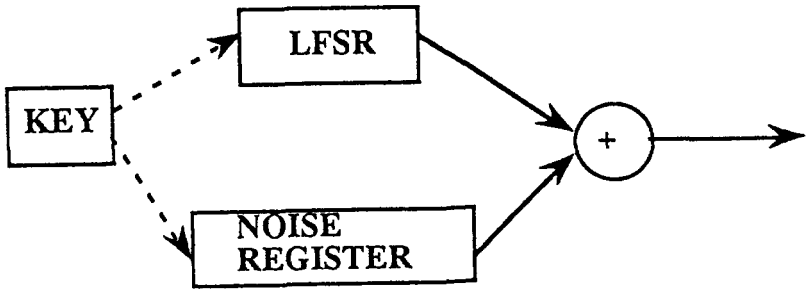



Figure 4.

The *LFSR* generates an $m$-sequence of period $T$ which is added modulo two to a noise sequence of the same period. The noise register contains a class of noise vectors of length $T$ that have high linear equivalence and are easy to generate (easy to describe). The class is parametrized by a piece of the key information. A common key

in transmitter and receiver produces the same *PR*-sequence at both ends. This is an easy and fast way of enhancing the linear complexity of an $m$-sequence. A large number of noise sequences stored in the register will prevent a cryptanalyst from recovering the $m$-sequence easily. Propositions 3 and 4 imply that all vectors of weight one and a large number of vectors of weight two (the linear equivalence of the vector should be determined) can be included in the noise register. Using Proposition 5 it can be seen that finding the proper noise vector is equivalent to finding polynomials which are relatively prime to $1+D^T$, or have a gcd of small degree. Proposition 6 specifies a class of polynomials that correspond to vectors of high linear complexity. This result can be combined with the corollary to Proposition 5 to characterise suitable noise vectors. We need the following definition.

*Definition* ([3]). The *cyclotomic coset* mod $n$ over $GF(q)$ which contains $s$ is

$$C_s = \{s, sq, sq^2, ....., sq^{T_s}\}, \text{ where } sq^{m_s} \equiv s \bmod n,\ T_s = m_s - 1.$$

**Proposition 6.** *A sequence of length* $2^m - 1$ *that consists of* $t$ *consecutive ones, where* $t$ *is a prime such that* $\gcd(t, 2^m - 1) = 1$ *and* $t$ *has only two cyclotomic cosets, has linear equivalence* $2^m - 1$.

*Proof.* Let $T = 2^m - 1$. Since, by hypothesis, $\gcd(t, 2^m - 1) = 1$, it follows that $\gcd(1 + D^t, 1 + D^T) = 1$. Further, since $t$ has only two cyclotomic cosets,

$$1 + D^t = (1 + D) \cdot \sum_{i=0}^{t-1} D^i.$$

Now it is easy to see that

$$\gcd\left(1 + D^{2^m-1}, \sum_{i=0}^{t-1} D^i\right) = 1,$$

where the second polynomial corresponds to a vector of $t$ consecutive zeros.

*Security.* It is easy to see that the *PR* generator discussed in this section satisfies Golomb's criteria [1] closely, if we restrict the weight of the noise sequences used in the noise register to small values (compared to the length of the *LFSR*). It is also noted that the proper selection of noise vectors ensures a lower bound on the linear complexity of the output sequence. However the small weight of the noise vectors implies that the first period of the sequence can be approximated by a *LFSR* sequence and would be highly predictable after $2m$ ($m$ is the length of *LFSR*) bits of it were intercepted. So the sequence is not secure.

## 4. Concluding Remarks.

The study of linear complexity of a Reed-Muller code and its cosets suggests a new way of increasing the linear equivalence of an $m$-sequence. The results are applied to devise a *PR* generator. The added complexity is the result of adding a noise vector

with high linear equivalence to the $m$-sequence. If the weight of the noise vector is one, the statistical parameters of the original $m$-sequence would not be greatly affected and the resulting sequence would closely satisfy the known criteria of security. It is noticed that in general for noise vectors of small weight the generator is not secure because the output can be approximated by an $m$-sequence. On the other hand, noise vectors of higher weight deteriorate the statistical properties of the $m$-sequence.

The $PR$ sequences obtained from the elements of cosets of weight one demonstrate the insufficiency of the criteria used for assessing the $PR$ generators of stream ciphers. Providing criteria to measure the security of such systems remains an open problem.

## References

1. H. Becker and F. Piper, *Cipher Systems.* (North Books, London, 1982.)
2. F. J. MacWilliams and N. J. Sloane, *The Theory of Error-Correcting Codes.* (North-Holland Publishing Company, 1978.)
3. R. A. Rueppel, *Analysis and Design of Stream Ciphers.* (Springer-Verlag, Berlin 1986.)
4. R. S. Safavi-Naini and J. R. Seberry, 'Pseudo-Random Sequences from Codes' Technical Report CS89/10, University College, University of New South Wales Canberra, Australia.

*Department of Computer Science, University College, The University of New South Wales, Australian Defence Force Academy, Canberra, ACT 2600, AUSTRALIA.*

# 11

# PRIVACY FOR MACNET

**Michael Warner**

This paper examines the security of the *MACNET* shared fibre access network. The major threats to security are identified, and an encryption and key management scheme is proposed which is shown to bring the security to a similar level as that provided by dedicated fibre access. The robustness of the scheme to transmission errors, as well as the ease of implementation are also considered.

## 1. Introduction.

The Customer Access Network (*CAN*) is that part of the telecommunications network located between the local exchange and the customer's premises. Traditionally, a star network architecture based on twisted copper pairs has been used; however, this is unable to meet the requirements of future broadband services. Due to the number of premises connected, the *CAN* represents a substantial proportion of the total capital investment in the Australian telecommunications network. To prevent the need for subsequent upgrading, it is desirable to 'future-proof' any alterations to the *CAN* by providing a transmission medium capable of meeting future needs.

Over the next few years, optical fibre will be progressively introduced into the *CAN*, providing broadband access to the customer. Current costs associated with optical equipment make the cost of a dedicated fibre *CAN* (ie a simple star architecture) prohibitive. One of the most promising schemes for early introduction of fibre to the *CAN* is the Multiple Access Customer Network (*MACNET*) [5,6,7]. *MACNET* uses passive optical couplers to service a number of customers from the same fibre (or fibre pair). While this structure has cost advantages over dedicated fibre access, it presents a unique environment in terms of network security. To satisfy customer demands, the *MACNET* system should provide a similar level of security as dedicated fibre.

This report highlights a number of potential threats associated with the *MACNET* topology, and outlines a possible solution to the most pressing concern—that of downstream privacy. The technical feasibility and level of security provided by this solution are investigated.

## 2. The MACNET Architecture.

*MACNET* is a network architecture which can provide a group of customers who are located geographically close to each other with an optical fibre connection between themselves and their local exchange [5,6,7]. This architecture is based on the use of a passive splitter, which allows the exchange equipment and main optical fibre cable to

be shared among a number of customers (typically 16). The passive optical splitter is located out in the *CAN* near the group of customers. Dedicated fibres connect the customers to the splitter, which in turn is connected to the exchange by a single, shared main fibre (or fibre pair). The optical coupler was chosen to be passive so as to avoid the costs associated with introducing active electronics into the external plant.

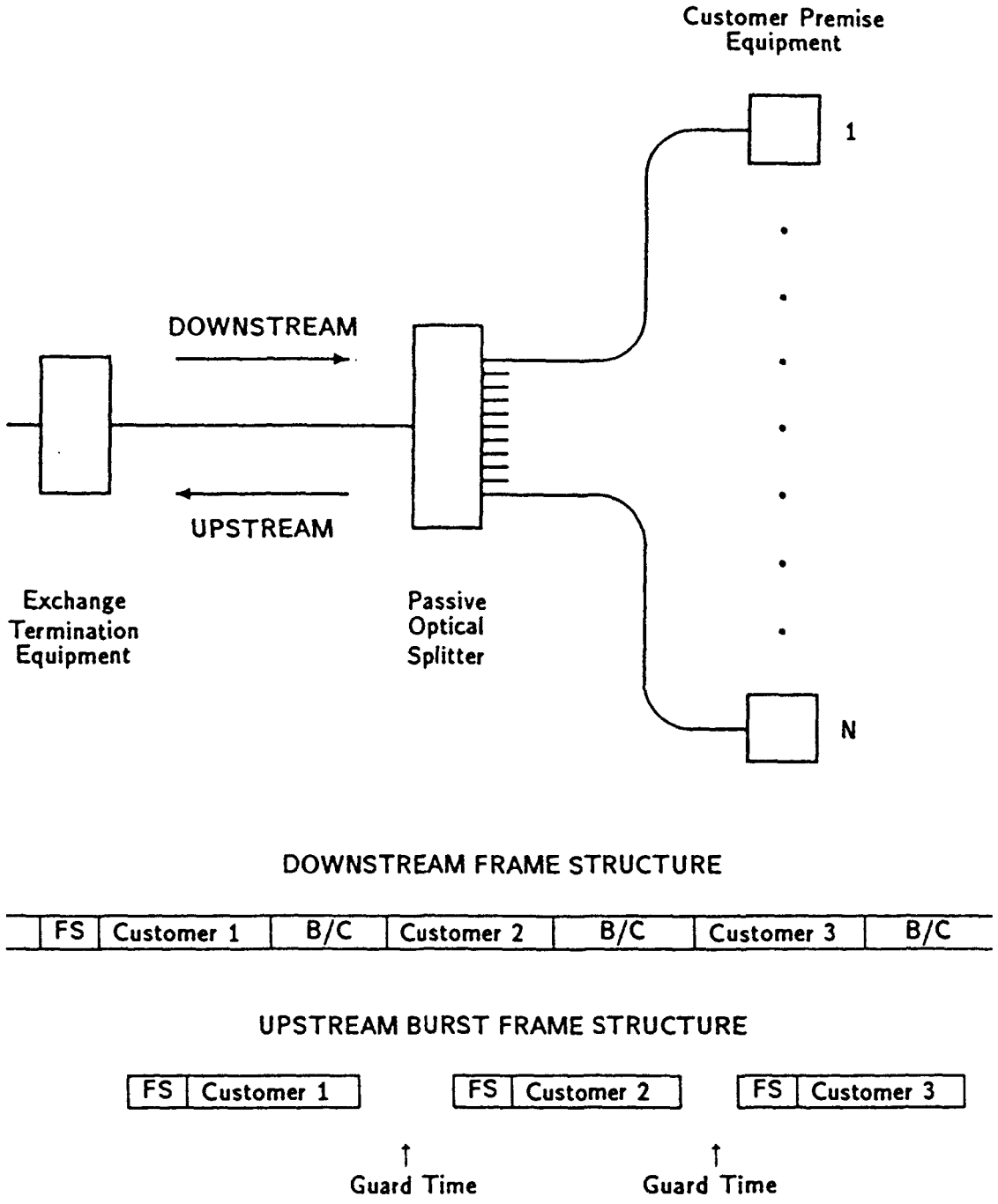



Figure 1. *MACNET* architecture and *TDM* frame structure.

The topology of *MACNET* is summarised in figure 1. In the downstream direction, the optical signal from the exchange equipment is split evenly by the passive coupler and carried to each customer. Thus the time division multiplexed (*TDM*) downstream

traffic of all customers in the group served by the *MACNET* reaches each customer's receiver. Time slot selection is used in the customer's equipment so that he only receives the downstream traffic specifically intended for him. The upstream channel is private; the optical signal incident on the coupler from one customer reflected to another customer suffers at least 50 dB of attenuation, placing it well below the noise floor of the system, thereby rendering it effectively undetectable.

In the initial implementation of *MACNET*, a full-duplex Time Division Multiplex (*TDM*) scheme is used. In the downstream direction, the transmission rate is 8·192 Mbit/s and based on 1 ms frames. The frames are divided into twenty 160 kbit/s and one 2 Mbit/s slots. These channels are assigned to customers connected to the *MACNET*. Broadcast information amounting to another 2 Mbit/s is transmitted as padding between the slots. The start of a customer's slot in the downstream frame is his cue to start transmitting back upstream to the exchange. Customers transmit upstream for the duration of their downstream slot. The padding in the downstream frame is required in order to provide a guard band to make allowances for the variation in distances (and hence propagation delays) from the passive coupler to different customers' locations.

The prototype *MACNET* equipment, which was developed under a Telecom Australia research and development contract by Alcatel-STC [2], is based on this *TDM* scheme. A single shared fibre is used (as opposed to a fibre pair) for both upstream and downstream paths. In the future, the use of sub-carrier multiplexing and wavelength division multiplexing will also be considered as an alternative or complementary way of sharing the fibre. The use of two fibres per customer (one for upstream; one for downstream) will also be considered as a way of increasing the power budget and therefore allowing a greater degree of splitting in the network [8]. Future implementations are likely to concentrate on 2 Mbit/s channels rather than 160 kbit/s.

In the prototype, a simple security mechanism was implemented to guard against the possibility of data mis-delivery due to equipment failure. This paper presents a re-working of that scheme such that a significantly higher level of security is provided.

## 3. Security Issues in MACNET.

While the *MACNET* architecture has many economic and technical advantages as described above, it also has a number of security implications. These stem both from the use of the passive optical coupler, and also, indirectly, from some of the possible methods of operation.

### 3.1. *Privacy.*

The confidentiality or privacy of a communications system refers to the ease with which an opponent can eavesdrop, gaining information to which he is not entitled. This is perhaps the most obvious aspect to communications security. There are a number of ways in which threats to communications privacy can occur in the *MACNET* system.

Firstly, data may be read at either the customer or exchange end, by unauthorised persons with conventional access to the terminal equipment (for example visitors reading faxes which have not been removed from the machine). This threat is obviously not

unique to the *MACNET* system and must be addressed by providing adequate physical security at both the exchange (by the network provider) and at the customer's premises (by the customer).

A second threat to privacy relates to an opponent intercepting data somewhere between the exchange and customer's premises. This would take the form of a fibre tap. Again, this threat is not unique to the *MACNET* system, and applies equally well to dedicated fibre access. Furthermore, while it is not impossible to tap a fibre undetectably [14], it is considered sufficiently difficult to deter all but the most determined opponent. In virtually all normal circumstances (excluding military and large electronic funds transfer applications), the cost of implementing such an attack would greatly exceed the potential benefits to the opponent.

A third threat which is unique to the *MACNET* system is posed by other customers connected to the same *MACNET*. By modification of the *MACNET* terminal equipment, an opponent may be able to read the downstream channel of another customer. While thoughtful design practises (for example locating the time slot selection operation on a single custom integrated circuit, and sealing the customer terminal equipment) may help to alleviate this threat, other considerations are likely to reduce the degree to which this is successful [1]. As a result, this is likely to remain the most severe threat to network privacy, and an encryption scheme is necessary to provide security.

3.1.1. *Upstream Privacy.* Of the three threats identified above, the first two are equally applicable to a dedicated fibre system as they are to *MACNET*. The upstream channel is considered to be at least as secure as a dedicated fibre access network. Reflections from one customer port to another via the coupler will typically suffer around 50 dB attenuation, placing them below the noise floor of the system. As a result, threats to a customer's upstream channel privacy from other customers are no greater than the first two threats described.

Attempts at passive fibre taps on the upstream channel may be detected by monitoring the power levels received from individual customers at the exchange. This may also detect attempts to 'shout down' (or jam) legitimate users.

3.1.2. *Downstream Privacy.* The main security concern from a customer perspective is the use of passive optical couplers in the downstream channel. The optical coupler must remain passive in order to reap the considerable cost savings achieved by avoiding the need for active electronics in the external plant. As a result, whatever is transmitted from the exchange in the downstream direction will be received by every customer terminal connected to the network. Normally, only the appropriately addressed information from the composite signal will be delivered to the customer; however, malicious tampering, or indeed certain error conditions, may result in unauthorised data becoming available. To prevent the disclosure of private information under these circumstances, all downstream data must be encrypted.

In accordance with ISO recommendations [3], and in order to ensure that the security mechanisms are transparent to the user, encryption should be performed at the

[1] For example, to provide service flexibility, it may be desirable that the receiver can be programmed to accept additional channels when further capacity is required by the customer.

lower two layers of the OSI model. In this way all user data is encrypted immediately prior to transmission on the downstream *MACNET* channel.

The choice of a suitable encryption algorithm and key management protocol are influenced by cost, implementation details and the level of security provided, which must be sufficient for the majority, but not necessarily all, users. In cases where the level of security provided is not considered adequate, such as for Electronic Funds Transfer (*EFT*), additional security mechanisms may be implemented by the user on an end-to-end basis at the Application level (layer 7).

Stream ciphers are considered to be the most suitable encryption scheme due to their relative simplicity and low cost, even when operating at quite high bit rates. For most applications, key transfer can be performed in the clear (unencrypted) on the upstream channel, which is considered secure. The use of a more sophisticated key transfer mechanism adds considerably to the implementation costs and is not warranted in most instances. More complete details of a possible solution are described in a subsequent section.

### 3.2. *Impersonation.*

Another way in which a communication system's security may be compromised is by means of impersonation; that is, an opponent may pretend to be someone he is not. There are two motives for this. Firstly, to deceive the other party involved in the communication, and secondly, to avoid being charged for the communication service obtained. Since *MACNET* customers may communicate with any other customer in the network (who will not necessarily be connected to a *MACNET*), the first type of impersonation cannot be addressed by *MACNET* in isolation. Such a peer-entity authentication service must be implemented end-to-end if it is required.

The second type of threat, that of an opponent who is connected to the *MACNET* impersonating another customer in order to avoid being charged, must be considered. A possible mechanism for this is for the opponent to transmit in a time-slot other than that to which he is entitled. The time-slot in which he transmits may either be an idle one (in which case nobody will be charged) or that of another customer, in which case that customer will be charged. In addition, the legitimate customer's transmissions will be corrupted (as may that of the impersonator).

This threat may be addressed in a number of ways. Firstly, channels which are supposed to be idle should be ignored at the exchange, thereby eliminating the possibility of an opponent using them. Since all customers will transmit in their time-slot, regardless of whether data is being sent, an opponent will have to transmit at considerably higher power in order to drown out the legitimate customer. This increase in received power can be detected at the exchange, and alarms raised.

## 4. Initial Security Services.

As discussed earlier, the most pressing concern for a basic *MACNET* service is the issue of downstream privacy. In order to gain customer acceptance, eavesdropping must be seen to be at least as difficult as in a dedicated fibre access system. It has been shown that the upstream channel does indeed provide this level of security, and may thus be used to transmit keying information in the clear. This considerably simplifies the process of key management, and allows frequent key changes.

4.1. *Stream Cipher Design.*

A stream cipher generally encodes the plaintext on a bit by bit basis to form the ciphertext. This is done by an exclusive OR (XOR) operation between the plaintext and a pseudo random bit sequence known as the key stream. The key stream is generated using a deterministic algorithm and a truly random key. Decryption is performed by the same XOR operation between the ciphertext and the same key stream. Thus the algorithm for key stream generation and the accompanying key must be available at both the transmitting and receiving ends.

In designing a suitable cipher for the *MACNET* system, a number of properties must be considered. In addition to the universal considerations of simplicity and cost, the level of security provided must be adequate. Due to the standard forms of many data transmissions, cipher systems must be able to withstand known plaintext attacks [13]. Since the *MACNET* encryption would be performed on all data at the local exchange, an opponent may send his own data to a particular customer, and observe the encrypted data being transmitted. In this way he can perform what is known as a chosen plaintext attack; however, for a stream cipher system, the distinction between known and chosen plaintext attacks is irrelevant, since they both yield the same information, that is, a portion of the key stream. Ideally, even under a known plaintext attack, the computational effort required to break the cipher should be the same as that of guessing the key.

In order to be cryptographically secure, the future key stream must be unpredictable, regardless of the number of past key stream bits observed. This implies that the key stream, which in all practical cases will be periodic, should have a very long period. If the key stream satisfies some linear or non-linear recursion relation (which is normally the case), then knowledge of the relation and its initial state will specify the complete key stream. While it is infeasible to determine the shortest non-linear recursion which the key stream satisfies, efficient methods exist in the form of the Berlekamp-Massey algorithm [4] to determine a linear recursion relation to generate the sequence. Consequently, to satisfy the requirement of unpredictability, the linear complexity of the key stream, which is defined as the minimum length linear recursion relation able to generate the sequence, must be large. In order to ensure that all sub-sequences of the key stream are unpredictable, the linear complexity profile must exhibit certain 'typical' characteristics, namely that it take the form of a random staircase function with given mean and variance for step length and height [10].

Perhaps the most simple of all key stream generators is based on the non-linear filtering of the stages of one or more Linear Feedback Shift Registers (*LFSR*) [2]. In such generators it is essential to provide correlation immunity between the output and the driving sequences to prevent a divide and conquer style attack [11,12]. In general, there exists a trade-off between correlation immunity and the degree of non-linearity (and hence linear complexity) provided. This compromise can be overcome by the addition of a memory element [10].

(An *LFSR* is a hardware realisation of a linear recursion relation, with the shift register providing the necessary delays required to access 'past' values of the sequence.

[2] Each *LFSR* implements a linear recursion relation to generate a pseudo random sequence. The order of the linear recursion relation can be at most equal to the length of the *LFSR*.

The length of the shift register will thus limit the maximum order of the recursion relation. A linear feedback function, often expressed as a connection polynomial, is used to generate the next value in the sequence from past values stored in the shift register. This feedback function is uniquely defined by the characteristic polynomial of the recursion relation. A complete discussion of the theory of *LFSRs* is beyond the scope of this paper—see for example [1,10,13].)

One of the simplest key stream generators to implement which satisfies the above conditions is based on the integer addition of a number of *LFSR* sequences, as shown in figure 2. Although perhaps counter-intuitive, integer addition is in fact highly non-linear when viewed in the Galois Field $GF(2)$. The carry mechanism introduces the necessary memory element required to decouple the correlation immunity from linear complexity.

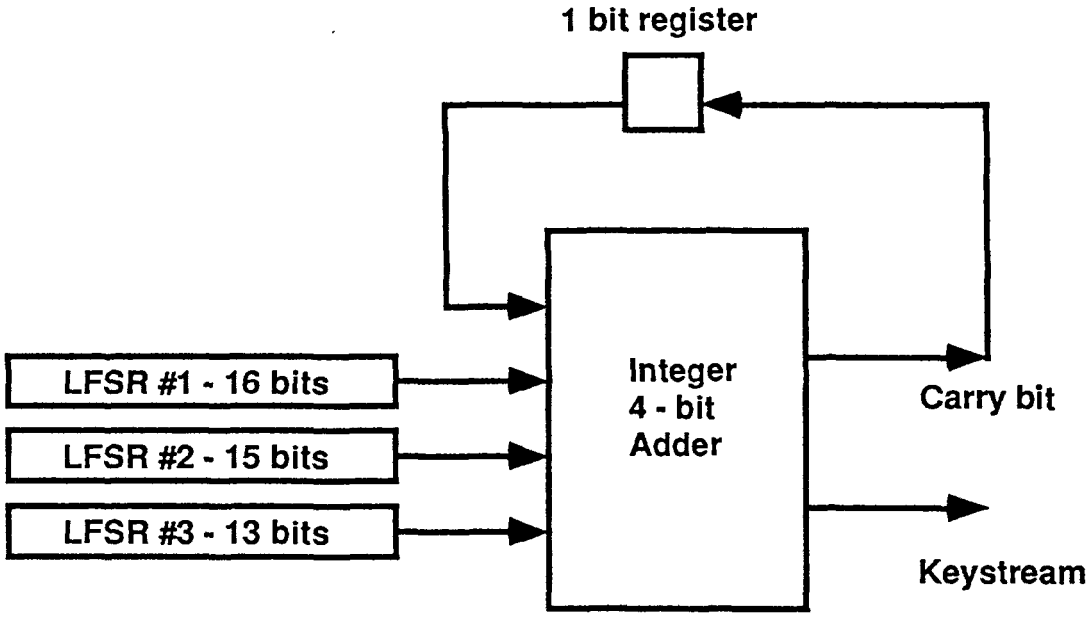



Figure 2. Proposed form of key stream generator.

Rueppel [10] shows that such a generator provides maximum correlation immunity, near maximum period for a given number of key bits (if the *LFSR* characteristic polynomials are chosen to be of relatively prime orders), and, in virtually all cases, very high linear complexity.

It may be noted that a known plaintext attack may be reduced to guessing the characteristic polynomials and initial states of all but the largest *LFSR*. The output sequence of the final *LFSR* can then be derived, and the Berlekamp-Massey algorithm used to determine its connectivity and initial state. At this point the rest of the key stream sequence can be generated.

This suggests that the driving *LFSRs* should all be of similar size, and that there should be more than two of them. To retain simplicity, three *LFSRs* are proposed, each being 16 bits in length, with connection polynomials of order 13, 15 and 16 bits respectively. This results in a potential sequence length of around $10^{13}$ bits, with a linear complexity of the same order. The characteristic polynomials should be primitive to ensure maximum possible sequence length for the system [1], and may

be stored in non-volatile memory (for example Programmable Read Only Memory–*PROM*) in the customer premise equipment, and downloaded to the exchange end during initialisation. As such they may be considered as a fixed part of the key, with the initial states of the *LFSRs* being changed on a regular basis (for example, every frame). This simplifies the (regular) key generation process, which may take the form of a truly random number generator. While the full integer addition of three *LFSR* sequences requires two carry bits, it is possible to implement only one while maintaining a high linear complexity. This may be desirable in implementations using combinatorial logic in order to reduce the number of gates required.

4.2. *Key Management.*

As stated earlier, the key for the proposed cipher takes two parts: one fixed and the other part being frequently changed. The fixed part specifies the connectivity (or characteristic polynomial) of the driving *LFSRs*. Since the characteristic polynomials should be primitive, these must be carefully chosen, requiring some computational effort. It can be shown [1] that the number of primitive polynomials of order $m$ is given by

$$N_m = \frac{\phi(2^m - 1)}{m}$$

where Euler's totient function $\phi(x)$ is defined as the number of positive integers less than and relatively prime to $x$, and is given by

$$\phi(x) = x \prod_{\substack{p|x \\ p \text{ prime}}} \left(1 - \frac{1}{p}\right).$$

Given the suggested polynomial orders of 13, 15 and 16, the number of possible choices of primitive characteristic polynomials is given by

$$N = \frac{\phi(2^{13} - 1)}{13} \cdot \frac{\phi(2^{15} - 1)}{15} \cdot \frac{\phi(2^{16} - 1)}{16} \approx 10^9$$

Due to the computational effort required to generate suitable primitive polynomials, it may be convenient to store the *LFSR* feedback connections in *PROMs* in the Customer Premise Equipment (*CPE*), and change them relatively infrequently (perhaps only when a security breach is suspected). To avoid having to match specific equipment (*PROMs*) between the exchange and customer ends, this portion of the key should be transmitted from the *CPE* to the exchange during initialisation, and stored there until the next key change.

The second part of the key changes the state of the driving *LFSRs*. Since all states other than the all-zero state are acceptable, these can be chosen randomly. A truly random number generator [3] can be used to generate this part of the key, which should be changed more frequently, for example each 1 ms frame. Key transfer may be performed in the clear in overhead bits in the upstream channel header. The number

[3] This may be obtained by measuring some noisy signal, or by sampling an unsynchronised high frequency oscillator.

of bits required depends on the length of the key and the period for which the key remains active. While for security considerations the key period should be as short as possible, efficiency and implementational factors may require longer key periods. In order to minimise the number of keys stored at any one time, and, in addition, to avoid the need for frame numbering, it is desirable to ensure that only one key period worth of data is in transit at any one time. If the key period consists of more than a single frame, a beginning of key period marker is required.

Under all foreseen circumstances, only one 1 ms frame will be in transit at any one time, and hence key changes may be performed each frame. Using the 48 bit key proposed (three 16 bit *LFSRs*), 48 overhead bits are required in the upstream channel header for key transfer, which, for a 2 Mbit/s channel, represents a 2% overhead.

As it stands, single bit errors in the received upstream channel header would result in different keys being used for the encryption and decryption processes. Consequently, the downstream data could not be correctly decrypted for one full key period. In order to reduce the likelihood of this occurrence, error detection and/or correction can be performed on the upstream channel header. An eight bit Cyclic Redundancy Check (*CRC*) may be used to detect all errors of less than four bits in the header if it is limited to 120 bits in length or less [8]. When an error is detected, the next key change is cancelled, and instead the last key continues to be used for a second key period (that is the key stream generator is allowed to continue to run without a new key being loaded). This change must be signalled to the customer end in the downstream channel header. A three bit majority voting scheme is suggested in order to provide robustness against random errors in this signalling channel. It should be noted that error bursts may go undetected by this scheme, but are also likely to corrupt the data and framing bits, thus invalidating large portions of data anyway. The addition of the eight bit *CRC* to the upstream channel header will increase the key transfer overheads to just under 3%. The *CRC* may also be used to perform error detection/correction over the rest of the header.

### 4.3. *Summary of Design.*

The proposed system for providing downstream privacy consists of three driving *LFSRs* configured with primitive characteristic polynomials of order 13, 15 and 16. The coefficients of these polynomials (or alternatively the truth tables of the feedback function) are held in non-volatile memory. The outputs of these *LFSRs* are summed together with the previous carry bit to form a two bit integer (the third and most significant bit of the sum need not be generated). The least significant bit is XOR-ed with the plaintext to form the ciphertext, while the second bit is fed back to provide the next carry bit. The states of the driving *LFSRs* are changed each 1 ms frame, with new values being generated randomly by the customer end and transmitted to the exchange end in a 48 bit field in the channel header. An eight bit *CRC* protects the entire channel header against single and two bit errors [4]. In the event of an error being detected by the exchange end in the channel header, the impending key change is NOT performed, and instead the cipher is allowed to continue the current sequence

[4] Single bit error correction is proposed at the expense of detecting all three bit errors, since the detection of three bit errors will not significantly reduce the frame loss rate (see Table 1).

for another frame. This fact is signalled to the customer end in a three bit field in the downstream channel header, which causes the customer end cipher to ignore the impending key change also. In order to prevent large portions of the key stream being transmitted in the clear, idle channels in the downstream direction should be scrambled.

4.4. *Analysis of Design.*

The suggested system may be implemented in LSI using approximately 15 packages. Alternatively, it could be implemented in custom silicon using a couple of hundred gates. Neither of these estimates includes the random number generator used for key generation, because this may be able to be derived from existing circuits with very little additional hardware.

It may be shown that this cipher provides a very high linear complexity, as well as a high degree of correlation immunity [10], thus eliminating the possibility of correlation attacks or *LFSR* synthesis in the form of the Berlekamp-Massey algorithm. In the case of a known plaintext attack, the opponent may attempt to guess all but the largest *LFSR*, and then derive this final one. Even in the case where all the characteristic polynomials are known, there remain 28 state bits (the initial states of the two shortest *LFSR*s) which must be guessed. For each guess, the two driving sequences must be generated and then compared with the known part of the key stream in order to derive the output sequence of the third *LFSR*. The Berlekamp-Massey algorithm must then be used to derive the remaining *LFSR* state, thereby testing the validity of the guess. If all this could be performed in 200 instructions on a 1 Mips processor, and assuming that on average the opponent must search through half the possible combinations of states, the average time taken to decode one millisecond of data is over seven hours. Thus to decode ten seconds worth of data, an opponent would require over eight years of dedicated processing. This is considered to be sufficient protection for the majority of cases. In cases where the characteristic polynomials are not known, the attacker must also guess the correct combination of primitive polynomials for the two smaller *LFSR*s out of a total of over one million, thereby increasing the average computational effort required to obtain data by a similar factor, making it virtually impossible.

Another important aspect of the performance of the encryption system is its robustness to transmission errors. While bit errors in the received ciphertext will simply result in errors in the deciphered plaintext (with no error multiplication), difficulties arise when undetected errors occur in the received key at the exchange end. In such cases, encryption will be performed with the wrong key, resulting in the loss of a full frame. The error correction/detection capabilities of the upstream channel header *CRC* reduce the probability of such errored keys. An eight bit *CRC* operating on a 120 bit header will correct all single bit errors, detect all two bit errors, and detect errors of more than two bits with probability, $p_d$ given by

$$p_d = \frac{255}{256} = 0{\cdot}9961.$$

A frame will be lost (due to incorrect keying) if the *CRC* fails to detect a (three or more bit) error in the upstream header in which one or more of the errored bits occurred in the 48 bit key. The probability of frame loss due to this mechanism is given in the second column of Table 1 for different transmission bit error rates.

Another mechanism for incorrect keying is the corruption of the errored key flag in the downstream channel header. This requires two or more of the three bits to be corrupted. The probability of frame loss due to this mechanism is shown in the third column of Table 1.

Table 1 also shows the bit error rate due to frame loss by the above two mechanisms. This calculation is based on 2 Mbit/s channel rates and a 1 ms frame length. In this case, frame loss due to either of the two mechanisms will result in the loss of 2048 data bits. It can be seen that for all realistic bit error rates for a fibre system (that is, $< 10^{-6}$) the increased bit error rate (or error multiplication) resulting from the use of the cipher is negligible.

| TX BER | U/S Header | D/S Header | Total BER |
|---|---|---|---|
| $10^{-3}$ | $9{\cdot}3 \times 10^{-7}$ | $3 \times 10^{-6}$ | $8{\cdot}0 \times 10^{-3}$ |
| $10^{-4}$ | $1{\cdot}0 \times 10^{-9}$ | $3 \times 10^{-8}$ | $6{\cdot}3 \times 10^{-5}$ |
| $10^{-5}$ | $1{\cdot}0 \times 10^{-12}$ | $3 \times 10^{-10}$ | $6{\cdot}2 \times 10^{-7}$ |
| $10^{-6}$ | $1{\cdot}0 \times 10^{-15}$ | $3 \times 10^{-12}$ | $6{\cdot}1 \times 10^{-9}$ |
| $10^{-7}$ | $1{\cdot}0 \times 10^{-18}$ | $3 \times 10^{-14}$ | $6{\cdot}1 \times 10^{-11}$ |
| $10^{-8}$ | $1{\cdot}0 \times 10^{-21}$ | $3 \times 10^{-16}$ | $6{\cdot}1 \times 10^{-13}$ |
| $10^{-9}$ | $1{\cdot}0 \times 10^{-24}$ | $3 \times 10^{-18}$ | $6{\cdot}1 \times 10^{-15}$ |

Table 1. Probabilities of frame corruption due to errors in upstream header and downstream error flag with resulting bit error rate.

## 5. Conclusions.

The security of the *MACNET* shared optical fibre access network has been considered. The most significant threat to security has been identified as privacy on the downstream channels. A stream cipher based solution has been proposed which is intended to match the level of security provided by a dedicated fibre access network. Analysis of the design confirms that the level of security provided is sufficient for most applications. The computational effort required to break the cipher has been shown to be very large.

The encryption scheme can be implemented relatively simply and can operate at quite high speeds—far in excess of the 2 Mbit/s data channels favoured for use in a commercial *MACNET* system. It is quite feasible for the cipher to be used for broadband applications if these are to be provided via *MACNET*. The proposed scheme requires manageable overheads for key management, and does not significantly increase the error rate over the network.

The use of such an encryption scheme in *MACNET* could well represent the first use of cryptography as a standard service in the Australian telecommunications network.

## 6. Acknowledgements.

The author would like to thank his colleagues Tim Batten and Ed Zuk for their contribution to this work. The permission of the Executive General Manager, Telecom Australia Research Laboratories, to publish this paper is also hereby acknowledged.

