Metaheuristic Search as a Cryptological Tool

John Andrew Clark

.
Submitted for the degree of Doctor of Philosophy

.
University of York
Department of Computer Science
December, 2001

Abstract

Cryptology is a thriving research area of great practical importance. It is a fundamental building block of communications security. Metaheuristic optimisation techniques such as simulated annealing and genetic algorithms have found successful application in a huge number of fields. However, their application to leading edge industrial-strength cryptology has been slight. The power of metaheuristic search is, however, greatly under-estimated. The research reported here shows how a range of modern-day cryptological problems can be attacked successfully using metaheuristic search. Along the way, the work provides the cryptological researcher with many new approaches to applying metaheuristic search techniques.

Acknowledgements

The author would like to thank Dr Jeremy Jacob for his support and encouragement to complete this thesis and to the Department of Computer Science for the sabbatical during which much of the groundwork was laid. I should like to thank Professor Colin Runciman for advice given in his role as assessor; this had a significant impact on my actually completing a thesis. I should like to thank: Dr Subhamoy Maitra for providing advice on correlation immune functions (and change of basis in particular); Dr William Millan for information on best achieved correlation immunity results to date, for providing fast ANF code and for alerting me to leading-edge literature on Boolean functions (and indeed for carrying out much of the work that interested me in this field in the first place); and DERA for sponsoring work that led to the heuristic evolution of security protocols. Thanks must also go to IBM Hursley, DERA, the UK Civil Service, the EPSRC's SEMINAL network and the Security Research Centre in Brisbane for invitations to speak on cryptology and metaheuristic search.

John A Clark
December 2001

Author's Declaration

This thesis is the work of John Andrew Clark and was carried out at the University of York. Work appearing here (or closely relating to it) has appeared in print as follows:

- Searching for a Solution: Engineering Tradeoffs and the Evolution of Provably Secure Protocols. John A Clark and Jeremy L Jacob. In Proceedings of the 2000 IEEE Symposium on Security and Privacy. May 2000 [22].
- Two-Stage Optimisation in the Design of Boolean Functions. John A Clark and Jeremy L Jacob. In Proceedings of the 5th Australasian Conference on Information Security and Privacy. July 2000 [23].
- Protocols are Programs Too: the Metaheuristic Search for Security Protocols. John A Clark and Jeremy L Jacob. Information and Software Technology. Special Issue on Metaheuristic Search for Software Engineering. December 2001 [24].

The review work preceding the research in Chapter 6 gave rise to the following publications:

- On The Security of Recent Protocols. John A Clark and Jeremy L Jacob. Information Processing Letters 56(3). October 1995 [20].
- Attacking Authentication Protocols. John A Clark and Jeremy L Jacob. High Integrity Systems 1(5). August 1996 [21].

The major review of protocols 'A Survey of Authentication Literature' is available on the author's web-site (www.cs.york.ac.uk/ jac) and is widely used by the international protocols community. Its library of protocols has now been formalised by Clóvis Freire Júnior of the University of Brasilia. [1] Parts of the work have also been presented at the following seminars and workshops: Security Research Center Brisbane (May 2000); DERA EU-US Security Workshop (June 2000); UK Civil Service (Dec 2000); IBM Hursley Institute of Technology Week (June 2001); and the SEMINAL network meeting (July 2001).

[1] Available at http://www.csl.sri.com/users/millen/capsl/library.html

Contents

List of Tables

List of Figures

Chapter 1

Introduction — A Tale of Two Communities

This chapter asks why the cryptology and metaheuristic search communities have interacted so little. It proposes that the power of metaheuristic search for cryptological applications is significantly underestimated.

1.1 Metaheuristic Search and Cryptology

The research reported in this thesis concerns two topics: metaheuristic search and cryptology. Metaheuristic search is concerned with the development and application of general purpose optimisation techniques and has been successfully used across many scientific, engineering and commercial domains. Cryptology is concerned with the making (cryptography) and breaking (cryptanalysis) of schemes to guarantee certain properties of data (confidentiality, integrity, authenticity etc.) and contributes significantly to the practical and intellectual underpinnings for communications security. Each is a vibrant area of research in its own right with its own research community, its own acknowledged pioneers, its own body of fundamental results and its own priorities. There are very few applications of metaheuristic search techniques to modern-day cryptological design or analysis problems. This is a little surprising since the metaheuristic search and cryptology research communities seem, at a fundamental level, to share one major interest — solving computationally 'hard' problems. The research reported in this thesis shows that the power of metaheuristic search techniques for cryptological applications is significantly underestimated and that current apparent ambivalence is misguided.

1.1.1 Metaheuristic Search — When Very Good is Usually Good Enough

Mathematics remains the most powerful tool in science and engineering. A vast number of techniques have been developed to solve problems posed. These techniques often provide exact answers. There is, for example, a *formula* for the roots of a quadratic equation. Yet many practical problems do not seem amenable to such clinical dispatch. The well-known Traveling Salesman Problem (TSP) is a good example:

> Consider a set of N cities, indexed $1 \ldots N$. Each pair $(i,\ j)$ of cities is connected by a road of length d_{ij}. A salesman lives in town 1. Starting from town 1, the salesman must carry out a tour, visiting each town in turn, and then return home. In what order should he visit the cities to give the shortest round tour?

There is no known efficient method for finding a minimal length tour of a large number of cities. Enumeration over all tours would reveal the answer eventually but since there are $\frac{(N-1)!}{2}$ possible tours this approach rapidly becomes infeasible. In practice, an optimal solution to such problems is not expected. Rather, the solution space is navigated in a practically effective way to reach excellent, but not *necessarily* optimal, solutions. This is sensible since such problems are often concerned with efficient use of resources. In practice, a planner is not asked 'What is the shortest length tour?' He or she is asked 'What is the best tour you can suggest within a reasonable time ?'

Exchanging guarantees of optimality for computational tractability in this way is at the heart of metaheuristic search. Often drawing loose inspiration from natural processes, researchers have created combinatorial search techniques that can produce effective answers where other techniques fail. Techniques such as simulated annealing [64] (based loosely on the cooling process of molten metals) and genetic algorithms [41] (based loosely on Darwinian evolution) have seen effective application across a huge range of disciplines. Metaheuristic search is a success.

1.1.2 Cryptology — Delivering Service in the Face of Opposition

Digital communication is increasingly replacing face to face contact and direct physical exchange in transactions. Internet shopping is now high profile, many major payments in shops and supermarkets are made by credit or debit card, electronic cash (e.g. Mondex) is now emerging on the horizon, auctions are being held over the World Wide Web and a great deal of day-to-day communication is effected by email. The non-digital world has developed mechanisms to ensure interactions take place in an appropriate manner. We send confidential messages by special courier, we have passports against which our faces may be checked, we sign documents in the presence of esteemed members of society who may subsequently confirm any agreements made if there is a dispute. Stockbrokers routinely have their telephone calls taped so that disputes about what was agreed at some point may be resolved. The notes in one's wallet may be held to the light to reveal watermarks and other indicators of authenticity. Moving to the digital world does not relieve us of providing similar guarantees. We cannot see the people with whom we are interacting and consequently issues of trust must arise. Since communications media are generally shared between many parties, many of whom we may have little reason to trust, we must cater for the possible subversion of our communications in transit. We need to develop means of transacting that ensure legitimate expectations are met despite a potentially very hostile environment that is the medium. There will also be limits to how far legitimate parties in transactions will be trusted.

Cryptology is at the heart of providing such guarantees. The task is not easy. The very nature of security makes more difficult the task at hand. Whereas most disciplines solve tasks unimpeded by external agents, the cryptographer must develop techniques that are resilient to perverse, malicious and potentially well-funded attempts to subvert his or her efforts (i.e. break the system). In contrast, although a genetic algorithms researcher might well compete with colleagues for computation time, it is unlikely he will face malicious attempts to subvert his techniques in action!

Cryptographers are, in a sense, concerned with creating problems that are artificially hard, so hard that an enemy will not be able to solve them. Suppose an Embassy encrypts diplomatic communications using a particular cryptosystem and a particular secret key K. Without knowledge of the secret key information it should be impractical for an enemy to determine the contents of any message sent within its useful life. Having intercepted the encrypted text (ciphertext) in transit, an enemy could decrypt with each possible secret key in turn (generally referred to as a 'brute force' attack) to determine the one actually used for encryption (the correct key will produce the original and presumably intelligible text). If the secret key space is of sufficient size, this attack is infeasible. The problem is just too hard to solve in this way. Brute force, however, is the least sophisticated of attacks. There is an armory of devices available to the professional cryptanalyst and a successful cryptosystem must resist each. A large keyspace may protect against brute force attack, but is no guarantee that a system cannot be broken by more sophisticated means. In practice, cryptosystem designers aim to make breaking systems using known types of attack infeasible (and in some cases provably so), aim to reduce features that might form the basis of an attack, or else rely on past experience to justify unproven assumptions (e.g. the difficulty of factoring).

Cryptographers aim to create systems (encryption systems, protocols etc.) that will provide adequate security now and also for some time in the future. These systems must also be efficient. There are tradeoffs to be made. For example, RSA with 1024 bit keys is more secure than RSA with 256 bit keys but there is a performance price to be paid. Security drives one to use longer keys and efficiency drives one to use shorter ones. Cryptographers steer a careful course between these (perhaps dangerously) competing objectives. The mainstay of commercial cryptography, the Data Encryption Standard [90], has now fallen to the raw computing power available today [39]. Inability to foresee future computational resources is a major issue as we shall see later.

1.1.3 Engagement but No Marriage Prospects

One might think that the metaheuristic search and cryptographic communities would work closely. A survey of the literature would suggest that this is not case. Metaheuristic searchers occasionally foray into cryptography and cryptographers occasionally return the compliment. Yet there has been little application of metaheuristic search techniques to modern-day ‘industrial-strength’ cryptological problems.

Cryptographic security is a very exciting and commercially highly relevant area. It is high profile and attracts some of the best mathematical minds. Breaking the security of a major cryptosystem such as RSA would make news world-wide. If a modern metaheuristic search technique could achieve results where the best cryptologists in the world have failed this would be a significant boost to the reputation of the field. Why then has the metaheuristic search community shown so little real interest? Similarly, if professional cryptologists really believed these search techniques were powerful, or even useful crypotological tools why haven’t they used them more?

It appears that neither community has much confidence that these techniques are of use for serious modern-day cryptology. It is this perceived lack of real confidence in the techniques that motivates the hypothesis below. The work reported in this thesis is intended to convince you, the reader, that the hypothesis is true.

1.2 The Thesis

1.2.1 Statement and Interpretation of the Hypothesis

The hypothesis is stated below:

> **The power of metaheuristic search as a tool for modern-day cryptological research is significantly greater than currently evidenced in publicly available literature.**

The domain of application is ‘modern-day cryptological research’. There has been a good deal of successful research exploring the use of metaheuristic search to classical cryptological problems (e.g. the cryptanalysis of simple substitution or transposition ciphers). However, such systems are readily breakable by available means, as noted by Bagnall et al. [1]. If professional cryptologists are to become interested in metaheuristic search some attempt must be made to tackle important problems of today. The success of metaheuristic search application so far warrants such a leap. **Only problems of current cryptological research will be addressed.** This is how ‘modern-day’ has been interpreted.

The hypothesis claims that the power of metaheuristic search for modern-day cryptological research is ‘significantly greater than currently evidenced in publicly available literature.’ Where optimisation-based results of other researchers are available straightforward comparisons will be made. However, a professional cryptologist might well ask ‘They may be significantly more powerful than hitherto realised, but are they actually of any real use?’ To address this concern this thesis aims to demonstrate that the techniques are capable of generating results of real interest to cryptologists.

The aim has been to show significant (and in some cases dramatic) improvements by ‘stepping outside the box’. There is little that is ‘clever’ in this thesis. No great optimisation sophistication is needed to understand the work. The general aim is to convey the idea that simple techniques *used in the right way* can give significant and sometimes very surprising results. The thesis contains several novel ways of approaching particular problems. Many of the ideas presented will hopefully find further application. These new approaches enhance the current cryptological optimisation toolkit. Each technical research chapter in this thesis identifies specific toolkit contributions.

There are two target audiences for the research reported here: the metaheuristic search community and the cryptology community. The metaheuristic search community should find clear descriptions of some problems of relevance to modern-day cryptology. The results reported are offered as targets to be surpassed and various open research questions are identified too. A general aim is to interest the metaheuristic search community in modern-day cryptology. The specific problems attacked in this thesis and the associated results will hopefully be of interest in themselves to cryptologists but a major goal of this thesis is to interest the cryptology community in metaheuristic search.

1.2.2 Brief Overview of the Thesis Chapters

The subsequent chapters of this thesis are:

Chapter 2 — Search Techniques and Their Cryptological Uses. This chapter examines how search techniques (and metaheuristic search in particular) have been applied in cryptology. The rationale for attacking specific problems is also presented.

Chapter 3 — Evolving Boolean Functions. This chapter shows how simulated annealing can be used to synthesise Boolean functions with excellent cryptographic properties. This is an important problem for modern-day cryptological research. Simulated annealing is used to disprove several numerical conjectures on particular cryptographic properties. The approaches are extended to the case of functions with multiple outputs (commonly used as and referred to as substitution boxes, or S-boxes for short).

Chapter 4 — Correlation Immunity. In this chapter the work of Chapter 3 is adapted to evolve correlation immune functions (essentially functions where it is difficult to draw exploitable inferences on specific subsets of the inputs based on observation of the single output). A radically different approach is also presented and the text describes how some basic problems in mathematics can be solved using simulated annealing.

Chapter 5 — Side Channels on Analysis. This chapter presents the cryptanalysis of David Pointcheval's identification scheme based on a well-known NP-complete problem [98]. This has been attacked previously using simulated annealing by Lars Knudsen and Willi Meier [65]. The cryptanalyis notions of a timing channel and fault-injection are shown to apply to annealing-based search. Thus, there are 'side channels' on analysis techniques. The results show that the power of metaheuristic search is significantly underestimated for problems that might reasonably be considered 'home ground'.

Chapter 5 — The Heuristic Evolution of Security Protocols. Creating protocols with proven security is often cited as one of the most difficult problems in security research. This chapter shows how a protocol specification written in a belief logic (BAN Logic) can be automatically refined into a series of message specifications written in the same logic. This refinement is carried out using both simulated annealing and genetic algorithms. Furthermore, executing the refinement is a proof its own correctness. This is a huge jump in abstraction compared with other optimisation-based work in cryptography.

Chapter 7 — Evaluation and Conclusions. This chapter examines the achievements of the research reported in this thesis and evaluates the degree to which the hypothesis has been justified.

Appendix A — Supporting Material This contains miscellaneous examples of artifacts produced (principally examples of Boolean functions with interesting properties).

Chapter 2

Search Techniques and Their Cryptological Uses

This chapter provides a brief introduction to general purpose search techniques and how they have been used in cryptographic applications.

2.1 Overview

This chapter provides a brief introduction to guided search techniques and a review of their application to cryptology. Three main application areas have been identified: the cryptanalysis of classical ciphers; the evolution of cryptographic building blocks with desirable properties; and the analysis of crypto-schemes based on NP-complete problems. The more general search context is examined too. There are some very interesting developments afoot.

2.2 Search Problems and Search Methods

The general aim is to find optimal solutions to problems that are structured as a function of some decision variables, perhaps in the presence of some constraints. These can be formulated as:

$$\begin{array}{l}\text{Minimise } f(x) \\ \text{subject to} \\ x \in C \subset X.\end{array}$$

The set X of all possible vectors $x = (x_1, \ldots, x_n)$ of decision variables will generally be referred to as the *solution space* for the search problem at hand. The set C represents the imposition of constraints. Searches may be restricted to consider only elements of C. Alternatively, the problem may be recast as 'Minimise $g(x)$ subject to $x \in X$' where $g(x)$ contains a component that punishes x outside C. Such values of x are said to be 'priced out'. The function f (or g) is generally referred to as a *cost function*. When problems are similarly couched as maximisation problems the term *fitness function* is used. There is complete freedom over which functions are used for the problem at hand. Experience shows that the choice of function is an important success factor in applying many search techniques. The best functions are those that give the best results when used! Unfortunately, it is difficult to predict in advance which functions will work best. Experimentation would seem the order of the day (but the use of very direct or obvious functions seems widespread).

Solution vectors x may be designs (e.g. the truth table of a Boolean function used as a component in a cryptosystem) or analysis artifacts (e.g. a vector of 64 key bits sought by a cryptanalyst). To find them the designer or analyst is free to employ whatever techniques seem most suitable from the vast array available. Solution techniques span a range of sophistication. Cryptologists may often appeal to beautiful mathematics (most typically from the 'Queen of Mathematics', number theory) but they are not above the most gruesome of 'number crunching' approaches where search is concerned. That is a low but good place to start.

2.2.1 Brute Force and Statistical Sampling

Brute force is the least sophisticated of search methods. Given a solution space X suppose a solution possessing some value of a measurable characteristic $f(x)$ is sought. The value of $f(x)$ is evaluated for each $x \in X$ in turn. Either a solution with the sought characteristic value is found or no such solution exists. A secret key K may be sought that can be used to decrypt particular ciphertext blocks $C_1, C_2, \ldots, C_n$ to obtain known plaintext blocks $P_1, P_2, \ldots, P_n$. Exhaustively searching the key space to find such a secret K should be *practically infeasible* for a secure encryption algorithm.

For design tasks the naïve brute force approach will generally be infeasible. For example, consider the Data Encryption Standard (DES) encryption algorithm [90]. This uses eight substitution boxes (S-boxes). Each S-box takes six Boolean inputs and produces four Boolean outputs. An S-box is defined by a table with four rows, each row comprising a permutation of the numbers (0,1,..,15). Since there are $16!^4$ possible S-boxes, obtaining one with desirable cryptographic properties by brute force is clearly a non-starter. Since there are $(16!^4)^8 = (16!)^{32}$ possible sequences of eight S-boxes, brute force on a larger design scale is clearly impossible.

Mathematical construction plays a crucial role in certain design tasks (e.g.

there are many constructions for highly non-linear Boolean functions) but for others (e.g. S-box design with multiple criteria) designers may also resort to random sampling of the design space followed by checking of properties. This has been a standard method for many years. It is still in common use. The Mars algorithm, IBM's candidate for the Advanced Encryption Standard (AES) in 2000, used S-boxes generated by random search [27]. As shown later, even simple guided search techniques can outperform such approaches for some of these traditional tasks.

2.2.2 Computational Power and Emergent Parallelism

One must be careful in determining what one considers to be practically feasible. Indeed, the bounds of computational feasibility are ever increased and challenges are faced on several fronts. Some of these are outlined below.

Special purpose hardware. It may be possible to create special purpose hardware to carry out a search. From its earliest days the Data Encryption Standard has been the subject of proposed brute force and special purpose hardware searches. In 1977 Diffie and Hellman proposed one such brute force analysis [30]. In 1986 Desmedt et al. [28] proposed several more attacks (including ones seeking one of many keys simultaneously). In 1993 Wiener outlined a design for special purpose hardware for recovering DES keys (details can be found in [125]). For a cost of one million US dollars a DES key could be retrieved in at worst seven hours. The Electronic Frontier Foundation (EFF) has recently (1998) developed special purpose hardware ('Deep Crack') [39] to win RSA Laboratories' July 1998 DES Challenge II-2 (a known plaintext key search problem) in 56 hours. Special purpose hardware may be expensive but can clearly be effective — it may, however, be unnecessary, as argued below.

More power in ever more places. Notions of practical computability also face pressure from other directions. Computing platforms grow ever faster (the author's current personal computer has a clock speed over 2000 times faster than the first machine he bought in 1987) and cooperative distributed attacks using such resources are easy to organise. RSA Laboratories' DES Challenge II issued in January 1998 was won in 39 days by distributed.net. RSA Laboratories' January 1999 DES Challenge III was solved in just over 22 hours by EFF's Deep Crack in cooperation with distributed.net. Similar challenges for the 48-bit and 56-bit RC5 algorithms were dispatched in 13 and 270 days respectively by distributed.net and the Bovine group. An attack is currently under way for 64-bit RC5. The most high profile of distributed searches, however, have been the attempts to factorise RSA Laboratories' series of challenge numbers (for details of the challenges and attacks on them see [54]). The security of the RSA system of encryption is based on the (as yet unproven) computational difficulty of factorising products of two large primes (i.e. if $N = p \times q$, then given N it is computationally hard to find p and q). RSA Laboratories have published a series of increasingly large moduli N and offer cash prizes to anyone breaking an unfactorised one. In 1994 the factorization of RSA-129 (a 129 digit number) using a variation of the multiple polynomial quadratic sieve factoring method took approximately 5000 mips years and was carried out in 8 months by about 600 volunteers from more than 20 countries, on all continents (except Antarctica) revealing the corresponding secret message "THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE." In 1999, a group of researchers completed the factorization of the 155 digit (512 bit) RSA-155 using the General Number Field Sieve. This took approximately 8000 mips years (and calendar time of 3.7 months). Since 512-bit RSA had been widely used in general practice in earlier years this was a significant achievement. As increasing numbers of increasingly powerful computing platforms facilities are organised to address theoretically 'smarter' questions, current achievements should readily be surpassed. A long-running and well-organised consumer of distributed computing power is the SETI (Search for Extra-Terrestial Intelligence) initiative's signal processing [89] found as background tasks on computers across the world. (Unlike most cryptanalytic searches, no computational end-date has been offered.) For more information on brute force search results the reader is referred to [25].

Pervasive Computing. The United Kingdom is currently (2001) seeing the rise of digital television. (Indeed in 2010 non-digital television will be 'switched off'.) Televisions will shortly contain significant reprogrammable computational power (either directly or via 'set-top boxes') that could be harnessed for whatever needs. Ubiquitous and web-enabled computing generally will raise the amount of processing capabilities. The cryptanalytic potential of such platforms is obvious. This observation is the author's own but the idea has been seen before (in 1991) in the shape of Quisquater and Desmedt's 'Chinese Lottery Attack' [99]. Essentially, if every Chinese had a radio with a key search processor in it, a problem could be radioed to all sets to carry out partitioned key search in parallel. With a population of over 1 billion and with processors capable of searching 1 million keys per second recovering a DES key would take about 30 seconds. Schneier [110] quotes times for finding 56-bit and 64-bit keys using this technique for population sizes of different countries.

Cheap and Powerful Re-programmable Hardware. Recent years have seen the emergence of Field Programmable Gate Arrays (FPGAs) — 2-D arrays of simple cells that communicate only with their immediate neighbours — as a significant computational platform. A cell will take Boolean inputs from the cells above, below, left and right and deliver outputs to those cells. A cell is capable of computing and storing some simple Boolean function of the current state and inputs. Each output is either the current Boolean state or is the value of some current input. Each cell is therefore primitive but all cells execute in parallel. The principal characteristic of these platforms is that they are readily re-programmed. The way the state value and outputs are calculated can be changed in a few milliseconds (for the whole array of cells).