## References


1. S. W. Golomb, *Shift Register Sequences.* (Aegean Park Press, revised edition, 1982.)
2. K. House and B. Jones, 'Implementation of experimental optical local distribution system using a 16-way passive optical splitter', *Proceedings of 12th Australian Conference on Optical Fibre Technology*, IREE (Aust), December 1987, 255–258.
3. International Organisation for Standardisation, *Information Processing Systems—Open Systems Interconnection Reference Model—Part 2 : Security Architecture*, 7498-2, 1988.
4. J. L. Massey, 'Shift-register synthesis and BCH decoding', *IEEE Transactions on Information Theory*, **IT-15**, No. 1 (1969), 122–127.
5. I. McGregor and G. Semple, '*MACNET*- a shared customer access network', *Proceedings of 12th Australian Conference on Optical Fibre Technology*, IREE (Aust), 1987, 251–254.
6. I. McGregor, G. Semple and G. Nicholson, 'Implementation of a *TDM* passive optical network for subscriber loop applications', *IEEE Journal of Lightwave Technology*, **6**, No. 11 (1989).
7. I. McGregor, G. Semple and G. Nicholson, '*TDM MACNET*—an implementation of a shared fibre customer access network', *Telecommunications Journal of Australia*, **39**, No. 2 (1989).
8. G. Nicholson, G. Semple, I. McGregor, B. Clarke and C. Desem, 'Transmission techniques for broadband access on shared fibre networks', *Proceedings of 13th Australian Conference on Optical Fibre Technology*, IREE (Aust), 1988, 195–198.
9. W. W. Peterson and D. T. Brown, 'Cyclic Codes for Error Detection', *Proceedings of the IRE*, January 1961, 228–235.
10. R. A. Rueppel, 'Analysis and Design of Stream Ciphers', *Communications and Control Engineering.* (Springer-Verlag, 1986.)
11. T. Siegenthaler, 'Correlation-immunity of nonlinear combining functions for cryptographic applications', *IEEE Transactions on Information Theory*, **IT-30**, No. 5, September 1984, 776–780.
12. T. Siegenthaler, 'Decrypting a class of stream cipher using ciphertext only', *IEEE Transactions on Computers*, **C-34**, No. 1, January 1985, 81–85.
13. H. C. van Tilborg, *An introduction to Cryptology.* Kluwer international series in engineering and computer science, Kluwer Academic Publishers, 1988.
14. I. White, 'The security of optical fibre transmission systems', *Proceedings of the 13th Australian Conference on Optical Fibre Technology*, IREE (Aust), 1988, 295–297.


*Telecom Australia Research Laboratories, 770 Blackburn Road, Clayton, Victoria 3168, AUSTRALIA*

# 12

# AUTHENTICATION

**Bill Newman**

The two main concerns in the authentication of information are verification that the communication originated with the purported transmitter and that it has not subsequently been altered. A model for the authentication channel is presented with two types of attack, by substitution and impersonation. The ideas are illustrated by a practical authentication system which can be used when both the transmitter and receiver will cheat if they can get away with it.

## 1. Introduction.

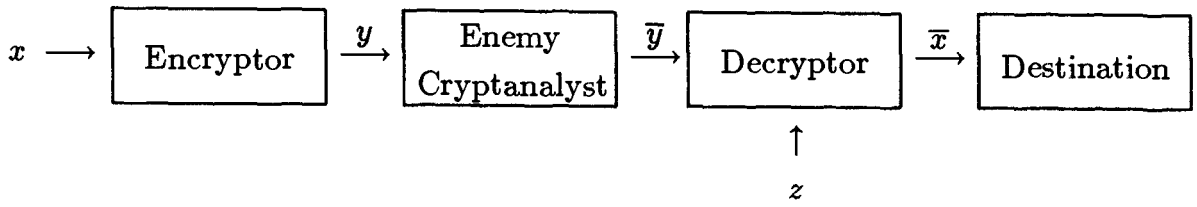
In both commercial and private transactions, authentication of messages is of vital concern. For example, a person accepting a cheque usually insists on some identification of the issuer or authentication of the originator, and the person issuing the cheque not only fills in the face amount in numerals but writes out the amount in words to make it more difficult for anyone to alter the face amount on a cheque bearing his signature. This example illustrates the two main concerns in the authentication of information, namely verification that the communication originated with the purported transmitter and that it has not been subsequently altered. The contemporary concern is with situations in which the exchange involves only information, as for example in Electronic Funds Transfer (*EFT*).

The Australian Standard for message authentication [1] for *EFT* is as follows. The information is broken into blocks of 64 bits each. The first block is encrypted using a secret *DEA* key (known to both the transmitter and the receiver). The resulting 64-bit cipher is then added bitwise (mod 2) with the second block of text and the result encrypted and so on until all blocks of the text have been processed (cipher-block chaining). The final 64-bit cipher is clearly a function of the secret key, and of every bit of the text. This cipher, called a message authenticating block (*MAB*), is used to form the message authentication code (*MAC*) by selecting the 32 most significant bits, which are appended to the information being authenticated and the resultant extended message normally sent in the clear. Anyone in possession of the secret key can easily verify the *MAC*, but an outsider cannot generate an acceptable authenticator to accompany a fraudulent message, nor can he separate an *MAC* from a legitimate message to use with an altered or forged message since the probability of it being acceptable is $2^{-32}$. Subsequent messages may include previously calculated *MAB*'s, for example the least significant 32 bits of the *MAB*.

Implicit in any authentication protocol is that the receiver will accept as authentic only a fraction of the total number of possible messages, and that the transmitter will only use some subset of this fraction to communicate with the authorized receiver.

## 2. Theory of authenticity.

A model for the authentication channel is as shown. (See [2].)




The enemy can choose either of two different attacks. He can choose to form his fraudulent cryptogram $\overline{y}$ without waiting to see the authentic cryptogram $y$ (the impersonation attack), or he can wait to see $y$ before he forms $\overline{y}$ (the substitution attack). Let $P_i$ and $P_s$ respectively denote the enemy's best-possible probability of success in these two attacks. Assuming that the enemy cryptanalyst will choose the attack that is more likely to succeed, define

$$P_d = \max(P_i, P_s)$$

to be the probability of deception.

Let $|Y|$ denote the number of cryptograms $y$ such that $\text{Prob}(Y = y) \neq 0$ and let $|X|$ and $|Z|$ be defined similarly as the number of plaintexts and keys, respectively, with non-zero probability. For every $z$, there must be at least $|X|$ different cryptograms $y$ such that $\text{Prob}(Y = y \mid Z = z) \neq 0$. Hence, if the enemy cryptanalyst in an impersonation attack selects $Y$ completely at random from the $|Y|$ cryptograms, his probability of success will be at least $|X|/|Y|$. That is

$$P_i \geq |X|/|Y|.$$

This shows that good protection against an impersonation attack demands that $|Y|$ be much greater than $|X|$, and shows also that complete protection is impossible.

Define the authentication function $\phi(y, z)$ to be 1 if $y$ is a valid cryptogram for the secret key $z$ and to be 0 otherwise. The probability that a particular $y$ is a valid cryptogram can then be written

$$\text{Prob}(y \text{ valid}) = \sum_z \phi(y, z)\text{Prob}(z).$$

The best possible impersonation attack is for the enemy cryptanalyst to choose $\overline{Y} = y$ for that $y$ that maximizes $\text{Prob}(y \text{ valid})$. Thus

$$P_i = \max_y \text{Prob}(y \text{ valid}).$$

Noting that $\text{Prob}(Y = y, Z = z) = \text{Prob}(z)\text{Prob}(y|z) \neq 0$ implies $\phi(y, z) = 1$, we can write

$$\text{Prob}(y) = \sum_z \text{Prob}(z)\text{Prob}(y|z)\phi(y, z).$$

**Simmon's Theorem.** $\log P_d \geq -I(Y;Z)$.

*Proof.* Introduce a new probability distribution on $Z$,

$$Q_y(z) = \frac{\text{Prob}(z)\phi(y,z)}{\text{Prob}(y \text{ valid})}$$

Then

$$\sum_z Q_y(z)\text{Prob}(y|z)\log\text{Prob}(y|z) \geq \left(\sum_z Q_y(z)\text{Prob}(y|z)\right)\log\left(\sum_z Q_y(z)\text{Prob}(y|z)\right)$$

with equality if and only if $\text{Prob}(y|z)$ has the same value for all $z$ for which $Q_y(z) \neq 0$. Now

$$\text{Prob}(y) = \text{Prob}(y \text{ valid})\sum_z Q_y(z)\text{Prob}(y|z)$$

so that

$$\text{Prob}(y)\log\text{Prob}(y) = \text{Prob}(y)\log\text{Prob}(y \text{ valid}) + \text{Prob}(y)\log\left(\sum_z Q_y(z)\text{Prob}(y|z)\right). \tag{1}$$

The second term on the right becomes

$$\begin{aligned}\text{Prob}(y \text{ valid})\left(\sum_z Q_y(z)\text{Prob}(y|z)\right)\log\left(\sum_z Q_y(z)\text{Prob}(y|z)\right) &\\ \leq \text{Prob}(y \text{ valid})\sum_z Q_y(z)\text{Prob}(y|z)\log\text{Prob}(y|z) &\\ = \sum_z \text{Prob}(z)\text{Prob}(y|z)\phi(y,z)\log\text{Prob}(y|z) &\\ = \sum_z \text{Prob}(z)\text{Prob}(y|z)\log\text{Prob}(y|z) &\end{aligned}$$

using the fact that $\text{Prob}(y|z)\text{Prob}(z) \neq 0$ implies $\phi(y,z) = 1$. Substituting this expression in (1) and summing over $y$ gives

$$-H(Y) \leq \sum_y \text{Prob}(y)\log\text{Prob}(y \text{ valid}) - H(Y|Z)$$

or

$$\sum_y \text{Prob}(y)\log\text{Prob}(y \text{ valid}) \geq -I(Y;Z).$$

But the average of $\log\text{Prob}(y \text{ valid})$ cannot exceed $\max\log\text{Prob}(y \text{ valid})$ so

$$\log P_i \geq -I(Y;Z).$$

Since $P_d \geq P_i$, $\log P_d \geq -I(Y;Z)$. Moreover, the necessary and sufficient conditions for equality to hold are

(i) Prob($y$ valid) is independent of $y$,

(ii) for each cryptogram $y$, Prob($y|z$) has the same value for all $z$ for which $\phi(y, z) = 1$,

(iii) $P_d = P_i$.

An authentication system is defined as perfect (see [3]) if equality holds, that is

$$\log P_d = -I(Y; Z).$$

Even with perfect authenticity, however, it must be remembered that the probability of deception $P_d$ will be small only when $I(Y; Z)$ is large, that is, only when the cryptogram provides the enemy cryptanalyst with much information about the key! The information that $Y$ gives about $Z$ is a measure of how much of the secret key is used to provide authenticity.

For a substitution attack, the effective equivocation to the enemy cryptanalyst when secrecy applies (that is $Y$ gives no information about $X$) is no greater than $H(Z|Y)$. Thus $\log P_s \geq -H(Z|Y)$. Now

$$\begin{aligned}\log P_d &\geq \max\{\log P_i, \log P_s\} \\ &\geq \max\{-I(Y; Z), -H(Z|Y)\} = \max\{-H(Z) + H(Z|Y), -H(Z|Y)\} \\ &\geq \tfrac{1}{2}(-H(Z) + H(Z|Y) - H(Z|Y)) = -\tfrac{1}{2}H(Z)\end{aligned}$$

Thus

$$P_d \geq \frac{1}{\sqrt{|Z|}}.$$

The inescapable conclusion that must be drawn from the theoretical results above is that a large number of encoding rules must be available in any secure authentication code—of the order of $P_d^{-2}$ at least—in order to realize a security of $P_d$, and that these encoding rules must also have a well-defined structural interdependence to insure that the conditional entropy conditions be met to make this level of security achievable at all.

It is not known in general under what conditions systems offering perfect authenticity exist. We will give some simple examples that illustrate the theory.

In the following examples, the plaintext is always a single binary digit $X$, the cryptogram $Y = \{Y_1, Y_2\}$ is a binary sequence of length 2, the key $Z = \{Z_1, \ldots, Z_K\}$ is a completely random binary sequence so that Prob($Z = z$) $= 2^{-K}$ and $H(Z) = K$ bits.

*Example 1.*

| $Z$ | $X$ | |
|---|---|---|
| | 0 | 1 |
| 0 | 00 | 10 |
| 1 | 01 | 11 |

Thus $Y = 10$ when $X = 1$ and $Z = 0$. In this example the key is appended as an *MAC* to the plaintext. Notice the system provides no secrecy at all. Moreover, $H(Z|Y) = 0$ so that $I(Y; Z) = 1$ bit and the theoretical bound is $P_i \geq \frac{1}{2}$. But upon observing $Y = y$, the enemy cryptanalyst always knows the other valid cryptogram so that he

can always succeed in a substitution attack. Hence $P_s = 1 = P_d > 2^{-I(Y;Z)} = \frac{1}{2}$. That is, the authenticity is not perfect.

*Example 2.*

| $Z$ | $r$ | $X$ | |
|---|---|---|---|
| | | 0 | 1 |
| 0 | 0 | 00 | 10 |
| 0 | 1 | 01 | 11 |
| 1 | 0 | 00 | 11 |
| 1 | 1 | 01 | 10 |

Here, $r$ is a randomized encipherment. Note that $Y_1 = X$ so again there is no secrecy. Given $Y = y$, the 2 possible values of $Z$ are equally likely, so that $H(Z|Y) = 1$ and thus $I(Y;Z) = 0$. Then $P_i = 1 = P_d = 2^{-I(Y;Z)}$ and thus trivially provides perfect authenticity. But, upon observing, say, $Y = 00$, the enemy cryptanalyst is faced with 2 equally likely alternatives 10 and 11 for the other valid cryptogram, only one of which will be accepted by the receiver, who knows $Z$, as authentic. This $P_s = \frac{1}{2}$. This shows that $-I(Y;Z)$ is not in general a lower bound on $\log P_s$.

*Example 3.* Consider the same system as in Example 2 except that $Z$ and $r$ are now the 2 digits $z_1, z_2$ respectively of the secret key and hence both are known to the legitimate receiver. There is still no secrecy. Given $Y = y$, there are still 2 equally likely possibilities for $Z$ so that $H(Z|Y) = 1$ and hence $I(Y;Z) = 1$ bit. But Prob($y$ valid) $= \frac{1}{2}$ for all four cryptograms $y$ and thus $P_i = \frac{1}{2}$. Also $P_s = \frac{1}{2}$. Thus $P_d = \frac{1}{2} = 2^{-I(Y;Z)}$ and hence this system offers perfect authenticity, no matter what the statistics of the plaintext $X$ may be.

*Example 4.*

| $Z$ | $X$ | |
|---|---|---|
| | 0 | 1 |
| 00 | 00 | 11 |
| 01 | 01 | 10 |
| 10 | 10 | 01 |
| 11 | 11 | 00 |

Because $P(Y = y|X = x) = \frac{1}{4}$ for all $x$ and $y$, the system provides perfect secrecy. Here, $I(Y;Z) = 1$ and $P_i = \frac{1}{2}$. But upon observing $Y = y$, the enemy can always succeed in a substitution attack by choosing $Y$ to be the complement of $y$. Thus $P_s = 1 = P_d$ and hence the system provides no protection against deception by substitution.

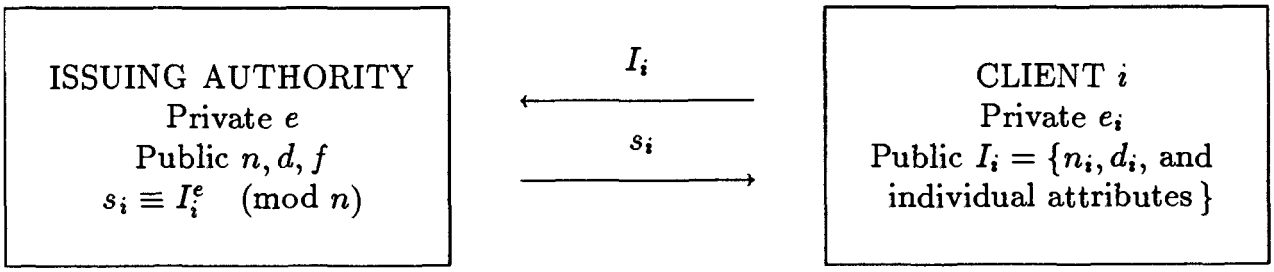
## 3. Practical authentication.

In the *MAC* authentication system used in electronic funds transfer, a single-key cryptoalgorithm is used for the encryption/decryption functions. The important point is that in a single key system, the transmitter and receiver must be mutually trusting. A more challenging case arises when the transmitter and receiver will cheat if they can get away with it.

Consider the needs of the various participants in a credit card transaction at the point of sale. The merchant (or automated teller machine) will give up merchandise, money, or services in exchange for a record of credit due from the customer. The

merchant must be able to satisfy himself as to the customer's identity, the validity of his claimed account and, if necessary, its level of credit. The customer, on the other hand, needs to be able to satisfy himself that the record of the transaction is accurate, that it can only be presented for collection on the correct date and the amount cannot be altered, and that the merchant cannot later, as a result of any number of transactions with the customer, impersonate him by fraudulently making purchases or withdrawals on the customer's account. Since the resolution of a dispute over the validity of a claimed transaction will necessarily involve third parties such as a bank or court, a satisfactory solution should address all of the concerns of all of the participants and produce a record which can be logically evaluated to assign liability to the most probable guilty party.

One system to provide such security uses the Rivest-Shamir-Adelman two-key cryptographic system (*RSA*) with modulus $n_i$, decryption exponent $d_i$, and encryption exponent $e_i$. This string of information $I_i$, excluding $e_i$, but possibly including a description of individual attributes such as height, colour of hair, eyes, and so on, must also include redundant information such as the message format. A central issuing authority calculates $s_i = m_i^e \pmod{n}$ where $m_i = f(I_i)$ for a polyrandom function $f$ (that is, $f$ cannot be distinguished from a truly random function by any polynomially bounded computation). The issuer gives the credential $(I_i, s_i)$ to customer $i$. No part of this credential need be kept secret.




When customer $i$ wishes to prove his identity to merchant $j$, he presents his credential $(I_i, s_i)$ concatenated with a string of symbols $u_i$ that describes the transaction. The merchant replies with his identification credentials $(I_j, s_j)$ concatenated with a string of symbols that describe the transaction from his standpoint. Both $i$ and $j$ form the concatenation $u$ of $u_i$ and $u_j$ and calculate $z = f(u)$. The merchant $j$ accepts the credential $(I_i, s_i)$ as valid if and only if

$$f(I_i) = s_i^d \pmod{n}.$$

If required, the customer $i$ can carry out a similar calculation to verify that the credential $(I_j, s_j)$ was indeed issued by the issuing authority. At this point in the protocol, $j$ is not yet confident that the customer $i$ knows $e_i$. If the customer is who he claims to be, he can calculate $t_i = z^{e_i} \pmod{n_i}$ which he communicates to $j$, who in turn calculates $t_j = z^{e_j} \pmod{n_j}$ and sends $t_j$ to the customer. The merchant accepts $i$ as valid if and only if $t_i^{d_i} = z \pmod{n_i}$. If $i$ is not in possession of $e_i$, he would have to find a number $x$ such that $x^{d_i} = z \pmod{n_i}$, but as $d_i, n_i$ are values signed by the issuer in $I_i$, and $z$ is a pseudorandom number jointly determined by both $i$ and $j$, then solving this equation is equivalent to breaking the RSA cryptoalgorithm from ciphertext alone.

| CUSTOMER $i$<br>Private $e_i$<br>Credential $(I_i, s_i)$ | | MERCHANT $j$<br>Private $e_j$<br>Credential $(I_j, s_j)$ |
|---|---|---|
| STEP 1<br>Present credential and customer transaction | $(I_i, s_i), u_i \longrightarrow$ | Accept credential if<br>$s_i^d \equiv I_i \pmod{n_i}$ |
| STEP 2<br>Accept credential if<br>$s_j^d \equiv I_j \pmod{n_j}$ | $\longleftarrow (I_j, s_j), u_j$ | Present credential and merchant transaction |
| STEP 3<br>Authenticate customer:<br>$z = f(u_i u_j)$<br>$t_i \equiv z^{e_i} \pmod{n_i}$ | $t_i \longrightarrow$ | Accept customer if<br>$t_i^{d_i} \equiv z \pmod{n_i}$ |
| STEP 4<br>Accept merchant if<br>$t_j^{d_j} \equiv z \pmod{n_j}$ | $\longleftarrow t_j$ | Authenticate merchant:<br>$z = f(u_i u_j)$<br>$t_j \equiv z^{e_j} \pmod{n_j}$ |
| STEP 5<br>Record<br>$((I_j, s_j) : u_i u_j, t_j)$ | | Record<br>$((I_i, s_i) : u_i u_j, t_i)$ |

The merchant keeps the 4-tuple $((I_i, s_i) : u, t_i)$ as his certified receipt for the transaction while the customer keeps the 4-tuple $((I_j, s_j) : u, t_j)$. Anyone, using only the publicly available information, $n, d$ and $f$ can later verify that the merchant's 4-tuple satisfies $t_i^{d_i} = z \pmod{n_i}$, which validates the transaction description and verifies that it was endorsed by customer $i$. The customer's 4-tuple can be validated in a like manner.

## References

1. Australian Standard 2805.4–1985, Electronic Funds Transfer–Requirements for interfaces, Part 4–Message Authentication.
2. J. L. Massey, 'An Introduction to Contemporary Cryptology', *Proc. IEEE* **76**, No.5 (May, 1988), 533–549.
3. G. J. Simmons, 'A Survey of Information Authentication', *Proc. IEEE* **76**, No.5 (May, 1988), 603–620.

*Department of Mathematics, James Cook University of North Queensland, Townsville, Queensland 4811, AUSTRALIA.*

13

# INSECURITY OF THE KNAPSACK ONE-TIME PAD

R. T. Worley

O'Connor proposed an approximation to the one-time pad which has similarities to the knapsack cryptosystems, but which he hoped would circumvent the low density attacks on these systems. In this report it will be shown the method proposed is insecure, being susceptible to attacks based on the short lattice vector algorithm.

## 1. Introduction.

Public key cryptosystems based on the knapsack trapdoor have been considered insecure since a polynomial time algorithm was found for breaking instances of the code. In particular, the Lenstra, Lenstra and Lovasz algorithm for producing short basis vectors of a lattice has proved useful in attacking the diophantine approximation problems which arise in attempts to break the codes. O'Connor [3] has proposed an interesting variant of the knapsack cryptosystem. Unfortunately, as will be shown, the diophantine equations associated with the system fall to the short basis vector attack. All instances of the proposed system that have been generated to test the system have been broken. It seems that there are many sets of parameters that will generate any instance of the code, and it is too easy to find such a set.

## 2. Construction of the system.

The one-time pad proposed by O'Connor is constructed in the following way. Suppose we wish to encode messages $m$ with $0 \leq m \leq B$. Select numbers $b_1, \ldots, b_n$, select numbers $p, q$ so that

$$q > pB, \tag{1}$$

and select a number $M$ having no factor in common with $pq$, so that

$$M \geq q(b_1 + \cdots + b_n + 1). \tag{2}$$

Use the extended euclidean algorithm to determine $p^*$ so that $pp^* \bmod M = 1$. Finally, calculate numbers $b_i^*$ as $b_i^* = qp^* b_i \bmod M$. The numbers $b_i^*$ and $B$ are made public.

## 3. Encryption.

Numbers $x_1, \ldots, x_n$, where $x_i = 0$ or 1, are selected at random. Then the encoded message is

$$C = m + (x_1 b_1^* + \cdots + x_n b_n^*).$$

## 4. Decryption.

Decryption is based on the identity

$$\begin{aligned} pC &= pm + p(x_1 b_1^* + \cdots + x_n b_n^*) \\ &= pm + p(x_1 p^* q b_1 + \cdots + x_n p^* q b_n) \\ &= pm + pp^* q(x_1 b_1 + \cdots + x_n b_n) \end{aligned}$$

But $pp^* \bmod M = 1$, so

$$pC \bmod M = pm + q(x_1 b_1 + \cdots + x_n b_n) \bmod M.$$

The inequalities of Section 2 show

$$pm + q(x_1 b_1 + \cdots + x_n b_n) < M$$

and so

$$pm + q(x_1 b_1 + \cdots + x_n b_n) = pC \bmod M.$$

Since $pm \leq pB < q$,

$$pm = pm + q(x_1 b_1 + \cdots + x_n b_n) \bmod q = (pC \bmod M) \bmod q.$$

Finally, since $m \leq B \leq q < M$ and $pp^* \bmod M = 1$, we obtain

$$m = p^* pm \bmod M = p^*((pC \bmod M) \bmod q) \bmod M.$$

## 5. Breaking the code.

We can break the code if we can recover $p, q$ and $M$, which satisfy the requirements of Section 2 from the publicly available $b_i^*$. Reducing a number mod $M$ will on average give a number of size $M/2$, so we expect the $b_i^*$ to be of this size and, accordingly, much greater than the $b_i$. Suppose we can find $A^*$ such that $A^* q p^* \bmod M = 1$. Then $A^* b_i^* \bmod M = A^* q p^* b_i \bmod M = b_i$. This means there exist integers $k_i$ such that

$$A^* b_i^* - k_i M = b_i, \tag{3}$$

or, on dividing by $M$ and rearranging,

$$\frac{A^*}{M} b_i^* = k_i + \frac{b_i}{M}. \tag{4}$$

Using equation (3) we find

$$k_1(A^* b_i^* - k_i M) - k_i(A^* b_1^* - k_1 M) = k_1 b_i - k_i b_1,$$

or

$$k_1 b_i^* - k_i b_1^* = \frac{k_1 b_i - k_i b_1}{A^*}.$$

Under the assumption that $A^*$ is large, the right hand side of this equation is approximately zero, so

$$k_1 b_i^* - k_i b_1^* \approx 0.$$

If we write this in vector form, we discover that

$$k_1(b_2^*, \ldots, b_n^*) - k_2(b_1^*, 0, \ldots, 0) - \cdots - k_n(0, \ldots, 0, b_1^*) \approx (0, \ldots, 0).$$

In other words, for $r \leq n$, there is a combination of the rows of the integer matrix

$$G_r = \begin{bmatrix} cb_2^* & \cdots & cb_r^* & 1 \\ cb_1^* & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & cb_1^* & 0 \end{bmatrix},$$

namely

$$\mathbf{v} = k_1(cb_2^*, \ldots, cb_r^*, 1) - \cdots - k_r(0, \ldots, 0, cb_1^*, 0) \tag{5}$$

which has its first $r-1$ entries $c(k_1 b_i^* - k_i b_1^*)$ small and its last entry $k_1$. If we assume $k_1/b_1^* \approx A^*/M \approx 1/2$ and $k_1 b_i^* - k_i b_1^* \approx 1$ then taking $c = [b_1^*/2]$ will ensure all entries of $\mathbf{v}$ are of approximately the same size.

The above matrix has two short vectors which can be formed from its rows. One will be $(0, \ldots, 0, b_1^*)$, and the other the vector $\mathbf{v}$. It can be expected that the basis reduction algorithm of Lenstra, Lenstra and Lovasz [2] (perhaps in the form in [4] or [5]) will produce these vectors. The value of $k_1$ can be read off as the last component of the second of the vectors. Using the value of $k_1$ we can determine all the remaining $k_i$ as the integer nearest $k_1 b_i^*/b_1^*$. This gives a set of $n$ rational approximations $k_i/b_i^*$ to the rational $A^*/M$. From this set of rational approximations we can hope to determine $A^*/M$ and then determine $p$ and $q$.

In practice, the components of the vector $\mathbf{v}$ may have a common factor, $d$, say, and we may find a rational multiple, say $l/d$ times the vector $\mathbf{v}$. This means that the last component of the short vector we find by the basis reduction algorithm may have its last component $lk_1/d$. To cope with this we divide the short vector we have found by the greatest common divisor of its components, and then take the last component, $v$ say, to be $k_1/d$. We then need to determine the possible values of $d$. The requirement that $dvb_i^*/b_1^*$ be close to an integer (the integer $k_i$) will eliminate many values of $d$. Normally, there will not be many values of $d$ less than $b_1^*/v$ to try. If there are more than a few it may be worthwhile rearranging the $b_i^*$ used in the matrix $G_r$.

From the set of approximations we want to get possible candidates for $A^*/M$. Of course $M$ is bigger than all the $b_i^*$, and probably less than $2b_i^*$. To generate candidates we can use 'extrapolating mediants'. Equation (4) indicates that $A^*/M$ is larger than all the $k_i/b_i^*$, so if $k_j/b_j^*$ is the largest then any number of the form $(mk_j - k_i)/(mb_j^* - b_i^*)$ where $i \neq j$ and $m \geq 2$ will be a candidate for $A^*/M$. One can also generate candidates from $k_j/b_j^*$ using the extended euclidean algorithm. If the greatest common divisor $g$ of $k_j$ and $b_j^*$ is written as $g = xk_j - yb_j^*$, then

$$\frac{k_j}{b_j^*} = \frac{g}{xb_j^*} + \frac{y}{x}.$$

If $xb_j^*$ is positive then $y/x < k_j/b_j^*$ and we can take extrapolating mediants in the same way as before. On the other hand, if $xb_j^*$ is negative then interpolating mediants $(mk_j + |y|)/(mb_j^* + |x|)$, for $m \geq 1$ are candidates for $A^*/M$.

A little investigation shows that further candidates for $A^*/M$ can be obtained from fractions $C/M$, where $C/M$ is smaller than all the $k_i/b_i^*$, by setting $A^* = M - C$. Of course, these small fractions can be obtained by using mediants based on the smallest $k_i/b_i^*$.

Having obtained candidates for $A^*/M$ it now remains to try to obtain $p$ and $q$. The $b_i$ can be obtained as $A^*b_i^* \bmod M$ and the basis reduction method can again be used to check if the candidate $A^*/M$ gives valid $p, q$. This can be done by observing that $pb_i^* - qb_i \bmod M = 0$, which means there exist integers $h_i$ such that

$$p(b_1^*, \ldots, b_n^*) - q(b_1, \ldots, b_n) - h_1(M, \ldots, 0) - \cdots - h_n(0, \ldots, M) = (0, \ldots, 0).$$

In addition, $p = qA^* \bmod M$ implies $p - qA^* - hM = 0$ for a suitable $h$, and $p$ is small, so the matrix

$$H_r = \begin{bmatrix} eb_1^* & eb_2^* & \cdots & eb_r^* & e & f & 0 \\ eb_1 & eb_2 & \cdots & eb_r & eA^* & 0 & 1 \\ eM & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & eM & 0 & 0 \end{bmatrix}.$$

has the small vector $(0, \ldots, 0, pf, -q)$ as a linear combination of its rows. We try to ensure this vector will be detected by the basis reduction algorithm. If we take $f = B$ then the dominant entry is $-q$. On the other hand, suppose we take $e = b_1^*$. Since any nonzero entry in the first $r$ places is divisible by $e$ and the order of magnitude of $b_1^*$ is $M$ and $q$ is much less than $M$, we can certainly expect the basis reduction algorithm to produce the vector $(0, \ldots, 0, pf, -q)$, from which we can obtain $p$ and $q$. All that remains is to confirm or reject $p, q, M$ by checking the requirements (1), (2) and $\gcd(pq, M) = 1$. If we reject one set, we can try another candidate $A^*/M$ or take another candidate $dv$ for $k_1$ and generate further candidates for $A^*/M$.

The matrix above can be reduced to upper triangular form using integer row operations instead of applying the basis reduction algorithm . The last two rows will have the form $(0, \ldots, 0, x, z)$ and $(0, \ldots, 0, 0, y)$. From these all possible short vectors of the form $(0, \ldots, 0, \alpha x, \alpha z + \beta y)$ with the correct signs can be produced and tested to see if they satisfy the requirements. This is not recommended as it is rather like an exhaustive search. However, it was used to see how many possible sets $p, q, M$ generate the $b_i^*$ given in O'Connor's paper (see later).

## 6. An example of breaking the code.

The example in [3] uses 65161, 50306, 25153, 38489 as the $b_i^*$ and 8 as $B$. It is convenient to ensure the $b_i^*$ are ordered so $b_1^*$ is the largest as this is most likely to ensure that the presumed condition $M/2 < b_1^* < M$ holds. Applying the basis reduction algorithm to the rows of the matrix

$$\begin{bmatrix} 50306c & 25153c & 38489c & 1 \\ 65161c & 0 & 0 & 0 \\ 0 & 65161c & 0 & 0 \\ 0 & 0 & 65161c & 0 \end{bmatrix}$$

yields, for $c = 32580$, the small vector $(-6c, -3c, -2c, 25481)$. This shows that

$$(-6c, -3c, -2c, 25481) = k_1(50306c, 25153c, 38489c, 1) - k_2(65161c, 0, 0, 0) - k_3(0, 65161c, 0, 0) - k_4(0, 0, 65161c, 0).$$

Trying $d = 1$, that is, $k_1 = 25481$, we find that the numbers $50306k_1/65161 = 19671.99^{(+)}, 25153k_1/65161 = 9835.99^{(+)}, 38489k_1/65161 = 15050.99^{(+)}$ are nearly integers, so we take $k_1 = 25481$, $k_2 = 19672$, $k_3 = 9836$, $k_4 = 15051$.

Now $A^*/M \approx k_i/b_i^*$ for any $i$, as mentioned in the previous section, so for $A^*/M$ we have the approximations 25481/65161, 19672/50306, 9836/25153 and 15051/38489. The smallest of these is 25481/65161. Applying the extended euclidean algorithm to 65161 and 25481 yields $13336 * 25481 - 5215 * 65161 = 1$, so 5215/13336 is close to 25481/65161, and is in fact smaller. The mediant, $(5215 + 25481)/(13336 + 65161) = 30696/78497$, is therefore close to 25481/65161 and leads to $(78497 - 30696)/78497 = 47801/78497$ as a candidate for $A^*/M$. Other candidates can be obtained by taking extrapolating mediants, for example $(2(25481) - 19672)/(2(65161) - 50306) = 31290/80016$, which gives the candidate 48726/80016. Further candidates can be obtained by working from the largest approximation, and still more candidates could be obtained by considering the possibility $d = 2$, that is, $k_1 = 50962$.

Having obtained candidates for $A^*/M$ it now remains to check each candidate to try to obtain $p$ and $q$. As described earlier, we take $f = B$ and $e = b_1^*$ and apply the basis reduction algorithm to the matrix

$$\begin{bmatrix} eb_1^* & eb_2^* & \cdots & eb_r^* & e & B & 0 \\ eb_1 & eb_2 & \cdots & eb_r & eA^* & 0 & 1 \\ eM & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & eM & 0 & 0 \end{bmatrix}.$$

To try the candidate 47801/78497 we first calculate the $b_i$ as $47801b_i^*$ mod 78497, and we find the $b_i$ are $1, 8, 4, 3$. Taking $r = 1$ we apply the basis reduction algorithm to the matrix

$$\begin{bmatrix} 65161e & e & B & 0 \\ e & 47801e & 0 & 1 \\ 78497e & 0 & 0 & 0 \\ 0 & 78497e & 0 & 0 \end{bmatrix}.$$

Two short vectors of the expected form $(0, 0, \cdot, \cdot)$ were obtained. These vectors were $(0, 0, -424, -335)$ and $(0, 0, -800, 849)$. These give values $p = 53$, $q = -335$ and $p = 100$, $q = 849$. The first of these has sign problems, as well as not having $q$ big enough in size relative to $p$ and the sum of the $b_i$. However, the second seems acceptable. It is easy to verify that $p = 100$, $q = 849$, $M = 78497$ satisfy the constraints and form a set of parameters generating the $b_i^*$ of [3]. Indeed, calculating $p^* = 52593$ we find that

$$((90317 * 100) \bmod 78497) \bmod 849) * 52593 \bmod 78497 = 3$$

and we have correctly decoded the message in [3].

It will be noted that the method produces the same $p, q$ pair as the one in [3]. If, however, we subtract the first short vector from the second we are led to the solution $p = 47$, $q = 1184$ which also satisfies the constraints and so correctly decodes the message. In practice, when trying to produce decoding parameters it is desirable to try combinations of the short vectors produced by the basis reduction algorithm. In order to find out how many decoding sets $p, q, M$ correspond to the $b_i^*$ in [3], the matrix above was reduced to triangular form and all possible short vectors were generated. In this way 12 other pairs $p, q$ were found for the modulus 78497. Other candidates for $A^*/M$ were tried and a further 41 sets of parameters $p, q, M$ generating the given $b_i^*$ were found. It seems that the large number of sets of parameters that generate a given set of $b_i^*$ will make it easy to find one set and so break the encryption.

## 7. Remarks.

The approach given may not always break the code, at least with the first candidate tried for $A^*/M$. Candidates may yield $p, q$ that

- fail to satisfy (1),
- fail to satisfy (2),
- fail to satisfy the requirement that $p, q$ have no factor in common with $M$.

For example, the two weighted mediants $(2(25481) - 15051)/(2(65161) - 38489)$ and $(2(25481) - 19792)/(2(65161) - 50306)$ when tried as candidates for $A^*/M$ yield $p, q$ pairs which have factors in common with $M$. The mediant of these two candidates, 67201/171849, however, yields the short vectors $(0, 0, 13, 1519)$ and $(0, 0, 64, -5741)$ (after removing the scaling factor $B$). The value $q = 5741$ is too big to satisfy (2), but adding multiples of the first short vector to the second yields the short vectors $(0, 0, 77, -4222)$, $(0, 0, 90, -2703)$ and $(0, 0, 103, -1184)$. The first of these still has $q$ too big, the second has $p, q$ with a factor 3 in common with $M$, but the last yields $p = 103$, $q = 1184$ which satisfy the requirements and lead to another method of breaking the code.