Taking advantage of such huge fine-grained parallelism is tricky. For some FPGAs, however, tools are available to allow applications written in high-level parallel languages to be compiled to run on them. Handel-C [40], for example, is a C-like language with its origins in Occam (and so Communicating Sequential Processes [49]). A program could be written that specified 100 parallel encryption tasks. These tasks will be compiled to run on different areas of the array. Multiple round algorithms can also benefit from such parallelism (e.g. by forming a pipeline). The benefits of special purpose hardware can be obtained for the price of producing a software program. Of course, as FPGAs get bigger (allowing more parallel tasks) the program is simply recompiled for the new hardware. It would be a simple task to co-ordinate racks of several thousand such arrays. FPGAs are likely to become mainstream (they are obvious candidates for hosting downloaded applications for increasingly sophisticated mobile phones) and consequently ever cheaper (the author has seen some small FPGAs for about 80 Australian dollars). The cryptanalytic potential of these platforms is considerable [1]

E-Science Side Effects. Cryptographic security may be under threat from a highly unusual source — e-science. Computational mathematics is now being brought to bear on problems of biology, chemistry and physics. Smart ways of search are being developed to explore very large solution spaces. It is generally accepted that vast computing power is of great use in addressing these problems. The rewards are potentially huge. In the United Kingdom the Engineering and Physical Sciences Research Council (EPSRC) is promoting academic access to massive distributed computing power as part of its e-science initiative [33]. Although IBM have accrued a great deal of positive public relations from the development and exploits of the 256 node chess engine Deep Blue [56] (which beat Kasparaov in a six-game match in 1999) chess is no longer the major interest. In 1999 they announced a five-year plan for the development of Blue Gene — a computational facility capable of 1,000,000,000,000,000 (approximately 2^{50}) operations per second (using one million processors) and targeted at the study of biomolecular phenomena such as protein folding [55]. This is approximately 50 times the combined power of all existing (in 1999) supercomputers. The research priorities of computer science are changing and would appear to be moving in a direction that supports the creation and use of vast computing power. This has obvious cryptological implications.

[1] At the time of writing Richard Clayton (Cambridge) has announced a cracking engine for DES based on FPGA hardware.

2.2.3 The Threats from Physics

All computation examined so far has been ‘classical.’ It may be possible to bring huge amounts of hardware to bear on a problem but ultimately nothing can be achieved that could not be achieved with a personal computer given sufficient time and memory. The classical model of computation, proposed by intellectual pioneers of computer science such as Alan Turing and Alonso Church has served us tremendously well, but assumes particular properties of how information may be handled (e.g. that a value may be read without affecting it). Recently we have seen the emergence of new models of computation that use the way nature appears to work at a fundamental level. Perhaps the most significant in cryptological terms is quantum computing.

Quantum Computing. Quantum computing is one of the most exciting developments of modern science. The physicist Richard Feynman had speculated that that quantum effects themselves might be harnessed to provide a simulation capability for investigating quantum effects [36]. In 1985 Deutsch provided the first demonstration that quantum effects could achieve results more efficiently than classical computation [29]. Quantum computing might well have remained an interesting but somewhat speculative area of physics were it not for a specific result from Peter Shor. In 1994, he published the ‘killer application’ [112]. He showed how the notion of a Quantum Discrete Fourier Transform could be harnessed to extract in polynomial time the period of the function $f(x) = a^x \mod n$. Extraction of such periods may allow the number n to be factorised. (Success may require a few applications of the algorithm.) Since much public key cryptography is based around the supposed difficulty of factorisation this was a shock result. Of particular note, the RSA algorithm is rendered vulnerable.

Since the publication of Shor’s result, research into quantum computing has mushroomed. Perhaps the most significant subsequent result is Grover’s algorithm [46]. Given a predicate P over $0..2^n - 1$ suppose there is only a single value v such that $P(v)$ holds. Grover’s algorithm can typically find v with order $O(2^{n/2})$ iterations of a loop body (each body comprises a series of unitary transformations). Thus searching over an 80-bit key space could be done in $O(2^{40})$ iterations. If there are N values satisfying the predicate then the search becomes of order $O(2^{\frac{n}{2}} \times N^{-\frac{1}{2}})$. Although practical quantum computing is some way off, it is a very real possibility. Grover’s algorithm can be viewed to some extent as ‘quantum brute force’, the application of quantum computing brawn. Shor’s algorithm is very much quantum mathematical brain. Quantum effects are harnessed in a very smart manner guided by an understanding of the problem domain. Guided search techniques (described below) are a small step in this direction for conventional search. Interestingly, work has been carried out on exploiting problem structure in quantum search [50]. Somewhat ironically, as cryptographic security faces a powerful long-term threat from quantum computation, quantum cryptography looks very promising and practical. An excellent introduction to quantum computing is given by Rieffel and Polak [104].

Physical Properties of the Implementation The work on quantum computing is significant in that it deviates significantly from the standard computational model — the Turing machine. The computation is inherently wrapped up with the reality of physics. At this point an observation about much current cryptography may usefully be made: most of it is concerned with algorithms and not their implementation. Cryptology is a profoundly mathematical subject. It is interesting to note that three of the most powerful attack ideas of the past decade exploit features of the *implementation* of an algorithm rather than the algorithm itself. Boneh, De Milo and Lipton demonstrated a fault injection based attack on public key systems [5]. Paul Kocher [66] has shown how the times taken to perform specific exponentiations could be used to leak key material (e.g. from SHA-2 and RSA). Similarly, the power consumption of a smart card has been shown to be a very powerful leakage channel (see [57]). As the saying goes, 'It's not what you do, it's the way that you do it.' [2] Other observable physical properties, such as local magnetic fields and local temperatures, could potentially be exploited to leak secret information (and one can think of all manner of combinational effects, e.g. differences in power consumption at different temperatures etc.) The notions of fault injection and timing based attacks will re-emerge later in this thesis in a rather unusual guise — they can be applied to search techniques too.

2.3 Guided Search

For some problems there may be no alternative to enumerative or sampling-based approaches. This is generally due to lack of (approximate) continuity in the function $f(x)$, i.e. the value of $f(x)$ at a specific point x_0 gives little exploitable information. Cryptology revels in lack of continuity. Indeed, certain cryptographic goodness criteria can be thought of as discontinuity measures (e.g. for a 64-bit block cipher it might be required that keys which differ by a single bit should produce ciphertexts that differ on average by 32 bits: small input changes can have radical output effects). This thesis is concerned with functions for which elements of continuity can be exploited. Solutions (inputs to the cost functions) that are 'near-by' or 'close' will not give outputs that are radically different. Information gleaned from function evaluation will be used to influence the progress of the search. This is *guided search.* The notion of closeness can be formalised

[2] These attacks have become fairly high profile, largely due to the rise of the smart-card as a computational platform. Elements of all these forms of attack may have been well-known within Governmental circles.

as a function. For a specific value x the set of all points that are in the immediate neighbourhood can be defined by some function $N(x)$:

$$N : X \to 2^X.$$

The research in this thesis uses principally what is often termed 'local search'. Here the search moves through a series of points $x_0, x_1, x_2, \ldots, x_{final}$ with each point being in the neighborhood of the point which precedes it. At each point x_n the value of $f(x)$ is evaluated for one or more points in $N(x_n)$ and the information used to determine whether the search should 'move' to a particular point in that neighborhood. There are several strategies for selecting points in the neighborhood and deciding which move, if any, should be taken (several are examined below).

2.3.1 Gradient Ascent — Hill-climbing

Gradient ascent methods sample or enumerate the values of $f(x)$ in the neighborhood of the current solution x_{curr}. If the search moves only to a neighbour if it improves the value of $f(x)$ then the search is a form of 'hill-climbing' or gradient acsent. If the neighborhood is huge then sampling may be carried out to find an improving move. Accepting a move that makes the greatest improvement gives rise to what is known as steepest ascent. If the search takes the first improving move it encounters, it is said to be a 'greedy' gradient ascent. The terms gradient ascent and gradient descent are used depending on whether the problem at hand is couched as a maximisation or a minimisation problem. The problem with such techniques is obvious. If the search starts in the wrong place the result may be a local optimum. Hill-climbing remains an important technique nevertheless — sometimes one simply has a hill to climb. Furthermore, robust non-linear optimisation techniques may get close to optimal solutions but use hill-climbing to carry out the very final stages of optimisation efficiently.

2.3.2 Simulated Annealing

In 1983 Kirkpatrick et al. [64] proposed a new search technique based on the cooling processes of molten metals. The technique was *simulated annealing.* It has proved to be an extraordinarily simple, yet powerful, heuristic search technique. It merges hill-climbing with the probabilistic acceptance of non-improving moves. The basic algorithm is shown in Figure 2.1.

The search starts at some initial state $S = S_0$. There is a control parameter T known as the temperature. This starts 'high' at T_0 and is gradually lowered. At each temperature, a number MIL (Moves in Inner Loop) of moves to new states

```
$S = S_0$
$T = T_0$
**Repeat**
{
        for(int $i = 0; i < MIL; i++$)
        {
                Select $Y \in N(S)$
                $\delta = f(Y) - f(S)$
                if ($\delta < 0$) then
                        $S = Y$
                else
                        Generate $U = U(0,1)$
                        if ($U < \exp(-\delta/T)$) then $S = Y$
        }
        $T = T \times \alpha$
}
**Until stopping criterion is met**
```

Figure 2.1: Basic Simulated Annealing for Minimisation Problems

are attempted. A candidate state Y is randomly selected from the neighborhood $N(S)$ of the current state. The change in value, δ, of f is calculated. If it improves the value of $f(S)$ (i.e. if the $\delta < 0$ for a minimisation problem) then a move to that state is taken ($S = Y$); if not, then it is taken with some probability. The worse a move is, the less likely it is to be accepted. The lower the temperature T, the less likely is a worsening move to be accepted. Probabilistic acceptance is determined by generating a random value U in the range (0..1) and performing the indicated comparison. Initially the temperature is high and virtually any move is accepted. As the temperature is lowered it becomes ever more difficult to accept worsening moves. Eventually, only improving moves are allowed and the process becomes ‘frozen’. The algorithm terminates when the stopping criterion is met. Common stopping criteria, and the ones used for the work in this thesis, are to stop the search after a fixed number $MaxIL$ of inner loops have been executed, or else when some maximum number MUL of consecutive unproductive inner loops have been executed (an inner loop is termed unproductive if no move is accepted within it). Generally the best state achieved so far will also be recorded (since the search may actually move out of it and subsequently be unable to find a state of similar quality). At the end of each inner loop the temperature is lowered. The simplest way of lowering the temperature is shown. This is known as geometric cooling. The basic simulated annealing algorithm has proven remarkably effective over a range of problems. This technique will be used (with hill-climbing) in all the technical chapters of this thesis.

There are many nuances. For example other forms of cooling have been proposed, perhaps the most significant of these is logarithmic cooling. It is also possible to 'reheat' the process to allow escape from suspected local optima and sophisticated parallelisable variants have been proposed. Theoretical results based on Markov chains are available to show that under appropriate conditions the search is guaranteed to converge on the global optimum. Unfortunately, this may require more function evaluations than enumerative search and so is of little practical use. Problem specific trade-offs are made that allow excellent solutions to be obtained under regimes that are practically computable. For examples of the practical use of annealing variants and the various issues involved in using annealing the reader is referred to [102, 100].

An interesting diversion from the 'vanilla' annealing (and one of particular relevance in this thesis) is provided by Chardaire et al.'s thermo-statistical persistency [16]. This deals with problems whose solutions are vectors of binary values (i.e. 0/1 problems). Some interesting patterns emerge if one profiles the values taken by specific elements x_i in the current solution vector x during the search. Initially, an element x_i may repeatedly switch values during the search without any obvious preference. However, as the search progresses and the temperature cools, the x_i may assume one value, say 1, more than the other. When such biases become significant, e.g. when an element spends 95 per cent of recent time at a particular value, it is highly likely that in the final solution the element will have that value too. Thermo-statistical persistency acknowledges this and actually fixes such values. Only moves involving non-fixed values are subsequently allowed with significant gains in efficiency. If one simply profiles the normal annealing search it is clear that some solution bits get 'stuck' at some value earlier in the search than others. Why? The order in which elements get stuck is clearly telling us something about the structure of the problem instance. But what? Can this form the basis of a 'timing attack' when applied to cryptological problems? For the time being it is noted that analysis techniques (here annealing-based search) have temporal characteristics that might be exploitable. Thermo-statistical ideas have seen application elsewhere (e.g. [101]) but do not seem widely used.

2.3.3 Tabu Search

Tabu search is a widely used modern local search technique. The next move to take is decided using cost function values but also historical information (i.e. it uses memory of some form). This allows the search to escape from local optima and also to explore the search space in a productive fashion. Tabu search generally adopts a best improvement local search but moderates this policy using historical information.

If a particular solution S is reached then it becomes ‘tabu’ for some number T_s of transitions, generally referred to as the solution’s tabu tenure. If a solution is tabu, the search is normally prevented from moving to that solution, i.e. the local neighbourhood from which the next solution is chosen excludes those solutions that are currently tabu. Conceptually, the currently tabu solutions together with their remaining tabu tenures form a ‘tabu list’. In its simplest form, with common tabu tenure of T, the list becomes a FIFO queue. The most recently visited solution is added and the solution visited T moves ago is removed. The tabu list implements what is generally referred to as a *recency* criterion. It prevents the search revisiting solutions in the short term (and so short cycles are prevented). The higher the tabu tenure the more the search is forced to explore the solution space. The tabu tenure may be varied during the search. Figure 2.2 outlines a basic tabu search procedure (taken from [4], which provides an interesting consideration of metaheuristic techniques more generally).

```
$S = GenerateInitialSolution()$
$InitializeTabuLists(TL_1, \ldots, TL_r)$
$k = 0$
while termination conditions not met do
{
$AllowedSet(S, k) = \{z \in N(s) |$ no tabu condition is violated
or at least one aspiration criterion is satisfied $\}$
$S = BestImprovement(S, AllowedSet(S, K))$
$UpdateTabuListsAndAspirationConditions()$
k=k+1
}
end while
```

Figure 2.2: Basic Tabu Search Procedure

In practice maintaining lists of *solutions* is very inefficient. Much more common is to keep lists of solution attributes or moves. Consider an object permutation problem, i.e. where objects O_1,..,O_n must be arranged in some order (and there is a cost associated with each such order). If a move (i,j) (with i<j) is taken that swaps the positions of objects O_i and O_j then this could be made tabu for a period. A more stringent tabu criterion would make any move involving object O_i or object O_j tabu. Thus, taking move (1,4) would render tabu any move of the form (a,b) where either a or b is equal to 1 or 4. Other features may be taken into account. For example, the actual cost associated with a solution could be made tabu. The search would be prevented from visiting solutions with the same cost function value for the tabu tenure.

The tabu status of a move can be relaxed if taking that move would give rise to a particularly good solution, most typically a solution better than any reached so far (this is generally referred to as the *aspiration criterion*). Other aspects of history can also be taken into account, such as long-term frequencies of particular move types. The notion of *influence* is also used to guide the search; a move that causes greater change (measured in some fashion) is deemed to be more influential. Thus, influence criteria can be created and applied to diversify the search. For an excellent discussion of tabu search details the reader is referred to the chapter on tabu search by Glover in [102].

2.3.4 More Recent Local Search Methods

A number of local search based metaheuristics have emerged in recent times. Two of these now outlined below. The algorithm descriptions are taken from [4].

The *Greedy Randomized Adaptive Search Procedure (GRASP)* procedure is one of the simplest and combines constructive algorithms for generating feasible solutions with local search. The GRASP approach is summarised in Figure 2.3. A greedy heuristic is used to generate a starting solution S. This solution S is then subject to an improvement heuristic (i.e. a local search). This is repeated until some termination criterion is met — maximum number of iterations reached, CPU time limits reached etc. The construction heuristic procedure is outlined in Figure 2.4.

```
while termination conditions not met do
{
$S = ConstructGreedyRandomizedSolution()$
$ApplyLocalSearch(s)$
$MemorizeBestFoundSolution()$
}
end while
```

Figure 2.3: Basic Greedy Randomized Adaptive Search Procedure

Assume that a solution comprises a number of elements $\{x_1, x_2, \ldots, x_n\}$. The solution is constructed one element at a time by picking randomly from a candidate list. In the Traveling Salesman Problem for example, a solution could be constructed by adding one edge at a time. Possible next elements are ranked according to some heuristic measure of desirability. The candidate list comprises the first cl elements. The desirability of including an element may change as an element is added to the solution. The size cl of the list is a crucial parameter. The local search procedure can take a variety of forms:gradient ascent; simulated

```
$S = \emptyset$
while solution is incomplete do
{
$RCL = RestrictedCandidateList()$
$x = SelectElementAtRandom(RCL)$
$S = S \cup \{x\}$
$UpdateGreedyFunction(s)$
}
end while
```

Figure 2.4: Greedy Randomized Solution Construction

annealing; tabu search etc. A bibliography or GRASP literature is given by Resende [103]. Further descriptions can be found in [35].

Iterated Local Search (ILS) aims to search over the space of local optima. A random solution S_0 is generated and then local search is applied to reach a local optimum S^*. This local optimum is then perturbed in some way to obtain S' and local search is then applied to reach another local optimum $S^{*'}$. Some criterion is applied to determine whether the 'move' from S^* to $S^{*'}$ is accepted. Thus the high level moves are seen to be between local optima. The general approach is given in Figure 2.5. Aspects of memory can be incorporated into the perturbation and into the acceptance criterion. The *strength* of a perturbation is a measure of how much it changes the solution. This may be fixed or may vary dynamically. A variety of acceptance criteria can be adopted (always accept, accept only improving moves, accept probabilistically in an annealing-like manner etc.)

```
$S_0 = GenerateInitialSolution()$
$S^* = LocalSearch(S_0)$
while termination conditions not met do
{
            $S' = Perturbation(S^*, history)$
            $S^* = LocalSearch(S')$
            $S^* = ApplyAcceptanceCrtierion(S^*, S^{*'}, history)$
}
end while
```

Figure 2.5: Iterated Local Search

The research reported in this thesis has a strong local search bias and elements of iteration will be adopted in many places.

Blum and Roli [4] describe further methods with a local search basis such as Variable Neighbourhood Search (VNS) and Guided Local Search (GLS). The reader is referred there for details.

2.3.5 Genetic Algorithms

Genetic algorithms (GAs) are heuristic search techniques based loosely on natural selection. A population of candidate solutions is generated randomly and then successive generations are evolved using three evolutionary 'operators'. These are *selection* (survival according to fitness), *crossover* (where solutions 'mate', producing offspring) and *mutation* (where solutions may spontaneously change a characteristic). The general idea is that populations evolve according to rules that will in general support the emergence of ever fitter individuals (i.e. ones with higher evaluation value). For an introduction to genetic algorithms the reader is referred to the classic text by Goldberg [41]. A simple example is now given for illustration.

Suppose the goal is to maximise the function $f(x) = x$ over the range $0..7$. First a population of candidate solutions expressed as bit strings is generated randomly, say 010, 001, 010 and 000. These bits strings are often referred to as 'chromosomes'. Assume that the bit strings are a straightforward binary encoding of integers in the range. An obvious evaluation of the fitness of a solution x for this problem is simply the value $f(x)$. Thus the fitness values for the initial population are $(2, 1, 2, 0)$. The total fitness of the population (i.e. the sum of the fitness values of all its members) is 5.

A new population of size 4 is now selected from the current population. When the selection is made for each of the four members of the new population, the first solution 010 of the current population has a $2/5$ chance of being chosen, the solution 001 has a $1/5$ chance, the second 010 has a $2/5$ chance and the final solution 000 has no chance of being chosen. This is selection with replacement (and so it is possible, for example, that the same solution will be picked four times). Suppose this gives rise to the new population $(010, 001, 010, 010)$. The actual fitness values are $(2, 1, 2, 2)$ and the total fitness is now 7 (and so fitness-weighted selection — survival of the fittest — has aided the evolution of a healthier population).

Successive pairs are now 'mated' by swapping some randomly chosen subsequences of bits (usually termed *crossover*). Suppose that the final bit is chosen for the first pair of solutions and the last two bits are chosen for the second pair. A population $\{011, 000, 010, 010\}$ results. The fitness values for this population are $(3, 0, 2, 2)$. The total fitness has not increased in this case but a particular solution has emerged that is fitter than any previous one (and so has better chances of further survival).

Selection according to fitness and mating are powerful mechanisms for obtaining populations of high performing solutions but will never produce the optimum (111) in this example. A final operator, mutation, now allows each bit to change value with some small probability, e.g. 0.01. Suppose that the only bit to flip at this stage is the first bit of the second string giving rise to 011, 100, 010 and 010. One can now see how further selection and mating can lead to eventual appearance of even better solutions (and 111 in particular). After mutation the solutions are then evaluated. The select-mate-mutate-evaluate cycle repeats until convergence has been achieved, until no further progress is apparent, some practical upper bound on the number of generations has been reached, or else one candidate solution 'solves' the problem at hand.

The technique is heuristic. Mating is not guaranteed to produce better solutions and mutation can at times be unhelpful. In addition, for non-linear functions, convergence to a local optimum is possible (though the principal strength of genetic algorithms is their global optimisation ability over a great range of different problems). The above describes a variant of what is generally referred to as the standard *simple genetic algorithm* following Goldberg [41]. In practice certain enhancements are often made. Fitness values may be scaled before use and a variety of selection methods are available. The one described above is known as *roulette wheel selection*. A more common (and effective) means of selection in modern day genetic algorithm work is that of *tournament selection*. Here, members are selected according to fitness as before and the fittest is then chosen to appear in the next generation. This technique can be extended to selection from tournament groups of size greater than 2 but research in this thesis will use only tournament pairs (i.e. $G = 2$).

Crossover, i.e. swapping subsequences of chromosome elements (here bits), is not forced to happen when a pair is selected. Rather, it takes place with some probability. In the experiments reported in this thesis a range of values for this probability will be examined (and the bit mutation probabilities will be varied too). Simple variants use single-point cross-over (where the bits to the right of some point are swapped). More common is two-point crossover (where a random section of consecutive bits is swapped) and uniform crossover (where each bit is randomly chosen or rejected for exchange between chromosomes). It has also been found that the insertion of 'noise' into the fitness function may improve the performance of the search.

There is a huge amount of genetic algorithms literature and nuances abound. In some cases the children replace the previous population entirely. In other approaches only a fraction of the current population are replaced. Sometimes the best population member so far is guaranteed a place in the next population (so called elite survival). Various measures can be taken to bring about intensification (convergence on a solution), to prevent it happening too soon, or else to force the search to explore new areas (diversification). Groups of populations can form niches and evolve separately before inter-group cross-fertilisation is carried out. The reader is referred to texts such as Goldberg [41] and Michalewicz [78].

Representation is a crucial issue. Although the illustration above explains genetic algorithms with bit-encodings, other encodings are possible and often advantageous. Michalewicz [78] discusses this issue. Indeed, breaking away from bit encodings has been a feature of what is now termed *evolutionary programming*. Representation is an important issue in cryptological applications. Some problems, including those of Chapters 3 and 4, seem to have a natural bitwise encoding, yet this may actually cause significant problems due to isomorphism (loosely, it is possible to combine two isomorphic and excellent solutions to obtain very poor offspring under combination). More generally, the notion of *epistasis* may rear its head (dependencies between chromosome elements). A cryptographer presented with half the truth table of a Boolean function and asked 'Does this look promising from a cryptographic point of view?' would be very surprised. The cryptographic characteristics of the function would depend crucially on what was in the other half of the table. It is often difficult to isolate the notion of high performing 'building blocks', i.e. characteristics generally associated with good overall fitness and preferably mapping onto some compact subset of chromosome elements. [3] A good guide to the practical use of genetic algorithms is available on-line [68].