Further random sets of $b_i^*$ were generated, and then the decoding method above attempted. In all but one case a triple $p, q, M$ to break the code was detected, using $H_1$ and either $G_3$ or $G_4$. The tests were performed using moduli $M$ of up to 180 bits (about 2/5 the size proposed by O'Connor) using the M.I.R.A.C.L. library for long integer arithmetic written by M. Scott. The tests were performed on a microcomputer and took 30 minutes to 90 minutes to break the large instances, the time being dependent on how many approximations had to be tested before a suitable one was found.

## 8. Finding $A^*, M, q$ using continued fractions.

Although the above method of obtaining decryption parameters by generating and trying approximations has worked well in practice in all the tests I have performed, the use of continued fractions instead of applying the basis reduction algorithm to $H_r$ is more methodical. The continued fraction approach is capable of finding all sets of decrypting parameters, subject to certain conditions, once the approximations $k_i/b_i^*$ to $A^*/M$ have been found.

Firstly, the equations

$$b_i = A^* b_i^* - k_i M = M\left(\frac{A^*}{M} b_i^* - k_i\right)$$

imply

$$\sum b_i = M\left[\frac{A^*}{M}\sum b_i^* - \sum k_i\right].$$

The requirement that $M \geq q(1 + \sum b_i)$ forces

$$\frac{1}{q} > \frac{A^*}{M}\sum b_i^* - \sum k_i.$$

If we set $S = \sum b_i^*$, $T = \sum k_i$ and $k/b^* = \max_i k_i/b_i^*$, then we have the condition

$$\frac{k}{b^*} \leq \frac{A^*}{M} < \frac{qT+1}{qS}. \tag{6}$$

(The left hand inequality is needed to ensure $b^* \geq 0$.) For a rational $A^*/M$ to exist satisfying (6) we must certainly have

$$q < \frac{b^*}{kS - b^*T}. \tag{7}$$

Secondly, the equation

$$p = qA^* - hM = qM\left(\frac{A^*}{M} - \frac{h}{q}\right)$$

shows that

$$\left|\frac{A^*}{M} - \frac{h}{q}\right| = \frac{p}{qM}.$$

Since $M \geq q(1 + \sum b_i)$ the right side of the above equation will be less than $1/2q^2$ providing $2p < 1 + \sum b_i$. In this case $h/q$ will have to be a convergent to the simple continued fraction for $A^*/M$ (see [1], chapter 10). This condition need not hold. However, for the parameters O'Connor suggests as realistic, $2p \approx 2^{51}$ and $\sum b_i \approx 2^{107}$ and $h/q$ will be a convergent to the simple continued fraction for $A^*/M$ prior to a very large partial quotient. We therefore look for $A^*, M$ and $q$ such that $h/q$ is a convergent $h_i/q_i$ of the simple continued fraction for $A^*/M$. In this case we must have

$$\left|\frac{A^*}{M} - \frac{h_i}{q_i}\right| > \frac{1}{q_i(q_{i+1} + q_i)}$$

and the requirement $q \geq pB$ implies

$$M < \frac{q_i(q_{i+1} + q_i)}{B}. \tag{8}$$

Of course, we do not know the continued fraction for $A^*/M$, but (6) indicates we can work with the continued fraction of $k/b^*$.

The two inequalities (6) and (8) pull in opposite directions. For a given value of $q = q_i$ inequality (8) gives an upper bound on $M$. However, the continued fractions for the left and right sides of (6) may agree until the convergents have denominator larger than the bound on $M$, indicating no fraction can satisfy the inequality. Usually, only a small number of $q_i$ satisfying (7) are such that it is possible to find $A^*/M$ satisfying (6). We illustrate this with two examples.

In the case of the small example given by O'Connor in the report, where $B = 8$ and the $b_i^*$ are 65161, 50306, 25153, 38489, the largest approximation $k_i/b_i^*$ found earlier was 9836/25153. Inequality (7) gives $q \leq 6288$, and a portion of the continued fraction expansion of 9836/25153 is given in the following table.

| $a_i$ | 1 | 1 | 3 | 1 | |
|---|---|---|---|---|---|
| $q_i$ | 179 | 335 | 1184 | 1519 | 11817 |
| $M$−bound | 7518 | 49915 | 225996 | 2245271 | |

If we take $q = 179$ and examine the continued fractions

$$(qT + 1)/S = \langle 0, 2, 1, 1, 3, 1, 6, 1, 1, 3, 1, 4, \ldots \rangle$$
$$9836/25153 = \langle 0, 2, 1, 1, 3, 1, 6, 1, 1, 3, 1, 7, 2 \rangle$$

we see that the rational with smallest denominator satisfying (6) is

$$\langle 0, 2, 1, 1, 3, 1, 6, 1, 1, 3, 1, 5 \rangle = 3433/8779,$$

which has denominator exceeding the bound 7518 given by (8). However for $q = 335$ we have $(qT + 1)/S = \langle 0, 2, 1, 1, 3, 1, 6, 1, 1, 3, 1, 5, 1, \ldots \rangle$, and we can take $A^*/M$ to be the rational $4027/10298 = \langle 0, 2, 1, 1, 3, 1, 6, 1, 1, 3, 1, 6 \rangle$. Other choices with $M$ satisfying (8) can be made, such as

$$\langle 0, 2, 1, 1, 3, 1, 6, 1, 1, 3, 1, 6, 1 \rangle = 4621/11817$$
$$\langle 0, 2, 1, 1, 3, 1, 6, 1, 1, 3, 1, 6, 2 \rangle = 8648/22115$$
$$\langle 0, 2, 1, 1, 3, 1, 6, 1, 1, 3, 1, 5, 2 \rangle = 11487/29375.$$

Using $A^*/M = 4027/10298$, $q = 335$ gives $p = qA^* \bmod M = 7$, $b_1 = A^* b_1^* \bmod M = 9$, $b_2 = 6$, $b_3 = 3$, $b_4 = 5$, while using $A^*/M = 4621/11817$, $q = 335$ gives $p = 8$, $b_1 = 4$, $b_2 = 2$, $b_3 = 1$, $b_4 = 2$. It should be remarked that the validity of these decryption parameters is not affected by the fact that $M$ is less than the given $b_i^*$. It should also be remarked that only alternate convergents to $k/b^*$ may prove useful, since we need convergents smaller than $A^*/M$ and the convergents are on alternate sides of $k/b^*$. It depends on whether (6) allows the choice of $A^*/M$ bigger than the chosen convergent $h_i/q_i$ of $k/b^*$ if that convergent is bigger than $k/b^*$.

Consider a second example, in which the parameters are $q = 741665264141219084$, $p = 590558651$, $M = 1157410030370067346107610152031$ and $b_i, b_i^*$ given in the following table (these parameters were randomly generated and ensure $M$ much greater

than $q(1+\sum b_i)$).

| | |
|---|---|
| 25211821598 | 765920753460166959185655937840 |
| 26755779464 | 342501481761259505563660343932 |
| 21858401296 | 364149457483126148970196920796 |
| 34133988443 | 605333570948994319096229812 04 |
| 19829368450 | 110914061458958552385063284 9971 |
| 29752209669 | 102659016642604141478434463 8221 |
| 30528171930 | 182778266183736520468979797980 |
| 33767051582 | 821153591251424710229482435243 |
| 28736975919 | 106965361426088010382495827 8400 |
| 24008946934 | 54784275893001174940007606 8157. |

The basis reduction algorithm yields approximations $k_i/b_i^*$, of which the largest is

$$224575368810520674504206342 47/605333570948994319096229812 04.$$

The continued fraction for this has a portion $\langle \ldots, 5, 1, 2641, 1, \ldots \rangle$. The large partial quotient is the only large partial quotient, and taking $h/q$ to be the convergent prior to this point yields the value of $q$ which was used to generate the $b_i^*$. For this value of $q$ the continued fractions of the rationals bounding $A^*/M$ are

$$\langle \ldots, 2641, 1, 1, 8, 2, 1, 1, \ldots \rangle \quad \text{and} \quad \langle \ldots, 2641, 1, 1, 8, 2, 1, 2, \ldots \rangle$$

and so a candidate for $A^*/M$ is $\langle \ldots, 2641, 1, 1, 8, 2, 1, 2 \rangle$ which gives

$$A^* = 157779763713823354215651 \quad \text{and} \quad M = 425288794128445504822373.$$

This value of $M$ satisfies (8) for the given value of $q$, and it is easily checked that the $p, q, M$ triple derived from these values of $q, A^*, M$ decodes the $b_i^*$. If the bound (7) is calculated it is found that the value of $q$ used above is the largest $q_i$ that can be taken. Smaller $q_i$ can be used. The smallest for which there is a rational satisfying (6) of small enough denominator is $q = 19676184937045817$, and the $A^*/M$ value is the same one as used above.

## References


1. G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers* (Oxford University Press, 1965.)
2. A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovasz, 'Factoring polynomials with rational coefficients', *Math. Ann.* **261** (1982), 515–534.
3. L. O'Connor, 'An Approximation to the One-Time Pad', Technical Report 328, Dept. of Computer Science, University of Sydney, 1988).
4. C. P. Schnorr, 'A More Efficient Algorithm for Lattice Basis Reduction', *J. Algorithms* **9** (1988), 47–62.
5. B. M. M. De Weger, 'Solving Exponential Diophantine Equations Using Lattice Basis Reduction Algorithms', *J. Number Theory* **26** (1987), 325–367.


*Department of Mathematics, Monash University, Clayton, Victoria 3168, AUSTRALIA.*

14

# THE TACTICAL FREQUENCY MANAGEMENT PROBLEM: HEURISTIC SEARCH AND SIMULATED ANNEALING

**Lindsay Peters**

The Tactical Frequency Management Problem is an example of the general Consistent Labelling Search Problem where frequencies and other communications data need to be assigned to mobile radio users in a dense deployment on a daily basis, such that the resulting mutual Electromagnetic Interference is minimised. This paper presents a qualitative description of three quite different methods for solving this problem. The first is based on the manual procedure historically used by communications planners. The second is based on a classical heuristic search algorithm employing backtracking and forward checking—a typical Artificial Intelligence approach. The third approach uses an algorithm which is a mathematical analogue to the physical process of annealing a molten metal—heating it and then cooling it so as to let it crystallise into its lowest energy state.

## 1. Introduction.

The Frequency Management Problem is a special case of a general NP-Complete problem called the 'Consistent Labelling Problem' [8] or the 'Constraint Satisfaction Problem' [3], in either case abbreviated as *CLP*. This problem consists of a list of $N$ units (or 'variables'), $U = (u_1, u_2, \ldots, u_N)$, each unit having a set $L = (L_1, L_2, \ldots, L_M)$ of $M$ possible labels (or 'values'). A consistent labelling $f$ is a relation $f : U \rightarrow L$ such that for each pair of units and assigned labels, the required constraints are satisfied. That is,

$$(u, f(u), v, f(v)) \in R \text{ for all } u, v \in U,$$

where the constraint $R$ is represented as a subset of $(U \times L) \times (U \times L)$, the set of all possible pairs of unit/label assignments.

A simple example of the *CLP* type is the colour mapping problem where the units are the countries, the labels are the colours and the constraint is the condition that adjacent countries have to be coloured differently. Another well-known example is the travelling salesman problem where the units are cities, the labels are pointers to the next city (an ordering) and the constraint is that all cities have to be visited exactly once and the total distance travelled has to be minimised.

In the special case of the Frequency Management Problem, the units are radio nets (groups of radios sharing the same frequency in order to communicate with each other) and the labels are frequencies that are available for assignment to those nets. The constraint is basically the set of those rules that ensure that the Electromagnetic Interference (*EMI*) in the deployment is minimised.

In practice each net may require many alternate frequencies but for simplicity we will assume there is only one frequency required per net. Furthermore, the problem is complicated by the fact that there are usually several pre-assigned nets over which the assignment algorithm has no control, but which must nevertheless be taken into account when assigning all the other nets. Each net may consist of up to 25 radios. Over half of the radios on each net may be sited with radios on other nets, as shown in the trivial example in Figure 1. The siting of radios in clumps, that is, in 'cosites' (radios on the same vehicle) or 'co-locations' (radios within approximately a 5 km radius of each other) is the principal cause of interference.

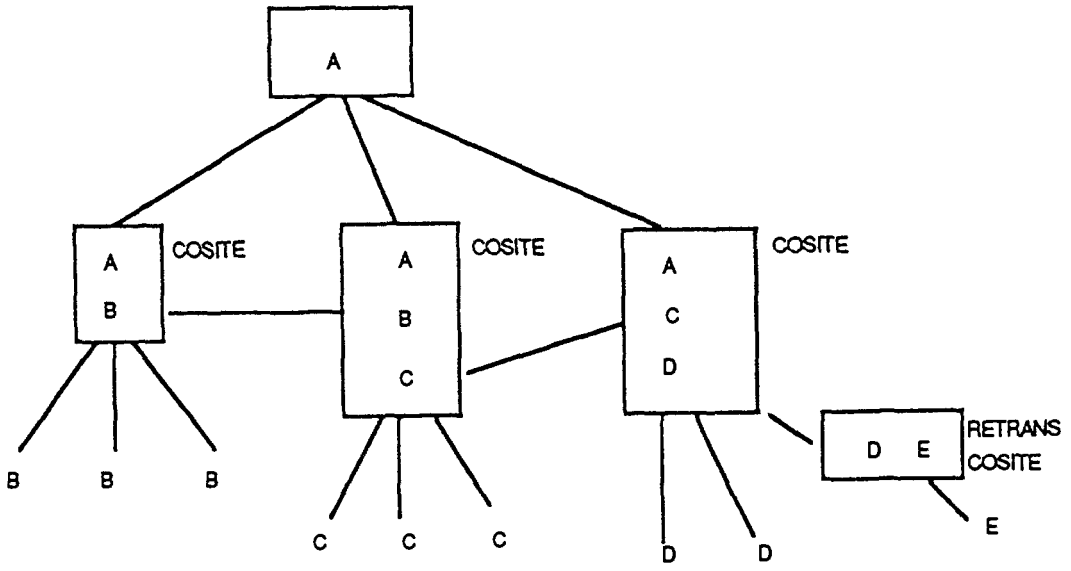



Figure 1. Example of a radio network.

A perfect assignment would be one where the resulting *EMI* due to all radio users does not exceed the prevailing noise floor. The noise floor is due to atmospheric, galactic, and man-made noise, and the noise of the radio equipment itself.

For the travelling salesman problem it is clear that a solution exists, (and similarly for the colour mapping problem it is well known that a solution exists for plane maps and 4 colours). However, for this problem it is also known that there is no faster algorithm for finding the solution than calculating all possible travel routes.

For the Frequency Management problem the situation is worse since for any non-trivial deployment of radios, no perfect assignment exists. For example, there are always fewer available frequencies than radio nets. Hence it is inevitable that some radios will interfere with each other since they will have to share the same frequency and typically will not be sufficiently separated from each other for the signal not to propagate from one net to the other.

Section 2 describes the major constraints relevant to the assignment of frequencies to nets, in particular the major interference effects relevant for tactical HF and VHF communications. Early attempts to solve this problem are outlined in Section 3. They involved manual graphic techniques historically used by communications planners. These were superseded by computer based methods based on these manual

procedures. Section 4 describes the initial attempt to improve on these early techniques by using algorithms that have been successful for game playing programs, in particular, backtracking and forward checking. This method conducts a series of local 'depth-first' searches to find small 'paths' (net/frequency allocations) which are good in the sense that they never exceed some prescribed average level of interference. Nodes in the search space from which there are no good paths are pruned forever, and the algorithm backtracks to find new paths until every net is assigned along some good path. Finally, Section 5 briefly describes the general method of simulated annealing and its success in dealing with the Frequency Management problem.

## 2. Constraints for frequency management.

The constraints are due to the following major interference mechanisms, and can be expressed in terms of equalities involving usually no more than 4 frequencies

$$\nu_i, \nu_j, \nu_k, \nu_l \quad \text{where} \quad i, j, k, l < M$$

and $M$ is the total number of available frequencies. For this problem, available frequencies may be in the range 2 to 88 MHz.

### 2.1. *Co-channel and Harmonic Interference.*

This is the worst possible mechanism and applies to radios even outside the cosites and colocations. Frequencies should not be either co-channel or harmonics of each other (unless they are sufficiently separated as already indicated). The corresponding relations to be avoided are:

$$\nu_i = n\nu_j \quad \text{for} \quad n = 1, 2, 3, \ldots .$$

### 2.2. *Adjacent Channel Interference.*

This is the second worst mechanism. Basically, a receiver assigned the wanted frequency $\nu_j$ will be blocked or desensitised if there is an unwanted signal $\nu_i$ close (typically within 5% to 20%) to the wanted signal. That is,

$$1 - \delta < \nu_i/\nu_j < 1 + \delta$$

where

$$0{\cdot}05 < \delta < 0{\cdot}2.$$

Here, $\delta$ depends on the wanted frequency $\nu_j$, the received power of the wanted and unwanted signals, and on the performance of the individual receiver.

### 2.3. *Spurious Emissions and Responses.*

All radio transmitters and receivers have characteristic spurious emissions and responses which should be avoided in cosite and co-location situations. The corresponding relations are:

$$\nu_i = m\nu_j + n\nu_{LO} + \nu_{spur}$$

where $\nu_{LO}$ is the local oscillator frequency, $m$ and $n$ are integers, and $\nu_{spur}$ is a constant offset frequency, all determined by the victim receiver or interfering transmitter.

2.4. *Receiver Intermodulation.*

Receiver intermodulation products occur when 2 or more input signals mix in a non-linear portion of the receiver to form a third (unwanted) frequency. If it happens to be close to the wanted frequency it may not be filtered out and so will cause interference. The worst products are the following:

$$\nu_i = 2\nu_j - \nu_k$$
$$\nu_i = 3\nu_j - 2\nu_k$$
$$\nu_i = \nu_j + \nu_k - \nu_l$$
$$\nu_i = 2\nu_j - 2\nu_k + \nu_l$$

2.5. *Sundry Constraints.*

Apart from these major *EMI* effects, there are many other constraints imposed on the Frequency Management problem. They include the following:

(i) Certain nets use the ionosphere for propagation, and so must have frequencies selected from within a narrow frequency range that varies continuously throughout the day.

(ii) Nets are usually assigned multiple frequencies and these must be separated by at least 10%.

(iii) Assignments must be changed daily, and must not be able to be predicted from previous assignments, that is, a bias towards any optimal 'preferred' solutions is not allowed.

(iv) Some nets are deemed to have higher priority than others, that is, they in particular should suffer minimal interference. To achieve this means that the total interference for the deployment may in fact be greater than if there were no privileged nets.

(v) In a tactical environment, the time available to perform assignments varies considerably, from a few minutes to several hours. This means that the assignment procedure must be able to degrade gracefully if a severe time limitation is imposed.

## 3. Early techniques.

Up to a decade ago, tactical frequency assignments were done manually by specialised communications planners. They used empirically derived graphs to try to avoid the worst interference mechanisms. Figure 2 is a typical graph for co-channel interference. It relates the wanted signal-to-noise ratio, the distance between the receiver and the wanted transmitter, and the distance to an unwanted transmitter.

The typical assignment procedure was as follows.

(i) Order the nets such that the most difficult problems are encountered first (the heavily cosited nets).

(ii) Randomly choose an available frequency and assign it to the first net.

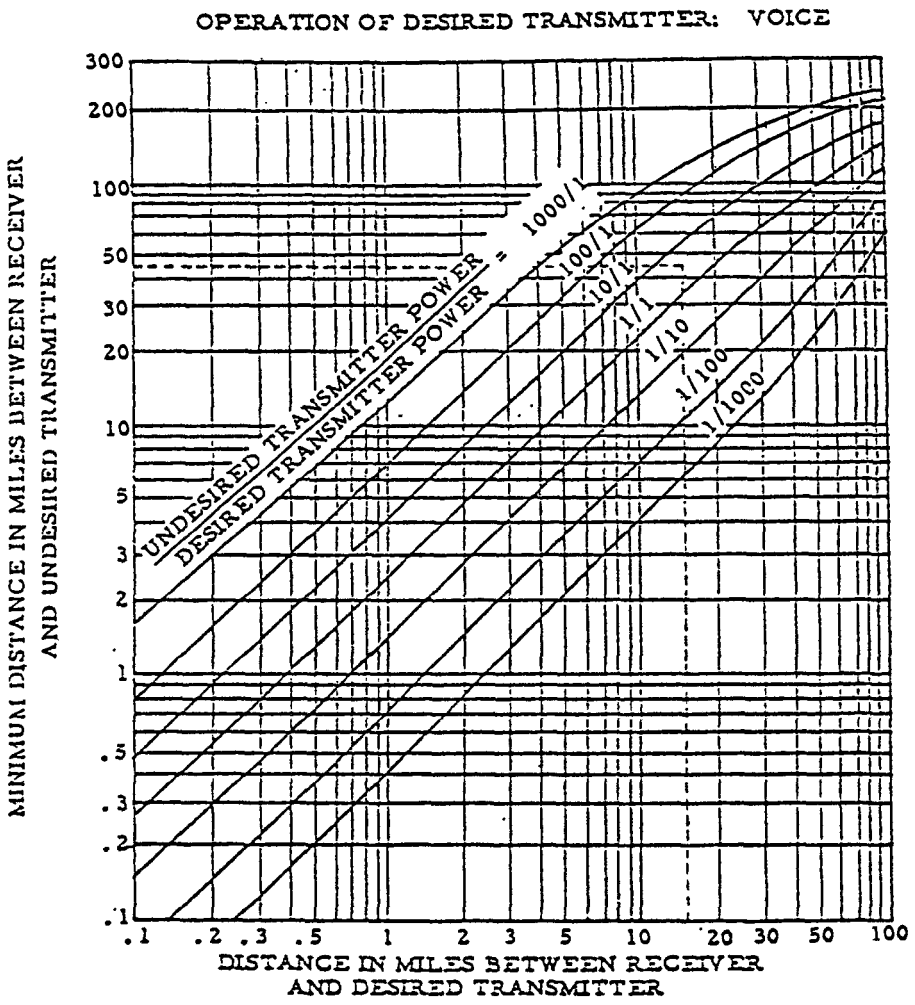



Figure 2. Co-channel interference chart.

(iii) Randomly choose another frequency, assign it to the second net on a trial basis only.

(iv) Check using the graphs to see if any of the stations on the second net will interfere with any of the stations on the first net, and vice versa (the constraints are not all symmetric). If so, reject the trial assignment and repeat step (iii) for the second net. If no interference is predicted then accept the assignment and repeat (iii) for the third net.

(v) Proceed in this fashion until all nets are assigned.

Apart from being an impractical procedure if there are large numbers of nets involved, there inevitably comes a stage when all possible subsequent assignments will produce high levels of interference. The communications planner is then left with the suspicion that the situation may have been improved if different frequencies had been chosen earlier on.

First attempts at automating the assignment of frequencies on an electromagnetic compatibility (*EMC*) basis tried to imitate the manual procedures of the communications planner, for example *FASTNET* [4]. This program highlighted three serious deficiencies of this procedure:

*The Complexity Problem*: Firstly, the algorithm is necessarily very complex as it tries to encapsulate all the knowledge of the communications planner into a deterministic procedure for choosing optimum frequencies in a sequential fashion. Complicated and unmaintainable decision trees that try to cope with all contingencies and special cases are necessary.

*The Innocent Frequency Problem*: Secondly, the suspicions of the communcations planner were completely justified—seemingly 'innocent' frequencies assigned to the first few nets could have disastrous consequences from an interference point of view for nets that are assigned much later. The effect of early decisions propagates through the cosite and co-location links so as to severely narrow the choices available for later decisions.

*The Graceful Degradation Problem*: Thirdly, with a sequential net assignment procedure there is no obvious way to degrade the assignment gracefully in the situation where there is only a very short time to perform the assignment, or alternatively, where an assignment that has already commmenced must now be completed very quickly.

Some attempts were made to alleviate the innocent frequency problem. Frequencies were selected in the early stages not entirely randomly but from pre-determined subsets of the total available list. Frequencies in each of these subsets were guaranteed to be compatible. (See for example Lustgarten [7] and Loxton [6].) While good for specific applications, these techniques worsened the complexity problem as the frequency pre-processing depends heavily on the interference constraints themselves. Furthermore, the interference-free subsets were never large enough to be of any practical use. This was due in part to the non-uniform distribution of available frequencies.

## 4. Classical heuristic search techniques.

The first attempt to address the fundamental cause of the 'innocent frequency problem' which is the sequential nature of the historical assignment procedure was to apply the basic game-playing strategy of backtracking. (See Golomb [1] for an overview of this method.) In this approach nets are still assigned sequentially, but if at any stage all the choices for the next net assignment will necessarily cause the interference threshold to be exceeded, the previous net assignment is undone and that node pruned from the search tree. In this way an exhaustive search strategy can be set up.

The interference threshold is an a priori estimate of the lowest interference level that could reasonably be expected for that stage of the assignment for that particular deployment of radio nets and for that particular number and distribution of available frequencies. The strategy is very sensitive to the threshold value. Too high a value will enable the search to complete but the resulting assignment will be sub-optimal. Too low a value will cause the search to backtrack wildly ('thrash') without any guarantee of being able to complete at all.

It has been shown (see Pearl [9]) that the efficiency of the search procedure may be improved by a combination of backtracking and forward-checking. Forward checking is used on a local and depth-first basis. That is, the search tree is examined not one level (that is, net) at a time, but $L$ consecutive levels at a time. The average interference increment $\chi$ allowed at each level is still required to be known a priori. Paths are

sought through the next $L$ levels such that the total allowed interference increment $\chi L$ is not exceeded. The local forward-checking allows nodes that are not viable to be pruned in advance, and also avoids the 'thrashing' that backtracking is susceptible to. Backtracking is used on a global basis, that is on whole paths of length $L$ found by the forward-checking, in order to restart the search if all subsequent nodes have been pruned. An example of this search procedure is shown in Figure 3 below:

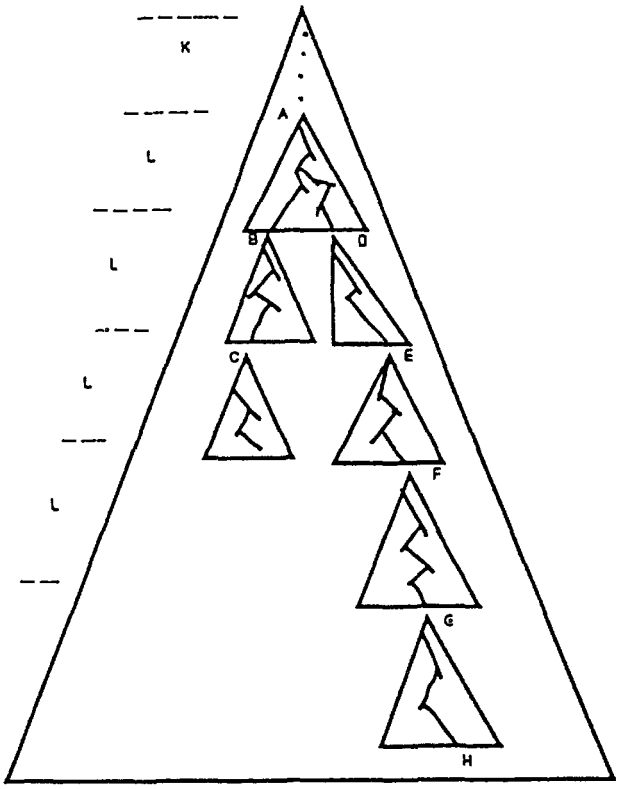



Figure 3. Example of a search tree.

(i) The search starts at node $A$, that is, after any nets that have been pre-assigned.

(ii) The search finds a 'good' path of length $L$ from $A$ to $B$, that is, where the interference increase is less than $\chi L$.

(iii) The search finds a good path from $B$ to $C$.

(iv) The search fails to find a good path from $C$.

(v) The search backtracks to $B$ and deletes node $C$ forever.

(vi) The search fails to find a good path from $B$.

(vii) The search backtracks to $A$ and deletes node $B$ forever.

viii) The search finds a good path from $A$ to $D$, then $D$ to $E$, then $E$ to $F$, then $F$ to $G$, and finally from $G$ to $H$.

In an unexpected way the heuristic search approach alleviated the 'complexity problem'. Formerly, the interference constraints were used as a complicated set of selection rules resulting in a complicated search algorithm. In the present approach, however, they were used to create an interference cost function that could be compared against the required threshold. The search itself is determined by the simple

backtracking and forward checking algorithm. Furthermore, it was proven that this approach did in fact overcome the innocent frequency problem. Optimal solutions to known Frequency Management problems were consistently obtained, no matter what the initial choice of frequencies was. However, the heuristic search approach to the Frequency Management problem suffers from the serious problem of being unable to be 'tuned'. The interference thresholds that are so critical to the performance of the algorithm are very difficult to determine a priori. This is because each day the Frequency Management problem will change significantly. Both the list of available frequencies and the deployment of radio nets will be different and it is very difficult to quantify how these changes should affect the interference thresholds before actually attempting the assignment. Furthermore, the graceful degradation problem still exists with this approach. Clearly if time runs out the search can be made to complete quickly by raising the interference thresholds and thereby reducing the backtracking. However, this is hardly 'graceful' as nets to be subsequently assigned are disadvantaged, and much of the processing time spent early on may now be wasted.

## 5. Simulated annealing.

Simulated annealing [5] is a general heuristic that has been successfully applied to many types of Consistent Labelling Problems. (The Travelling Salesman Problem was one of its first applications.) In this method the system of units and labels is likened to a molten metal or glass which is first heated and then cooled. The cooling stage of the annealing must be done slowly if it is to settle eventually into its lowest energy state (which represents the highest crystallisation structure). At each temperature the metal is in equilibrium which means that the set of possible micro-states it can be in follows a Boltzmann distribution. Cooling too quickly ('quenching') does not allow the metal to reach equilibrium at each temperature and results in defects being frozen into the structure.

Applied to the Frequency Managment problem, the system is given an intitial arbitrary assignment and the overall interference $E$ is evaluated using the cost function mentioned earlier. The algorithm then attempts to make random changes to the assignment—a randomly chosen net is assigned a randomly chosen available frequency. If the cost function decreases, that is, $\Delta(E) \leq 0$, the change is allowed. If, however, $\Delta(E) > 0$, then the change is allowed with a probability

$$\exp(-\Delta(E)/T),$$

where $T$ is the normalised 'temperature' of the system. The temperature is the most important control parameter of the system. At the initial high temperatures $\exp(-\Delta(E)/T) \approx 1$ since $\Delta(E) \ll T$. Hence most changes are allowed, even those that do not improve the assignment. At low temperatures $\exp(-\Delta(E)/T) \approx 0$ since $\Delta(E) \gg T$ and so far fewer 'bad' changes are allowed.

The cooling schedule

$$[T_0, n_0], [T_1, n_1], ..., [T_m, n_m]$$

defines the sequence of temperatures to be used, and the number of changes that have to be accepted at each temperature before it can be lowered. The larger each $n_i$

is, the better chance the system will have to reach equilibrium at the corresponding temperature $T_i$ and the better the overall assignment will be when 'frozen'.

The innocent frequency problem is now completely avoided as there is no concept of some nets being assigned earlier than others, that is, there is no net ordering for assignment. The complexity problem is further alleviated as the implementation of a cooling schedule and the calculation of the probability function is even simpler than the heuristic search strategy previously described. Furthermore, the performance of the algorithm is not particularly sensitive to the scaling of $T$. This means that it does not have to be rescaled for each different Frequency Management deployment.

Finally, the simulated annealing method offers a natural way to degrade gracefully. The cooling schedule can be amended to be of the form

$$[T_0, t_0], [T_1, t_1], ..., [T_m, t_m]$$

where $t_i$ is the processing time allowed at the temperature $T_i$. In this way the time to complete can be precisely controlled either before the assignment begins or at some stage after it has started. The degradation is graceful as no particular nets are disadvantaged by a faster cooling schedule, and the earlier processing time is not wasted if the schedule is speeded up during the assignment. This is because the major interference mechanisms such as co-channel and adjacent channel are resolved *for all nets* early on at the high temperatures.

The less important interference mechanisms such as the intermodulation products are resolved by a small number of changes at the last stages of the cooling schedule, without destroying the overall structure that enabled the major interference mechanisms to be resolved.

Simulated annealing consistently finds the optimal solutions to known Frequency Management problems, as does the heuristic approach described earlier. Unlike the heuristic approach, however, simulated annealing is easily applied to a wide range of realistic Frequency Management problems. The efficiency of the simulated annealing algorithm has been improved by using the standard practice of initialising the system with a quick (high interference threshold) forward checking search, instead of using a random initial state. The cooling schedule could then be started at a lower temperature than would otherwise have been required. Futher attempts at improving the efficiency have concentrated on reducing the number of rejected changes by introducing a bias on the frequency selection process, based on information gained from previous selections (for example, the 'rejectionless method' [2]).

## 6. Conclusion.

A description of the history of attempts to solve the Frequency Management problem has been presented. It has been demonstrated that to date, simulated annealing has proved to be by far the most powerful technique for solving this problem. Its success in this application is due to the fact that it is simple, its performance is not sensitive to the particular Frequency Management problem that it is applied to, and in dealing with all nets simultaneously it bypasses the problems that are inherent in any sequential assignment or ordering of nets.

**Acknowledgement.**

The author gratefully acknowledges the work of G. Wilkins and B. McDowall in proposing and implementing these algorithms. The author also wishes to thank R. Finch for many helpful discussions and Plessey Australia for providing the opportunity to investigate this problem.

## References

1. S. W. Golomb and L. D. Umert, 'Backtrack Programming', *JACM* **12** (1965), 516–524.
2. J. Greene and K. Supowit, 'Simulated Annealing Without Rejected Moves', *IEEE Trans on CAD*, **CAD-5**, No 1 (1986).
3. R. M. Haralick, and G. L. Elliott, 'Increasing Tree Search Efficiency for Constraint Satisfaction Problems', *Artificial Intelligence* **14** (1980), 263–313.
4. R. N. Johnson, 'A Computerised Method of Frequency Assignment for Tactical Net Radios—*FASTNET*', *SASG*, Note 35 (1978).
5. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, 'Optimization by Simulated Annealing', *Science* **220** (1983), 671–680.
6. J. Loxton, 'Report on a Problem on the Assignment of Radio Frequencies'. Report (FMF/1/28, Plessey Australia, 1985).
7. M. N. Lustgarten, 'A Method for Computing Intermodulation-Free Frequency Lists', *IEEE EMC Symposium Record* (1968).
8. B. Nudel, 'Consistent-Labelling Problems and their Algorithms', *Search and Heuristics*, J. Pearls (ed.). (North-Holland, 1983.)
9. J. Pearl and R. P. Karp, 'Searching for an Optimal Path in a Tree with Random Costs', *Search and Heuristics*, J. Pearl (ed.). (North-Holland, 1983.)

*Technical Computing, Plessey Australia,*
*Railway Road, Meadowbank, New South Wales 2114, AUSTRALIA.*

# 15

# REED-SOLOMON CODING IN THE COMPLEX FIELD

**Mark Rudolph**

Reed-Solomon error correction codes are powerful codes which allow efficient correction of multiple burst errors. The $(n, k)$ Reed-Solomon code takes strings of $n$ symbols, each containing $k$ information symbols and $n - k$ check symbols and can correct any occurrence of at most $\frac{1}{2}(n - k)$ incorrectly received symbols ([4]). The Reed-Solomon code is usually implemented via computation in a finite field of characteristic 2. However, this requires special circuits for encoding and decoding which have only recently become economic.

There is an alternative 'spectral' interpretation which uses ordinary complex arithmetic. This has the advantage that conventional floating point devices can be used for encoding and decoding. Its compensating disadvantage is that the arithmetic is no longer exact. For the purposes of this example, let us assume that we would like to code $k = \frac{1}{2}n$ message symbols and that we wish to correct up to $\frac{1}{4}n$ incorrectly received symbols. First, pad the $\frac{1}{2}n$ length message sequence with a 'prefix' of $\frac{1}{2}n$ zero symbols. We view the resulting sequence as a sequence of spectral magnitudes. Thus it has a characteristic 'high-pass' spectrum, that is all low frequency spectral magnitudes are zero. Next, take the inverse fourier transform of the high-pass spectrum and transmit the resulting time domain sequence. The characteristic high-pass spectrum, that is zero symbols in the first $\frac{1}{2}n$ spectral bands, is spread over the entire $n$ time domain components. At the decoder, the received message is transformed into the spectral domain. Since the first $\frac{1}{2}n$ spectral bands are known to be zero, the first $\frac{1}{2}n$ transformed components are a 'window' into the error sequence of length $n$. If these first $\frac{1}{2}n$ symbols are zero, we may be confident that the received message is correct. However, in any case, we may use the exact knowledge of the error in the first $\frac{1}{2}n$ spectral components in order to correct errors of fewer than $\frac{1}{4}n$ symbols in the entire received sequence. (An efficient method for finding the error location sequence is a Toeplitz matrix inversion algorithm called the Berlekamp-Massey algorithm ([1], page 176). It can be demonstrated ([5], page 169) that the $\frac{1}{4}n \times \frac{1}{4}n$ matrix of error 'syndromes' determines a non-singular solution matrix if the number of incorrectly coded symbols is less than $\frac{1}{4}n$. With these $\frac{1}{4}n$ error location coefficients and the known $\frac{1}{2}n$ error components, the other $\frac{1}{2}n$ error correction components can be obtained by recursion.)

The most important parameter which must be determined is the length of the sequence to be transformed. The longer the transform length, the greater the length of continuous burst error which can be completely corrected. However, the length of the transform is also proportional to the amount of computation 'noise' which accumulates

during the process of decoding which involves, in effect, a matrix inversion of size $\frac{1}{2}N$, where $N$ is the transform length. The situation can be improved by using Gaussian 'pivot' methods for solving matrix equations, instead of the Berlekamp-Massey algorithm ([2], page 673) and by an iterative post-processing technique referred to as the 'residual improvement' method ([2], page 680). After implementing these techniques, the complex Reed-Solomon decoder performed perfectly reliably in extensive tests at the theoretical coding limit of $\frac{1}{4}N$ errors for $N = 8$ and with only intermittent single or double bit errors at $N = 16$. (The errors always occurred in the last one or two message values computed.)

Based on these investigations and tests, we choose $N = 8$ for our illustration, since we can be fairly confident that no errors will occur as a result of computational noise. Further security against computation error can be obtained by appending zeros at the tail of the message words. The message then acquires a characteristic 'low pass' spectrum. Therefore, if the transformed signal received at the decoder is put through a low-pass filter prior to decoding, the effect should be to reduce error amplitude, since any high frequency energy in the received message is due solely to transmision error. The idea has been found to be effective and can be further refined. (Indeed, transform lengths of 32 may be possible.) With judicious packing, such a system with $N = 8$ can be designed to achieve a message rate of 0·25. If we assume a transmission rate of 4800 bits/sec, then the message rate is 1200 bits/sec, or 150 characters/sec.

In order to maximise error burst protection, interleaving should be used. For example, suppose the interleaving matrix is 8 rows by 8 symbols per row. The rows are loaded one at a time and, when the matrix is full, the 8 columns are read out and transmitted. Interleaving allows bursts of 8 times the length of burst tolerated in each row. Each row can tolerate errors in any 2 symbols and, in particular, in 2 consecutive symbols. Since each symbol is 64 bits long, an error of at least 65 bits is tolerable and the error burst within one frame could be as long as 128 bits. Therefore, interleaving allows a burst of at least 16 complex symbols, that is 1024 bits, or 0·213 seconds. According to studies by Lutz et al. ([7], figure 3, page 539), the probability of occurrence of an error burst of fade greater than −10 dB for greater than 0·213 seconds is about 0·05. Thus, fewer than 1 in 20 one-second messages require retransmission.

## References

1. R. E. Bluhat, *Theory and practice of error control codes.* (Addison-Wesley, Reading, Mass., 1984.)
2. E. Kreysig, *Advanced Engineering Mathematics.* (Wiley, New York, 1962.)
3. E. Lutz, W. Papke and E. Plochinger, 'Land mobile satellite communications–channel model, modulation and error control', *7-th ICDSC* (1986).
4. I. S. Reed and G. Solomon, 'Polynomial codes over certain finite fields', *J. Soc. Ind. Appl. Math.* **8**, number 2 (1960).
5. J. K. Wolf, 'Redundancy, the discrete Fourier transform and impulse noise cancellation', *IEEE Trans. Comm.* **COM-31** (1983), 458–461.

*Department of Communication and Electrical Engineering, Royal Melbourne Institute of Technology, Melbourne, Victoria, AUSTRALIA 3000.*

16

# CLASS NUMBER PROBLEMS FOR REAL QUADRATIC FIELDS

R. A. Mollin* and H. C. Williams**

## 1. Introduction.

The purpose of this paper is to give an overview of the main recent advances concerning Gauss's class number one problem for real quadratic fields, to describe the connections with prime-producing polynomials, continued fraction theory and the theory of reduced ideals, and to make the comparison with the development of the solution of Gauss's class number one problem for complex quadratic fields. This includes a description of the search for a real quadratic field analogue of the well-known Rabinowitsch result for complex quadratic fields.

Furthermore, we describe a criterion for class number 2 (in terms of continued fractions and reduced ideals) for general real quadratic fields. We also provide (for a specific class of real quadratric fields called Richaud-Degert types) class number 2 criteria in terms of prime-producing quadratic polynomials. This is the real quadratic field analogue of Hendy's result [9] for complex quadratic fields. Other related results including a solution of a problem of L. Bernstein [2], [3] are delineated as well.

## 2. Notation and preliminaries.

*Some remarks concerning ideals.* Before beginning the development of the main results in this paper, it will be necessary to review some of the properties of ideals. The material in this section is well known and can be found in a number of sources. We mention here the book of Cohn [4], the introduction to the table of Ince [11], and the paper of Williams and Wunderlich [39] which are particularly relevant to this discussion.

Let

$$\omega = \frac{(\sigma - 1 + \sqrt{d})}{\sigma},$$

where

$$\sigma = \begin{cases} 1 & \text{if } d \equiv 2,3 \pmod 4 \\ 2 & \text{if } d \equiv 1 \pmod 4 \end{cases}$$

* Research supported by NSERC Canada grant #A8484.

** Research supported by NSERC Canada grant #A7649.

and let $\mathcal{O}_\mathcal{K}$ be the maximal order of our quadratic field $\mathcal{K} = \mathbf{Q}(\sqrt{d})$. The *discriminant* $\Delta$ of $\mathcal{K}$ is given by

$$\Delta = \left(\frac{2}{\sigma}\right)^2 d.$$

Also, if by $[\alpha, \beta]$ we denote the module $\alpha\mathbf{Z} + \beta\mathbf{Z} = \{x\alpha + y\beta : x, y \in \mathbf{Z}\}$, then

$$\mathcal{O}_\mathcal{K} = [1, \omega].$$

If $\alpha \in \mathcal{K}$, we use $\overline{\alpha}$ to denote the element of $\mathcal{K}$ that is conjugate to $\alpha$. We also define the trace of $\alpha$ to be $\mathrm{Tr}(\alpha) = \alpha + \overline{\alpha}$ and the norm of $\alpha$ to be $\mathrm{N}(\alpha) = \alpha\overline{\alpha}$. Of course, $\alpha \in \mathcal{O}_\mathcal{K}$ if and only if $\mathrm{Tr}(\alpha) \in \mathbf{Z}$ and $\mathrm{N}(\alpha) \in \mathbf{Z}$.

Now any ideal $\mathcal{A}$ of $\mathcal{O}_\mathcal{K}$ can be written as

$$\mathcal{A} = [a, b + c\omega] \quad (a, b, c \in \mathbf{Z},\ a > 0,\ c \mid b,\ c \mid a,\ ac \mid \mathrm{N}(b + c\omega)). \tag{2.1}$$

The ideal conjugate to $\mathcal{A}$, denoted by $\overline{\mathcal{A}}$, is given by $\overline{\mathcal{A}} = [a, b + c\overline{\omega}]$. Furthermore, if $a, b, c \in \mathbf{Z}$, $c \mid b$, $c \mid a$, $ac \mid \mathrm{N}(b + c\omega)$, then $[a, b + c\omega]$ is an ideal of $\mathcal{O}_\mathcal{K}$. If $|c| = 1$ in (2.1), we say that $\mathcal{A}$ is a primitive ideal.

In the sequel, we will confine our attention to primitive ideals. For such an ideal $\mathcal{A}$ with $a > 0$, the norm of $\mathcal{A}$, denoted by $\mathrm{N}(\mathcal{A})$, has the same value as $a$.

A primitive ideal $\mathcal{A}$ is said to be *reduced* if it does not contain any non-zero element $\alpha$ satisfying

$$|\alpha| < \mathrm{N}(\mathcal{A}), \quad |\overline{\alpha}| < \mathrm{N}(\mathcal{A}).$$

Since $\mathrm{N}(\mathcal{A}) = \mathrm{N}(\overline{\mathcal{A}})$, it is easy to show that if $\mathcal{A}$ is a reduced ideal of $\mathcal{O}_\mathcal{K}$, then so is $\overline{\mathcal{A}}$. With the above definition of reduction, it is possible to prove (see, for example, [40]):

**Theorem 2.1.** *$\mathcal{A}$ is a reduced ideal of $\mathcal{O}_\mathcal{K}$ if and only if there exists some $\beta \in \mathcal{A}$ such that $\mathcal{A} = [\mathrm{N}(\mathcal{A}), \beta]$ with $\beta > \mathrm{N}(\mathcal{A})$ and $-\mathrm{N}(\mathcal{A}) < \overline{\beta} < 0$.*

This result has two useful corollaries:

**Corollary 2.1.** *If $\mathcal{A}$ is a reduced ideal of $\mathcal{O}_\mathcal{K}$, then $\mathrm{N}(\mathcal{A}) < \sqrt{\Delta}$.*

Unfortunately, this condition for reduction is only a necessary one and not sufficient. For sufficiency, we need:

**Corollary 2.2.** *If $\mathcal{A}$ is a primitive ideal of $\mathcal{O}_\mathcal{K}$ and $\mathrm{N}(\mathcal{A}) < \frac{1}{2}\sqrt{\Delta}$, then $\mathcal{A}$ is a reduced ideal of $\mathcal{O}_\mathcal{K}$.*

In $\mathcal{A} = [\mathrm{N}(\mathcal{A}), b + \omega]$, we may assume that $0 < b < \mathrm{N}(\mathcal{A})$, so we see by Corollary 2.2 that there can only be a finite number of reduced ideals of $\mathcal{O}_\mathcal{K}$.

Now let $\mathcal{A} = [\mathrm{N}(\mathcal{A}), b + \omega]$ be any reduced ideal of $\mathcal{O}_\mathcal{K}$. By Theorem 2.1, we may assume that

$$b + \omega > \mathrm{N}(\mathcal{A}), \quad -\mathrm{N}(\mathcal{A}) < b + \overline{\omega} < 0.$$

Let $q = [(b + \omega)/\mathrm{N}(\mathcal{A})]$ and put $c = q\mathrm{N}(\mathcal{A}) - b$. Then $\mathcal{A} = [\mathrm{N}(\mathcal{A}), c - \omega]$ and

$$(c - \overline{\omega})\mathcal{A} = (\mathrm{N}(\mathcal{A}))\mathcal{B}, \tag{2.2}$$

where $\mathcal{B}$ is the ideal $[-\mathrm{N}(c-\omega)/\mathrm{N}(\mathcal{A}), c-\overline{\omega}]$. Since $q = (b+\omega)/\mathrm{N}(\mathcal{A})-\eta$ with $0 < \eta < 1$, we have

$$c-\omega = -\eta \mathrm{N}(\mathcal{A}) < 0. \tag{2.3}$$

Also,

$$c-\overline{\omega} = q\mathrm{N}(\mathcal{A}) - b - \overline{\omega} > q\mathrm{N}(\mathcal{A}) > 0. \tag{2.4}$$

Thus, $\mathrm{N}(\mathcal{B}) = -\mathrm{N}(c-\omega)/\mathrm{N}(\mathcal{A})$ and $\mathcal{B} = [\mathrm{N}(\mathcal{B}), b'+\omega]$, where $b' = c - \mathrm{Tr}(\omega)$. By (2.3), $\mathrm{N}(\mathcal{B}) = -(c-\omega)(b'+\omega)/\mathrm{N}(\mathcal{A}) = \eta(b'+\omega) < b'+\omega$ and by (2.4)

$$\mathrm{N}(\mathcal{B}) = -\frac{(b'+\overline{\omega})(c-\overline{\omega})}{\mathrm{N}(\mathcal{A})} > -(b'+\overline{\omega}).$$

Hence $\mathcal{B}$ is a reduced ideal by Theorem 2.1.

If we put $Q = \sigma\mathrm{N}(\mathcal{A})$, $P - \sqrt{d} = \sigma(b'+\omega)$, $Q' = \sigma\mathrm{N}(\mathcal{B})$, $P' + \sqrt{d} = \sigma(b'+\omega)$, it is easy to show that

$$P' = qQ - P, \quad Q' = \frac{d - P'^2}{Q}.$$

These, of course, are the familiar formulae used in the expansion of $(P+\sqrt{d})/Q$ into a continued fraction. Thus, if we put $Q_0 = Q, P_0 = P, a_0 = [(P_0 + \sqrt{d})/Q_0]$ and we define

$$P_{n+1} = a_n Q_n - P_n, \quad Q_{n+1} = \frac{d - P_{n+1}^2}{Q_n}, \quad a_{n+1} = \left[\frac{P_{n+1}+\sqrt{d}}{Q_{n+1}}\right] \quad (n = 0, 1, 2, \ldots),$$

then the ideals

$$\mathcal{A}_{k+1} = \left[\frac{Q_k}{\sigma}, \frac{P_k + \sqrt{d}}{\sigma}\right]$$

are all reduced and by (2.2) are all equivalent.

However, as pointed out in [40], the continued fraction expansion of $(b+\omega)/\mathrm{N}(\mathcal{A})$ not only provides a technique for finding some reduced ideals equivalent to a given reduced ideal $\mathcal{A} = [\mathrm{N}(\mathcal{A}), b+\omega]$, but actually produces *all* such ideals. Thus, if $\mathcal{B} \sim \mathcal{A}$ and $\mathcal{B}$ is a reduced ideal, then $\mathcal{B} = \mathcal{A}_m$ for some $m \geq 1$. Further, as there are only a finite number of such ideals, we find that for some minimal $k$ $(> 0)$, we get $\mathcal{A}_{k+1} = \mathcal{A}_1 = \mathcal{A}$. The value of $k$ is the period length of the continued fraction expansion of $(b+\omega)/\mathrm{N}(\mathcal{A})$, but we see here that it also represents the exact number of reduced ideals in the class containing $\mathcal{A}$. We call the set of reduced ideals

$$\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \ldots, \mathcal{A}_k \quad (\mathcal{A}_{k+1} = \mathcal{A}_1)$$

a cycle of reduced ideals and we call the value of $k$ the *period length of the cycle.*

If $\mathcal{A}$ is any primitive ideal of $\mathcal{O}_\mathcal{K}$, then we can write $\mathcal{A} = [\mathrm{N}(\mathcal{A}), \beta]$ for some $\beta \in \mathcal{O}_\mathcal{K}$. Let

$$\Gamma(\mathcal{A}) = \{\gamma \in \mathcal{A} : \gamma \neq 0,\ |\gamma| < \mathrm{N}(\mathcal{A}),\ |\overline{\gamma}| < \mathrm{N}(\mathcal{A})\}.$$

For any $\gamma \in \mathcal{A}$, we have $\gamma = x\mathrm{N}(\mathcal{A}) + y\beta$ with $x, y \in \mathbf{Z}$. By solving $\gamma = x\mathrm{N}(\mathcal{A}) + y\beta$ and $\overline{\gamma} = x\mathrm{N}(\mathcal{A}) + y\overline{\beta}$ for $x$ and $y$, we can bound the values of $|x|$, $|y|$ by values that depend

only on $\mathrm{N}(\mathcal{A})$ and $\beta$. Hence, $|\Gamma(\mathcal{A})|$ is finite. Of course, if $\mathcal{A}$ is a reduced ideal, then $|\Gamma(\mathcal{A})| = 0$. In fact, if $|\Gamma(\mathcal{A})| \neq 0$, then $\mathcal{A}$ contains an element of the form $t\mathrm{N}(\mathcal{A}) \pm \beta$ with $t \in \mathbf{Z}$. For let $\gamma = x\mathrm{N}(\mathcal{A}) + y\beta \in \Gamma(\mathcal{A})$ and suppose $|y| \geq 2$. Define $l$ by $l \equiv x \pmod{|y|}$ and $|l| \leq \frac{1}{2}|y|$. Then $(\gamma - l)/y \in \mathcal{A}$ and

$$\left|\frac{\gamma - l}{y}\right| \leq \left|\frac{\gamma}{y}\right| + \frac{1}{2} \leq \frac{\mathrm{N}(\mathcal{A}) + 1}{2} \leq \mathrm{N}(\mathcal{A})$$
$$\left|\frac{\overline{\gamma} - l}{y}\right| \leq \left|\frac{\overline{\gamma}}{y}\right| + \frac{1}{2} \leq \mathrm{N}(\mathcal{A})$$

Since $(\gamma - l)/y$ has the form $t\mathrm{N}(\mathcal{A}) \pm \beta$, we have our result.

Thus if $\mathcal{A}$ is not reduced, there exists some $\lambda \in \Gamma(\mathcal{A})$ such that $\lambda = t\mathrm{N}(\mathcal{A}) \pm \beta$ and $\mathcal{A} = [\mathrm{N}(\mathcal{A}), \lambda]$. If we define the primitive ideal

$$\mathcal{B} = \left[\frac{|\mathrm{N}(\lambda)|}{\mathrm{N}(\mathcal{A})}, \overline{\lambda}\right],$$

we have

$$(\overline{\lambda})\mathcal{A} = (\mathrm{N}(\mathcal{A}))\mathcal{B}.$$

Suppose $\mu \in \Gamma(\mathcal{B})$. We must have some $\gamma \in \mathcal{A}$ such that $\overline{\lambda}\nu = \mathrm{N}(\mathcal{A})\mu$. That is,

$$|\nu| = \frac{\mathrm{N}(\mathcal{A})|\mu|}{|\overline{\lambda}|} < \frac{\mathrm{N}(\mathcal{A})\mathrm{N}(\mathcal{B})}{|\overline{\lambda}|} = \left|\frac{\mathrm{N}(\lambda)}{\overline{\lambda}}\right| = |\lambda|$$

and similarly $|\overline{\nu}| < |\overline{\lambda}|$. It follows that $|\Gamma(\mathcal{B})| < |\Gamma(\mathcal{A})|$. Since $\mathcal{B} \sim \mathcal{A}$, we see by repeating the above process that we must ultimately produce some ideal $\mathcal{C}$ such that $\mathcal{C} \sim \mathcal{A}$ and $\Gamma(\mathcal{C}) = 0$. Thus there is always at least one reduced ideal in every class. From this, we see that if $k_i$ is the period length of the cycle of reduced ideals in the $i$-th class, then

$$\sum_{i=1}^{h} k_i$$

is the total number of reduced ideals in $\mathcal{O}_{\mathcal{K}}$. We can also put this in the following way: there are exactly $h$ distinct cycles of reduced ideals of $\mathcal{O}_{\mathcal{K}}$. Here, $h = h(d)$ is the *class number* of the field $\mathcal{K} = \mathbf{Q}(\sqrt{d})$.

Next, we look at some of the analytic results concerning $h$. If $\epsilon$ $(> 1)$ is the fundamental unit of $\mathcal{K}$, then $R = \log \epsilon$ is the regulator of $\mathcal{K}$ and the analytic class number formula asserts that

$$2hR = \sqrt{\Delta}L(1, \chi_\Delta) \tag{2.5}$$

where

$$L(1, \chi_\Delta) = \sum_{n=1}^{\infty} \left(\frac{\Delta}{n}\right)\frac{1}{n}$$

and $\left(\frac{\cdot}{n}\right)$ is the Kronecker symbol. To obtain a lower bound on $h$, it is necessary to find an upper bound on $R$ and a lower bound on $L(1, \chi_\Delta)$. The former problem can

be easily dealt with for certain values of $d$; however, the latter problem is very difficult indeed. The best result at present is that of Tatuzawa [4] who showed that if $0 < \eta < \frac{1}{2}$ and $\Delta \geq \max(e^{1/\eta}, e^{11\cdot 2})$, then

$$L(1, \chi_\Delta) > 0{\cdot}655\eta\Delta^{-\eta}$$

with at most one possible exceptional value of $\Delta$. A completely effective lower bound on $L(1, \chi_\Delta)$ can also be derived by using Oesterlé's version of the Chebotarev density theorem; unfortunately, this result depends on the truth of the as yet unproved Extended Riemann Hypothesis for real quadratic fields. (See Mollin and Williams [31] for further details.)

Finally, since we will refer to the Extended Riemann Hypothesis (*ERH*) in the next section, we give a brief elucidation of it here. The Riemann zeta function is defined by

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s}$$

for a complex variable $s$ with $\mathrm{re}(s) > 1$ and extended by analytic continuation to the whole complex plane. Riemann's Hypothesis (1859) says that if $s$ is any non-trivial zero of $\zeta(s)$ (that is, $s \neq -2, -4, -6, \ldots$) then the real part of $s$, $\mathrm{re}(s)$, must be $\frac{1}{2}$. Because of the functional equation satisfied by $\zeta(s)$, it is known that all such zeros must lie in the 'critical strip', $0 < \mathrm{re}(s) < 1$ and that the zeros are symmetrically located about the real axis and about the 'critical line', $\mathrm{re}(s) = \frac{1}{2}$. Hence the Riemann Hypothesis can be reformulated as: $\zeta(s) \neq 0$ for any value of $s$ such that $\mathrm{re}(s) > \frac{1}{2}$.

Now, the Dirichlet $L$-function is defined by

$$L(s, \chi) = \sum_{n=1}^{\infty} \chi(n) n^{-s}$$

where $s$ is a complex variable. The *ERH* says that $L(s, \chi) \neq 0$ for any value of $s$ such that $\mathrm{re}(s) > \frac{1}{2}$. (See [31] for a more detailed description.)

## 3. Gauss's class number one problem for quadratic fields.

Gauss's class number one problem for complex quadratic fields has been settled for some time. The early part of this century saw the first significant progress toward a proof of the conjecture that there are only finitely many such fields—actually Gauss suspected that $\mathbf{Q}(\sqrt{-163})$ was the last. In 1918, Landau [12] published a result, attributed to Hecke, which showed that Gauss's conjecture for complex quadratic fields follows from *ERH*. Almost 50 years later, Baker [1] and Stark [38] (anticipated by Heegner [8]) established (unconditionally) that there are exactly 9 complex quadratic fields $\mathcal{K} = \mathbf{Q}(\sqrt{-d})$ with $h(-d) = 1$, the last being with $d = 163$ as Gauss suspected. For an excellent detailed account of the solution of this problem, as well as the solution of Gauss's general class number problem for complex quadratic fields, see Goldfeld [6]. (Goldfeld, Gross and Zagier won the 1987 Cole Prize in number theory for this work in which they gave an effective bound for the discriminants of all complex quadratic fields with a given class number.)

Now we turn to recent advances concening Gauss's conjecture that there are infinitely many real quadratic fields with class number one. It should be pointed out that, unlike the complex case, very little is known. In point of fact, we do not yet know whether there are infinitely many number fields with class number one.

Recent advances concerning a certain restricted set of real quadratic fields closely parallels the solution of the class number one problem for complex quadratic fields, so we outline it here.There are several conjectures in the literature which motivated much of this research. We list them in chronological order.

**Conjecture 3.1** (S. Chowla, 1976). *If $p = m^2 + 1$ is prime and $m > 26$ then $h(p) > 1$.*

**Conjecture 3.2** (H. Yokoi, 1986). *If $n = m^2 + 4$ is square-free and $m > 17$ then $h(n) > 1$.*

**Conjecture 3.3** (R. A. Mollin, 1987). *If $n = m^2 - 4$ is square-free and $m > 21$ then $h(n) > 1$.*

**Conjecture 3.4** (R. A. Mollin and H. C. Williams, 1987). *If $n = m^2 \pm 2$ is square-free and $m > 20$ then $h(n) > 1$.*

**Conjecture 3.5** (R. A. Mollin and H. C. Williams, 1987). *If $n = m^2 - p$ is square-free, where $p$ is an odd prime dividing $m$ and $m > 42$, then $h(n) > 1$.*

**Conjecture 3.6** (R. A. Mollin and H. C. Williams, 1987). *If $n = m^2 \pm 4l$ is square-free, where $l > 1$ is an odd integer dividing $m$ and $m > 39$, then $h(n) > 1$.*

What do the forms in conjectures 3.1 to 3.6 have in common? They are all of *extended Richaud-Degert type* (*ERD*-type), that is, $n = m^2 + r$ where $r$ divides $4m$. From results of R. A. Mollin [23], [24], it follows that if $h(n) = 1$ and $n$ is of *ERD*-type, then $n$ must be one of the forms in conjectures 3.1 to 3.6. In [26], [27], Mollin and Williams were able to determine all *ERD*-types with class number one under the assumption of the *ERH*. Subsequently, they were able to drop the *ERH* assumption in [28] and prove:

**Theorem 3.1.** *If $n$ is of ERD-type, then $h(n) = 1$ if and only if $n$ is an element of the set*

$$\{2, 3, 5, 6, 7, 11, 13, 14, 17, 21, 23, 29, 33, 37, 38, 47, 53, 62, 69, 77, 83, 93, 101, 141, 167, 173, 197, 213, 227, 237, 293, 398, 413, 437, 453, 573, 677, 717, 1077, 1133, 1253, 1283, 1757\},$$

*with possibly only one more value remaining.*

*Remark 3.1.* Theorem 3.1 thus proves that 5 of the 6 conjectures 3.1 to 3.6 are true (but we do not know which ones) and the remaining one fails for at most one value. Moreover, given [26] and [27], if the exceptional value exists, then is provides a counterexample to the Riemann Hypothesis.

*Remark 3.2.* In a recent letter, Zhang Ming-yao of the People's Republic of China stated the following result which he has proved: If $p = 4N^2 + 1 > 677$ is prime and $h(p) = 1$ then $p > \exp(8{\cdot}8 \times 10^7)$. At the time of writing, the above result has not yet been published. As a consequence of Theorem 3.1, if the Chowla conjecture fails, then it fails for exactly one value $p > \exp(8{\cdot}8 \times 10^7)$.

*Remark 3.3.* Thus the situation for real quadratic fields of *ERD*-type of class number one is at exactly the same position as was the class number one problem for complex quadratic fields before the Baker-Stark-Heegner solution. However, to eliminate the possibility of one more value in Theorem 3.1 appears to be extremely difficult at this time.

Now we turn to a different approach to the class number one problem for real quadratic fields. Throughout, $d$ is assumed to be positive and square-free. Recall the notation on continued fractions from Section 2. The following table illustrates that when $d$ is of *ERD*-type and $h(d) = 1$, then $k \leq 4$.

| $k$ | $d$ | |
|---|---|---|
| 1 | $(2l+1)^2+1$ | $(l \geq 0)$ |
| 3 | $4l^2+1$ | $(l \geq 2)$ |
| 2 | $4l^2-1$ | $(l \geq 1)$ |
| 2 | $l^2+2$ | $(l \geq 1)$ |
| 4 | $l^2-2$ | $(l \geq 3)$ |
| 1 | $(2l+1)^2+4$ | $(l \geq 0)$ |
| 2 | $(2l+1)^2-4$ | $(l \geq 2)$ |
| 2 | $(lm)^2+4l$ | $(l \geq 1)$ |
| 4 | $(2lm)^2-l$ | $(l \geq 1)$ |
| 4 | $(lm)^2-4l$ | $(l \geq 1)$ |

Table 3.1.

*Remark 3.4.* In [25], we pushed the computational and number-theoretic techniques so that we can now list all real quadratic fields of class number one and $k \leq 24$, (with possibly only one more value remaining). The largest such value of $d$ is 49013 where $k = 20$. Our calculations suggest that the number of $d$ with $h(d) = 1$ and with $\omega$ of period $k$ tends to infinity as $k \to \infty$. This then is a reformulation of the Gauss conjecture.

We were also able to show

**Theorem 3.2.** *For a fixed period $k$, there are at most finitely many $d$ with $h(d) = 1$.*

Moreover, for the special case where $d \equiv 1 \pmod 8$, we were able to prove that those $d$ we found with $h(d) = 1$ and $k \leq 24$ are precisely the ones with no exceptional value possible. This case was settled by means of the following result from [25] derived from the theory of reduced ideals.

**Theorem 3.3.** *If the prime $p$ splits in $\mathbf{Q}(\sqrt{d})$ and $\Delta > 4p^{k+1}$, then $h(d) > 1$.*

*Remark 3.5.* Since 2 splits in $\mathbf{Q}(\sqrt{d})$ for $d \equiv 1 \pmod 8$, then $h(d) > 1$ when $d > 2^{k+3}$. Since we checked all $d$'s up to $2^{31} - 1$ on a computer and we were interested in $k \leq 24$, we then have the result.

We conclude this section with a list of known class number one criteria.

(I) $h(d) = 1$ if and only if for all non-zero $\alpha, \beta \in \mathcal{O}_K$ with $\beta$ not dividing $\alpha$ and $|\mathrm{N}(\alpha)| \geq |\mathrm{N}(\beta)|$, there exist $\sigma, \delta \in \mathcal{O}_K$ with $0 < |\mathrm{N}(\alpha\sigma - \beta\delta)| < |\mathrm{N}(\beta)|$.

*Remark 3.6.* (I) is attributed to Dedekind and Hasse (see Pollard [33]). However, it seems to have been rediscovered by Kutsuna. (I) is virtually useless as a class number one test.

(II) $h(d) = 1$ if and only if whenever a prime $p$ divides

$$f_d(x) = \begin{cases} -x^2 + x + \dfrac{d-1}{4} & \text{if } d \equiv 1 \pmod 4 \\ -x^2 + d & \text{if } d \not\equiv 1 \pmod 4 \end{cases}$$

for $0 \leq x \leq \frac{1}{2}\sqrt{d}$ and $p < \frac{1}{2}\sqrt{d}$, then $p = Q_i/Q_0$ for some $i$ with $0 < i < k$.

*Remark 3.7.* (II) is also well-known (see, for example, Hendy [10]). However, (II) also seems to have been rediscovered by Louboutin in [14] where the result is stated as a real quadratic field analogue of the Rabinowitsch result for complex quadratic fields. (See Section 4 for a discussion.)

III) (H. Lu [16].) $h(d) = 1$ if and only if $\sum_{i=1}^{k} a_i + \theta = \lambda_1(d) + \lambda_2(d)$, where $\lambda_1(d)$ and $\lambda_2(d)$ are respectively the numbers of solutions of the diophantine equations $x^2 + 4yz = \Delta$ $(x, y, z \geq 0)$ and $x^2 + 4y^2 = \Delta$ $(x, y \geq 0)$ and

$$\theta = \begin{cases} 0 & \text{if } d \equiv 1 \pmod 4,\ k \text{ even},\ k = 2n,\ a_n \text{ odd} \\ 1 & \text{if } d \equiv 1 \pmod 4,\ k \text{ even},\ k = 2n,\ a_n \text{ even} \\ 1 & \text{if } d \equiv 1 \pmod 4,\ k \text{ odd} \\ 1 & \text{if } d \not\equiv 1 \pmod 4,\ k \text{ even},\ k = 2n,\ a_n \text{ odd} \\ 2 & \text{if } d \not\equiv 1 \pmod 4,\ k \text{ even},\ k = 2n,\ a_n \text{ even} \\ 2 & \text{if } d \not\equiv 1 \pmod 4,\ k \text{ odd} \end{cases}$$

For example, consider the special case where $d \equiv 1 \pmod 4$ is prime and $k$ is odd. Set $\alpha = \frac{1}{2}\sqrt{d-1}$. Then $h(d) = 1$ if and only if

$$\sum_{i=1}^{k} a_i = \sum_{x=1}^{\alpha} \tau(f_d(x))$$

where $\tau$ counts the number of divisors and $f_d(x)$ is as in (II). (III) is a useful criterion which we were able to utilise in our search for a precise real quadratic field analogue of the Rabinowitsch result which we discuss in the next section.

## 4. Rabinowitsch conditions and prime-valued polynomials.

**Theorem 4.1** (Rabinowitsch [34]). *Let $d \equiv 3 \pmod 4$ be positive and square-free. Then $x^2 - x + (d+1)/4$ is prime for all integers $x$ with $1 \leq x \leq (d-3)/4$ if and only if $h(-d) = 1$.*

In an effort to understand the Chowla Conjecture 3.1 better, Mollin discovered the following in [20].

**Theorem 4.2.** *Let $n = 4m^2 + 1$ be square-free. Then the following are equivalent:*
(1) $h(n) = 1$;
(2) *$p$ is inert in $\mathbf{Q}(\sqrt{n})$ for all primes $p < m$;*
(3) $f(x) = -x^2 + x + m^2 \not\equiv 0 \pmod{p}$ *for all integers $x$ and all primes $p$ satisfying $0 < x < p < m$;*
(4) *$f(x)$ is prime for all integers $x$ with $1 < x < m$.*

*Remark 4.1.* The equivalence of Theorems 4.2 (1) and (4) was also independently discovered by H. Yokoi [42].

Subsequently, we were able to generalise Theorem 4.2 in [27]. We maintain $f_d(x)$ as in Section 3 and set

$$\alpha = \begin{cases} \frac{1}{2}\sqrt{d-1} & \text{if } d \equiv 1 \pmod{4} \\ \sqrt{d} & \text{if } d \not\equiv 1 \pmod{4} \end{cases}$$

and we label the following statements for reference.
*Conditions*:

(I) $p$ is inert in $\mathbf{Q}(\sqrt{d})$ for all primes $p < \alpha$;

(II) $f_d(x) \not\equiv 0 \pmod{p}$ for all integers $x$ and primes $p$ such that $0 \leq x < p < \alpha$;

(III) $f_d(x)$ is prime for all integers $x$ with $1 < x < \alpha$;

(IV) $h(d) = 1$.

**Theorem 4.3** (Mollin and Williams [27]). (I) $\leftrightarrow$ (II) $\rightarrow$ (III) $\rightarrow$ (IV). *Additionally, if $d \equiv 1 \pmod{4}$ then* (III) $\rightarrow$ (II).

Although this seems like a general Rabinowitsch-like result, we discovered in [27]

**Theorem 4.4.** *If $d \equiv 1 \pmod{4}$ then* (III) *holds only if $d = m^2 + r$ with $|r| \in \{1, 4\}$ (called narrow ERD-types).*

In point of fact, Mollin and Williams showed

**Theorem 4.5.** (III) $\leftrightarrow$ (IV) *if and only if*

$$d \in \{2, 3, 5, 6, 7, 11, 13, 17, 21, 29, 37, 53, 77, 101, 173, 197, 293, 437, 677\}$$

*with possibly only one value remaining.*

*Remark 4.2.* Other polynomials may be used but insisting on consecutive initial prime values seems to force *ERD*-types. For example,

**Theorem 4.6** (Mollin and Williams [32]).
(a) *If $d = 2p$ where $p \equiv 3 \pmod{4}$ is prime and $-2x^2 + p$ is 1 or prime for all integers $x$ with $0 \leq x < \frac{1}{2}\sqrt{d}$, then $h(d) = 1$.*
(b) *If $p \equiv 3 \pmod{4}$ is prime and $-2x^2 + 2x + \frac{1}{2}(p-1)$ is 1 or prime for all integers $x$ with $0 < x < \frac{1}{2}(\sqrt{p} + 1)$, then $h(p) = 1$.*
(c) *If $p \equiv 1 \pmod{4}$ is prime and $px^2 + px + \frac{1}{4}(p-1)$ is 1 or prime for all integers $x$ with $0 < x < \frac{1}{2}(\sqrt{p-1} - 1)$, then $h(p) = 1$.*
(d) *If $d = pq$ where $p < q$, $p, q \equiv 3 \pmod{4}$ are primes and $d \equiv 5 \pmod{8}$, then*

$|px^2 + px + \frac{1}{4}(p-q)|$ *is prime or* 1 *for all integers* $x$ *with* $0 \leq x < \frac{1}{4}\sqrt{d-1} - \frac{1}{2}$ *only if* $h(d) = 1$.

**Conjecture 4.1.** *If any of the hypotheses of Theorem 4.6 hold, then* $d$ *is of ERD-type.*

Since we posed this Conjecture 4.1 in [32], Louboutin [15] has settled it affirmatively for all but the case of Theorem 4.6(d).