2.3.6 Combining Techniques

The GRASP and Iterated Local Search procedures outlined above might be considered hybrid techniques in their own right. Indeed, since they are obviously strategies for exploiting lower-level local search procedures they may be considered to be operating in a more genuinely 'metaheuristic' fashion than simulated annealing, tabu search and genetic algorithms (as described earlier). Techniques such as simulated annealing, genetic algorithms and tabu search need not be used in isolation. Indeed, these and other techniques are often used in combination, or else ideas arising in one technique are borrowed and adapted for use in another.

It is possible to use problem specific heuristics to determine a good initial starting solutions for annealing runs. Lower starting temperatures are then adopted to avoid the benefits of good initial solutions being destroyed by excessive random fluctuations. Another approach is to incorporate annealing into a construction heuristic which works by building on a previous partial solution. Chams et al. adopt such an approach for colouring graphs [15]. Finally, incorporation of a local search heuristic (such as gradient-ascent) after applying annealing is very common. This is adopted for most applications in this thesis. Elements of tabu search have been incorporated in simulated annealing to manipulate the temperature in a

[3] More technically, "building blocks" is the term usually used to describe schemata that are low-order, well-defined, and have above average fitness.

strategic fashion (standard annealing reduces the temperature monotonicly, as in geometric cooling or logarithmic cooling) [93].

The application of a genetic algorithm to a problem is often followed by local search. This local search may be of a general form (e.g. the application of a gradient-ascent) or else be problem-specific (for example, the Kernighan-Lin optimisation algorithm for the Travelling Salesman Problem). The use of gradient-ascent to carry out the final 'fine-tuning' is very common. Glover and Laguna (see [102] indicate several ways in which tabu search ideas could be incorporated in genetic algorithms. Further hybridisation of population-based approaches and exploitation of local search can be found in some *memetic algorithms*. The reader is referred to [26] for details.

The notion of hybridisation is a powerful one and has been readily adopted for emerging techniques. Ant Colony Optimisation (ACO), a technique based loosely on the self-organisation characteristics of real ant colonies (see [26]), has recently received considerable attention. Once again, following the basic ACO algorithm with a local search procedure has been found to improve results for many problems.

The above is by no means exhaustive; it is intended only to illustrate the need for a flexible approach to optimisation. This observation is not restricted to current-day approaches — the author believes that long-term hybrids of quantum search and metaheuristic search will be of considerable use.

2.4 The Cryptanalysis of Classical Ciphers

2.4.1 General Background

Classical ciphers are based around the notions of character substitution and transposition. Messages are sequences of characters taken from some plaintext alphabet (e.g. the letters A to Z) and are encrypted to form sequences of characters from some ciphertext alphabet. The plaintext and ciphertext alphabets may be the same. Subsitution ciphers replace plaintext characters with ciphertext characters. For example, if the letters of the alphabet $A \ldots Z$ are indexed by $0 \ldots 25$, then a Caesar cipher might replace a letter with index k by the letter with index $(k + 3)\ mod\ 26$. Thus, the word "JAZZ" would become "MDCC". Transposition ciphers work by shuffling the plaintext in certain ways. Thus, reversing the order of letters in successive blocks of four would encrypt "CRYPTOGRAPHY" as "PYRCRGOTYHPA".

Modern cryptosystems have now supplanted the classical ciphers but cryptanalysis of classical ciphers is the most popular cryptological application for metaheuristic search research. Why is this so? The reasons are probably mixed. The basic concepts of substitution and transposition are still widely used today (though typically using blocks of bits rather than characters) and so these ciphers form simple but plausible testbeds for exploratory research. Problems of varying difficulty can easily be created (e.g. by altering the key size). They seem also to be natural candidates for metaheuristic solution as argued below.

Consider a simple substitution cipher on the letters $A \ldots Z$ indexed by $0 \ldots 25$ as above. The keyspace for this type of system is the set of bijective functions $f : 0 \ldots 25 \rightarrow 0 \ldots 25$. Given ciphertext C, decryption can be thought of as a function $f_C(K)$ from the keyspace to the space of plaintext messages. Decrypting ciphertext using keys that are 'nearly the same' gives rise to plaintexts that are nearly the same. Similarly, keys that are 'nearly correct' give rise to plaintexts that are nearly correct. With respect to correctness the decryption operation is reasonably continuous over the keyspace. This is crucial to the general use of heuristic search since some means of homing in on the solution is required. It is this continuity that makes these problems natural candidates for guided search techniques.

One cannot know how correct a decrypted text is without knowing the plaintext. Instead, the degree to which decrypted text has the distributional properties of standard English is taken as a surrogate measure of correctness of the decryption key. In English text the letter "E" will usually occur more than any other. Similarly, the pair (bigram) "TH" will occur frequently, as will the triple (trigram) "THE". In contrast, the occurrence of the pair "AE" is less common and the occurrence of "ZQT" is either a rare occurrence of an acronym or else indicates an inability to spell. The frequencies with which these various N-grams appear in plaintext are used as the basis for determining the correctness of the key which produced that plaintext. The more the frequencies resemble expected frequencies, the more correct the underlying decryption key is assumed to be.

More advanced cipher variants seek to hide such statistical patterns. For example, multiple substitution alphabets can be used to hide such gross statistical characteristics. However, other patterns emerge and successful manual analysis techniques have been available for some time. The above argument gives a flavour of how automated cryptanalysis works. Perhaps the last character based cipher to capture the public imagination was the Enigma cipher (essentially a sophisticated polyalphabetic substitution cipher). For information on classical ciphers and means of cryptanalysing them, the reader is referred to [97].

6	3	1	2	5	4
N	O	W	I	S	T
H	E	T	I	M	E
F	O	R	A	L	L
G	O	O	D	M	E
N					

Figure 2.6: Transposition Cipher

2.4.2 Research on Automated Cryptanalysis of Classical Ciphers

Most major optimisation techniques have been applied to classical cipher cryptanalysis. Progress remained sluggish until 1993 when numerous papers appeared. Spillman et al. [120] showed how simple substitution ciphers could be attacked using genetic algorithms. Mechanisms for representation, mutation and crossover are discussed as might be expected, but perhaps the most interesting feature is the fitness function used:

$$Fitness = \left(1 - \sum_{i=1}^{26} \left\{ |SF[i] - DF[i]| + \sum_{j=1}^{26} |SDF[i,j] - DDF[i,j]| \right\} /4 \right)^{8}. \tag{2.1}$$

The letters $A \ldots Z$ are referenced by the indices $1 \ldots 26$. Here $SF[i]$ is the standard frequency of character i in English. $DF[i]$ is the measured frequency of the character i in the decoded ciphertext. Similarly $SDF[i,j]$ is the standard frequency of bigram character i followed by character j in English. $DDF[i,j]$ is the corresponding frequency of that bigram of the decoded ciphertext. The really unusual element is the exponent of 8, included ‘to amplify small differences.’ Experimentation with such exponentiation parameters is almost universally ignored in much subsequent work. Spillman emphasises the importance of experimentation with genetic algorithm parameters. Also, he suggests possibilities of more sophisticated cost functions involving trigrams, i.e. three letter strings such as “THE”.

Independently, Matthews [74] was also investigating the use of genetic algorithms for transposition ciphers. He describes the operation of GENALYST — a flexible scheduling type GA. The particular transposition cipher examined is a familiar one. A key is some permutation of $1 \ldots N$, for example $(6,3,1,2,5,4)$ for $N = 6$. Plaintext is written in rows of length N under the key and enciphered by reading it off in columns in the order dictated by the integers making up the key. Using the key $(6,3,1,2,5,4)$ the phrase “NOWISTHETIMEFORALLGOODMEN” is written as shown in Figure 2.6 and hence enciphered as “WTROIIADOEOOTEWLESMLMNHGFGN”. This was one of the earliest papers and exhibits considerable originality and sophistication. Firstly, the standard frequency based cost was replaced with a points scoring system. Six bigrams and four trigrams were considered. With each a number of points was associated reflecting the likelihood of its occurrence in successfully deciphered text. The (N-gram, points) pairs were (“TH’,+2), (“HE”,+1) (“IN”,+1), (“ER”,+1), (“AN”,+1), (“ED”, +1), (“THE”,+5),(“ING”,+5),(“AND”,+5) and (“EEE”,-5). If the text length is L then the fitness function is given by

$$\langle F_L \rangle = L \sum_{i=1}^{Q} (P_i S_i / 100), \tag{2.2}$$

where P_i is the percentage frequency of the ith bi– or trigram tested for, S_i is its score and Q is the number of bigrams or trigrams checked for. This approximate means seems to work effectively (at least for the experiments reported). The system can be used to determine the keylength. Essentially attempts at wrong key lengths are limited in the fitnesses that can be achieved. Trying runs at various key lengths readily reveals the most effective length. The text notes that random testing also is quite effective in this respect. The real power of GAs comes when the actual permutation is sought. The work describes various enhancements that have been brought to bear, such as elite survival. Another notion advanced is human interaction to aid what he terms ‘perming’. Essentially, manual analysis of the schedules resulting from various runs reveals little groups of columns that regularly appear somewhere in the key. The actual solution is likely to be a permutation that maintains such groups. This is the first paper to espouse real hybridisation techniques. Examining the results of repeated runs is an excellent idea and will reappear in the work of Knudsen and Meier (see Section 2.7).

Giddy and Safavi-Naini [60] use simulated annealing to attack simple transposition ciphers where sections of n letters are each shuffled according to a key permutation. The work places the problem very clearly within the theoretical domain of applicability of simulated annealing (showing the search space to be connected, arguing that the cost surface is reasonably smooth, giving general theoretical advice on cooling schedule, appealing to theory to justify the number of iterations within a temperature cycle etc). The authors demonstrate a new move function that is intended to increase the smoothness of transitions. The cost function used, based as usual on expected $p_{\alpha\beta}$ and actual (i.e. under decryption) $c_{\alpha\beta}$ plaintext bigram frequencies is

$$C(s) = \sum_{\alpha\in\Lambda} \sum_{\beta\in\Lambda} \left| \frac{p_{\alpha\beta} - c_{\alpha\beta}}{\epsilon + p_{\alpha\beta}} \right| . \tag{2.3}$$

(The $p_{\alpha\beta}$ and $c_{\alpha\beta}$ would in Spillman’s notation be SF[α,β] and DF[α, β]). The authors note that the value of ϵ significantly affects results. Such parameters are often referred to informally as ‘fiddle factors’ and highlight the need for general experimentation when applying metaheuristic search techniques.

Jakobsen [59] attacks simple and polyalphabetic subsitution ciphers (assuming that the number of alphabets used is obtained by standard means such as Kasiski or Index of Coincidence, see [97]). The work is essentially a form of hill-climbing. It shows marked efficiency gains on previous work, due in part to clever manipulation of the matrix of bigrams obtained under decryption. In the conclusion he states ‘this approach is not immediately useful for the more modern type of encryption algorithms (IDEA, DES etc.)’ echoing a widely held view.

Clark and Dawson have carried out the most extensive research on classical cipher cryptanalysis. The work is reported in various places and covers applications of genetic algorithms, simulated annealing and the more recently developed tabu search technique. The work has attacked substitution, transposition and polyalphabetic substitution ciphers (the latter using a parallel genetic algorithm). The journal paper [18] provides a comparison of simulated annealing, genetic algorithms and tabu search attacks on simple substitution ciphers. There appears to be little to choose between them according to correctness of final keys produced (TS comes out marginally on top) but there are significant differences with respect to efficiency. TS again comes out on top (roughly twice as fast as SA) with GAs markedly worst (roughly twice as slow as SA). The work of Jakobsen [59] above indicates strongly that local search is effective for the simple substitution cipher (of English) and so it is not so surprising that all techniques work well (with GAs essentially hill-climbing at the end via mutation.) Where hill-climbing has merit then SA is an inefficient way of achieving gradient-ascent, TS will take the form of steepest gradient-ascent. The ability to calculate delta-costs for local search and not for GAs may also explain the relative inefficiency of GAs for this problem. The GAs uses a mating operation that is far more intuitive that that used by Spillman. The authors experiment (via enumeration) with cost function weighting for unigram, bigram and trigram costs. Bigrams are chosen for much of the comparative study. Would higher level parametric optimisation be of any use?