*Remark 4.3.* If $p = 199$ in 4.6(a), we get the polynomial $f_d(x) = -2x^2 + 199$ claimed by Karst to 'supplant' Euler's polynomial $x^2 - x + 41$ in terms of initial prime-producing capacity. However, $|f(10)| = 1$ whereas $x^2 - x + 41$ is prime for all $x$ with $1 \leq x \leq 40$. However, we have found polynomials which do exceed Euler's polynomial in terms of the number of initial distinct primes produced. Gilbert Fung (a graduate student of H. C. Williams) found $f_d(x) = 47x^2 - 1701x + 10181$ with $d = 979373$ and $|f_d(x)|$ prime for all $x$ with $0 \leq x \leq 42$, so we have 43 initial consecutive distinct primes. Also, Russell Ruby of Oregon State University recently informed the first author that he found $36x^2 - 810x + 2753$ with $d = 259668 = 2^2 \cdot 3^2 \cdot 7213$ and this is prime for $0 \leq x \leq 44$ yielding 45 initial consecutive distinct primes. Here are some further examples.

| $d$ | $a$ | $b$ | $c$ | Number of distinct initial primes |
|---|---|---|---|---|
| 21188 | 8 | −298 | 2113 | 40 |
| 21188 | 8 | −326 | 2659 | 40 |
| 1398053 | 59 | −1873 | 8941 | 40 |
| 4978797 | 111 | −3123 | 10753 | 40 |
| 979373 | 47 | −1965 | 15329 | 40 |
| 1398053 | 59 | −2729 | 25633 | 40 |
| 259668 | 36 | −522 | 89 | 41 |
| 979373 | 47 | −1513 | 6967 | 41 |
| 1398053 | 103 | −3533 | 26903 | 41 |
| 259668 | 36 | −594 | 647 | 42 |
| 979373 | 47 | −1607 | 8527 | 42 |
| 259668 | 36 | −666 | 1277 | 43 |
| 979373 | 47 | −1701 | 1018 | 43 |
| 259668 | 36 | −738 | 1979 | 44 |
| 259668 | 36 | −810 | 2753 | 45 |

Table 4.1.

Now we return to the Rabinowitsch criteria. We were searching for a precise prescription for the factorisation of $f_d(x)$ which is tantamount to class number one. We did this by pushing the algebraic techniques to the limit in [30]. The following result shows that such a prescription comes at the expense of simplicity. There may be an algorithm for the general case, but it eludes us at this point in time. Perron has given the parametrisation for general $d$ (as in the special case that follows), but the general prescription for factorisation is not clear.

**Theorem 4.7.** *Suppose* $d \equiv 1 \pmod 4$ *is square-free,* $\omega = \langle a, \overline{b, c, b, 2a-1} \rangle$ *is represented by a continued fraction expansion of period* 4, $d = (2a-1)^2 + 4(c(fb-c)+f)$ *and* $2a-1 = b^2cf - bc^2 - c + 2bf$ *for some positive integers* $a, b, c$ *and* $f$. *Then* $h(d) = 1$ *if and only if the following conditions* (1)—(6) *all hold.*

(1) $b(fb-c)+1$ *is prime,*

(2) $c(fb-c)+f$ *is prime,*

(3) $f_d(x)/(b(fb-c)+1) = -x^2 - x + \frac{1}{4}(d-1)$ *is* 1 *or prime for all integers* $x$ *with* $0 \le x \le a-1$ *and* $x \equiv -2^{-1} \pmod{b(fb-c)+1}$,

(4) $f_d(x)/(c(fb-c)+f)$ *is prime for all integers* $x$ *satisfying* $0 \le x \le a-1$ *and* $x \equiv -2^{-1}(fb-c+1) \pmod{c(fb-c)+f}$,

(5) $f_d(x)/(c(fb-c)+f)$ *is* 1 *or prime for all integers* $x$ *with* $0 \le x \le a-1$ *and* $x \equiv 2^{-1}(fb-c+1) \pmod{c(fb-c)+f}$,

(6) $f_d(x)$ *is prime for all integers* $x$ *with* $0 \le x \le a-1$ *and* $x$ *is not congruent to* $-2^{-1}(fb-c+1) \pmod{c(fb-c)+f}$, $2^{-1}(fb-c+1) \pmod{c(fb-c)+f}$ *and* $-2^{-1} \pmod{b(fb-c)+1}$.

The following result is useful in determining certain $h(d) = 1$ from continued fraction techniques.

**Theorem 4.8.** *Let* $d \equiv 1 \pmod 4$, $\omega = \langle a, \overline{a_1, a_2, \ldots, a_{k-1}, 2a-1} \rangle$ *and* $f_d(x) = -x^2 + x + \frac{1}{4}(d-1)$. *There are exactly* $a_i + 1$ *values of* $x$ *with* $1 \le x \le a$ *and* $0 < i < k$ *such that* $f_d(x) \equiv 0 \pmod{Q_i}$.

*Application.* Let $d \equiv 1 \pmod 8$ and $\omega = \langle a, \overline{b, c, b, c, \ldots, b, 2a-1} \rangle$ where '$b, c$' is repeated $\frac{1}{2}k - 1$ times and $k$ is even. Then $h(d) = 1$ if and only if $d = 33$. To get this we use the fact that $f_d(x)$ is even for all $x$ when $d \equiv 1 \pmod 8$ and, since 2 splits in $\mathbf{Q}(\sqrt{d})$, then $Q_i = 2$ for some $i$ when $h(d) = 1$. Thus $a_i = a - 1$.

*Remark 4.4.* The following was of interest to other authors including L. Bernstein where the forms for which all $Q_i$'s are powers of 2 were investigated. However, Bernstein was interested in the continued fraction expansion of $\sqrt{d}$ only and was interested in using these results to compute the fundamental unit. We are interested in the continued fraction expansion of $\frac{1}{2}(1 + \sqrt{d})$ for $d \equiv 1 \pmod 4$ and in its connection with the class number one problem. Our goal is to classify those $d \equiv 1 \pmod{4p}$ for which the $Q_i$ are all powers of a single prime $p$.

**Theorem 4.9.** *If* $d \equiv 1 \pmod{4p}$ *then all* $Q_i$*'s are powers of* $p$ *if and only if* $d = (2a-1)^2 + 4p^s$ *with* $s \ge 0$, $2a - 1 = p^s a_1 + g$ *where* $0 \le g < p^s$, $s \equiv 0 \pmod{\frac{1}{2}(k-1)}$ *for* $k \ge 1$ *and odd, and* $a_1 g + 1 = p^{2s/(k-1)}$ *when* $k > 1$ *and* 1 *when* $k = 1$.

*Example 4.1.* (1) $d = 494317$ is prime; all $Q_i$'s are powers of 3 and the $\mathcal{O}_K$ primes above 3 are not principal. (2) $d = 60997 = 181 \cdot 337$; all $Q_i$'s are powers of 3 and the $\mathcal{O}_K$ primes above 3 are principal. However $h(d) > 1$. (3) $d = 61$; all $Q_i$'s are powers of 3 and $h(d) = 1$.

**Corollary 4.1.** *If* $p$ *does not divide* $a$ *and all* $\mathcal{O}_K$ *primes above* $p$ *are principal then all* $Q_i$*'s are powers of* $p$ *if and only if* $d = (2a-1)^2 + 4p^s = (p^{s+1} - p^s + p)^2 + 4p^s$ *and* $k = 2s + 1$. *Moreover,* $[\sqrt{d}] = 2a$ *if and only if* $p = 2$.

**Corollary 4.2.** *Suppose* $d \equiv 1 \pmod{4p}$. *If* $p \mid a$ *and all the primes in* $\mathcal{O}_K$ *above* $p$ *are principal, then all* $Q_i$*'s are powers of* $p$ *if and only if* $d = (2a-1)^2 + 4p^s = (p^s + p - 1)^{2s} + 4p^s$ *and* $k = 2s+1$, $s \geq 1$. *Moreover,* $[\sqrt{d}] = 2a$.

**Conjecture 4.1.** *If* $d \equiv 1 \pmod 8$ *and all* $Q_i$*'s are powers of* 2 *then* $h(d) = 1$ *if and only if* $d \in \{17, 41, 113, 353, 1217\}$.

*Problems.* (1) Find a precise prescription for factorisation (over $\mathbf{Z}$) of $f_d(x)$ equivalent to $h(d) = 1$ and so provide a real quadratic field analogue of the Rabinowitsch result. (2) Approach the Gauss problem by using the ideas elucidated herein to construct a list of forms $d$ (say for $d \equiv 1 \pmod 8$) such that $h(d) = 1$ for period $k$ as $k$ increases.

## 5. Higher class number and a criterion for $h(d) = 2$.

**Proposition 5.1.** *If* $d = m^2 + r$ *with* $|r| \in \{1, 4\}$ *then* $p$ *is inert in* $\mathbf{Q}(\sqrt{d})$ *for all primes* $p$ *with* $p^{h(d)} < \alpha$ ($\alpha$ *as in section* 4) *unless* $h(d)$ *is even in which case* $p$ *may be ramified.*

*Applications.* When $d \equiv 1 \pmod 8$ then $d$ of narrow *ERD*-type means $d = 4m^2 + 1$. Since 2 splits in $\mathbf{Q}(\sqrt{d})$, then Proposition 5.1 implies that $2^{h(d)} \geq m$. Hence it is easy to check that

$h(d) = 1$ if and only if $d = 17$,

$h(d) = 2$ if and only if $d = 65$,

$h(d) = 3$ if and only if $d = 257$,

$h(d) = 4$ if and only if $d = 145$.

**Proposition 5.2.** (I) $\rightarrow$ (II) $\leftrightarrow$ (III) *where*

(I) $f_d(x)$ *is a product of at most* $h(d)$ *primes for all integers* $x$ *with* $1 < x < \alpha$;

(II) $p$ *is inert for all primes* $p$ *with* $p^{h(d)} < \alpha$;

III) $f_d(x) \not\equiv 0 \pmod{p^{h(d)}}$ *for all integers* $x$ *and primes* $p$ *with* $0 < x < p^{h(d)} < \alpha$.

Now we give a general class number 2 criterion for real quadratic fields which is much in the flavour of Section 3, criterion (II) for $h(d) = 1$. Refer to Section 2 for the notation.

**Theorem 5.1.** *If* $d \not\equiv 5 \pmod 8$ *and* $f_d(x) \neq 2$ *for any* $x$ *then* $p = 2$. *Otherwise, we let* $p$ *be the least odd prime such that both* $(d/p) = 1$ *and* $f_d(x) \neq p$ *for any* $x$. *Then* $h(d) = 2$ *if and only if whenever* $q \mid f_d(x)$ *for any prime* $q < \frac{1}{2}\sqrt{\Delta}$ *and* $1 \leq x \leq \frac{1}{2}\sqrt{\Delta}$, *then* $q = Q_i/\sigma$ *in either the continued fraction expansion of* $\omega$ *or* $(b_p + \omega)/p$ *where* $\mathcal{P} = [p, b_p + \omega]$ *is an* $\mathcal{O}_K$ *ideal above* $p$.

*Example 5.1.* Let $d = 395 = 5 \cdot 79 = 20^2 - 5$ of *ERD*-type with $h(d) = 2$. The expansion of $\omega = \sqrt{395}$ begins

| $i$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $P_i$ | 0 | 19 | 15 | 15 |
| $Q_i$ | 1 | 34 | 5 | 34 |
| $a_i$ | 19 | 1 | 6 | ... |

so $k = 4$. The expansion of $\frac{1}{2}(1 + \sqrt{d})$ begins

| $i$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $P_i$ | 1 | 19 | 15 | 15 |
| $Q_i$ | 2 | 17 | 10 | 1 |
| $a_i$ | 10 | 2 | 3 | ... |

again showing that $k = 4$. Values of $f_d(x)$ are

| $x$ | $f_d(x) = 395 - x^2$ |
|---|---|
| 2 | $391 = 17 \cdot 23$ |
| 3 | $386 = 2 \cdot 193$ |
| 4 | 379 |
| 5 | $370 = 2 \cdot 5 \cdot 37$ |
| 6 | 359 |
| 7 | $346 = 2 \cdot 173$ |
| 8 | 331 |
| 9 | $314 = 2 \cdot 157$ |
| 10 | $295 = 5 \cdot 59$ |
| 11 | $274 = 2 \cdot 137$ |
| 12 | 251 |
| 13 | $226 = 2 \cdot 113$ |
| 14 | 199 |
| 15 | $170 = 2 \cdot 5 \cdot 17$ |
| 16 | 139 |
| 17 | $106 = 2 \cdot 53$ |
| 18 | 71 |
| 19 | $34 = 2 \cdot 17$ |

We now look at criteria for $h(d) = 2$ in terms of the factorisation of $f_d(x)$ in the same manner as Hendy [9] accomplished for complex quadratic fields. We can use Theorem 5.1 to do this for *ERD*-types. See [19] for proofs.

**Proposition 5.3.** *Let* $d \equiv 2 \pmod 4$ *and* $d = l^2 + 1$. *Then* $h(d) = 2$ *if and only if* $d - x^2$ *is prime or twice a prime for all* $x$ *with* $1 < x < l$.

Proposition 5.3 has a generalisation of sorts as follows.

**Proposition 5.4.** *Let* $q$ *be a fixed prime dividing* $d$ (*assumed positive and square-free*). *If* $f_d(x)$ *is prime or* $q$ *times a prime for all* $x$ *with* $1 < x < \alpha$ *then* $h(d) \leq 2$.

**Proposition 5.5.** *Let* $d = l^2 + r \equiv 2 \pmod 4$, $r \mid 4l$, $r > 2$ *and* $[\sqrt{d}] = a = l$. *Then* $h(d) = 2$ *if and only if,*

(i) *for* $l$ *odd* $(2 < x < l)$, $d - x^2$ *is prime, twice a prime,* $r$ *times a prime,* $2r$ *times a prime, or the product of two primes* $(l + \frac{1}{2}(r-1))(l - \frac{1}{2}(r-1))$ (*at* $x = \frac{1}{2}(r+1)$), *and* $r$ *is prime.*

(ii) *for* $l$ *even* $(1 < x < l)$, $d - x^2$ *is a prime, twice a prime,* $r$ *times a prime, or* $\frac{1}{2}r$ *times a prime, and* $\frac{1}{2}r$ *is prime.*

**Proposition 5.6.** *Let $d = l^2 + r \equiv 2 \pmod 4$, $r \mid 4l$, $r < 0$ and $[\sqrt{d}] = a = l - 1$. Then $h(d) = 2$ if and only if,*
(i) *for $l$ even $(1 < x < l)$, $d - x^2$ is prime, twice a prime, $-\frac{1}{2}r$ times a prime, $-r$ times a prime, or $\frac{1}{2}(r + 4l - 4)$ times a prime, and $\frac{1}{2}r$ and $\frac{1}{2}(r + 4l - 4)$ are both prime.*
(ii) *for $l$ odd $(1 < x < l)$, $d - x^2$ is a prime, twice a prime, $-r$ times a prime, or $-2r$ times a prime, or a product of two primes $(l + \frac{1}{2}(r-1))(l - \frac{1}{2}(r-1))$ $(at\ x = \frac{1}{2}(r+1))$.*

**Proposition 5.7.** *Let $d = l^2 + r \equiv 2 \pmod 4$, $r < 0$ and $[\sqrt{d}] = a = l$ and $r \neq -2$. Then $h(d) = 2$ if and only if,*
(i) *for $l$ even $(1 < x < l)$, $d - x^2$ is prime, twice a prime, $-\frac{1}{2}r$ times a prime, or $-r$ times a prime, and $-\frac{1}{2}r$ is prime.*
(ii) *for $l$ odd $(1 < x < l)$, $d - x^2$ is a prime, twice a prime, $-r$ times a prime, or $-2r$ times a prime, or a product of two primes $(l + \frac{1}{2}(r-1))(l - \frac{1}{2}(r-1))$ $(at\ x = \frac{1}{2}(r+1))$ and $r$ is prime.*

**Proposition 5.8.** *Let $d = l^2 + r \equiv 3 \pmod 4$, $r > 0$, $r \mid 2l$ and $r \neq 2$. Then $h(d) = 2$ if and only if,*
(i) *for $l$ odd $(1 < x < l)$, $d - x^2$ is prime, twice a prime, $\frac{1}{2}r$ times a prime, or $r$ times a prime for all $x$.*
(ii) *for $l$ even $(1 < x < l)$, $d - x^2$ is a prime, twice a prime, $\frac{1}{2}r$ times a prime, $r$ times a prime, or a product of two primes $(l + \frac{1}{2}(r-1))(l - \frac{1}{2}(r-1))$ $(at\ x = \frac{1}{2}(r+1))$.*

**Proposition 5.9.** *Let $d = l^2 + r \equiv 3 \pmod 4$, $r < 0$, $r \mid 4l$ and $r \neq -2$ and $[\sqrt{d}] = l - 1$. Then $h(d) = 2$ if and only if,*
(i) *for $l$ even $(1 < x < l)$, $d - x^2$ is a prime, twice a prime, $-r$ times a prime, or a product of two primes $(l + \frac{1}{2}(r-1))(l - \frac{1}{2}(r-1))$ $(at\ x = \frac{1}{2}(r+1))$ and $-r$ is a prime.*
(ii) *for $l$ odd $(1 < x < l)$, $d - x^2$ is prime, twice a prime, $-\frac{1}{2}r$ times a prime, or $\frac{1}{2}(r + 4l - 4)$ times a prime and both $-\frac{1}{2}r$ and $\frac{1}{2}(r + 4l - 4)$ are primes.*

**Proposition 5.10.** *Let $d = l^2 + r \equiv 3 \pmod 4$, $r < 0$, $r \mid 4l$ and $r \neq -2$ and $[\sqrt{d}] = l$. Then $h(d) = 2$ if and only if,*
(i) *for $l$ odd $(1 < x < l)$, $d - x^2$ is prime, twice a prime, or $-\frac{1}{2}r$ times a prime, and $-\frac{1}{2}r$ is prime.*
(ii) *for $l$ even $(1 < x < l)$, $d - x^2$ is a prime, twice a prime, $-r$ times a prime, or a product of two primes $(l + \frac{1}{2}(r-1))(l - \frac{1}{2}(r-1))$ $(at\ x = \frac{1}{2}(r+1))$ and $-r$ is prime.*

*Remark 5.1.* The above characterise $h(d) = 2$ for $d \not\equiv 1 \pmod 4$ of *ERD*-type (except for the troublesome forms $d = l^2 + 2$) in terms of the factorisation of $d - x^2$. We now determine all *ERD*-types $d \equiv 1 \pmod 8$ with $h(d) = 2$.

**Theorem 5.2.** *If $d \equiv 1 \pmod 8$ and $d$ is of ERD-type, then $h(d) = 2$ if and only if $d = 65$ or $105$.*

## 6. Miscellaneous related results and summary.

In [17], [18], Mollin used techniques concerned with solutions for diophantine equations to settle certain class number problems for real quadratic fields of *RD*-type. A fundamental tool was the following:

**Lemma 6.1.** *Let $d$ and $t$ be positive integers with $d$ square-free. Suppose that $(A + B\sqrt{d})/\sigma$ is the fundamental unit of $\mathbf{Q}(\sqrt{d})$ and $A^2 - dB^2 = \sigma^2\delta = \pm\sigma^2$. If there exists a non-trivial solution to the diophantine equation $x^2 - dy^2 = \pm\sigma^2 t$ then $t \geq ((2A/\sigma) - \delta - 1)/B^2$.*

This completed an omission in the proof of a theorem of Yokoi ([43], Theorem 3, page 147). Lemma 6.1 was used to prove the following result for $RD$-types in [17].

**Theorem 6.1.** *Let $d = l^2 + r > 7$ be a square-free integer with $r \mid 4l$ and $-l \leq r \leq l$ and either $d \not\equiv 1 \pmod 4$ or $|r| \in \{1, 4\}$. Suppose that there exists a prime $q$ dividing $l$ such that $q < l$ whenever $|r| = 4$. Then $h(d) > 1$ whenever any of the following conditions is satisfied:*

(i) $\gcd(q, r) = 1$, $q > 2$ *and* $(r/q) = 1$. *Moreover, if $r = 1$ and $l$ is even, then $l > 2q$.*
(ii) $q = 2$ *and* $r \neq 1$ *is odd.*
(iii) $q = 2$, $r = 1$, $l \equiv 0 \pmod 4$ *and* $l > 4$.
(iv) $q \mid r$, $|r| > q$ *and* $|r| \neq 4$.
(v) $|r| = q > 2$.

Theorem 6.1 extended the work of Yokoi ([41], Theorem 3, page 157), Hasse [7], S. D. Lang [13] and Mollin [18], [22], among others.

Another connection with the Chowla Conjecture 3.1 and stemming from Mollin's Theorem 4.2 is with Fibonacci numbers. First we need some definitions.

**Definition 6.1.** *Let $g$ and $n$ be positive integers. The $n$-th Fibonacci sequence, $\{F_i(n)\}$, with base $g$ is defined by $F_0(n) = 1$, $F_1(n) = g$ and $F_i(n) = F_{i-1}(n) + nF_{i-2}(n)$ for $i > 1$.*

*Remark 6.1.* The first Fibonacci sequence with base 1 is the ordinary Fibonacci sequence.

**Definition 6.2.** *Let $p$ be a prime and let $g$ be a primitive root modulo $p$. We call $g$ an $n$-th Fibonacci primitive root modulo $p$ if it satisfies*

$$x^2 \equiv x + n \pmod p \quad \text{with} \quad \gcd(p, n) = 1. \tag{$*$}$$

*Remark 6.2.* The case $n = 1$ yields the ordinary Fibonacci primitive roots introduced by Shanks [35] and for which properties were developed in [31] and [37] and generalised by Mollin in [21].

Now we are in a position to state the connection between the Chowla conjecture and these new Fibonacci numbers.

**Theorem 6.3** (Mollin [21]). *Suppose $n$ is a positive integer relatively prime to $p$. Then $g$ is a solution of $(*)$ if and only if the $n$-th Fibonacci sequence with base $g$ satisfies $F_{i+1}F_{i-1} \equiv F_i^2 \pmod p$ for some $i > 1$. Moreover, if $g$ is a solution of $(*)$, then $F_{i+1}F_{i-1} \equiv F_i^2 \pmod p$ for all $i > 0$.*

**Conjecture 6.1.** *If $n = q^2$ where $q > 13$ is an odd prime and $4q^2 + 1$ is prime then there is an $n$-th Fibonacci sequence, $\{F_i(n)\}$, to base $g$ satisfying $F_{i+1}F_{i-1} \equiv F_i^2 \pmod p$ for a prime $p$ with $0 < g < p < q$.*

*Example.* Suppose that $d = 4p + 1 = m^2 + 4$ where $p$ is a prime and $m$ is an odd positive integer. If $s < \sqrt{p}$ is an odd prime, then $p \equiv t \pmod{s}$ for $0 \leq t < s$. If there exists an integer $u > 0$ such that $1 + 4t \equiv (2u - 1)^2 \pmod{s}$, then $-u^2 + u + p \equiv 0 \pmod{s}$ where $0 < u < s < \sqrt{p}$. This violates condition II of Theorem 4.3, so in view of Theorem 4.5, we have $h(d) > 1$.

The connection between Theorem 4.3 (I) and (III) was investigated by Mollin and Williams in [29]. A sample result is:

**Theorem 6.4.** *Let* $d \equiv 2, 3 \pmod{4}$ *and*

$$b = \begin{cases} \sqrt{d/2} & \text{if } d \equiv 2 \pmod{4} \\ \sqrt{(d-1)/2} & \text{if } d \equiv 3 \pmod{4}. \end{cases}$$

*If the Extended Riemann Hypothesis holds and* $(d/p) = -1$ *for all odd primes* $p < b$, *then* $d$ *is an entry in either Table* 6.1 *or Table* 6.2 *below.*

| $d$ | $f_d(x) = -2x^2 + \frac{1}{2}d$ for $0 \leq x < \frac{1}{2}\sqrt{d}$ |
|---|---|
| 6 | 3, 1 |
| 10 | 5, 3 |
| 14 | 7, 5 |
| 26 | 13, 11, 5 |
| 38 | 19, 17, 11, 1 |
| 62 | 31, 29, 23, 13 |
| 122 | 61, 59, 53, 43, 29, 11 |
| 362 | 181, 179, 173, 163, 149, 131, 109, 83, 53, 19 |
| 398 | 199, 197, 191, 181, 167, 149, 127, 101, 71, 37 |

Table 6.1.

| $d$ | $f_d(x) = -2x^2 + +2x + \frac{1}{2}(d-1)$ for $0 \leq x < \frac{1}{2}\sqrt{(d-1)}$ |
|---|---|
| 3 | – |
| 7 | 3 |
| 11 | 7 |
| 23 | 11, 7 |
| 35 | 17, 13 |
| 47 | 23, 19, 11 |
| 83 | 41, 27, 29 , 17 |
| 143 | 71, 67, 59, 47, 31 |
| 167 | 83, 79, 71, 59, 43, 23 |
| 227 | 113, 109, 101, 89, 73, 53, 29 |

Table 6.2.

In summary, the topics discussed herein in connection with Gauss's conjectures are of interest in their own right and may shed light upon a difficult problem. However, the solution of Gauss's conjecture for real quadratic fields seems a long way off. There is much work yet to be done.

## References


1. A. Baker, 'Linear forms in the logarithms of algebraic numbers', *Mathematika* **13** (1966), 204–216.
2. L. Bernstein, 'Fundamental units and cycles', *J. Number Theory* **8** (1976), 446–491.
3. L. Bernstein, 'Fundamental units and cycles in the period of real quadratic number fields, Part II', *Pacific J. Math.* **63** (1976), 63–78.
4. H. Cohn, *A second course in number theory.* (Wiley, New York, 1962.)
5. G. Degert, 'Uber die Bestimmung der Grundeinheit gewisser reell-quadratischer Zahlkörper' *Abh. Math. Sem. Univ. Hamburg* **22** (1958), 92–97.
6. D. Goldfeld, 'Gauss' class number problems for imaginary quadratic fields', *Bull. Amer. Math. Soc.* **13** (1985), 23–37.
7. H. Hasse, 'Uber mehrklassige, aber eingeschlechtige reell-quadratische Zahlkörper', *Elem. Math.* **20** (1965), 49–59.
8. K. Heegner, 'Diophantische Analysis und Modulfunktionen', *Math. Z.* **56** (1952), 227–253.
9. M. D. Hendy, 'Prime quadratics associated with complex quadratic fields of class number two', *Proc. Amer. Math. Soc.* **43** (1974), 253–260.
10. M. D. Hendy, 'Applications of a continued fraction algorithm to some class number problems', *Math. Comp.* **28** (1974), 267–277.
11. E. L. Ince, 'Cycles of reduced ideals in quadratic fields', *Math. Tables* **IV** (Brit. Assoc. Adv. Sci., London, 1939).
12. E. Landau, 'Uber die Klassenzahl imaginar-quadratischen Zahlkörper', *Gottingen Nach.*, (1918), 285–295.
13. S. D. Lang, Note on the class number of the maximal real subfield of a cyclotomic field', *J. reine angew. Math.* **290**, (1977), 70–72.
14. S. Louboutin, 'Continued fractions and real quadratic fields', *J. Number Theory* **30** (1988), 167–176.
15. S. Louboutin, 'Prime producing quadratic polynomials and class numbers of real quadratic fields', *Canad. J. Math.* (to appear).
16. H. Lu, 'On the class number of real quadratic fields', *Sci. Sinica*, Special issue (II) (1979), 118–130.
17. R. A. Mollin, 'On the insolvability of a class of diophantine equations and the nontriviality of the class numbers of related quadratic fields of Richaud-Degert type', *Nagoya Math. J.* **105** (1987), 39–47.
18. R. A. Mollin, 'Diophantine equations and class numbers', *J. Number Theory* **24** (1986), 7–19.
19. R. A. Mollin, 'Class number two criterion and real quadratic fields' (to appear).
20. R. A. Mollin, 'Necessary and sufficient conditions for the class number of a real quadratic field to be one, and a conjecture of S. Chowla', *Proc. Amer. Math. Soc.* **102** (1988), 17–21.

21. R. A. Mollin, 'Generalised Fibonacci primitive roots, and class numbers of real quadratic fields', *Fib. Quarterly* **26** (1988), 46–53.

22. R. A. Mollin, 'Lower bounds for class numbers of real quadratic fields and biquadratic fields', *Proc. Amer. Math. Soc.* **101** (1987), 439–444.

23. R. A. Mollin, 'Class number one criteria for real quadratic fields I' *Proc. Japan Acad., Ser. A* **63** (1987), 121–125.

24. R. A. Mollin, 'Class number one criteria for real quadratic fields II' *Proc. Japan Acad., Ser. A* **63** (1987), 162–164.

25. R. A. Mollin and H. C. Williams, 'Real quadratic fields of class number one and related continued fractions' (to appear).

26. R. A. Mollin and H. C. Williams, 'Prime-producing quadratic polynomials and real quadratic fields of class number one', Proc. International Conf. on Number Theory (Quebec, 1987), (to appear).

27. R. A. Mollin and H. C. Williams, 'On prime-valued polynomials and class number of real quadratic fields', *Nagoya Math. J.* **112** (1988), 143–151.

28. R. A. Mollin and H. C. Williams, 'Solution of the class number one problem for real quadratic fields of extended Richaud-Degert type (with one possible exception)', *Proc. First Conf. Canadian Number Theory Assoc.* (Banff, 1988) (to appear).

29. R. A. Mollin and H. C. Williams, 'Quadratic non-residues and prime-producing polynomials', *Canad. Math. Bull.* (to appear).

30. R. A. Mollin and H. C. Williams, 'Real quadratic fields of class number one and continued fraction period less than six', *C. R. Math. Rep. Acad. Sci. Canada* **XI** (1989), 51–56.

31. R. A. Mollin and H. C. Williams, 'Computation of the class numbers of a real quadratic field', *Advances in the theory of computing and comp. math.* (to appear).

32. R. A. Mollin and H. C. Williams, 'Class number one for real quadratic fields, continued fractions and reduced ideals'. R. A. Mollin (ed.), *Number Theory and Applications* (Kluwer Academic Publishers, 1989), 481–496.

33. H. Pollard, *Algebraic numbers* (Carus monograph **9**, American Mathematical Society, 1950).

34. G. Rabinowitsch, 'Eindeutigkeit der Zerlegung in Primzahlfaktorin in quadratischen Zahlkörpern', *Proc. fifth international congress math. (Cambridge)* **I** (1913), 418–421.

35. D. Shanks, *Solved and unsolved problems in number theory.* (2nd edition, Chelsea, New York, 1978.)

36. D. Shanks, 'Fibonacci primitive roots', *Fibonacci Quarterly* **10** (1972), 163–168, 181.

37. D. Shanks and L. Taylor, 'An observation on Fibonacci primitive roots', *Fibonacci Quarterly* **11** (1973), 159–160.

38. H. Stark 'A complete determination of the complex quadratic fields of class number one', *Michigan Math. J.* **14** (1967), 1–27.

39. T. Tatuzawa, 'On a theorem of Siegel', *Japan J. Math.* **21** (1951), 163–178.
40. H. C. Williams and M. C. Wunderlich, 'On the parallel generation of the residues for the continued fraction factoring algorithm', *Math. Comp* **48** (1987), 405–423.
41. H. Yokoi, 'On the diophantine equation $x^2 - py^2 = \pm 4q$ and the class number of real quadratic subfields of a cyclotomic field', *Nagoya Math. J.* **91** (1983), 151–161.
42. H. Yokoi, 'Class number one problem for certain kind of real quadratic fields', *Proc. International Conf. on class numbers and fundamental units of algebraic number fields* (Kattata, Japan, June 24-28, 1986).
43. H. Yokoi, 'On real quadratic fields containing units with norm $-1$', *Nagoya Math. J.* **33** (1988), 139–152.

*Department of Mathematics and Statistics, University of Calgary, Calgary, Alberta, CANADA T2N 1N4.*

*Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, CANADA R3T 2N2*

# 17

# NUMBER THEORETIC PROBLEMS INVOLVING TWO INDEPENDENT BASES

**Teturo Kamae**

Let $r$ and $s$ be integers not less than 2 which are multiplicatively independent, that is, $\log_r s$ is irrational. For $n \in \mathbf{N}$, $\mathbf{N}$ being the set of nonnegative integers, let

$$n = \sum_{i=0}^{\infty} n_i r^i \quad (n_i \in \{0, 1, \ldots, r-1\})$$

be the expansion of $n$ to base $r$. Also, for $x \in [0, 1)$, let

$$x = \sum_{i=1}^{\infty} x_i r^{-i} \quad (x_i \in \{0, 1, \ldots, r-1\})$$

be the expansion of $x$ to base $r$. We are interested in some problems which involve the expansions of $n$ or $x$ to base $r$ and $s$ at the same time.

I. Is the sequence $(\{3^n/2^n\} : n = 1, 2, \ldots)$ uniformly distributed?

II. Let $\alpha : \mathbf{N} \to \{-1, 1\}$ be the *Morse sequence*. That is,

$$\alpha(n) = (-1)^{n_0 + n_1 + \cdots},$$

where the $n_i$'s are the digits of $n$ to base 2. Is it true that

$$\lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N-1} \alpha(3^n) = 0?$$

III. (See [1].) Let

$$\alpha(n) = \xi^{n_0 + n_1 + \cdots} \quad (n \in \mathbf{N}),$$

where the $n_i$'s are the digits of $n$ to base $r$ and $\xi$ is a complex number with $|\xi| = 1$ and $\xi^{r-1} \neq 1$. Let

$$\beta(n) = \eta^{n_0 + n_1 + \cdots} \quad (n \in \mathbf{N}),$$

where in this case the $n_i$'s are the digits of $n$ to base $s$ and $\eta$ is a complex number with $|\eta| = 1$ and $\eta^{s-1} \neq 1$. Then, the spectral measures on [0,1) related to $\alpha$ and $\beta$

are continuous but singular with respect to the Lebesgue measure on [0,1). Moreover, they are mutually singular.

IV. (W. Schmidt [2].) Let $B_r$ be the set of *r-normal numbers*. That is,

$$B_r = \{x \in [0,1) : (\{r^n x\} : n = 0,1,2,\ldots) \text{ is uniformly distributed}\}.$$

Then $B_r \setminus B_s$ has the cardinality of the continuum.

In this report, we prove two results related to problems I and II based on the fact that the multiplicative order of 3 modulo $2^k$ is $2^{k-2}$ for any $k \geq 3$. We also give a simple proof for the essential part of III in a special case that there exist prime numbers $p$ and $q$ such that $p \mid r$, $p \nmid s$, $q \mid s$ and $q \nmid r$. In fact, IV follows from this proof.

Since the multiplicative order of 3 modulo $2^k$ is $2^{k-2}$ for any $k \geq 3$, the sequence of fractional parts $(\{3^n/2^k\} : n = 1,2,\ldots)$ is a periodic sequence with period $2^{k-2}$ and the points in a period are distinct. So, exactly $\frac{1}{4}$ of the $2^k$ points of the form $i/2^k$ $(i = 0,1,\ldots,2^k-1)$ appear in the sequence. Moreover, the set of points in the sequence is invariant under multiplication by 3 modulo 1. Since 'almost all' numbers $i \in \{0,1,\ldots,2^k-1\}$ are 'random' in some sense if we look at the first $k$ digits to base 2, then 'almost all' numbers $3^n$ $(n = 1,2,\ldots,2^{k-2})$ are 'random' if we look at the first $k$ digits to base 2. These observations lead to Theorems 1 and 2 stated below. Theorem 1 might be well known, but I do not know a reference.

**Theorem 1.** *The sequence* $(\{3^n/2^{\beta(n)}\} : n = 1,2,\ldots)$ *is uniformly distributed if* $\beta(n)$ *is a nondecreasing sequence of positive integers such that* $\beta(n) \to \infty$ *as* $n \to \infty$ *and* $2^{\beta(n)} = O(n)$.

*Remark.* If $\beta(n) = n$, the assertion of Theorem 1 is still an open problem. I do not even know whether the theorem is true with the condition $\beta(n) = O(\log n)$ instead of $2^{\beta(n)} = O(n)$.

*Proof.* Suppose to the contrary that the sequence $(\{3^n/2^{\beta(n)}\} : n = 1,2,\ldots)$ is not uniformly distributed. Then there exists an infinite set $S$ of positive integers such that

$$\mu = \operatorname*{wlim}_{\substack{N\to\infty \\ N\in S}} \frac{1}{N} \sum_{n=1}^{N} \delta_{\{3^n/2^{\beta(n)}\}}$$

exists and is not equal to the Lebesgue measure $\lambda$ on [0,1], where $\delta_x$ is the unit measure at $x$ and wlim implies the weak convergence of Borel measures.