As Bagnall et al. note [1] the ciphers attacked are generally simple ones. The application of heuristic search gives no surprises and to some extent the body of research is much of a muchness. There seems to be a stark lack of ambition. It is pleasing to see something a little different and harder attacked. The Enigma variants attacked by Bagnall et al. are arguably the most sophisticated classical ciphers attacked (both odometer and other rotor rotation strategies are considered) using metaheuristic search. Their technique involves solving for the last rotor of a three-rotor machine (and then solving for the remaining two rotors using a known technique). The first two rotors give rise to a cipher with period q^2 (where q is the cardinality of the alphabet). If the ciphertext is mapped through the correct third rotor, and the resulting intermediate ciphertext is split into q^2-

ciphertext strings, these strings will exhibit the statistical characteristics of mono-alphabetically enciphered text. The degree to which this is actually holds can be taken as a measure of the correctness of the current solution for the final rotor (and so used to guide the search appropriately).

Parallelism is little exploited in heuristic cryptanalysis research. Clark and Dawson [17] use a parallised GA to attack a polyalphabetic substituion cipher. This combines several individual substitution ciphers. Given a plaintext message, the letters at positions $1, 6, 11, \ldots$ might be encrypted using the first cipher, those at $2, 7, 12, \ldots$ using the second and those at $5, 10, 15, \ldots$ using the fifth etc. The individual ciphers are farmed out to various processes. Calculation of unigram costs can be carried out in isolation for each such cipher, but bigram and trigram statistics cannot. Initially, each local process calculates its estimate of its key based only on local unigram measures. Every so often each process communicates its best local key to neighbouring processes to enable such costs to be calculated (e.g. a process solving cipher 2, would receive the best keys so far for ciphers 1 and 3 and use this to calculate appropriate bigram costs.) Eventually, the process converges. The results show that this is a highly effective approach. Lebedko and Topcy [87] comment that the use of parallel GAs is ‘not original’. It is hard to say whether this comment applies to polyalphabetic substitution ciphers (no other reference is given) or to parallel genetic algorithms in general (which seems true, but irrelevant). Whatever, this paper seems a useful contribution.

2.4.3 General Commentary

All the work described above has served a useful purpose. Classical cipher cryptanalysis provides a simple testbed for examining the capabilities of the techniques. In addition, the cryptological knowledge needed is small and so makes these problems attractive to researchers outside the cryptographic community (understanding letter frequency characteristics is probably easier than understanding differential cryptanalysis). As noted earlier [1] most work has concentrated on ciphers readily breakable by other means. Several authors have commented that the techniques are not readily applicable for modern cryptanalysis. It is disappointing is that no-one appears to suggest any way forward in this respect.

On a technical level, the classical cryptanalysis work exhibits symptoms common to virtually all applications of the search techniques to cryptological problems. A major feature is that optimisation is a ‘one shot’ technique — the idea is to ‘solve’ the problem, to extract the whole key in one go (Matthews’ exploitation of multiple runs is highly unusual in the area.) This is not the way modern cryptanalysts work. Cryptanalysts typically work by exploiting small biases and using perhaps many billions of pieces of data (e.g. plaintext-ciphertext pairs). They generally don’t run a program for a few minutes and expect the result to pop out!

With respect to classical ciphers this is entirely understandable; after all, the direct approach appears to work reasonably. But there seems to be a general agreement that the techniques will not work when applied to modern cryptanalysis problems. Although such cryptanalysis is not addressed here, the results of various pieces of work in this thesis would suggest that moving away from this one-shot view has considerable potential. This author also suggests that any future cryptanalysis attempts should address 'hard' problems.

2.5 Cryptographically Strong Building Blocks

The design of Boolean functions and substitution boxes (S-boxes) with good cryptographic properties remains an important research challenge. In this section the notation used in the remainder of this thesis is provided, definitions of the properties we would require of these artifacts are presented together with informal motivation for why these properties are important. This should aid the reader who is not a specialist cryptographer.

2.5.1 Notation and Conventions

A Boolean function is a function $f : \mathbf{Z}_2^n \rightarrow \mathbf{Z}_2$ mapping each combination of n binary variables to some binary value ('0' denotes 'false' and '1' denotes 'true'.) Simple examples of Boolean functions abound. The NOT, AND, NAND, OR and XOR gates of digital logic are all Boolean functions. The XOR gate for example takes values of binary input variables P and K, say, and returns an output value C that is 1 if the values of P and K differ and 0 if they agree. The *polarity truth table* or *polar form* is a particularly useful representation for many purposes. The polar form of a function $f(x)$, denoted by $\hat{f}(x)$, is defined by

$$\hat{f}(x) = (-1)^{f(x)}. \tag{2.4}$$

Thus, a Boolean function of n variables can be represented by a vector of size 2^n with elements of value 1 or -1. Each input vector $x = (x_1, \ldots, x_n)$ of binary variables can be viewed as the simple binary encoding of an integer. Thus (0,0,0,0) corresponds to decimal 0, (0,0,1,1) corresponds to decimal 3, and (1,1,1,1) corresponds to decimal 15 etc. Binary vector values will often be referred to by their corresponding decimal value. This allows us to use common mathematical notations for summation etc. Thus $\sum_{x=0}^{2^n-1} \hat{f}(x)$ is the sum of $\hat{f}(x)$ over all possible (i.e. 2^n) input vectors x. Where the summation range is obvious a shorter form may be used, e.g. $\sum_x \hat{f}(x)$. In this thesis, such shortened forms are used regularly. The implied range will generally be $0..(2^n - 1)$.

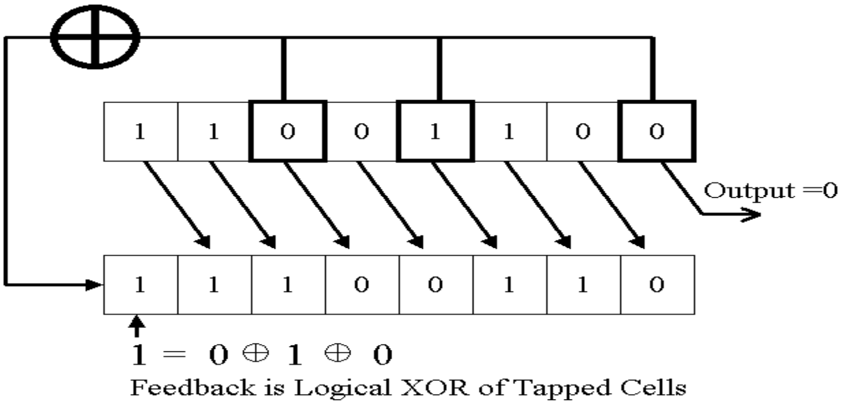

Figure 2.7: Linear Feedback Shift Register

2.5.2 Building Secure Streams

The simple XOR function is at the heart of the most secure method of encryption — the one time pad (or Vernam cipher). Here the sender and receiver share a random key bit sequence $\{K_i\}_{i=1}^{N}$. Let the plaintext message to be sent be $\{P_i\}_{i=1}^{M}$ (with M $\leq$ N). The ciphertext is given by $\{C_i = K_i \oplus P_i\}_{i=1}^{M}$. The receiver recovers each plaintext bit using $P_i = C_i \oplus K_i$. This method provides perfect secrecy (the mutual information between the plaintext and ciphertext is zero) but is just too inefficient for most encryption needs. The shared key needs to be as big as the message and must be distributed securely to both parties. The emphasis shifts therefore to the production of key streams that efficiently provide good *practical* security.

A first attempt to achieve this might be to use Linear Feedback Shift Registers (LFSRs). An LFSR is essentially a finite state transition machine with output but no input and is used to generate bit streams as shown in Figure 2.7. A shift feedback register has N binary registers (N is its ‘length’). A transition of the state machine causes the value in the rightmost register to be produced as the output. Each register except the leftmost now assumes the value of the register immediately to its left (i.e. there is a general right shift). The leftmost register assumes a value that is some function of the old register values. The precise dependency is given by the *feedback function*. The simplest form of feedback function returns a value that is equal to the XOR of the values of some subset of the registers. This form of Boolean function is said to be a linear function (a sum of products of size one). Formally, a *Linear Boolean Function* selected by $\omega \in \mathbf{Z}_2^n$, is defined by

$$L_\omega(x) = \omega_1 x_1 \oplus \omega_2 x_2 \cdots \oplus \omega_n x_n, \tag{2.5}$$

where $\omega_i x_i$ is the logical AND of ω_i and x_i. The set of affine functions is the set of linear functions and their complements

$$A_{\omega,c}(x) = L_\omega(x) \oplus c, \tag{2.6}$$

where c is either 0 or 1. A finite state transition machine with no input is periodic (i.e. the states it assumes, and the outputs it gives, recur periodically). For use in pseudo-random number generators it is desirable that this period be as great as possible. The particular subset of registers chosen for the XOR feedback (these registers are said to be 'tapped') is very important. Certain choices allow periods of maximum length $2^N - 1$ (the state where all registers are 0 is not allowed.) Technically, such choices correspond to *primitive polynomials* over $G(2^N)$ though this is a detail that need not bother us here. LFSRs in their raw form are not used to generate keystreams for direct use. If an eavesdropper knew or could guess N consecutive plaintext bits he could obtain the corresponding N keystream bits and so would be able to re-construct a state of the register (and so be able to get all subsequent bits). If the analyst does not know the registers tapped to form the feedback function, then knowing $2N$ consecutive bits allows the system to be similarly broken. LFSRs can be used successfully as components of a stream cipher as shown in Figure 2.8. Here they provide individual streams that are subtly combined by a Boolean function f to produce an effectively secure stream $Z_1 Z_2 Z_3 \ldots$. The question arises 'What properties should the function f have?'

2.5.3 Balanced Functions

The worst possible Boolean function would be a constant function — one that returned a 0 or a 1 every time, whatever the values of the inputs. If a 0 were the only possible output then the ciphertext is equal to the plaintext. If a 1 were produced every time simply inverting the ciphertext bits would produce the plaintext. Less dramatically, if more of the 2^n inputs produce a 1 than a 0 (or vice versa) then there is a bias in the stream produced that may be exploited. Consider a 3-input function with five input combinations giving a 1 and three giving a 0. XOR-ing ciphertext bits with 1 will give plaintext that is 62.5 per cent correct. Furthermore, such biases provide information on the LFSR streams that feed into the combining function (such information is very powerful, see below). This sort of bias is generally avoided. Most applications will use functions that return a 0 for exactly

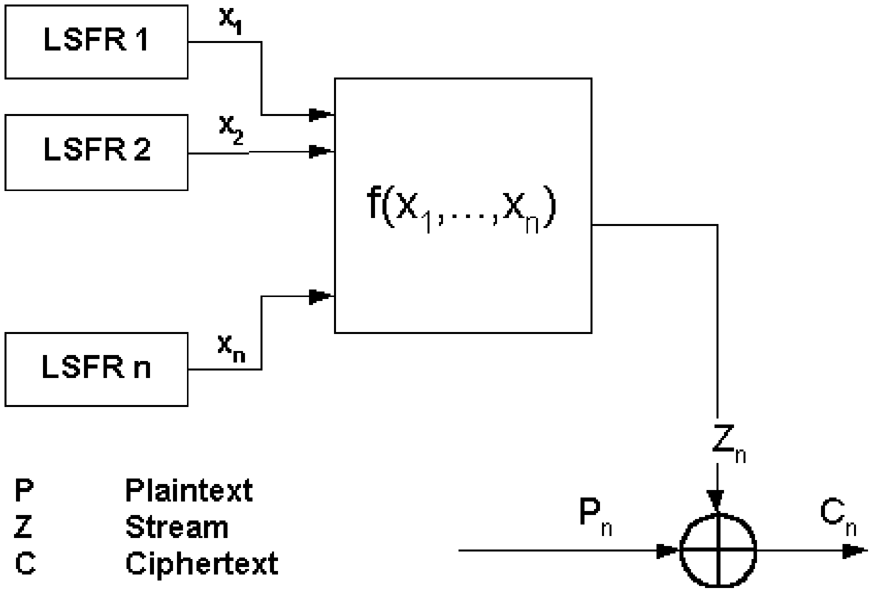

Figure 2.8: Synchronous Stream Cipher

half the inputs and a 1 for the other half. These functions are said to be *balanced*. Two functions f and g are said to be *uncorrelated* when

$$\sum_{x \in \mathcal{Z}_2^n} \hat{f}(x)\hat{g}(x) = 0. \tag{2.7}$$

If so, if f is approximated using g the result is right half the time and wrong half the time. Formally, a Boolean function $f(x)$ is said to be balanced if its polar form $\hat{f}(x)$ satisfies

$$\sum_{x=0}^{2^n-1} \hat{f}(x) = 0. \tag{2.8}$$

A balanced function is therefore uncorrelated with a constant function.

2.5.4 Correlation Immunity

The initial state of the stream cipher comprises the initial register values and the actual tap sequences used for feedback polynomials. This knowledge, which is the system's key, must not be obtained by an adversary. Suppose the output of the combining function f is *correlated* with the output of one LFSR, say $LFSR_1$ and that the cryptanalyst has access to the output of f (corresponding to a known plaintext attack). Each possible configuration of $LFSR_1$ (initial state S_0 and tap sequence T) can be considered in turn. The corresponding output streams can be generated and tested for correlation with the witnessed output of f. The correct choice for initial $LFSR_1$ state will exhibit stronger correlation than the rest. In practice, the cryptanalyst may have direct access only to ciphertext bits. However, biases in the plaintext may be exploited to reveal information about the actual output of f. Exaggerating to make a point, if the plaintext has 90 per cent 1s, then $\{1 \oplus C_i\}_{i=1}^{M}$ will be a 90 per cent correct estimate for the keystream $\{Z_i\}_{i=1}^{M}$. The same attack proceeds much as before, but now requires more ciphertext bits. If each shift register $LFSR_i$ has length N_i and R_i primitive polynomials (corresponding to tap sequences) then the total number of possible keys for such a system is given by

$$\prod_{i=1}^{k} R_i(2^{N_i} - 1). \tag{2.9}$$

Attacking each of the individual LFSRs in turn means that the total number of keys the cryptanalyst needs to consider is

$$\sum_{i=1}^{k} R_i(2^{N_i} - 1). \tag{2.10}$$

This 'divide and conquer' attack was demonstrated by Siegenthaler [114]. Of course, knowing the positions of the taps for the linear feedback decreases the complexity of any attack and increased register length makes it harder. In 1988 Meier and Staffelbach [76] demonstrated more sophisticated attacks that could attack registers up to length 1000 bits if the number of taps was small. A good deal of work has been carried out aiming to exploit similar correlations (e.g. that by Golic [43, 44, 45].)

A first step to avoiding such attacks would be to require no correlation between the output of the combining function f and any single input. Often this will in itself not be sufficient since more sophisticated relationships may be exploited. For example $I_1 \oplus I_2$ may be correlated with the output. This leads to the notion of correlation immunity of specific orders. A Boolean function f is said to be *correlation immune of order m* if every subset of m or fewer input variables is statistically independent of the value of $f(x)$. A balanced correlation immune function of order m is said to be *resilient*. A particularly useful characterisation of correlation immunity, due to Zhen and Massey, is given in the next section.

2.5.5 Algebraic Complexity and Nonlinearity

The relationship between inputs of the function f and its output must be sufficiently 'complex' if it is to resist attack. It is common to say that the relationship should be highly *non-linear*, though there are various interpretations and measures of nonlinearity. *One* measure of nonlinearity is *algebraic degree.* A Boolean function can be expressed as a minimal (XOR) sum of (AND) products:

$$\begin{aligned} f(x_1, \ldots, x_n) \quad = \quad & a_0 \oplus a_1 x_1 \cdots \oplus a_n x_n \oplus \\ & a_{1,2} x_1 x_2 \cdots \oplus a_{n-1,n} x_{n-1} x_n \\ & \ldots \\ & \oplus a_{1,2,\ldots,n} x_1 x_2 \ldots x_n . \end{aligned}$$

Here the a_s are Boolean constants. Given a function f it is a simple matter to construct the minimal sum of products, usually referred to as the *Algebraic Normal Form (ANF)* of the function (Siegenthaler [114] explains how). The highest number of x_i in a product term is the algebraic degree of the function. Thus $f(x_1, x_2, x_3) = x_1 \oplus x_2$ is linear (of degree 1), $f(x_1, x_2, x_3) = x_1 x_3 \oplus x_2$ is quadratic (degree 2) and $f(x_1, x_2, x_3) = 1 \oplus x_1 x_2 x_3 \oplus x_2$ is cubic (degree 3) etc. Siegenthaler [113] has shown that for functions with n inputs and with correlation immunity of order m and algebraic degree d it must follow that $m + d \leq n$. For balanced functions it must be the case that $m + d \leq n - 1$.

High correlation immunity implies low algebraic complexity. This has unfortunate consequences. Several attacks on the standard stream cipher model become easier the more linear the combining function is, e.g. the Best Affine Attack or Massey's multi-sequence equivalent LFSR generation (see [31]). Designers seek to reduce the susceptibility to such attacks by engineering functions that are highly nonlinear, i.e. that are resilient to any linear approximation. High algebraic degree is a common requirement. [4]

To give a precise definition of one very common measure of nonlinearity, it is necessary to introduce some additional concepts. For each $\omega \in 0..(2^n - 1)$ there is a linear function $L_\omega(x)$ defined by Equation 2.5. The polar forms of these functions, viewed as vectors $\hat{L}_\omega = (\hat{L}_\omega(0), \ldots, \hat{L}_\omega(2^n - 1))$ in $\mathbf{R}^{2^n}$, form an orthogonal basis for $\mathbf{R}^{2^n}$. For any function $\hat{f}$ in polar form, the degree to which it is approximated by a linear function $L_\omega(x)$ can be measured by the dot product of $\hat{f}$ with $\hat{L}_\omega$. The Walsh Hadamard Transform value for ω (or just Walsh value for short) captures this notion and is defined by

$$\hat{F}(\omega) = \sum_{x \in \mathbf{Z}_2^n} \hat{f}(x) \hat{L}_\omega(x). \tag{2.11}$$

[4] High algebraic degree is a common and sensible complexity requirement but high degree functions are not necessarily complex. For example, $x_1 \oplus x_2 \oplus \cdots \oplus x_n \oplus x_1 x_2 \cdots x_n$ is almost identical to the linear function $x_1 \oplus x_2 \oplus \cdots \oplus x_n$.

Dividing $\hat{F}(\omega)$ by 2^n gives the *correlation* between the two vectors $\hat{f}$ and $\hat{L}_\omega$. The Hamming distance $d(f,g)$ between two functions $f(x)$ and $g(x)$ is a count of the number of truth table positions in which they differ, i.e.

$$d(f,g) = \#\{x : f(x) \neq g(x)\}. \tag{2.12}$$

Since

$$\hat{F}(\omega) = \#\{x : f(x) = g(x)\} - d(f, L_\omega) \tag{2.13}$$

and also

$$2^n = \#\{x : f(x) = g(x)\} + d(f, L_\omega) \tag{2.14}$$

the Hamming distance $d(f, L_\omega)$ between f and L_ω is given by

$$d(f, L_\omega) = \frac{1}{2}(2^n - \hat{F}(\omega)). \tag{2.15}$$

Resilience to linear approximation is now captured by the formal measure of **nonlinearity**. The nonlinearity N_f of a Boolean function f is the minimum distance to *any affine* function. It is given by

$$N_f = \frac{1}{2}(2^n - WH_{max}(f)) = \frac{1}{2}(2^n - \max_\omega |\hat{F}(\omega)|). \tag{2.16}$$

Note, if f is an affine function then there is some ω for which $|\hat{F}(\omega)| = 2^n$ (i.e. f is either equal to a specific linear function L_ω or its complement) and so its nonlinearity is 0.

A major goal is to reduce the extent to which the function f is approximated by *any* affine function. The following well known theorem, due to Parseval

$$\sum_{\omega \in \mathbf{Z}_2^n} (\hat{F}(\omega))^2 = 2^{2n} \tag{2.17}$$

forces $WH_{max}(f) \geq 2^{\frac{n}{2}}$. This places bounds on the best nonlinearity that can be achieved. Functions achieving this bound are called *bent* functions [105]. They exist only for even numbers of inputs and are never balanced. Parseval's theorem motivates a new cost function family in Chapter 3.

In Section 2.5.4 correlation immunity for a function f was described in terms of statistical independence. An equivalent and very useful characterisation has been derived by Zhen and Massey [47]. This states that a function f is correlation immune of order m if and only if

$$(1 \leq |\omega| \leq m) \Longrightarrow |\hat{F}(\omega)| = 0. \tag{2.18}$$

Here $|\omega|$ denotes the number of bits set in natural binary encoding of the integer ω, i.e. its Hamming weight. The Zhen-Massey characterisation will be used throughout Chapter 4.

2.5.6 Propagation Characteristics and Autocorrelation

The reader familiar with time series will recall the notion of autocorrelation of lag k, which is essentially a correlation between samples taken at times that differ by k. A similar notion holds for Boolean functions. Here the 'lag' is represented by some binary offset vector. For binary input vector x and binary offset vector s the values of $f(x)$ and $f(x \oplus s)$ may be correlated. It may be possible to exploit this. Indeed, various forms of *differential cryptanalysis* do so. We will generally seek to keep such correlations low. Notions of nonlinearity and correlation immunity have been described above. Similar notions apply with respect to these differential characteristics. The autocorrelation function $\hat{r}(s)$ for $s \in 0..(2^n - 1)$ is given by

$$\hat{r}(s) = \sum_{x=0}^{2^n-1} \hat{f}(x)\hat{f}(x \oplus s). \tag{2.19}$$

A function f is said to be satisfy the *propagation characteristic of order* m if

$$(1 \leq |s| \leq m) \Longrightarrow |\hat{r}(s)| = 0. \tag{2.20}$$

Similarly, the autocorrelation $AC(f)$ of a function f is defined to be the modulus of the worst case value of $\hat{r}(s)$

$$AC(f) = \max_{s \neq 0} \left| \sum_x \hat{f}(x)\hat{f}(x \oplus s) \right| = \max_{s \neq 0} |\hat{r}(s)|. \tag{2.21}$$

Autocorrelations $\hat{r}(s)$ provide further conduits for information flow from inputs to outputs. In some cases, a particularly strong relationship may be present known as a linear structure (where it is always the case that $f(x) \oplus f(x \oplus s) = 1$ or always the case that $f(x) \oplus f(x \oplus s) = 0$. The cryptanalytic significance of these (for block ciphers) has been shown by Evertse [34]. They are generally to be avoided. The influential paper by Meier and Staffelbach [77] cites the minimum distance of a function to *any* function with a linear structure as a nonlinearity measure. The autocorrelation $AC(f)$ was offered by Zheng and Zhang as a *global avalanche characteristic* [129], since it considers all autocorrelations, not just those corresponding to s with particular (low) Hamming weights. In a similar vein, another global characteristic was offered: the sum of the squares of the various autocorrelations, i.e.

$$\sigma_f = \sum_s \hat{r}(s)^2. \tag{2.22}$$

Bent functions achieve zero autocorrelation (and indeed maximal nonlinearity).