Since $\beta(n) = o(n)$, the set of $n$ such that $\beta(n) \neq \beta(n+1)$ has density 0. Therefore,

$$\begin{aligned}
\mu \circ T_3^{-1} &= \operatorname*{wlim}_{\substack{N\to\infty \\ N\in S}} \frac{1}{N} \sum_{n=1}^{N} \delta_{\{3^n/2^{\beta(n)}\}} \circ T_3^{-1} \\
&= \operatorname*{wlim}_{\substack{N\to\infty \\ N\in S}} \frac{1}{N} \sum_{n=1}^{N} \delta_{\{3^{n+1}/2^{\beta(n)}\}} \\
&= \operatorname*{wlim}_{\substack{N\to\infty \\ N\in S}} \frac{1}{N} \sum_{n=1}^{N} \delta_{\{3^{n+1}/2^{\beta(n+1)}\}} = \mu,
\end{aligned}$$

where $T_3$ is multiplication by 3 modulo 1. Hence $\mu$ is $T_3$-invariant.

Let

$$\Gamma(n) = \#\{k : \beta(k) = \beta(n)\},$$

and for any $\epsilon > 0$, let

$$\Lambda_\epsilon = \{n : \Gamma(n)/2^{\beta(n)} < \epsilon\}.$$

Then we have

$$\#(\Lambda_\epsilon \cap [1, N]) \leq \sum_{i=1}^{\beta(N)} \epsilon 2^i < 2\epsilon \cdot 2^{\beta(N)} \leq 2\epsilon CN,$$

where $C$ is a constant such that

$$2^{\beta(N)} \leq CN \quad (N = 1, 2, \ldots).$$

There exists an infinite subset $S'$ of $S$ such that

$$\mu' = \operatorname*{wlim}_{\substack{N\to\infty \\ N\in S'}} \frac{1}{N} \sum_{\substack{n=1 \\ n\notin\Lambda_\epsilon}}^{N} \delta_{\{3^n/2^{\beta(n)}\}}$$

exists. Then, it is easy to see that the total variation of the measure $\mu - \mu'$ is less than $2\epsilon C$. Take $a, b \in [0, 1]$ with $a < b$. Then, we have

$$\begin{aligned}\mu'((a,b)) &\leq \limsup_{N\to\infty} \frac{1}{N} \sum_{\substack{n=1 \\ n\notin\Lambda_\epsilon}}^{N} \mathbf{1}_{\{3^n/2^{\beta(n)}\}\in(a,b)} \\ &= \limsup_{N\to\infty} \frac{1}{N} \sum_{\substack{n=N_0 \\ n\notin\Lambda_\epsilon}}^{N} \mathbf{1}_{\{3^n/2^{\beta(n)}\}\in(a,b)},\end{aligned}$$

where $N_0$ is a positive integer such that $\beta(N_0) \geq 3$ and

$$2^{\beta(N_0)}(b - a) \geq 1.$$

Let $\{\beta_1 < \beta_2 < \ldots < \beta_M\}$ be the set of all $\beta(n)$'s such that $N_0 \leq n \leq N$ and $n \notin \Lambda_\epsilon$. Let

$$\Xi_m = \{n : \beta(n) = \beta_m\}.$$

Then we have

$$\sum_{\substack{n=N_0 \\ n\notin\Lambda_\epsilon}}^{N} \mathbf{1}_{\{3^n/2^{\beta(n)}\}\in(a,b)} = \sum_{m=1}^{M} \sum_{n\in\Xi_m} \mathbf{1}_{\{3^n/2^{\beta_m}\}\in(a,b)}.$$

Since the maximum multiplicity of the value $\{3^n/2^{\beta_m}\}$ for $n \in \Xi_m$ is not bigger than

$$\frac{\#\Xi_m}{2^{\beta_m - 2}} + 1,$$

we have

$$\sum_{n\in\Xi_m} 1_{\{3^n/2^{\beta_m}\}\in(a,b)} \le \left(\frac{\#\Xi_m}{2^{\beta_m-2}}+1\right)\left(2^{\beta_m}(b-a)+1\right)$$
$$\le \left(\frac{\#\Xi_m}{2^{\beta_m-2}}+\frac{\#\Xi_m}{\epsilon 2^{\beta_m}}\right)2\cdot 2^{\beta_m}(b-a)$$
$$= \left(8+\frac{2}{\epsilon}\right)(b-a)\#\Xi_m.$$

Hence

$$\mu'((a,b)) \le \left(8+\frac{2}{\epsilon}\right)(b-a)$$

for any $a, b \in [0,1]$ with $a < b$. This implies that $\mu'$ is absolutely continuous with respect to $\lambda$. Since $\mu$ is the limit in the total variation norm of these $\mu'$, $\mu$ is absolutely continuous with respect to $\lambda$. Since $\mu$ is a $T_3$-invariant probability measure which is absolutely continuous with respect to $\lambda$, $\mu = \lambda$ holds by the ergodicity of $\lambda$ with respect to $T_3$. Thus we have a contradiction which proves the theorem.

Let

$$\alpha(n) = (-1)^{n_0+n_1+\cdots} \quad (n \in \mathbf{N})$$

be the Morse sequence, where the $n_i$'s are the digits of $n$ to base 2. Then, it is easy to see that

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}\alpha(n) = 0,$$

so that $\alpha(n) = \pm 1$ with the same frequency $\frac{1}{2}$. It is not known whether or not

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}\alpha(\beta(n)) = 0$$

for $\beta(n)$ equal to the $n+1$-st prime, $\beta(n) = n^2$, or $\beta(n) = 3^n$. However, we can answer the following related problem in the affirmative.

We imbed the Morse sequence in a dynamical system as an orbital function. Let $\mathbf{Z}_2$ be the set of 2-adic integers. That is,

$$\mathbf{Z}_2 = \{0,1\}^{\mathbf{N}}$$

with the additive group structure. An element of $\mathbf{Z}_2$ is written as $\omega = (\omega_0, \omega_1, \omega_2, \ldots)$ with $\omega_i \in \{0,1\}$. Let $\omega, \eta \in \mathbf{Z}_2$. Then $\omega + \eta \in \mathbf{Z}_2$ is defined by

$$\sum_{i=0}^{N-1}(\omega+\eta)_i 2^i \equiv \sum_{i=0}^{N-1}\omega_i 2^i + \sum_{i=0}^{N-1}\eta_i 2^i \pmod{2^N}$$

for any $N$. We consider $n \in \mathbf{N}$ as an element $(n_0, n_1, n_2, \ldots)$ of $\mathbf{Z}_2$ so that $\mathbf{Z}$ is embedded in $\mathbf{Z}_2$.

For $\omega \in \mathbf{Z}_2$ such that $\omega \neq (1,1,1,\ldots)$, we define

$$\phi(\omega) = (-1)^{\lim_{N\to\infty}(\sum_{i=0}^{N}(\omega+1)_i - \sum_{i=0}^{N}\omega_i)}.$$

Note that

$$\prod_{i=0}^{n-1}\phi(\omega+i) = (-1)^{\lim_{N\to\infty}(\sum_{i=0}^{N}(\omega+n)_i - \sum_{i=0}^{N}\omega_i)}$$

if the left hand side is defined. We abbreviate the exponent of $-1$ in the right hand side as

$$\sum(\omega+n)_i - \sum\omega_i.$$

The normalised Haar measure on $\mathbf{Z}_2$ is denoted by $d\omega$ and that on the multiplicative group $\{-1,1\}$ is denoted by $dg$. We define a measure preserving transformation $T$ on $\mathbf{Z}_2 \times \{-1,1\}$ with respect to the probability measure $d\omega \times dg$ by

$$T(\omega,g) = (\omega+1, g\phi(\omega)) \quad (\omega \in \mathbf{Z}_2, g \in \{-1,1\}).$$

Note that $T$ is defined at almost all points.

Let $\pi(\omega,g) = g$. The Morse sequence can be written as

$$\alpha(n) = \pi(T^n(0,1)) \quad (n \in \mathbf{N}).$$

In this sense, $\alpha$ is imbedded in the dynamical system $(\mathbf{Z}_2 \times \{-1,1\}, d\omega \times dg, T)$.

**Theorem 2.** *Suppose $\beta$ is strictly increasing with at most polynomial order, or $\beta(n) = 3^n$. Then*

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}\pi(T^{\beta(n)}(\omega,g)) = 0$$

*for almost all $\omega$ and all $g$.*

*Remark.* We prove this in the case $\beta(n) = 3^n$. In the general case, there are exactly $N$ different numbers which can be extracted from the $\beta(n)$'s $(n = 0,1,\ldots,N-1)$ if we look only at the first $m = [\log_2 \beta(n)]$ digits to base 2. Since $N \geq a2^{bm}$ for some positive constants $a$ and $b$, we can apply the same argument as in the proof of Theorem 3.

*Proof.* By a standard technique in probability theory, it is sufficient to show that

$$\int\left|\frac{1}{N}\sum_{n=0}^{N-1}\pi(T^{3^n}(\omega,g))\right|^2 d\omega = O((\log N)^{-1-\epsilon})$$

for some $\epsilon > 0$. We have

$$\begin{aligned}\int\left|\frac{1}{N}\sum_{n=0}^{N-1}\pi(T^{3^n}(\omega,g))\right|^2 d\omega &= \frac{1}{N^2}\sum_{n,m=0}^{N-1}\int(-1)^{\sum(\omega+3^n)_i - \sum(\omega+3^m)_i}\,d\omega \\ &= \frac{1}{N^2}\sum_{n,m=0}^{N-1}\int(-1)^{\sum(\omega+3^n-3^m)_i - \sum\omega_i}\,d\omega \\ &\leq \frac{1}{N^2}\sum_{n,m=0}^{N-1}\left|\int(-1)^{\sum(\omega+3^n-3^m)_i - \sum\omega_i}\,d\omega\right|.\end{aligned}$$

For $n \in \mathbf{N}$, let

$$\tau(n) = \#\{i : n_{2i} = 0 \text{ and } n_{2i+1} = 1\}.$$

Then it is proved in [1] that there exists $\delta$ with $0 < \delta < 1$ such that

$$\left| \int (-1)^{\sum(\omega+n)_i - \sum \omega_i} \, d\omega \right| < \delta^{\tau(|n|)}.$$

Let $k = [\log_2 N]$. Then, by the large deviation principle, there exists $\epsilon > 0$ such that

$$\#\left\{0 \le i < 2^k : \tau(3^n - i) < \frac{k}{9}\right\} = O(2^{(1-\epsilon)k})$$

holds uniformly in $n$ as $N \to \infty$. Therefore

$$\begin{aligned}\#\left\{0 \le m < N : \tau(3^n - 3^m) < \frac{k}{9}\right\} &\le \#\left\{0 \le i < 2^k : \tau(3^n - i) < \frac{k}{9}\right\} \times \left(\frac{N}{2^{k-2}} + 1\right) \\ &= O(N^{1-\epsilon})\end{aligned}$$

uniformly in $n$. Hence

$$\sum_{n,m=0}^{N-1} \left| \int (-1)^{\sum(\omega+3^n-3^m)_i - \sum \omega_i} \, d\omega \right| \le N(N - O(N^{1-\epsilon}))\delta^{\frac{k}{9}} + NO(N^{1-\epsilon}),$$

which implies that

$$\int \left| \frac{1}{N} \sum_{n=0}^{N-1} \pi(T^{3^n}(\omega, g)) \right|^2 d\omega = O((\log N)^{-1-\epsilon})$$

for some $\epsilon > 0$. This completes the proof.

Now let $\xi$ be a complex number such that $|\xi| = 1$ and $\xi^{r-1} \neq 1$. Let $T$ be a measure preserving transformation on $\mathbf{Z}_r \times G$, $G$ being the closure of $\{\xi^n : n \in \mathbf{Z}\}$, such that

$$T(\omega, g) = (\omega + 1, \, g\xi^{\sum(\omega+1)_i - \sum \omega_i}) \quad (\omega \in \mathbf{Z}_r, g \in G).$$

Let $\pi(\omega, g) = g$ and let $\Lambda$ be the spectral measure of $\pi$ with respect to $T$. That is, $\Lambda$ is a Borel measure on $[0, 1)$ such that

$$\int e^{2\pi i n x} \, d\Lambda(x) = \int \pi(T^n(\omega, g)) \overline{\pi(\omega, g)} \, d\omega \, dg$$

for any $n \in \mathbf{Z}$. It is clear that $\Lambda$ is a probability measure by considering the case $n = 0$ in this equality.

It is known [1] that $\Lambda$ is a continuous but singular measure for which

$$\Lambda(B_r) = 0 \text{ and } \Lambda(B_s) = 1.$$

This implies that $B_s \setminus B_r$ has the cardinalty of the continuum. Let $\Lambda'$ be a similar spectral measure with respect to the base $s$. Then, since

$$\Lambda'(B_s) = 0 \text{ and } \Lambda'(B_r) = 1,$$

$\Lambda$ and $\Lambda'$ are mutually singular. Here, we give a simple proof for $\Lambda(B_s) = 1$ in the special case in which there is a prime $p$ such that $p \mid r$ and $p \nmid s$.

**Theorem 3.** *Suppose there is a prime number $p$ such that $p \mid r$ and $p \nmid s$. Then $\Lambda(B_s) = 1$.*

*Proof.* It is known [2] that the multiplicative order of $s$ modulo $r^n$ is greater than $ar^{bn}$ where $a$ and $b$ are positive constants independent of $n$. By the Weyl criterion, it is sufficient to prove that for any $l \in \mathbf{Z} \setminus \{0\}$,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i l s^n x} = 0$$

holds for $\Lambda$-almost all $x \in [0,1)$. For the same reason as in the proof of Theorem 2, it is sufficient to prove that

$$\int \left| \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i l s^n x} \right|^2 d\Lambda(x) = O((\log N)^{-1-\epsilon})$$

for any $l \in \mathbf{Z} \setminus \{0\}$ with some $\epsilon > 0$. Now,

$$\begin{aligned}
\int \left| \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i l s^n x} \right|^2 d\Lambda(x) &= \int \left| \frac{1}{N} \sum_{n=0}^{N-1} \pi(T^{ls^n}(\omega, g)) \right|^2 d\omega\, dg \\
&= \frac{1}{N^2} \sum_{n,m=0}^{N-1} \int \xi^{\sum(\omega + ls^n - ls^m)_i - \sum \omega_i}\, d\omega \\
&\le \frac{1}{N^2} \sum_{n,m=0}^{N-1} \left| \int \xi^{\sum(\omega + ls^n - ls^m)_i - \sum \omega_i}\, d\omega \right| .
\end{aligned}$$

For $n \in \mathbf{N}$ with $n_i$'s as its digits to base $r$, let $\tau(n)$ be the maximum number $k$ such that there exists $i_1 < i_2 < \ldots < i_{2k}$ satisfying $n_{i_j} > 0$ for odd $j$'s and $n_{i_j} < r - 1$ for even $j$'s. Then it is proved in [2] that

$$\left| \int \xi^{\sum(\omega+n)_i - \sum \omega_i}\, d\omega \right| < \delta^{\tau(|n|)}$$

for some constant $\delta$ with $0 < \delta < 1$ which is independent of $n \in \mathbf{Z}$.

Suppose $k \to \infty$ and choose $\eta$ with $\eta \asymp (\log k)^2$. Then we have the following estimate:

$$\#\{0 \le n < r^k : \tau(n) < \tfrac{1}{2} k\eta \le k^{k\eta} r^{k\eta} = r^{k\phi(k)},$$

where

$$\phi(k) = \eta + \eta \log_r k = o(1).$$

Therefore, there exists $\epsilon > 0$ such that

$$\phi(k) - b < -\epsilon.$$

Let $k = [\log_r N]$. Then, as $N \to \infty$, we have

$$\#\left\{0 \le m < N : \left| \int \xi^{\sum(\omega + ls^n - ls^m)_i - \sum \omega_i} \, d\omega \right| \ge \delta^{\frac{k\eta}{2}} \right\} \le r^{k\phi(k)} \left( \frac{N|l|}{ar^{bk}} + 1 \right)$$
$$\le O(N^{1-\epsilon})$$

uniformly in $n$. Thus

$$\int \left| \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i l s^n x} \right|^2 d\Lambda(x) \le \delta^{\frac{k\eta}{2}} + \frac{1}{N} O(N^{1-\epsilon}) \le O((\log N)^{-1-\epsilon})$$

for some $\epsilon > 0$, which completes the proof.

## References

1. T. Kamae, 'Cyclic extensions of odometer transformations and spectral disjoint ness', *Israel J. Math.* **56** (1987), 41–63.
2. W. Schmidt, 'On normal numbers', *Pacific J. Math.* **10** (1960), 661–672.

*Department of Mathematics, Osaka City University, Sugimoto-cho, Osaka, 558 JAPAN.*

# A CLASS OF NORMAL NUMBERS II

Y.-N. Nakai and I. Shiokawa

## 1. Introduction.

Let $r \geq 2$ be a fixed integer and let $0{\cdot}a_1a_2a_3\ldots = a_1r^{-1} + a_2r^{-2} + a_3r^{-3} + \cdots$ be the $r$-adic expansion of a real number $\theta$ $(0 < \theta < 1)$. Then $\theta$ is said to be normal to base $r$ if, for any block $b_1 \ldots b_l \in \{0, 1, \ldots, r-1\}^l$,

$$\frac{1}{n}N_r(\theta; b_1 \ldots b_l; n) = r^{-l} + o(1)$$

as $n \to \infty$, where $N_r(\theta; b_1 \ldots b_l; n)$ is the number of indices $i \leq n - l + 1$ such that $a_i = b_1, a_{i+1} = b_2, \ldots, a_{i+l-1} = b_l$. Various kinds of constructions of normal numbers have been found. However, most of them are very complicated and by no means easy to write down (see, for example, [4]). One of the simplest algorithms which gives normal numbers is the following: Let $\mathbf{Q}[x]$ denote the set of polynomials in $x$ with rational coefficients. Let $f(x) \in \mathbf{Q}[x]$ with $1 \leq f(n) \in \mathbf{Z}$ $(n = 1, 2, \ldots)$. Then Davenport and Erdős [1] proved that the decimal $0{\cdot}f(1)f(2)f(3)\ldots$ is normal to base 10, where each $f(n)$ is written in the scale of 10 and the digits of $f(1)$ are succeeded by those of $f(2)$, and so on.

A pseudo-polynomial with real coefficients is a function $g(x)$ of the form

$$g(x) = \alpha x^{\beta} + \alpha_1 x^{\beta_1} + \cdots + \alpha_d x^{\beta_d}, \tag{1}$$

where $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_d$ are nonzero real numbers and $\beta = \beta_0 > \beta_1 > \ldots > \beta_d \geq 0$. In this paper, we always assume that $g(x) > 0$ for all $x \geq 0$. The set of all such pseudo-polynomials will be denoted by $\mathcal{R}$. For each $g(x) \in \mathcal{R}$, we define the number

$$\theta_r = \theta_r(g) = 0{\cdot}[g(1)][g(2)][g(3)]\ldots,$$

to be the infinite $r$-adic decimal $0{\cdot}a_{11}a_{12}\ldots a_{1k_1}a_{21}a_{22}\ldots a_{2k_2}a_{31}\ldots$, obtained from the $r$-adic expansion $[g(n)] = a_{n1}a_{n2}\ldots a_{nk_n} = a_{n1}r^{k_n-1} + a_{n2}r^{k_n-2} + \cdots + a_{nk_n}$ of the integral part of $g(n)$.

In [5], we proved the normality to base $r$ of the number $\theta_r(g)$, when $g(x) \in \mathcal{R}\backslash\mathbf{R}[x]$, that is when at least one of the exponents $\beta, \beta_1 \ldots, \beta_d$ in (1) is not an integer. Here $\mathbf{R}[x]$ denotes the set of polynomials with real coefficients. In the present paper, we shall prove that the number $\theta_r(g)$ is normal to base $r$ for any $g(x) \in \mathbf{R}[x]$. By combining this with our results from [5], we see that $\theta_r(g)$ is normal to base $r$ for all $g(x) \in \mathcal{R}$.

In particular, the number $\theta_r = 0{\cdot}[\alpha][\alpha 2^\beta][\alpha 3^\beta]\ldots$ is normal to base $r$ for all $\alpha > 0$ and $\beta > 0$.

More precisely, we obtained in [5] the following estimate: For any $g(x) \in \mathcal{R} \setminus \mathbf{R}[x]$ and any $b_1 \ldots b_l \in \{0, 1, \ldots, r-1\}^l$, we have

$$R_n = \frac{1}{n} N_r(\theta_r(g); b_1 \ldots b_l; n) - r^{-l} = O\left(\frac{1}{\log n}\right).$$

To prove this, we used tricky estimates for exponential sums of the Vinogradov type. For $g(x) \in \mathbf{Q}[x]$, Schiffer [6] showed that $R_n = O(1/\log n)$.

Let $\sigma$ be a finite string of $r$-adic digits and let $N_r(\sigma; b_1 \ldots b_l)$ denote the number of occurrences of the block $b_1 \ldots b_l \in \{0, 1, \ldots, r-1\}^l$ in the string $\sigma$. In this paper, we prove the following theorem.

**Theorem.** *For any $g(x) \in \mathbf{R}[x]$ and any block $b_1 \ldots b_l \in \{0, 1, \ldots, r-1\}^l$, we have*

$$\sum_{n \le x} N_r([g(n)]; b_1 \ldots b_l) = r^{-l} x \log_r g(x) + O(x \log\log x)$$

*as $x \to \infty$, where the implied constant depends only on the $\alpha$'s, $\beta$'s, $r$ and $l$.*

As an immediate consequence, we have

**Corollary.** *For any $g(x)$ and $b_1 \ldots b_l$ as in the theorem, we have*

$$R_n = O\left(\frac{\log\log n}{\log n}\right)$$

*as $n \to \infty$. In particular, the number $\theta_r(g)$ is normal to base $r$.*

Our method of proof, which is different from that of Schiffer [6], makes use of estimates for Weyl sums in a somewhat unusual manner and of simple remarks on diophantine approximation. The error term estimate $R_n = O(1/\log n)$ is best possible for $g(x) = x$, in the sense that it cannot be replaced by $o(1/\log n)$ (see the remark in [5]). There remains the problem of replacing the estimate $R_n = O(\log\log n/\log n)$ by $O(1/\log n)$ for $g(x) \in \mathbf{R}[x] \setminus \mathbf{Q}[x]$.

## 2. A lemma.

**Lemma.** *Let $f(x)$ be a polynomial with real coefficients and leading term $Ax^b$ where $A \neq 0$ and $b \ge 2$. Let $a/q$ be a rational number such that $(a, q) = 1$ and $|A - a/q| \le q^{-2}$. Let $V \ge 1$ be a real number. Then*

$$\left| \sum_{1 \le n \le Q} e(f(n)) \right| \ll Q\left(Q^{-1} + V^{-1}(\log Q)^B + V(q^{-1} + Q^{-1}\log q + Q^{-b+1} + Q^{-b} q \log q)\right)^\delta$$

*where $\delta = 2^{-b+1}$ and $e(t) = \exp(2\pi i t)$. Here, $B$ is any constant satisfying*

$$\sum_{n \le x} \tau_{b-1}(n)^2 \ll x(\log x)^B$$

*as* $x \to \infty$, *and* $\tau_{b-1}(n)$ *is the number of representations of* $n$ *as a product of* $b-1$ *positive integers* ($\tau_1(n) = 1$).

It is known that the choice $B = (b-1)^2 - 1$ is sufficient (compare [3], chapter III, problem 8, page 60).

**Corollary.** *Make the same assumptions as in the lemma and let* $q$ *be such that*

$$(\log Q)^H \ll q \ll Q^b (\log Q)^{-H}, \tag{2}$$

*where* $H = B + 2^{b-1} \cdot 2G + 1$ *and* $G$ *is a non-negative real number. Then*

$$\left| \sum_{1 \le n \le Q} e(f(n)) \right| \ll Q(\log Q)^{-G}. \tag{3}$$

*Remark.* If $b = 1$, the corollary still holds with $B = 0$.

*Proof of the lemma.* As is usual in treating Weyl sums (compare Lemmas 3.3 and 3.4 in [2], or Lemmas 2.3 and 2.4 in [7]), we have

$$\left| \sum_{1 \le n \le Q} e(f(n)) \right|^{2^{b-1}} \ll (2Q)^{2^{b-1}-b} \left( (2Q)^{b-1} + \sum_{1 \le y \le b! q^{b-1}} 2^{b-1} \tau_{b-1}(y) \min(Q, \|Ay\|^{-1}) \right),$$

where $\|t\| = \min(t - [t], 1 + [t] - t)$. For $k = 0, 1, 2, \ldots$, we have

$$\sum_{\substack{1 \le y \le b! Q^{b-1} \\ \tau_{b-1}(y) \ge 2^k V}} 1 \ll (2^k V)^{-2} \sum_{1 \le y \le b! Q^{b-1}} \tau_{b-1}(y)^2 \ll (2^k V)^{-2} Q^{b-1} (\log Q)^B$$

and then

$$\begin{aligned} \sum_{\substack{1 \le y \le b! Q^{b-1} \\ \tau_{b-1}(y) \ge V}} \tau_{b-1}(y) &\ll \sum_{k=0}^{\infty} 2^{k+1} V \sum_{\substack{1 \le y \le b! Q^{b-1} \\ \tau_{b-1}(y) \ge 2^k V}} 1 \\ &\ll \sum_{k=0}^{\infty} 2^{k+1} V (2^k V)^{-2} Q^{b-1} (\log Q)^B \\ &\ll V^{-1} Q^{b-1} (\log Q)^B, \end{aligned}$$

so that

$$\sum_{\substack{1 \le y \le b! Q^{b-1} \\ \tau_{b-1}(y) \ge V}} \tau_{b-1}(y) \min(Q, \|Ay\|^{-1}) \ll V^{-1} Q^b (\log Q)^B.$$

As for $y$'s with $\tau_{b-1}(y) \le V$, we have

$$\begin{aligned} \sum_{\substack{1 \le y \le b! Q^{b-1} \\ \tau_{b-1}(y) \le V}} \tau_{b-1}(y) \min(Q, \|Ay\|^{-1}) &\ll V \sum_{1 \le y \le b! Q^{b-1}} \min(Q, \|Ay\|^{-1}) \\ &\ll V(Q^{b-1} q^{-1} + 1)(Q + q \log q) \end{aligned}$$

by routine arguments in treating Weyl sums (compare Lemmas 3.5 and 3.6 in [2]). These inequalities imply the lemma.

## 3. Proof of the theorem.

Let the leading term of $g(x)$ be $\alpha x^b$. Let $j_0$ be a positive integer chosen sufficiently large. Then, for each integer $j \geq j_0$, there is a positive integer $n_j$ such that

$$r^{j-2} \leq g(n_j) < r^{j-1} \leq g(n_j+1) < r^j.$$

It follows that $n_j < n \leq n_{j+1}$ if and only if $r^{j-1} \leq g(n) < r^j$ and that $n_j \gg\ll r^{j/b}$ and $n_{j+1} - n_j \gg\ll r^{j/b}$, where the implied constants are independent of $j$. Let $J$ be a positive integer such that $n_J < x \leq n_{J+1}$, so that

$$J = \log_r g(x) + O(1) = O(\log x).$$

Put $X_J = x - n_J$ and $X_j = n_{j+1} - n_j$ for $(j_0 \leq)\ j < J$ and introduce the abbreviation $N(g(n)) = N_r([g(n)]; b_1 \dots b_l)$. Then

$$\sum_{n \leq x} N(g(n)) = \sum_{j=j_0}^{J} \sum_{n=n_j+1}^{n_j+X_j} N(g(N)) + O(1).$$

Using the periodic function $I(t)$ with period 1 defined by

$$I(t) = \begin{cases} 1 & \text{if } \sum_{k=1}^{l} b_k r^{-k} \leq t - [t] < \sum_{k=1}^{l} b_k r^{-k} + r^{-l} \\ 0 & \text{otherwise,} \end{cases}$$

we have

$$\sum_{n=n_j+1}^{n_j+X_j} N(g(n)) = \sum_{m=l}^{j} \sum_{n=n_j+1}^{n_j+X_j} I\left(\frac{g(n)}{r^m}\right).$$

Up to this point, our proof is the same as that in [5]. Choose now a sufficiently large constant $C_0$. Then we have

$$\left(\sum_{l \leq m \leq C_0 \log j} + \sum_{j - C_0 \log j \leq m \leq j}\right) \sum_{n=n_j+1}^{n_j+X_j} I\left(\frac{g(n)}{r^m}\right) = O(X_j \log j) = O(X_j \log\log x).$$

In what follows, we treat those $m$ with $C_0 \log j \leq m \leq j - C_0 \log j$. For each $j$, there are functions $I_-(t)$ and $I_+(t)$, periodic with period 1, such that $I_-(t) \leq I(t) \leq I_+(t)$, having Fourier expansions of the form

$$I_\pm(t) = r^{-l} \pm j^{-1} + \sum_{\substack{\nu=\infty \\ \nu \neq 0}}^{\infty} A_\pm(\nu) e(\nu t)$$

with $|A_{\pm}(\nu)| \ll \min(|\nu|^{-1}, j|\nu|^{-2})$. Then

$$\sum_{n=n_j+1}^{n_j+X_j} N(g(n)) = r^{-l}jX_j + O(X_j) + O(X_j \log\log x) + O\Bigg( \sum_{C_0 \log j \le m \le j - C_0 \log j} \sum_{\nu=1}^{j^2} \min(\nu^{-1}, j\nu^{-2}) \Bigg| \sum_{n=n_j+1}^{n_j+X_j} e\Big(\frac{\nu g(n)}{r^m}\Big) \Bigg| \Bigg), \tag{6}$$

where the implied constants are independent of $j$. We shall estimate the exponential sums

$$S(j,m,\nu) = \sum_{n=n_j+1}^{n_j+X_j} e\Big(\frac{\nu g(n)}{r^m}\Big),$$

where $J \ge j \ge j_0$, $j - C_0 \log j \ge m \ge C_0 \log j$ and $1 \le \nu \le j^2$. Here, the leading coefficient of $\nu r^{-m} g(x)$ is $\nu r^{-m}\alpha$. Assume first that $j < J$. For any pair $(m,\nu)$ for which there is a rational number $a/q$ such that

$$(a,q) = 1, \quad \Big|\frac{\nu\alpha}{r^m} - \frac{a}{q}\Big| \le \frac{1}{q^2} \quad \text{and} \quad (\log X_j)^H \le q \le X_j^b (\log X_j)^{-H} \tag{7}$$

with $G = 3$ and $H$ as in the lemma, we have

$$|S(j,m,\nu)| \ll X_j (\log X_j)^{-3} \ll X_j j^{-3}$$

by the corollary to the lemma. Hence, denoting by $\sum'$ the sum over all pairs $(m,\nu)$ having this property, we have the following estimates

$$\begin{aligned} \sum_m \sum_\nu{}' \min(\nu^{-1}, j\nu^{-2}) |S(j,m,\nu)| &\ll \sum_{m=l}^{j} \sum_{\nu=1}^{j^2} \min(\nu^{-1}, j\nu^{-2}) \cdot X_j j^{-3} \\ &\ll j \log j \cdot X_j j^{-3} \ll X_j \ll r^{j/b}. \end{aligned}$$

If $j = J$, there are two cases. Assume first that $X_J = O(r^{J/b} J^{-3})$. Then we have the trivial estimates

$$\sum_{m=l}^{J} \sum_{\nu=1}^{J^2} \min(\nu^{-1}, J\nu^{-2}) |S(J,m,\nu)| \ll \sum_{m=l}^{J} \sum_{\nu=1}^{J^2} \min(\nu^{-1}, J\nu^{-2}) \cdot r^{J/b} J^{-3} \ll r^{J/b} J^{-1}.$$

Otherwise, $X_J \gg r^{J/b} J^{-3}$ and we have $\log X_J \gg\ll J$, so that we can repeat the same argument as above for $j < J$. In any case, we get

$$\sum_m \sum_\nu{}' \min(\nu^{-1}, j\nu^{-2}) |S(j,m,\nu)| \ll r^{j/b} \tag{8}$$

for $(j_0 \le) j \le J$. It remains to estimate the sums over all pairs $(m,\nu)$ satisfying $j - C_0 \log j \ge m \ge C_0 \log j$ and for each of which there is no rational number $a/q$

satisfying the conditions in (7). (If $j = J$, $X_J$ is supposed to be $\gg r^{J/b}J^{-3}$.) It will turn out that there is no such pair $(m, \nu)$. To show this, we choose for each pair $(m, \nu)$ in question, a rational number $a/q$ such that

$$(a, q) = 1, \quad 1 \le q \le X_j^b (\log X_j)^{-H} \quad \text{and} \quad \left|\frac{\nu\alpha}{r^m} - \frac{a}{q}\right| \le \frac{(\log X_j)^H}{qX_j^b} \quad \left(\le \frac{1}{q^2}\right).$$

This is done by using an appropriate Farey approximant. If $2 \le q \le X_j^b(\log X_j)^{-H}$, then $2 \le q \le (\log X_j)^H$, since (7) is no longer satisfied. This implies that

$$\left|\frac{\nu\alpha}{r^m}\right| > \frac{1}{q} - \frac{1}{q^2} \ge \frac{1}{2q} \gg (\log X_j)^{-H},$$

so that

$$r^m \ll |\nu\alpha|(\log X_j)^H \ll j^2(\log X_j)^H \ll j^{H+2}$$

and therefore

$$(C_0 \log j \le) m \le (H+2)\log j + O(1)$$

which fails if $C_0$ is sufficiently large. Now let $q = 1$. Then $\|\nu r^{-m}\alpha\| < X_j^b(\log X_j)^H$. If $|\nu r^{-m}\alpha| \ge \frac{1}{2}$, then $r^m \ll \nu \ll j^2$, which is impossible again by the same reasoning as before. Otherwise, $|\nu r^{-m}\alpha| < \frac{1}{2}$ and $|\nu r^{-m}\alpha| = \|\nu r^{-m}\alpha\| < X_j^{-b}(\log X_j)^H$. This implies that

$$r^m > |\nu\alpha|\, X_j^b(\log X_j)^{-H} \gg (r^{j/b})^b(\log r^{j/b} - O(1))^{-H} \gg r_j j^{-H},$$

so that

$$(j - C_0 \log j \ge) m > j - O(\log j)$$

which is also impossible. Combining (4), (6) and (8), we have

$$\sum_{n=n_j+1}^{n_j+X_j} N(g(n)) = r^{-l} j X_j + O(X_j \log\log x) + O(r^{j/b}).$$

Therefore

$$\begin{aligned}\sum_{n\le x} N(g(n)) &= \sum_{j=j_0}^{J}(r^{-l}jX_j + O(X_j\log\log x) + O(r^{j/b})) \\ &= r^{-l}Jx + O(r^{J/b}\log\log x) \\ &= r^{-l}x\log_r g(x) + O(x\log\log x)\end{aligned}$$

and the proof of the theorem is completed.

## References

1. H. Davenport and P. Erdős, 'Note on normal decimals', *Canadian J. Math.* **4** (1952), 58–63.
2. L.-K. Hua, *Additive theory of prime numbers.* (Translations of Mathematical Monographs, American Math. Soc., **13**, 1965.)
3. A. A. Karacuba, *Foundations of the analytic theory of numbers.* (2nd edition, Nauka, 1983 (in Russian).)
4. L. Kuipers and H. Niederreiter, *Uniform distribution of sequences.* (Wiley, New York, 1974.)
5. Y.-N. Nakai and I. Shiokawa, 'A class of normal numbers', *Japanese J. Math.* **16** (1990). (To appear.)
6. J. Schiffer, 'Discrepancy of normal numbers', *Acta Arith.* **47** (1986), 175–186.
7. R. C. Vaughan, *The Hardy-Littlewood method.* (Cambridge Tracts in Mathematics, Cambridge University Press, **80**, 1981.)

*Department of Mathematics, Faculty of Education, Yamanashi University, Kofu, 400 JAPAN.*

*Department of Mathematics, Keio University, Hiyoshi, Yokohama, 223 JAPAN.*

# NOTES ON UNIFORM DISTRIBUTION

G. Myerson and A. Pollington

Let $u = (u_1, u_2, \ldots)$ be a sequence of real numbers. We say that $u$ is uniformly distributed (modulo 1) if for every subinterval $J$ of $[0,1)$ we have

$$\lim_{N\to\infty} \frac{1}{N} \#\{n \le N : \{u_n\} \in J\} = \mu(J),$$

where $\mu(J)$ is the length of $J$. For motivation, examples, applications and much more, we refer the reader to [1].

The following results are quite well known.

**Proposition 1.** *If $u$ is u.d.* (mod 1) *then so is $ku$ for any non-zero integer $k$.*

**Proposition 2.** *Fix $k \ge 1$. If for all $j$ with $1 \le j \le k$ the sequence $(u_{kn+j})$ is u.d.* (mod 1), *then $u$ is u.d.* (mod 1).

We prove that even very weak converses of these propositions are false.

**Theorem 1.** *There is a sequence $u$ which is not u.d.* (mod 1) *even though $ku$ is u.d.* (mod 1) *for every integer $k \ge 2$.*

**Theorem 2.** *There is a sequence $u$ which is u.d.* (mod 1) *even though no subsequence of the form $(u_{kn+j})$ with $k \ge 2$ is u.d.* (mod 1).

Our proofs are constructive. An example illustrating Theorem 1 appears below. The proofs will appear in [2].

The following result may be seen as complementary to Proposition 2.

**Theorem 3.** *Let $P$ be a set of primes such that $\sum_{p\in P} \frac{1}{p}$ diverges. If $(u_{kn})$ is u.d.* (mod 1) *for every $k \ge 2$ composed of primes from $P$, then $u$ is u.d.* (mod 1).

The proof uses the Weyl criterion for uniform distribution [1] and a simple sieve argument; details will appear in [2].

We now move to higher dimensions. Let $u = (u_1, u_2, \ldots)$ be a sequence of $m$-tuples of real numbers. We say that $u$ is uniformly distributed (modulo 1) if for every "subinterval" $J$ of $[0,1)^m$ we have

$$\lim_{N\to\infty} \frac{1}{N} \#\{n \le N : \{u_n\} \in J\} = \mu(J),$$

where $\mu(J)$ is the measure of $J$. Here $J = I_1 \times I_2 \times \ldots \times I_m$ where each $I_j$ is a subinterval of $[0,1)$, and $\{u_n\} = (\{u_n^{(1)}\}, \{u_n^{(2)}\}, \ldots, \{u_n^{(m)}\})$.

The following results are quite well known.

**Proposition 3.** *If $u$ is u.d.* (mod 1) *then so is $Au$ for any non-singular integer matrix $A$.*

**Proposition 4.** *If $Au$ is u.d.* (mod 1) *and* $\det A = \pm 1$ *then $u$ is u.d.* (mod 1).

By way of contrast to Theorem 1, we show that if $Au$ is u.d. (mod 1) for enough matrices $A$, then $u$ is u.d. (mod 1). Let $S$ be a set of $m \times m$ integer matrices. We call $S$ a *covering* if for every integer row $m$-vector $h$ there exists an $A$ in $S$ and an integer row $m$-vector $k$ such that $kA = h$.

**Theorem 4.** *If $Au$ is u.d.* (mod 1) *for all $A$ in a covering set of matrices, then $u$ is u.d.* (mod 1).

*Remark.* It is not difficult to find coverings. Indeed, in each dimension (greater than 1) there is a covering with just three elements.

*Proof.* Note that $\sum_{n=1}^{N} e(h \cdot u_n) = \sum_{n=1}^{N} e(kA \cdot u_n) = \sum_{n=1}^{N} e(k \cdot Au_n)$, where, as usual, $e(x) = e^{2\pi i x}$. Now apply the Weyl criterion.

*Example illustrating Theorem* 1. Let $g(t) = t + \frac{1}{2\pi}\sin 2\pi t$. Let $h$ be the function inverse to $g$. Let $x$ be any sequence uniformly distributed (modulo 1). Let $u = h(\{x\})$, that is, $u_n = h(\{x_n\})$ for $n = 1, 2, \ldots$. It is not hard to show that

$$\lim_{N\to\infty} \frac{1}{N}\#\{n \le N : \alpha \le u_n < \beta\} = \beta - \alpha + \frac{1}{2\pi}(\sin 2\pi\beta - \sin 2\pi\alpha),$$

so $u$ is not u.d. (mod 1), whereas, making use of the identity $\sum_{r=0}^{k-1} \sin 2\pi(\gamma + \frac{r}{k}) = 0$ for all real $\gamma$ and all $k \ge 2$, we find $\lim_{N\to\infty} \frac{1}{N}\#\{n \le N : \alpha \le \{ku_n\} < \beta\} = \beta - \alpha$ for all $k \ge 2$, so $ku$ is u.d. (mod 1).

## References

1. L. Kuipers and H. Niederreiter, *Uniform Distribution of Sequences.* (Wiley, 1974.)
2. G. Myerson and A. Pollington, 'Notes on uniform distribution modulo one', *J. Austral. Math. Soc. (A)*, to appear.

*School of Mathematics, Physics, Computing and Electronics, Macquarie University, New South Wales 2109, AUSTRALIA.*

*Department of Mathematics, Brigham Young University, Provo, Utah 84602, USA.*

# THUE EQUATIONS AND MULTIPLICATIVE INDEPENDENCE

B. Brindza

## 1. Introduction.

In this paper, we demonstrate the multiplicative independence of certain algebraic numbers. By combining this with an inequality for simultaneous linear forms in logarithms, we derive new sharp effective bounds for the solutions of a class of Thue equations.

Let $\alpha$ be an algebraic integer of degree $n > 1$ and denote the conjugates of $\alpha$ over the rational field, $\mathbf{Q}$, by $\alpha_1 = \alpha, \alpha_2, \ldots, \alpha_n$. Further, let $\beta_2, \ldots, \beta_n$ be non-zero elements of the field $\mathbf{Q}(\alpha_1, \ldots, \alpha_n)$ and let $p$ be a positive number. For brevity, we write $M = M(\alpha)$ for the Mahler height of $\alpha$, that is

$$M = \prod_{i=1}^{n} \max\{1, |\alpha_i|\}.$$

Let $\Gamma$ denote the Galois group of the extension $\mathbf{Q}(\alpha_1, \ldots, \alpha_n)/\mathbf{Q}$ and set

$$B_1 = \min_{\substack{2 \le i \le n \\ \phi \in \Gamma}} |\phi(\beta_i)|, \qquad B_2 = \max_{\substack{2 \le i \le n \\ \phi \in \Gamma}} |\phi(\beta_i)|.$$

Then we have the following theorem.

**Theorem 1.** *Suppose that* $|x - \alpha_1 y| \le p$ *and* $|y| > c(n) p M^{(2n-1)(n-2)} (B_2/B_1)^{n-1}$, *where* $c(n) = 2^{n^2+n-5} n^{\frac{1}{2}(n-1)(n-2)}$, *and* $x$ *and* $y$ *are rational integers. Then the numbers* $\beta_2(x - \alpha_2 y), \ldots, \beta_n(x - \alpha_n y)$ *are multiplicatively independent.*

Let $F(X, Y) = (X - \gamma_1 Y) \cdots (X - \gamma_n Y)$ be an irreducible binary form of degree $n$ with rational integer coefficients and let $\epsilon_1, \ldots, \epsilon_r$ be a fundamental system of units for the splitting field of the polynomial $F(X, 1)$. For an arbitrary rational integer solution $(x, y)$ of the equation

$$|F(x, y)| = 1, \tag{1}$$

the factors $x - \gamma_i y$, $i = 1, \ldots, n$ are units. Therefore

$$(x - \gamma_i y)^m = \epsilon_1^{k_{1i}} \cdots \epsilon_r^{k_{ri}},$$

where $m$ is the rank of the group of roots of unity in the splitting field of $F(X,1)$ and the exponents are rational integers. We may assume, without loss of generality, that $|x-\gamma_1 y| \leq 1$. If $r < n-1$, then the vectors $\mathbf{k}_i = (k_{1i}, \ldots, k_{ri})$, $i = 2, \ldots, n$ are linearly dependent over $\mathbf{Z}$, hence $x - \gamma_2 y, \ldots, x - \gamma_n y$ are multiplicatively dependent. Using Theorem 1 with $p = B_1 = B_2 = 1$ and the notation introduced above, we have the following immediate consequence.

**Theorem 2.** *If the unit rank of the splitting field of $F(X,1)$ is less than $n-1$, then all the solutions of (1) in rational integers satisfy*

$$|y| \leq c(n) M(\gamma)^{(2n-3)(n-2)}.$$

To see an example satisfying the condition imposed on the unit rank in Theorem 2, let $K$ be a normal extension of $\mathbf{Q}$ of degree $k \geq 3$ and let $\gamma$ be a primitive element of $K$. If $K$ is not totally real, then its unit rank is less than $k-1$, hence the binary form

$$F(X,Y) = N_{K/\mathbf{Q}}(X - \gamma Y)$$

satisfies the condition. If $K$ is totally real, then there seems to be no way to derive an upper bound for all the solutions without Baker's method. However, by combining Theorem 1 with an inequality of Loxton for simultaneous linear forms in logarithms, we can prove the following theorem.

**Theorem 3.** *Suppose $\gamma$ is an algebraic integer of degree at least 3 and a primitive element of the number field $K$. All the solutions of the equation*

$$|N_{K/\mathbf{Q}}(x - \gamma y)| = 1 \qquad (1^*)$$

*in non-zero rational integers $x$ and $y$ satisfy*

$$\max\{\log|x|, \log|y|\} < C_1 M(\gamma)^3,$$

*where $C_1$ is an effectively computable constant depending only on $k = [K : \mathbf{Q}]$.*

We note that general effective results for Thue equations (for example, [5]) give similar bounds with $M(\gamma)^{\frac{1}{2}k(k-1)}$. In our theorem, the essential improvement is that the exponent of $M(\gamma)$ does not depend on $k$. Actually, our method yields $1 + \frac{1}{k-2} + \epsilon$ instead of 3.

## 2. Proof of Theorem 1.

Suppose, contrary to the assertion of Theorem 1, that

$$\prod_{i=2}^{n} (\beta_i(x - \alpha_i y))^{r_i} = 1,$$

with some rational integers $r_2, \ldots, r_n$ not all zero. It is more convenient to write this relation in the form

$$\sum_{i=2}^{n} r_i \log|\beta_i(x - \alpha_i y)| = 0.$$

Let $|r_j|$ be the largest element of the set $\{|r_2|,\dots,|r_n|\}$. The natural isomorphism betwen the fields $\mathbf{Q}(\alpha_j)$ and $\mathbf{Q}(\alpha_1)$ (for which $\alpha_j \mapsto \alpha_1$) can be extended to the field $\mathbf{Q}(\alpha_1,\dots,\alpha_n)$. Let $\phi \in \Gamma$ denote the resulting automorphism. Obviously,

$$\prod_{i=2}^{n} \left(\phi(\beta_i)(x-\phi(\alpha_i)y\right)^{r_i} = 1$$

and

$$-r_j(\log|\beta_j(x-\alpha_j y)| - \log|\phi(\beta_j)(x-\alpha_1 y)|) = \sum_{\substack{i=2\\ i\neq j}}^{n} r_i\big(\log|\beta_i(x-\alpha_i y)| - \log|\phi(\beta_i)(x-\phi(\alpha_i)y)|\big).$$

Hence

$$\big|\log|\beta_j(x-\alpha_j y)| - \log|\phi(\beta_j)(x-\alpha_1 y)|\big| \le \sum_{\substack{i=2\\ i\neq j}} \left|\log\left|\frac{\beta_i(x-\alpha_i y)}{\phi(\beta_i)(x-\phi(\alpha_i)y}\right|\right| \tag{2}$$

For any $i$ with $2 \le i \le n$, we obtain

$$|\beta_i(x-\alpha_i y)| = |y|\left|\beta_i(\alpha_1-\alpha_i) + \frac{\beta_i}{y}(x-\alpha_1 y)\right| \le 2|y||\beta_i||\alpha_1-\alpha_i|$$

and

$$|\beta_i(x-\alpha_i y)| \ge \tfrac{1}{2}|y||\beta_i||\alpha_1-\alpha_i|$$

provided that

$$|y| \ge \frac{2p}{|\alpha_1-\alpha_i|}.$$

Now we need an upper bound for $|\alpha_1-\alpha_i|^{-1}$. We could apply the usual 'Liouville type argument' (see, for example, [5], page 10), but in our case a sharper result can be obtained from the inequality

$$|F'(\alpha_1,1)| = |\alpha_2-\alpha_1|\cdots|\alpha_n-\alpha_1| \ge (M\sqrt{n})^{2-n} \tag{3}$$

which is a special case of Lemma 1 in [1]. From (3), we have

$$|\alpha_i-\alpha_1| \ge 2^{2-n}n^{\frac{1}{2}(2-n)}M^{4-2n}. \tag{4}$$

In the sequel, we may assume that

$$|y| \ge 2^{n-1}n^{\frac{1}{2}(n-2)}M^{2n-4}p,$$

since this is implied by the hypotheses of the theorem. Thus, for $i \notin \{1,j\}$,

$$\left|\frac{\beta_i(x-\alpha_i y)}{\phi(\beta_i)(x-\phi(\alpha_i)y)}\right| \le \frac{2|\beta_i||\alpha_1-\alpha_i|}{\frac{1}{2}|\phi(\beta_i)||\alpha_1-\phi(\alpha_i)|} \le \frac{B_2}{B_1}\,2^{n+2}n^{\frac{1}{2}(n-2)}M^{2n-3}$$

and similarly

$$\left|\frac{\beta_i(x-\alpha_i y)}{\phi(\beta_i)(x-\phi(\alpha_i)y)}\right| \geq \frac{B_1}{B_2}\, 2^{-(n+1)} n^{\frac{1}{2}(2-n)} M^{3-2n}.$$

These inequalities with (2) imply

$$\left|\log|\beta_j(x-\alpha_j y)| - \log|\phi(\beta_j)(x-\alpha_1 y)|\right| \leq (n-2)\log\left(\frac{B_2}{B_1} 2^{n+2} n^{\frac{1}{2}(n-2)} M^{2n-3}\right).$$

The elementary inequality

$$a \leq b\exp(|\log a - \log b|) \quad \text{for} \quad a, b > 0,$$

yields

$$|\beta_j(x-\alpha_j y)| \leq |\phi(\beta_j)||x-\alpha_1 y|\left(\frac{B_2}{B_1} 2^{n+2} n^{\frac{1}{2}(n-2)} M^{2n-3}\right)^{n-2},$$

therefore

$$|x-\alpha_j y| \leq 2^{n^2-4} n^{\frac{1}{2}(n-2)^2} M^{(2n-3)(n-2)} p\left(\frac{B_2}{B_1}\right)^{n-1} \tag{5}$$

and, finally, combining (5) with $|x-\alpha_1 y| \leq p$ and (4) completes the proof.

## 3. Preparation for the proof of Theorem 3.

To prove Theorem 3, we need some preliminaries. Consider the linear forms

$$\Lambda_i = b_{i0} + b_{i1}\log a_1 + \cdots + b_{in}\log a_n \quad (1 \leq i \leq t)$$

where the $a$'s are algebraic numbers and the $b$'s are rational integers and the logarithms have their principal values. Set

$$A_i = \max\{4, M(a_i)\} \quad (i = 1, \ldots, n), \quad B = \max_{\substack{1\leq i\leq t \\ 0\leq j\leq n}} (4, |b_{ij}|)$$

and

$$\Omega = \log A_1 \cdots \log A_n.$$

The following lemma is a simple consequence of a result of Loxton ([3], Theorem 4).

**Lemma 1.** *If the $a$'s are non-zero and multiplicatively independent and the matrix $(b_{ij})$ formed by the $b$'s has rank $t$, then*

$$\max_{1\leq i\leq t} |\Lambda_i| > \exp\{-C_2(\Omega\log\Omega)^{1/t}\log(B\Omega)\},$$

*where $C_2$ is an explicit constant depending only on $n$ and the degree of $\mathbf{Q}(a_1, \ldots, a_n)$ over $\mathbf{Q}$.*

**Lemma 2** (Loxton and van der Poorten [4]). *Let $a_1 \ldots, a_n$ be multiplicatively dependent algebraic numbers. Then there are rational integers $q_1, \ldots, q_n$, not all zero, such that*

$$a_1^{q_1} \cdots a_n^{q_n} = 1$$

*and*

$$|q_k| < C_3 \prod_{i \neq k} \log\left(2M(a_i)\right)$$

*where $C_3$ is an explicit constant depending only on $n$ and the degree of $\mathbf{Q}(a_1, \ldots, a_n)$ over $\mathbf{Q}$.*

Let $K$ be an algebraic number field of degree $n$ with regulator $R$ and unit rank $r$.

**Lemma 3.** *There is a fundamental system of units $\{\epsilon_1, \ldots, \epsilon_r\}$ for $K$ such that*

$$\prod_{i=1}^{r} \log M(\epsilon_i) \leq 2r! r^r R.$$

*Proof.* Following the usual notation, let $s$ and $t$ denote, respectively, the number of real and pairwise non-conjugate complex embeddings of $K$ into $\mathbf{C}$. For an element $\alpha \in K$, let $\alpha^{(1)} = \alpha, \ldots, \alpha^{(n)}$ denote the conjugates of $\alpha$ over $\mathbf{Q}$, where $\alpha^{(1)}, \ldots, \alpha^{(s)}$ are real and $\alpha^{(i+t)} = \overline{\alpha}^{(i)}$ for $i = s+1, \ldots, s+t$. For $\alpha \neq 0$, set

$$L(\alpha) = \max_{1 \leq i \leq s+t-1} \left| e_i \log |\alpha^{(i)}| \right|,$$

where $e_i = 1$ for $1 \leq i \leq s$ and $e_i = 2$ for the remaining $i$. It is well known (see, for example, [5], page 22) that there are multiplicatively independent units $\eta_1, \ldots, \eta_r$ in $K$ for which

$$L(\eta_1) \cdots L(\eta_r) \leq R.$$

The function $L$ can be considered as a convex distance function on the logarithmic space and then the geometry of numbers (see [2], Lemma 8, page 135) provides a fundamental basis $\{\epsilon_1, \ldots, \epsilon_r\}$ satisfying

$$L(\epsilon_i) \leq \max\{L(\eta_i), \tfrac{1}{2}[L(\eta_1) + \cdots + L(\eta_i)]\}$$

for $i = 1, \ldots, r$. We may assume without loss of generality that

$$L(\eta_1) \leq \ldots \leq L(\eta_r).$$

Then the simple inequalities

$$0 < \log M(\epsilon_i) \leq 2r L(\epsilon_i) \leq 2r \max\{1, \tfrac{1}{2}i\} L(\eta_i)$$

imply Lemma 3. Here, we have used the fact that $\epsilon_i$ is a unit to estimate the conjugate not counted in the definition of $L(\epsilon_i)$.

*Remark.* The existence of a fundamental system of units for $K$ with

$$L(\epsilon_1)\cdots L(\epsilon_r) \le \left(\frac{2}{\sqrt{3}}\right)^{\frac{1}{2}r(r-1)} R$$

was proved by Silverman [6]. Our proof provides $r!2^{1-r}$ instead of $(2/\sqrt{3})^{\frac{1}{2}r(r-1)}$.

## 4. Proof of Theorem 3.

We may assume that all the conjugates $\gamma_1,\dots,\gamma_k$ of $\gamma$ (over $\mathbf{Q}$) are real, for otherwise, Theorem 2 provides a much better bound. Let $\epsilon_1,\dots,\epsilon_r$ be a fundamental system of units of $K$ with

$$\log M(\epsilon_1)\cdots\log M(\epsilon_r) \le c_1 R,$$

where $R$ is the regulator of $K$. In the sequel, $c_1,\dots,c_{10}$ will denote effectively computable positive constants depending only on $k = [K : \mathbf{Q}]$. Let $(x,y)$ be a solution of (1*). We can assume that $|y| > \exp M(\gamma)^3$ and $M(\gamma) > c_2$, where the constant $c_2$ will be determined later. The factors $x - \gamma_i y$ are units and can be written in the form

$$x - \gamma_i y = \pm\epsilon_1^{t_{1i}}\cdots\epsilon_r^{t_{ri}}, \quad i = 1,\dots,k,$$

where the exponents are rational integers. The field $K = \mathbf{Q}(\gamma)$ is a normal extension of $\mathbf{Q}$, so it is generated by each of the $\gamma_i$. Hence, we may assume that $|x - \gamma_1 y| \le 1$. From the well-known identity

$$(\gamma_i - \gamma_2)(x - \gamma_1 y) + (\gamma_2 - \gamma_1)(x - \gamma_i y) + (\gamma_1 - \gamma_i)(x - \gamma_2 y) = 0, \quad i = 3,\dots,k,$$

we have

$$\left|\beta_i \epsilon_1^{t_{1i}-t_{12}}\cdots\epsilon_r^{t_{ri}-t_{r2}} \pm 1\right| = \left|\frac{\gamma_i - \gamma_2}{\gamma_1 - \gamma_i}\right|\left|\frac{x - \gamma_1 y}{x - \gamma_2 y}\right|, \quad i = 3,\dots,k, \tag{6}$$

where $\beta_i = (\gamma_1 - \gamma_2)/(\gamma_1 - \gamma_i)$. By taking $c_2$ large enough, we obtain

$$\left|\frac{\gamma_1 - \gamma_2}{\gamma_1 - \gamma_i}\right|\left|\frac{x - \gamma_1 y}{x - \gamma_2 y}\right| \le \frac{1}{\sqrt{|y|}}. \tag{7}$$

By applying the 'regulator argument' (see, for example, [5], page 103), we see that

$$\max_{\substack{1\le j\le r\\ 1\le i\le k}} |t_{ji}| < c_3 \log A,$$

where $A = \max\{2, |y|, M(\gamma)\}$. In our case, we can suppose $A = |y|$, for otherwise the theorem is proved. Let $\{\delta_1,\dots,\delta_s\}$ be a maximal multiplicatively independent subset of $\mathcal{S} = \{\beta_3,\dots,\beta_k,\epsilon_1,\dots,\epsilon_r\}$. For an arbitrary $\mu \in \mathcal{S}$, there are rational integers $e_0(\mu), e_1(\mu),\dots,e_s(\mu)$ such that $e_0(\mu) \ne 0$ and

$$\mu^{e_0(\mu)} = \delta_1^{e_1(\mu)}\cdots\delta_s^{e_s(\mu)}.$$

By Lemma 2, we may asume that

$$\max_{\substack{0\le j\le s\\ \mu\in\mathcal{S}}} |e_j(\mu)| < c_4\Big(\prod_{i=3}^{k}\log M(\beta_i)\Big)\Big(\prod_{i=1}^{r}\log M(\epsilon_i)\Big).$$

Let $D$ denote the discriminant of $K$. The inequalities

$$R < c_5 D^{1/2}(\log D)^{k-1}, \quad D < c_6 M(\gamma)^{2k-2},$$

of Siegel and Mahler, respectively, imply

$$\max_{\substack{0\le j\le s\\ \mu\in\mathcal{S}}} |e_j(\mu)| < c_7 M(\gamma)^{k-1}\big(\log M(\gamma)\big)^{2k-3}. \tag{8}$$

Set

$$e = 2\prod_{\mu\in\mathcal{S}} e_0(\mu)$$

and note that $e < c_8 M(\gamma)^{k(2k-3)}$. It will be more convenient to work with the forms

$$\Lambda_i = \beta_i^e \epsilon_1^{e(t_{1i}-t_{12})}\cdots\epsilon_r^{e(t_{ri}-t_{r2})} - 1, \quad i = 3,\dots,k.$$

We rewrite $\Lambda_i$ as

$$\Lambda_i = \delta_1^{f_{1i}}\cdots\delta_s^{f_{si}} - 1,$$

where the exponents are rational integers and (8) yields

$$\max_{\substack{1\le j\le s\\ 3\le i\le k}} |f_{ji}| < c_9 M(\gamma)^{2k^2}\log|y| < \big(\log|y|\big)^{3k^2},$$

provided that $c_2$ is large enough. From the elementary inequality

$$|z^n - 1| \le n|z-1|(1+|z-1|)^n,$$

we get

$$\max_{3\le i\le k} |\Lambda_i| \le \frac{e}{\sqrt{|y|}}\left(1+\frac{1}{\sqrt{|y|}}\right)^e < \frac{1}{|y|^{1/3}}, \tag{9}$$

if $c_2$ is large enough.

Preparatory to an application of Lemma 1, we prove that the rank of the matrix formed by the row vectors $\mathbf{f}_i = (f_{1i},\dots,f_{si})$, $i = 3,\dots,k$, is $k-2$ so that these vectors are linearly independent. Supposing the contrary, we have

$$\sum_{i=3}^{k} n_i\mathbf{f}_i = \mathbf{0}$$

with some rational integers $n_3, \ldots, n_k$, not all zero. This implies

$$\prod_{i=3}^{k} \left( \beta_i \frac{x - \gamma_i y}{x - \gamma_2 y} \right)^{n_i e} = 1,$$

so that the elements $(x - \gamma_2 y), \beta_3(x - \gamma_3 y), \ldots, \beta_k(x - \gamma_k y)$ are multiplicatively dependent in contradiction to Theorem 1, provided $c_2$ is large enough. Finally, we can apply Lemma 1 to obtain the lower bound

$$\max_{3 \le i \le k} |\Lambda_i| > \exp\left( - c_{10} \left[ \left( M(\gamma) \right)^{k-1} \left( \log M(\gamma) \right)^{2k-3} \right]^{\frac{1}{k-2}} \log\log |y| \right),$$

and comparison of this inequality with (9) gives Theorem 3.

I would like to thank Professors Alf van der Poorten and John Loxton for their valuable remarks.

## References

1. E. Bombieri and W. M. Schmidt, 'On Thue's equation', *Inv. Math.* **88** (1987), 69–81.
2. J. W. S. Cassels, *An introduction to the geometry of numbers.* (Springer, Berlin, 1959.)
3. J. H. Loxton, 'Some problems involving powers of integers', *Acta Arith.* **46** (1987), 113–123.
4. J. H. Loxton and A. J. van der Poorten, 'Multiplicative dependence in number fields', *Acta Arith.* **42** (1983), 291–302.
5. T. N. Shorey and R. Tijdeman, *Exponential diophantine equations.* (Cambridge University Press, 1986.)
6. J. H. Silverman, 'An inequality relating the regulator and discriminant of a number field', *J. Number Theory* **19** (1984), 437–443.

*School of Mathematics, Physics, Computing and Electronics, Macquarie University, New South Wales 2109, AUSTRALIA.*

# 21

# A NUMBER THEORETIC CRANK ASSOCIATED WITH OPEN BOSONIC STRINGS

**Frank G. Garvan**

A Dyson-type crank is given which explains Moreno's congruence for the number of open bosonic strings. This crank is in terms of 24-coloured partitions.

## 1. Introduction.

Let $q = e^{2\pi iz}$, where $\mathrm{Im}(z) > 0$ so that $|q| < 1$. The well-known discriminant modular form $\Delta(z)$ has $q$-expansion

$$(2\pi)^{-12}\Delta(z) = \sum_{n=1}^{\infty} \tau(n)q^n = q\prod_{n=1}^{\infty}(1-q^n)^{24}. \tag{1.1}$$

The reciprocal of this function is essentially the string function associated to the affine Lie algebra $A_{24}^{(1)}$ ([12], page 137, [13], section 3.2). We let $\tilde{\tau}(n)$ denote the $n$-th Fourier coefficient of this function so that

$$\tilde{\Delta}(z) = \sum_{n=-1}^{\infty} \tilde{\tau}(n)q^n = \frac{1}{q\prod_{n=1}^{\infty}(1-q^n)^{24}}. \tag{1.2}$$

As noted by Moreno and Rocha-Caridi ([13], page 144), $\tilde{\Delta}(z)$ has a physical interpretation from the light cone formulation of string theory. In fact, the number of open string states with mass $M$ such that $\alpha' M^2 = n$ is $\tilde{\tau}(n)$, where $\alpha'$ is the Regge slope ([11], page 117). The coefficients $\tilde{\tau}(n)$ can also be interpreted in terms of the weight multiplicities of the vertex algebra associated to the unique Lorentzian lattice of signature $(25,1)$ and the No-Ghost Theorem of Brower, Goddard and Thorn ([11], page 102). Moreno and Rocha-Caridi [13] also found Hardy-Ramanujan-Rademacher expansions for string functions associated with affine Lie algebras and thus found such an expansion for $\tilde{\tau}(n)$. In [14], Moreno explored congruence and combinatorial properties of $\tilde{\tau}(n)$.

The coefficients $\tilde{\tau}(n)$ have a natural combinatorial interpretation in terms of coloured partitions. The coefficients $p_r(n)$ are defined by

$$\sum_{n\geq 0} p_r(n)q^n = \prod_{n=1}^{\infty}(1-q^n)^r. \tag{1.3}$$

For negative $r$, say $r = -d$, $p_r(n)$ counts the number of partitions of $n$ taken from $d$ copies of the natural numbers. We call such partitions $d$-coloured partitions. In view of (1.2) we see that $\tilde{\tau}(n)$ counts 24-coloured partitions:

$$\tilde{\tau}(n) = p_{-24}(n+1). \tag{1.4}$$

It is this interpretation of $\tilde{\tau}(n)$ that we use in this paper.

Moreno ([14], theorem 12) proved the following congruence

$$p_{-24}(n) \equiv 0 \pmod 5 \quad \text{for} \quad n \equiv 1 \pmod 5 \quad \text{and} \quad n \neq 1, \tag{1.5}$$

and asked for a combinatorial interpretation analogous to Dyson's interpretation ([2], [3]) of Ramanujan's congruences ([15]) for the partition function $p(n)$ $(= p_{-1}(n))$. We give such an interpretation below in Theorem 1. We also explain similar congruences modulo 2, 3 and 25. Our combinatorial interpretation is in terms of the crank of 24-coloured partitions and is described in the next section. Our method is elementary and was first introduced in [5]. We note other interpretations of partition congruences have been found in [1], [6], [7], [8], [9] and [10].

## 2. The crank for 24-coloured partitions.

As well as (1.5) the following congruences hold

$$p_{-24}(n) \equiv 0 \pmod 2 \quad \text{for} \quad n \not\equiv 0 \pmod 8 \tag{2.1}$$

$$p_{-24}(n) \equiv 0 \pmod 3 \quad \text{for} \quad n \not\equiv 0 \pmod 3 \tag{2.2}$$

$$p_{-24}(n) \equiv 0 \pmod{25} \quad \text{for} \quad n \equiv 3 \text{ or } 4 \pmod 5 \tag{2.3}$$

$$p_{-24}(n) \equiv 0 \pmod 7 \quad \text{for} \quad n \equiv 1 \pmod 7 \quad \text{and} \quad n \neq 1. \tag{2.4}$$

The congruences (2.1) and (2.2) are trivial. The proof of (2.4) is analogous to Moreno's proof of (1.5) and also follows from [4], Lemma (3.12). (2.3) has a very simple proof.

$$\begin{aligned}\sum_{n\geq 0} p_{-24}(n)q^n &= \prod_{m=1}^{\infty} \frac{(1-q^m)}{(1-q^m)^{25}} \\ &\equiv \prod_{m=1}^{\infty} \frac{(1-q^m)}{(1-q^{5m})^5} \pmod{25} \\ &= \frac{\sum_{-\infty}^{\infty}(-1)^n q^{n(3n-1)/2}}{\prod_{m=1}^{\infty}(1-q^{5m})^5}.\end{aligned} \tag{2.5}$$

The result follows since $n(3n-1)/2 \not\equiv 3, 4 \pmod 5$.

We now define a *crank* that explains (1.5), (2.1), (2.2), (2.3), but not (2.4). In this section we concentrate on the more interesting congruences (1.5), (2.3) and leave the remaining congruences (2.1) and (2.2) to section 3. We number the 24 colours $1, 2, \ldots, 24$. For a 24-coloured partition $\tilde{\pi}$ let

$$c_i(\tilde{\pi}) := \text{the number of parts of } \tilde{\pi} \text{ coloured } i.$$

We define the crank of $\tilde{\pi}$ as

$$\operatorname{crank}(\tilde{\pi}) := \sum_{i=1}^{24} i c_i(\tilde{\pi}). \tag{2.6}$$

Let $N_{24}(m,n)$ denote the number of 24-coloured partitions $\tilde{\pi}$ of $n$ with crank $m$ and let $N_{24}(k,t,n)$ denote the number of 24-coloured partitions $\tilde{\pi}$ of $n$ with crank congruent to $k$ modulo $t$. Unfortunately it is *not* true that

$$N_{24}(-m,n) = N_{24}(m,n). \tag{2.7}$$

We could have defined our crank differently so that (2.7) holds. However, this other crank explains (1.5) and (2.3) but fails to explain (2.1) and (2.2). Luckily, an analogue of (2.7) holds for $N_{24}(k,t,n)$ for the values of $t$ we are concerned with. By considering each of the following two recolouring involutions:

$$\begin{aligned} &\text{Involution}_1: \quad i \mapsto 25-i \quad \text{for} \quad 1 \le i \le 24 \\ &\text{Involution}_2: \quad i \mapsto 24-i \quad \text{for} \quad 1 \le i \le 23 \quad \text{and} \quad 24 \mapsto 24, \end{aligned}$$

we have

$$N_{24}(-k,t,n) = N_{24}(k,t,n) \quad \text{when } t \text{ is a divisor of 24 or 25.} \tag{2.8}$$

Then we have

**Theorem 1.** *For $n \equiv 1,3$ or 4 (mod 5) and $n \neq 1$ we have*

$$N_{24}(k,25,n) = \tfrac{1}{5} N_{24}(k,5,n). \tag{2.9}$$

This provides a natural way of dividing the 24-coloured partitions of $n$ into 5 equal classes for $n \equiv 1,3$ or 4 (mod 5) and $n \neq 1$.

**Corollary 1.** *For $0 \le j \le 4$, let $M_{24}(j,n)$ denote the number of 24-coloured partitions of $n$ with crank congruent to $5j$, $5j+1$, $5j+2$, $5j+3$, or $5j+4$ (mod 25). Then*

$$M_{24}(j,n) = \tfrac{1}{5} p_{-24}(n) \tag{2.10}$$

*when $n \equiv 1,3$ or 4 (mod 5) and $n \neq 1$.*

For $n \equiv 3$ or 4 (mod 5) more is true.

**Theorem 2.** *For $n \equiv 3$ or 4 (mod 5) we have*

$$N_{24}(k,5,n) = \tfrac{1}{5} p_{-24}(n), \qquad 0 \le k \le 4. \tag{2.11}$$

Combining the results of Theorems 1 and 2 we find that the residue of the crank mod 25 divides the partitions of $n$ (for $n \equiv 3$ or 4 (mod 5)) into 25 equal classes.

**Theorem 3.** *For $n \equiv 3$ or 4 (mod 5) we have*

$$N_{24}(k,25,n) = \tfrac{1}{25} p_{-24}(n), \qquad 0 \le k \le 24. \tag{2.12}$$

*Proof of Theorem 1.* Let $\zeta = e^{2\pi i/25}$ so that $\zeta^{25} = 1$ and $1+\zeta^5+\zeta^{10}+\zeta^{15}+\zeta^{20} = 0$. The generating function for $N_{24}(m,n)$ is

$$\sum_{n\geq 0}\sum_{m=0}^{24n} N_{24}(m,n)z^m q^n = \prod_{i=1}^{24}(z^i q;q)_\infty^{-1}, \tag{2.13}$$

where $(a;q)_\infty = \prod_{m=1}^{\infty}(1-aq^{m-1})$ and $|q| < 1$. Substituting $z = \zeta$ in (2.13) and proceeding as in [6], Section 2, we find that

$$\sum_{n\geq 0}\left(\sum_{k=0}^{24} N_{24}(k,25,n)\zeta^k\right)q^n = \prod_{n=1}^{\infty}\frac{(1-q^n)}{(1-q^{25n})}. \tag{2.14}$$

But the coefficient of $q^n$ on the left side of (2.14) is

$$\begin{aligned}(n_0-n_5)+\ &(n_1-n_4)\zeta\ +\ (n_2-n_3)\zeta^2\ +\ (n_3-n_2)\zeta^3\ +(n_4-n_1)\zeta^4\\ &+\ (n_6-n_4)\zeta^6\ +\ (n_7-n_3)\zeta^7\ +\ (n_8-n_2)\zeta^8\ +(n_9-n_1)\zeta^9\\ +(n_{10}-n_5)\zeta^{10}&+(n_{11}-n_4)\zeta^{11}+(n_{12}-n_3)\zeta^{12}+(n_{12}-n_2)\zeta^{13}+(n_{11}-n_1)\zeta^{14}\\ +(n_{10}-n_5)\zeta^{15}&+\ (n_9-n_4)\zeta^{16}\ +\ (n_8-n_3)\zeta^{17}\ +\ (n_7-n_2)\zeta^{18}\ +(n_6-n_1)\zeta^{19},\end{aligned}$$

where $n_i = n_i(n) = N_{24}(i,25,n)$. The result then follows from

$$\begin{aligned}\prod_{n=1}^{\infty}\frac{(1-q^n)}{(1-q^{25n})} &= \prod_{n=1}^{\infty}\frac{(1-q^{25n-15})(1-q^{25n-10})}{(1-q^{25n-20})(1-q^{25n-5})}\\ &\quad -q-q^2\prod_{n=1}^{\infty}\frac{(1-q^{25n-20})(1-q^{25n-5})}{(1-q^{25n-15})(1-q^{25n-10})},\end{aligned} \tag{2.15}$$

which is [6], Lemma (3.18), was known to Ramanujan and has been generalised by Atkin and Swinnerton-Dyer ([2], Lemma 6).

*Proof of Theorem 2.* Let $\eta = e^{2\pi i/5}$. We substitute $z = \eta$ into (2.13) to find

$$\sum_{n\geq 0}\left(\sum_{k=0}^{4} N_{24}(k,5,n)\eta^k\right)q^n = \prod_{n=1}^{\infty}\frac{(1-q^n)}{(1-q^{5n})^5}. \tag{2.16}$$

Since the series expansion of $\prod_{n=1}^{\infty}(1-q^n)$ has no terms with exponent congruent to either 3 or 4 modulo 5, as in the proof of (2.3), the result follows.

Now Theorem 3 follows immediately from Theorems 1 and 2.

## 3. Remarks.

We remark that our crank also explains the congruences (2.1) and (2.2). We omit the proof.

**Theorem 4.** *We have*

$$\sum_{k=0}^{3} N_{24}(k,8,n) = \tfrac{1}{2}p_{-24}(n) \qquad \textit{for} \quad n \not\equiv 0 \pmod 8, \tag{3.1}$$

$$N_{24}(k,3,n) = \tfrac{1}{3}p_{-24}(n) \qquad \textit{for} \quad n \not\equiv 0 \pmod 3 \quad \textit{and} \quad 0 \leq k \leq 2. \tag{3.2}$$

Unfortunately our crank fails to explain the mod 7 congruence (2.4).

For restricted $n$ stronger congruences than (2.1) and (2.2) hold:

$$p_{-24}(n) \equiv 0 \pmod{2^7} \quad \text{for } n \equiv 1 \pmod 2 \text{ and } n \neq 1, \tag{3.3}$$

$$p_{-24}(n) \equiv 0 \pmod{3^3} \quad \text{for } n \not\equiv 0 \pmod 3 \text{ and } n \neq 1. \tag{3.4}$$

(3.3) follows from the following $q$-series identity

$$\sum_{n\geq 0} p_{-24}(2n+1)q^n = 24\prod_{n=1}^{\infty}\frac{(1-q^{2n})^{24}}{(1-q^n)^{48}} + 2^{11}q\prod_{n=1}^{\infty}\frac{(1-q^{2n})^{48}}{(1-q^n)^{72}}, \tag{3.5}$$

which follows from a certain well known quadratic modular equation ([16], page 470). (3.4) follows first by observing that the generating function for $p_{-24}(n)$ is congruent to $(q;q)_\infty^3/(q^3;q^3)_\infty^9 \pmod{27}$ and then by using Jacobi's identity for $(q;q)_\infty^3$ ([2], (3.6)). Our crank fails to explain either (3.3) or (3.4). For (3.3) the best we can do is

$$N_{24}(k,8,n) = \tfrac{1}{8}p_{-24}(n) \quad \text{for } n \equiv 1 \pmod 2 \text{ and } 0 \leq k \leq 7. \tag{3.6}$$

We can explain a weaker form of (3.4) but with a different crank. If we define crank$'$ by

$$\text{crank}'(\tilde{\pi}) := \sum_{i=1}^{24}(i+1)c_i(\tilde{\pi}) \tag{3.7}$$

and define $N'_{24}$ in the obvious way then

$$N'_{24}(k,27,n) = \tfrac{1}{27}p_{-24}(n) \qquad \text{for } n \not\equiv 0 \pmod 3,\ n \not\equiv 1 \pmod 9 \text{ and } 0 \leq k \leq 26. \tag{3.8}$$

We omit the proof.

## References


1. G. E. Andrews and F. G. Garvan, 'Dyson's crank of a partition', *Bull. Amer. Math. Soc.* **18** (1988), 167–171.
2. A. O. L. Atkin and P. Swinnerton-Dyer, 'Some properties of partitions', *Proc. London Math. Soc.* (3) **4** (1954), 84–106.
3. F. J. Dyson, 'Some guesses in the theory of partitions', *Eureka* (Cambridge) **8** (1944), 10–15.
4. F. G. Garvan, 'A simple proof of Watson's partition congruences for powers of 7', *J. Austral. Math. Soc. (Series A)* **36** (1984), 316–334.
5. F. G. Garvan, *Generalizations of Dyson's Rank.* (Ph. D. thesis, Pennsylvania State University, 1986.)
6. F. G. Garvan, 'New combinatorial interpretations of Ramanujan's partition congruences mod 5, 7 and 11', *Trans Amer. Math. Soc.* **305** (1988), 47–77.
7. F. G. Garvan, 'Combinatorial interpretations of Ramanujan's partition congruences', *Ramanujan Revisited*, Proc. Centenary Conference, Urbana, Ill. (Academic Press, 1988), 29–45.

8. F. G. Garvan, 'The crank of partitions mod 8, 9 and 10', *Trans Amer. Math. Soc.*, to appear.
9. F. G. Garvan and D. Stanton, 'Sieved partition functions and $q$-binomial coefficients', *Math. Comp.*, to appear.
10. F. G. Garvan, D. Kim and D. Stanton, 'Cranks and $t$-cores', Inventiones Math., to appear.
11. M. Green, J. Schwarz and E. Witten, *Superstring Theory*, Vol. I. (Cambridge University Press, New York, 1987.)
12. V. G. Kač, *Infinite Dimensional Lie Algebras.* (Cambridge University Press, New York, 1985.)
13. C . J. Moreno and A. Rocha-Caridi, 'The exact formula for the weight multiplicities of affine Lie algebras, I', *Ramanujan Revisited*, Proc. of Centenary Conference, Urbana, Ill. (Academic Press, 1988), 111–152.
14. C. J. Moreno, 'Partitions, congruences and Kac-Moody Lie algebras', preprint.
15. S. Ramanujan, 'Some properties of $p(n)$, the number of partitions of $n$', *Collected Papers of S. Ramanujan*, paper 25. (Cambridge Univ. Press, London and New York, 1927; reprinted: Chelsea, New York, 1962.)
16. E. T. Whittaker and G. N. Watson, *A course of Modern Analysis.* (Cambridge University Press, New York, 1927.)

*School of Mathematics, Physics, Computing and Electronics, Macquarie University, New South Wales 2109, AUSTRALIA.*

# UNIVERSAL FAMILIES OF ABELIAN VARIETIES

Alice Silverberg

## Introduction.

In this paper we give a survey of the problem of determining Mordell-Weil groups of universal abelian varieties.

We begin with the pioneering work of Shioda in the early 1970's on elliptic modular surfaces. In many ways, this is the most difficult and intriguing case. Shioda showed that the universal elliptic curve of level $N$ in characteristic zero has only the $N$-torsion in its Mordell-Weil group over the field of elliptic modular functions of level $N$. The elliptic modular case is also the only case in which nontrivial examples are known of universal abelian varieties (in positive characteristic) with infinite Mordell-Weil group.

We then turn to abelian varieties (in characteristic zero) of higher dimension, paying special attention to two simple cases. The first, a direct generalization of the elliptic case, is that of the universal principally polarized abelian variety of dimension $d > 1$ and level $N \geq 3$. Two proofs are sketched (Sections 2.2 and 2.7) of Shioda's conjecture that the Mordell-Weil group over the field of Siegel modular functions of level $N$ is exactly the $N$-torsion ($\cong (\mathbf{Z}/N\mathbf{Z})^{2d}$). These proofs are given as illustrative examples of some general techniques which were given in [12], [13], and [14]. The second case concerns abelian varieties whose endomorphism algebras contain an order in an indefinite division quaternion algebra over $\mathbf{Q}$. The associated moduli spaces are compact Shimura curves. This example and the elliptic modular case considered by Shioda show that more sophisticated techniques are required to deal with the cases of low-dimensional moduli spaces than with high-dimensional ones.

In the final section we give results on Mordell-Weil groups of universal abelian varieties over towers of fields obtained by taking higher and higher level structure.

I would like to thank the mathematicians at Macquarie University for their kind hospitality.

*Notation.* We denote by $I_d$ the $d \times d$ identity matrix.

## 1. Universal elliptic curves.

We will recall some of the work of Shioda and others on elliptic modular surfaces.

### 1.1. *Elliptic modular surfaces over* $\mathbf{C}$.

Fix a positive integer $N \geq 3$, let $\Gamma(N) = \{\gamma \in SL_2(\mathbf{Z}) : \gamma \equiv I_2 \pmod{N}\}$ be the principal congruence subgroup of level $N$, and let $H$ be the Poincaré upper half plane. Let $\Delta = H/\Gamma(N)$ and $W = (H \times \mathbf{C})/(\Gamma(N) \ltimes \mathbf{Z}^2)$ where

$$(\gamma,(m,n))(z,u) = (\gamma(z),(cz+d)^{-1}(u+mz+n))$$

for $\gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma(N), (m,n) \in \mathbf{Z}^2, z \in H$, and $u \in \mathbf{C}$.

We can compactify $\Delta$ by adjoining finitely many cusps, to obtain the modular curve $X(N)$. Similarly, Kodaira's construction of elliptic surfaces ([2], Section 8) yields $B(N)$, the elliptic modular surface of level $N$, a compactification of $W$. Shioda studied elliptic modular surfaces in Part II of [9]. (In fact, Shioda studied the more general case where $\Delta$ is a quotient of $H$ by a subgroup of $SL_2(\mathbf{Z})$ of finite index and not containing $-1$, but for simplicity we will restrict to the cases where the group is a principal congruence subgroup.)

We will omit the dependence on $N$ in our notation, and write $B$ for $B(N), X$ for $X(N)$, etc. One of Shioda's results is:

**Theorem 1** (Shioda). *The group of global sections of the elliptic surface $B$ over the modular curve $X$ is a finite group, isomorphic to $\mathbf{Z}/N\mathbf{Z} \times \mathbf{Z}/N\mathbf{Z}$.*

The generic fibre $E$ of $B$ is an elliptic curve defined over $\mathbf{C}(X)$, the field of modular functions of level $N$. One can identify global sections with $\mathbf{C}(X)$-rational points of the generic fibre. Thus, the above theorem is equivalent to:

**Theorem 1'** (Shioda). *The Mordell-Weil group $E(\mathbf{C}(X))$ is exactly the $N$-torsion on the elliptic curve $E$.*

1.2. *Level three.*

In fact, the universal elliptic curve $E$ of level $N \geq 3$ can be constructed in any characteristic not dividing $N$ (Igusa, [1]). For example, for $N = 3$, and any field $k$ with characteristic not equal to 3, the universal elliptic curve $E \subset \mathbf{P}^2(\overline{k})$ of level three can be defined over $k(\mu)$ by the projective equation:

$$x^3 + y^3 + z^3 = 3\mu xyz$$

in $\mathbf{P}^2$. If $k$ contains three cube roots of unity, then the 3-torsion on $E$ is rational over $k$, and gives the full Mordell-Weil group of $E$ over $k(\mu)$ (as proved by Igusa, [1]).

1.3. *Level four.*

We might expect that in general, the Mordell-Weil group of the universal elliptic curve $E$ of level $N$ will be exactly the $N$-torsion. However, this is not the case, as was shown by Shioda for level four.

For a field $k$ of odd characteristic $p$, the universal elliptic curve of level four can be given by

$$y^2 = x(x-1)(x - \tfrac{1}{4}(\sigma + 1/\sigma)^2),$$

an elliptic curve defined over $k(\sigma)$. If $k$ contains a square root of $-1$, then the torsion in $E(k(\sigma))$ is exactly the 16 points on $E$ of order dividing four. When $p \equiv 1 \pmod 4$, then this is all of $E(k(\sigma))$ ([10]). However, Shioda also proved:

**Theorem 2** (Shioda [11]). *When $p \equiv 3 \pmod 4$, then $E(k(\sigma))$ has rank two, and is isomorphic to $\mathbf{Z} \times \mathbf{Z} \times \mathbf{Z}/4\mathbf{Z} \times \mathbf{Z}/4\mathbf{Z}$.*

For example, for $p = 3$, the points

$$(\sigma^2, \sigma^2 - 1) \quad \text{and} \quad ((1-i)(\sigma - i), (1+i)(\sigma+1)(\sigma - i)(\sigma - 1 + i)/\sigma)$$

are independent points of infinite order in $E(k(\sigma))$ ([10]).

Further, if $4 \mid N$ and $3 \nmid N$, then the rank of the Mordell-Weil group of the universal elliptic curve of level $N$ over the field of elliptic modular functions of level $N$ in characteristic three is at least two ([10], Corollary, p. 156).

1.4. *Levels two and one.*

For any field $k$ of characteristic $\neq 2$, there is an "almost" universal elliptic curve of level two, namely the Legendre cubic:

$$y^2 = x(x-1)(x-\lambda)$$

defined over the field $k(\lambda)$. See [1] and [4] for proofs that the $k(\lambda)$-rational points on this elliptic curve are exactly the 2-torsion.

For level one, there is no universal elliptic curve. However, the curve:

$$E_j : y^2 = x^3 - 27j(x-1)/4(j-1)$$

is an elliptic curve defined over $\mathbf{Q}(j)$, with absolute invariant $J = 1728j$, and every elliptic curve over $\mathbf{C}$ of $J$-invariant $J_0 = 1728j_0$ is isomorphic to $E_{j_0}$ if $j_0 \in \mathbf{C} \setminus \{0,1\}$. Hazama [3] proved that $E_j(\mathbf{Q}(j))$ has rank one, and further, for every $j_0 \in \mathbf{Q} \setminus \{0,1\}$ the point (1,1) is a point of infinite order on the specialization curve $E_{j_0}$. However, one can show that for the curve

$$E_g : y^2 = 4x^3 - g_2 x - g_3,$$

defined over the function field in two variables $\mathbf{Q}(g_2, g_3)$, we have $E_g(\mathbf{Q}(g_2, g_3)) = 0$.

## 2. Abelian varieties.

We can perform similar constructions for universal abelian varieties (at least in characteristic zero). Here, along with some level structure, we must fix additional structure (a polarization).

2.1. *Polarized abelian varieties of level $N$.*

The fundamental example is the universal principally polarized abelian variety of dimension $d$ and level $N$. Fix integers $d > 1$ and $N \geq 3$. Let $J = \begin{pmatrix} 0 & I_d \\ -I_d & 0 \end{pmatrix}$, and let $Sp(d, \mathbf{R})$ denote the symplectic group $\{\gamma \in M_{2d}(\mathbf{R}) : {}^t\gamma J \gamma = J\}$. Let

$$Sp(d, \mathbf{Z}) = Sp(d, \mathbf{R}) \cap M_{2d}(\mathbf{Z}),$$

and let

$$\Gamma = \{\gamma \in Sp(d, \mathbf{Z}) : \gamma \equiv I_{2d} \pmod{N}\}.$$

The group $Sp(d, \mathbf{R})$ acts by fractional linear transformations on the Siegel upper half space

$$H_d = \{Z \in M_d(\mathbf{C}) : {}^tZ = Z \text{ and } \mathrm{Im}(Z) \text{ is positive definite}\}.$$

Let

$$\Delta = H_d/\Gamma \text{ and } W = (H_d \times \mathbf{C}^d)/(\Gamma \ltimes \mathbf{Z}^{2d}).$$

The complex manifolds $\Delta$ and $W$ can be realized as quasi-projective varieties [8]. (For $d = 1$, we recover the situation of Section 1.1.)

Analytically, the fibre in $W$ over the class of the point $Z \in H_d$ is equal to the complex torus $\mathbf{C}^d/(Z \ \ I_d)\mathbf{Z}^{2d}$, with a principal polarization given by the Riemann form $E_Z : \mathbf{C}^d \times \mathbf{C}^d \to \mathbf{R}$ defined by:

$$E_Z((Z \ \ I_d)a, (Z \ \ I_d)b) = {}^taJb, \text{ for } a, b \in \mathbf{R}^{2d},$$

and with a basis for the $N$-torsion given by the points $(Z \ \ I_d)e_i/N$ where $\{e_i\}_{i=1}^{2d}$ is the canonical basis for $\mathbf{Z}^{2d}$. (We are using the fact that $\mathbf{C}^d = (Z \ \ I_d)\mathbf{R}^{2d}$ for every $Z \in H_d$.) Thus, the fibres are principally polarized abelian varieties of dimension $d$, with level $N$ structure.

The group of holomorphic sections of $W$ over $\Delta$ is isomorphic to the group of $\mathbf{C}(\Delta)$-rational points of the generic fibre $A$ ([12], Corollary 2.2). In [9], Shioda conjectured:

**Theorem 3** ([12]). *If $A$ is the universal principally polarized abelian variety of level $N$ as above, then $A(\mathbf{C}(\Delta)) \cong (\mathbf{Z}/N\mathbf{Z})^{2d}$.*

We sketch two proofs of this theorem, one in Section 2.2 and one in Section 2.7.

2.2. *Sketch of proof of Theorem 3.*

We will pull back holomorphic sections of $W$ over $\Delta$, to sections of elliptic modular surfaces over modular curves, to reduce Theorem 3 to Shioda's Theorem (Theorem 1). For every $\gamma \in Sp(d, \mathbf{Q})$, define a map $\varepsilon_\gamma : H \to H_d$ by $\varepsilon_\gamma(z) = \gamma(zI_d)$. The abelian variety $A_{\varepsilon_\gamma(z)}$ which is the fibre over $\varepsilon_\gamma(z)$ is isogenous to $E_z^d$, where $E_z = \mathbf{C}/(\mathbf{Z}z + \mathbf{Z})$, with an isogeny, $\lambda_{z,\gamma}$, defined by multiplication by ${}^t(Cz + D)M$, where $\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ and $M$ is a positive integer such that $M\gamma$ and $M\gamma^{-1}$ are in $M_{2d}(\mathbf{Z})$.

If $f$ is a section of the fibre variety, $f$ can be lifted to a holomorphic map $H_d \to \mathbf{C}^d$. Let $\pi_j : \mathbf{C}^d \to \mathbf{C}$ be projection of $\mathbf{C}^d$ onto the $j$-th factor for $j = 1, \cdots, d$. Then the compositions

$$g_{j,\gamma}(z) = \pi_j \circ \lambda_{z,\gamma} \circ f \circ \varepsilon_\gamma(z)$$

induce rational sections of the elliptic modular surface $B(NM^2)$ over the modular curve $X(NM^2)$ of Section 1.1 above. Since rational sections over a curve extend to global holomorphic sections, we can apply Theorem 1 to show that the images of the maps $g_{j,\gamma}$ are points of order $NM^2$ on the elliptic curves $E_z$.

We can conclude that the images under the section $f$ of elements in the set $Q = \{\varepsilon_\gamma(z) : z \in H, \gamma \in Sp(d, \mathbf{Q})\}$ are points of finite order in the fibres $A_{\varepsilon_\gamma(z)}$. However,

these orders depend on the numbers $M$, the "denominators" of the group elements $\gamma$, and these integers $M$ must go to infinity in any subset of $Q$ dense in $H_d$. Initially, this appears to be an obstruction to proving $f$ is a section of finite order.

Nevertheless, by restricting the points $z \in H$ to belong to imaginary quadratic fields (or even one imaginary quadratic field), we obtain a subset $P$ of $Q$, dense in $H_d$, whose associated fibres are abelian varieties with complex multiplcation. The theory of complex multiplication allows us to uniformly bound the orders of the images under the section $f$ of the points in $P$ [12]. Thus, $A(\mathbf{C}(\Delta))$ is finite. Theorem 3 then follows on observing that $A(\mathbf{C}(\Delta))_{\text{torsion}} \cong H^0(\Gamma, \mathbf{R}^{2d}/\mathbf{Z}^{2d}) \cong (\mathbf{Z}/N\mathbf{Z})^{2d}$.

The above sketch is the simplest case of a very general method for proving finiteness of Mordell-Weil groups of universal abelian varieties (see [12] and [13]).

2.3. *Quaternionic multiplication.*

To give another example, which must be treated by different methods, let us consider universal two-dimensional polarized abelian varieties with quaternionic multiplication (and with level struture). Let $\mathcal{L} \subset M_2(\mathbf{R})$ be an indefinite quaternion algebra over $\mathbf{Q}$, let $\mathcal{O}$ be an order in $\mathcal{L}$, suppose $3 \leq N \in \mathbf{Z}$, and let

$$\Gamma = \{\gamma \in \mathcal{L} : \gamma\mathcal{O} = \mathcal{O}, (1-\gamma)\mathcal{O} \subset N\mathcal{O}\}.$$

Let $\Delta = H/\Gamma$ and let $W = (H \times \mathbf{C}^2)/(\Gamma \times \mathcal{O})$, where

$$(\gamma, \alpha)(z, u) = (\gamma(z), (cz+d)^{-1}(u + (z \;\; 1)\alpha))$$

for $\gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma, \alpha \in \mathcal{O}, z \in H$, and $u \in \mathbf{C}^2$.

The fibres of $W$ over $\Delta$ are two-dimensional abelian varieties. The fibre over $z \in H$ is given analytically by the complex torus $\mathbf{C}^2/(z \;\; 1)\mathcal{O}$ with polarization given by the Riemann form

$$E_z((z \;\; 1)\alpha, (z \;\; 1)\beta) = \mathrm{Tr}\left(\alpha \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} {}^t\beta\right),$$

for $\alpha, \beta \in M_2(\mathbf{R})$ (or more precisely by a real multiple of $E_z$ such that the image of $(z \;\; 1)\mathcal{O} \times (z \;\; 1)\mathcal{O}$ is in $\mathbf{Z}$). Note that $(z \;\; 1)M_2(\mathbf{R}) = \mathbf{C}^2$ for every $z \in H$. The generic fibre may be thought of as a universal polarized abelian variety with level $N$ structure and with "quaternionic" multiplication by the order $\mathcal{O}$.

If $\mathcal{L}$ is a division algebra, then $\Delta$ is a compact curve; otherwise, $\mathcal{L}$ is isomorphic to $M_2(\mathbf{Q})$ and we are essentially in the case of Section 1.1 (the fibre over $z$ is then two copies of the elliptic curve $\mathbf{C}/(\mathbf{Z}z+\mathbf{Z})$). Let us restrict to the case where $\mathcal{L}$ is a division algebra. Then $W$ can be realized as a projective variety, and the group of holomorphic sections is isomorphic to the group of points of the generic fibre $A$ rational over the function field of the curve $\Delta$ (see [12]).

**Theorem 4** ([12]). *$A(\mathbf{C}(\Delta))$ is isomorphic to the finite group $H^0(\Gamma, M_2(\mathbf{R})/\mathcal{O})$.*

The group $H^0(\Gamma, M_2(\mathbf{R})/\mathcal{O})$ will be at least of order $N^4$, but may be larger (see [14], Section 5) for situations where there is extra 2-torsion.

2.4. *Sketch of proof of Theorem 4.*

A holomorphic section is given by a holomorphic map $h : H \to \mathbf{C}^2$ such that for every $\gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma$, there exists $y_\gamma \in \mathcal{O}$ with

$$(cz + d)h(\gamma(z)) = h(z) + (z \quad 1)y_\gamma. \tag{1}$$

Set $h = (h_1, h_2)$. Then (1) implies that the second derivatives $h_1''$ and $h_2''$ are in the space $S_3(\Gamma)$ of modular (cusp) forms of weight three with respect to $\Gamma$. In fact, we have:

**Lemma 5.** *If $f : H \to \mathbf{C}$ is holomorphic, then the following are equivalent:*
(a) $f \in S_3(\Gamma)$,
(b) *$f = g''$ for a holomorphic map $g : H \to \mathbf{C}$ such that for every $\gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma$, there exist complex numbers $p$ and $q$ with $(cz + d)g(\gamma(z)) = g(z) + pz + q$.*

One can use (1) to show ([12], Theorem 3.3) that $h''$ must be in the kernel of the isomorphism $S_3(\Gamma) \oplus S_3(\Gamma) \xrightarrow{\sim} H^1(\Gamma, M_2(\mathbf{R}))$, induced by the Eichler-Shimura cohomology isomorphism. Therefore $h'' = 0$, and so $h(z) = (z \quad 1)\alpha$ for some $\alpha$ in $M_2(\mathbf{R})$. Now (1) says exactly that $\alpha \in H^0(\Gamma, M_2(\mathbf{R})/\mathcal{O})$.

An embellishment of this argument ([12]), which uses Eichler-Shimura isomorphisms for modular forms of weights two and three, can be used to show the finiteness of the group of holomorphic sections for other fibre system of abelian varieties, when the universal cover of the base variety is a product of upper half planes $H$ (this includes the classical Hilbert modular varieties as a special case).

2.5. *Symplectic embeddings.*

More generally, suppose $V$ is a real vector space of dimension $2d$, $E$ is a nondegenerate alternating bilinear form on $V$ and $L$ is a lattice in $V$ such that $E(L, L) \subset \mathbf{Z}$. Suppose $I_0$ is a complex structure on $V$ such that $E(u, I_0 v)$ is symmetric and positive definite, and let $K'$ be the centralizer of $I_0$ in the symplectic group $Sp(V, E)$. Suppose $G$ is a connected semisimple real Lie group defined over $\mathbf{Q}$ of hermitian type, $K$ is a maximal compact subgroup of $G$, and $D = G/K$ is the associated symmetric domain. Suppose we have a faithful irreducible representation $\rho : G \to Sp(V, E)$, defined over $\mathbf{Q}$ and preserving the Cartan decompositions (thus, $\rho(K) \subset K'$ and $\rho$ induces a holomorphic map on the associated hermitian symmetric spaces). Suppose $\Gamma$ is a torsion-free arithmetic subgroup of $G$ such that $\rho(\Gamma)L \subset L$. Then one can view $W = (D \times V)/(\Gamma \times L)$ as a complex manifold (see [7] and [8]), and also as a (quasi-projective) fibre variety over the quasi-projective variety $\Delta = D/\Gamma$, where the fibres are $d$-dimensional abelian varieties.

Let $R$ be the complex manifold given by $(D \times V)/\Gamma$, let $Z = (D \times L)/\Gamma$, let $S(W)$ (respectively, $S(R)$) be the sheaf of germs of holomorphic sections of $W$ (respectively, $R$) over $\Delta$, and let $S(Z)$ be the sheaf of germs of locally holomorphic sections of $Z$ over $\Delta$. The short exact sequence of sheaves $0 \to S(Z) \to S(R) \to S(W) \to 0$ induces the long exact sequence in cohomology:

$$0 \to H^0(\Delta, S(Z)) \to H^0(\Delta, S(R)) \to H^0(\Delta, S(W)) \to H^1(\Delta, S(Z)) \to H^1(\Delta, S(R)) \to \cdots$$

Suppose that either $\dim(\Delta) > 1$ or $\Delta$ is compact. Then the group of holomorphic sections of $W$ over $\Delta$, $H^0(\Delta, S(W))$, is isomorphic to the group of $\mathbf{C}(\Delta)$-rational points on the generic fibre $A$, an abelian variety defined over $\mathbf{C}(\Delta)$ (see [12], Corollary 2.2).

The short exact sequence $0 \to L \to V \to V/L \to 0$ induces a long exact sequence in group cohomology, which relates to the sequence above as follows:

$$\begin{array}{ccccccccc}
L^\Gamma & \to & V^\Gamma & \to & H^0(\Gamma, V/L) & \to & H^1(\Gamma, L) & \to & H^1(\Gamma, V) \\
\downarrow\cong & & \downarrow & & \downarrow & & \downarrow\cong & & \downarrow f \\
H^0(\Delta, S(Z)) & \to & H^0(\Delta, S(R)) & \to & H^0(\Delta, S(W)) & \to & H^1(\Delta, S(Z)) & \to & H^1(\Delta, S(R)) \\
 & & & & \downarrow\cong & & & & \\
 & & & & A(\mathbf{C}(\Delta)) & & & &
\end{array} \tag{2}$$

where the second and third vertical arrows are injections. (The reader may supply extra terms '$0 \to$' on the left and '$\to \cdots$' on the right.)

**Proposition 6** ([14]). *The following are equivalent:*
(a) $A(\mathbf{C}(\Delta))$ *is a finitely generated abelian group.*
(b) $H^0(\Gamma, V) = 0$.
(c) $H^0(\Delta, S(R)) = 0$.

Let $V_{\mathbf{Q}} = L \otimes_{\mathbf{Z}} \mathbf{Q}$. We can identify the torsion subgroup of $A(\mathbf{C}(\Delta))$ with $H^0(\Gamma, V_{\mathbf{Q}}/L)$ ([12], Proposition 2.6). If $V^\Gamma = 0$, then $H^0(\Gamma, V/L)$ is a finite group and is isomorphic to $H^0(\Gamma, V_{\mathbf{Q}}/L)$. Therefore we have:

**Proposition 7.** $A(\mathbf{C}(\Delta))$ *is finite if and only if*

$$V^\Gamma = 0 \quad \text{and} \quad A(\mathbf{C}(\Delta)) \cong H^0(\Gamma, V/L).$$

2.6. *Some conjectures.*

We can conjecture vertical isomorphisms in diagram (2). Conjectures 1 and 2 below are true whenever $A(\mathbf{C}(\Delta))$ is finite, and so in particular are true in the examples of Sections 2.1 and 2.3.

**Conjecture 1.** $A(\mathbf{C}(\Delta)) \cong H^0(\Gamma, V/L)$.

If Conjecture 1 is true, then $A(\mathbf{C}(\Delta))$ is finite if and only if it is finitely generated (by Propositions 6 and 7). Conjecture 1 implies (by diagram (2)):

**Conjecture 2.** $V^\Gamma$ *is isomorphic to* $H^0(\Delta, S(R))$.

We have a natural map $f : H^1(\Gamma, V) \to H^1(\Delta, S(R))$ in diagram (2), and we can conjecture

**Conjecture 3.** *The map* $f$ *is injective.*

Conjectures 2 and 3 imply Conjecture 1. In particular, Conjecture 3 and $V^\Gamma = 0$ imply Conjecture 1. Moreover, we could make the stronger conjecture:

**Conjecture 4.** *The map $f$ is an isomorphism.*

2.7. *Finiteness results.*

Let $\delta$ be the composition:

$$H^0(\Delta, S(W)) \to H^1(\Delta, S(Z)) \xrightarrow{\sim} H^1(\Gamma, L) \to H^1(\Gamma, V).$$

Then $V^\Gamma = 0$ and $\delta = 0$ imply Conjecture 1 and the finiteness of $A(\mathbf{C}(\Delta))$. In many cases $H^1(\Gamma, V)$ is known to vanish (by theorems of Raghunathan), so the map $\delta$ is automatically zero ([5], [6]). In particular, this is the case when $\Gamma \subset Sp(d, \mathbf{Z})$, $L = \mathbf{Z}^{2d}$ and $A$ is the universal principally polarized abelian variety of level $N$ and dimension $d > 1$, as in Section 2.1. Here, $H^0(\Gamma, V/L) \cong (N^{-1}\mathbf{Z}/\mathbf{Z})^{2d}$. This yields a second proof of Shioda's Conjecture (Theorem 3).

One case in which $H^1(\Gamma, V)$ does not vanish is the case of abelian varieties with quaternionic multiplication given in Section 2.3. Here, $H^1(\Gamma, V) \cong S_3(\Gamma) \oplus S_3(\Gamma)$, but we have already shown Conjecture 1 in this case (Theorem 4) in Section 2.4.

2.8. *Mordell-Weil groups in towers of function fields.*

Suppose we have data $V, L, \Gamma$, etc. as in Section 2.5. For each arithmetic subgroup $\Gamma'$ of $G$ which is normal in $\Gamma$, consider the system over $\Delta' = D/\Gamma'$ associated to the data with $\Gamma'$ replacing $\Gamma$. The generic fibre $A'$ of this new fibre system is isomorphic over $\mathbf{C}(\Delta')$ to $A$, the generic fibre of the original fibre system. Let $K$ be the compositum of all the fields $\mathbf{C}(\Delta')$, as $\Gamma'$ ranges over arithmetic subgroups of $G$ normal in $\Gamma$.

**Theorem 8.** *Suppose $A(\mathbf{C}(\Delta')) \cong H^0(\Gamma', V_\mathbf{Q}/L)$ for every $\Gamma'$ as above. Then $A(K) \cong V_\mathbf{Q}/L$.*

*Proof.* $A(K) = \bigcup_{\Gamma'} A'(\mathbf{C}(\Delta')) = \bigcup_{\Gamma'} H^0(\Gamma', V_\mathbf{Q}/L) = V_\mathbf{Q}/L.$

The hypothesis of Theorem 8 is equivalent to the finiteness of all the groups $A(\mathbf{C}(\Delta'))$. We can apply this theorem to all the examples (see Sections 1.1, 2.1, 2.3) of this paper, and also to the fibre systems in [12], [13], and [14], and obtain results of the following type.

**Corollary 9.** *If $N \geq 3$, $E_N$ is the universal elliptic curve of level $N$ and $K$ is the field of elliptic modular functions (of all levels), then $E_N(K) \cong (\mathbf{Q}/\mathbf{Z})^2$.*

*Proof.* Here

$$K = \bigcup_{M \in \mathbf{Z}^+} \mathbf{C}(X(MN)) = \bigcup_{M \in \mathbf{Z}^+} \mathbf{C}(X(M)),$$

since $\mathbf{C}(X(M)) \subset \mathbf{C}(X(MN))$ and now Shioda's Theorem (Theorem 1′) allows us to apply Theorem 8.

For universal principally polarized abelian varieties of dimension $d$ (and level structure), the Mordell-Weil group over the compositum of fields is $(\mathbf{Q}/\mathbf{Z})^{2d}$. For universal

two-dimensional abelian varieties with quaternionic multiplication by an order $\mathcal{O}$ in an indefinite quaternion algebra $\mathcal{L}$ over $\mathbf{Q}$, the Mordell-Weil group over the compositum is $\mathcal{L}/\mathcal{O}$.

## References


1. J. Igusa, 'Fibre systems of Jacobian varieties (III. Fibre systems of elliptic curves)', *Amer. J. Math.* **81** (1959), 561–577.
2. K. Kodaira, 'On compact analytic surfaces. II, III', *Ann. of Math.* **77** (1963) 563–626, **78** (1963), 1–40.
3. F. Hazama, 'On the Mordell-Weil group of certain elliptic curves', *Proc. Japan Acad. Ser. A* **55** (1979), 412–416.
4. M. Ohta, 'The rational points of $Y^2 = X(X-1)(X-\lambda)$ over $k(\lambda)$ with transcendental $\lambda$', *J. Fac. Sci. Univ. Tokyo Sect. IA Math.* **20** (1973), 383–385.
5. M. S. Raghunathan, 'On the first cohomology of discrete subgroups of semisimple Lie groups', *Amer. J. Math.* **87** (1965), 103–139.
6. M. S. Raghunathan, 'Cohomology of arithmetic subgroups of algebraic groups, I', *Ann. of Math.* **86** (1967), 408–424; II, *ibid.* **87** (1968), 279–304.
7. I. Satake, *Algebraic structures of symmetric domains.* (Publ. Math. Soc. Japan **14** (1980), Iwanami, Tokyo and Princeton University Press.)
8. G. Shimura, 'Moduli and fibre systems of abelian varieties', *Ann. of Math.* **83** (1966), 294–338.
9. T. Shioda, 'On elliptic modular surfaces', *J. Math. Soc. Japan* **24** (1972), 20–59.
10. T. Shioda, 'On rational points of the generic elliptic curve with level $N$ structure over the field of modular functions of level $N$', *J. Math. Soc. Japan* **25** (1973), 144–157.
11. T. Shioda, 'Algebraic cycles on certain $K3$ surfaces in characteristic $p$', *Manifolds–Tokyo 1973, Proc. Internat. Conf. Tokyo*, 357–364. (Univ. Tokyo Press, Tokyo, 1975.)
12. A. Silverberg, 'Mordell-Weil groups of generic abelian varieties', *Invent. Math.* **81** (1985), 71–106.
13. A. Silverberg, 'Mordell-Weil groups of generic abelian varieties in the unitary case', *Proc. Amer. Math. Soc.* **104** (1988), 723–728.
14. A. Silverberg, 'Cohomology of fibre systems and Mordell-Weil groups of abelian varieties', *Duke Math. J.* **56** (1988), 41–46.


*Thomas J. Watson Research Centre, IBM,*
*PO Box 218, Yorktown Heights, New York 10598, USA.*