A fair amount of work has appeared using the two global avalanche criteria (autocorrelation and sum-of-squares) since they were proposed. This has either targeted the creation of low values of the properties directly, e.g. [83, 81, 80] or else investigated relationships between properties, e.g. [116, 122, 71].

2.5.7 Extension of Properties to Block Ciphers

Some of the criteria we have spoken of above find application in block cipher design. Block ciphers encrypt bits a block at a time rather than on a bit by bit basis. They may be thought of as functions $B : \mathbf{Z}_2^n \times \mathbf{Z}_2^m \to \mathbf{Z}_2^n$. A plaintext block of n bits is encrypted using a key of m bits to produce a ciphertext block of n bits. The best known example is the Data Encryption Standard (DES) which has been used world-wide. It is based on 64 bit block encryption and 56 bit keys.

Some block ciphers are product ciphers comprising a number of rounds. The output from a round forms the input to the next. DES, for example, comprises 16 rounds with each round using a different 48 bit key derived from the overall 56-bit key [90]. The precise functionality of each round is clearly crucial. Two very powerful techniques have been developed based on exploiting particular types of structure in the round function: linear cryptanalysis and differential cryptanalysis.

Matsui proposed an attack based on linear approximations of the round functions [75]. A linear expression of input bits P, keybits K, and output bits C of the form

$$P_{r_1} \oplus \cdots \oplus P_{r_a} \oplus K_{s_1} \oplus \cdots \oplus K_{s_b} \oplus C_{t_1} \oplus \cdots \oplus K_{t_c} \tag{2.23}$$

may hold with some probability different from 0.5 (when exercised over all input combinations). If so, then given sufficiently many inputs and corresponding outputs, it will be possible to deduce $K_{s_1} \oplus \cdots \oplus K_{s_b}$. Such a linear expression is said to approximate the functionality of the round. The stronger the deviation from 0.5 the better the approximation. Approximations can be built up to cover the whole n-round cipher allowing one bit of key information to be deduced. (Technically it is usually applied to one or two rounds less than the full cipher and it is possible to deduce more bits but the details are omitted here.)

Matsui gives approximations over a variety of rounds for DES. In fact linear cryptanalysis was the first attack on DES that was better than brute force key enumeration. Many other algorithms have been subjected to linear cryptanalysis. Linear cryptanalysis works by exploiting linear relationship biases between inputs to a round and its outputs. Reducing the biases of available linear relationships protects against this form of attack. In short, we need to make the mapping non-linear. We would now seek to reduce the degree to which linear combinations of the inputs are correlated with linear combinations of the outputs.

Differential cryptanalysis [3] is another significant attack on block ciphers. Linear cryptanalysis exploits relationships between inputs and outputs of a round, differential cryptanalysis exploits how *differences* in inputs are related to differences in the outputs produced. As Langford and Hellman state [69] two chosen plaintexts, P and P', which XOR to a carefully chosen differential plaintext $D = P \oplus P'$ can produce two ciphertexts C and C' such that $DC = C \oplus C'$ takes on a specific value with non-negligible probability.

Within each round the most important components are generally the substitution boxes, or S-boxes. The DES round function for example, makes use of eight 6-input 4-output substitution boxes. Reducing the susceptibility of S-boxes to linear approximation is a significant design goal. Similarly reducing any differential biases as much as possible is also a goal (to protect against differential cryptanalysis). [5] High nonlinearity and low autocorrelation are highly relevant in this respect. The difference here is that any linear combination of the m outputs can participate in an approximation. For each S-box with n inputs and m outputs there are $2^m - 1$ non-trivial linear combinations of the outputs each of which can be used to define an n input single output Boolean function. The measure of nonlinearity for the S-box is taken to be the lowest nonlinearity of any of these linear output combination functions. Similarly the autocorrelation is the highest autocorrelation achieved by any of the linear output combinations.

FEAL has been cryptanalysed by Ohta and Aoki [91]. Kalisksi and Yin have attacked RC-5 [62]. Kaliski and Robshaw have shown how using multiple linear approximations can reduce the amount of data needed to conduct an attack [61]. Ohta et al. [92] have provided improved algorithms for finding best linear approximations (i.e. with greatest biases). Chabaud and Vaudenay have shown links between linear cryptanalysis and differential cryptanalysis [14]. Nyberg and Knudsen have shown how to characterise the susceptibility of DES-like ciphers in terms of the nonlinearity of the round function [86].

2.6 Evolving Boolean Functions

2.6.1 Efficient Hill-climbing for Design

The design of Boolean functions and S-boxes with desirable cryptographic properties remains an important area of cryptological research. The application of heuristic search techniques to these tasks has almost exclusively been carried out by the Security Research Center at the Queensland University of Technology in Brisbane. Several papers have emerged from that group in the past five years.

The early work on efficient hill-climbing is the most important. It is a useful technique in its own right and is used as a component in more sophisticated searches. Millan et al. show that small changes to a Boolean function does not radically alter its nonlinearity (and may not alter it at all) and so some form of guided local search is worth consideration [83]. For any index x, flipping the value of $\hat{f}(x)$ from -1 to 1 or vice-versa causes each Walsh-Hadamard value

[5] Interestingly, DES is highly resilient to differential cryptanalysis. It was designed to be so. The designers must have known in the early-mid 1970s about differential cryptanalysis. Schneier [110] gives a flavour of the controversy surrounding DES.

$\hat{F}(\omega)$ to change by $+2$ or -2. Similarly, if $f(x) = 1$ and $f(y) = -1$ then flipping both values (to -1 and $+1$ respectively) causes each $\hat{F}(\omega)$ to change by $+4$, -4 or else stay the same. Flipping two bits in this way preserves the balance of a Boolean function (assuming it starts as a balanced function).

The authors propose a hill-climbing approach to maximise nonlinearity and compare it with random generation. Hill-climbing is radically better. To improve nonlinearity a move must reduce the absolute value of the maximum $|\hat{F}(\omega)|$. To check whether a move does this, one need only consider the effect on $|\hat{F}(\omega)|$ with extreme or near extreme values. Consider for example single bit flipping. If $M = \max_\omega |\hat{F}(\omega)|$ then if $|\hat{F}(\omega)| = M$ this value must be reduced by 2 by the move. If $|\hat{F}(\omega)| < M - 2$ then any single bit flip will result in $|\hat{F}(\omega)| < M$ and so can be ignored. Similar arguments apply to the balanced case. Restricting attention to the near extreme cases greatly enhances the speed of hill-climbing. Similar ideas and efficiency gains can also be applied to improving autocorrelation.

The most high profile of the hill-climbing papers [81] documents precisely efficient hill-climbing for nonlinearity and autocorrelation and investigates a variety of joint property hill-climbing strategies. It considers nonlinearity and autocorrelation as goals and characterises a set of search strategies according to the restrictions they impose on the acceptance of moves around the state. Thus, nonlinearity strategies may be characterised as *strong*, *weak* or *none*. A strong strategy allows only moves that strictly improve nonlinearity. A weak strategy requires that a move does not worsen nonlinearity. Finally, it is possible not to place a restriction on the search. The terms apply also to autocorrelation strategies. This gives nine combinations of strategy. Thus a ‘strong-strong’ strategy will only allow moves that strictly improve both criteria.

The results show the importance and power of basic hill-climbing. The authors note that nonlinearity is improved over random generation when strong autocorrelation rules are applied (even when no restrictions with respect to nonlinearity are imposed). They state that this is due to the ‘qualitative connection between the maximum values of WHT [Walsh Hadamard Transform] and AC [autocorrelation]’ [81]. Once again the cost functions are direct expressions of the property desired (nonlinearity and autocorrelation).

2.6.2 Criteria Targeted and Cost Functions Used

The initial work was aimed solely at nonlinearity. This was quickly extended to cover autocorrelation. In both cases the objective function was itself used as the fitness or cost function. Thus, when high nonlinearity or low autocorrelation was the goal the fitness (cost) functions were:

$$\mathbf{fitness}(f) = N_f = \frac{1}{2}\left(2^n - \max_\omega |\hat{F}(\omega)|\right)\text{; and} \qquad (2.24)$$

$$\mathbf{cost}(f) = AC_f = \max_s |\hat{r}(s)|. \tag{2.25}$$

Later work [84] sought Boolean functions that were correlation immune (of degrees 1 and 2) or which satisfied the so-called *strict avalanche criterion* (or, equivalently, the propagation criterion of order 1). To couch the search for such functions as optimisation problems notions of correlation deviation and propagation deviation were defined:

$$cidev_m(f) = \max_{|\omega| \leq m} |\hat{F}(\omega)|; \text{and} \tag{2.26}$$

$$pcdev_m(f) = \max_{1 \leq |s| \leq m} |\hat{r}(s)|. \tag{2.27}$$

This is of particular interest since it addresses tradeoffs that can be made by automated search. It would seem that a multi-criteria optimisation is where the techniques may potentially have greatest benefit compared with other approaches.

What links all of the above functions is their *directness*. Although each indicated cost or fitness function may characterise well the goal of a particular search (for example, a zero-cost solution of Equation 2.26 is correlation immune of order m), it does not follow that it is a good cost or fitness function for guiding a search. Consider the nonlinearity fitness function of Equation 2.24. Suppose $\max_\omega |\hat{F}(\omega)| = M$. If there is a single value of ω with $|\hat{F}(\omega)| = M$ then there is greater possibility of improving the nonlinearity than if there are, say thirty or more ω with this value. Similar considerations apply to all the cost functions above. Although the fitness or cost values are expressed as functions of particular extreme elements, the ability to reach better values depends on the values of *other* elements. This observation forms the crux of the work in Chapters 3 and 4.

2.6.3 Optimisation Techniques Used

The Brisbane work has made use of random generation (for comparison purposes), hill-climbing and genetic algorithms [82]. Hill-climbing has generally been found to be a useful final stage to any optimisation based approach. This is consistent with application of genetic algorithms in other fields where an element of local search is often brought to bear.

Hill-climbing is conceptually straightforward. The efficiency savings of the smart hill-climbing are very considerable. It is harder to see why the genetic algorithms work. Standard genetic algorithms do not work. The work often makes use of a problem specific cross-over method and occasionally incorporates elements of intermediate hill-climbing. Although the genetic algorithms have shown themselves to be better than hill-climbing, it appears to this author that they are not obviously natural candidates for boolean function design. Interesting representational issues arise with the use of genetic algorithms. With the vector representations indicated earlier (truth tables) it is entirely possible to mate two excellent functions to get awful children. Indeed, two optimally non-linear functions may be isomorphic (under relabelling of variables) and yet mate to give children with low nonlinearity. Also combination like this must preserve balance. Millan et al. [84] give balance preserving crossover approaches.

There would appear to be no application of more ‘local’ search techniques (such as simulated annealing).

2.6.4 Generalisation

The work on balanced functions was generalised to encompass bijective [79] and regular [80] S-Boxes. A bijective S-Box is an invertible function $f : \{0,1\}^n \rightarrow \{0,1\}^n$. In a regular S-box on n input and m outputs each output occurs precisely 2^{n-m} times. Regularity is the vector-valued output version of balance. In both cases the cost or fitness functions used are very direct. The nonlinearity of an S-box is just the worst nonlinearity of any linear combination of the outputs (similarly for autocorrelation). These objective functions measures are used directly as the fitness and cost functions.

The MARS S-box work of Burnett et al. [9] is clearly very significant. IBM has a great deal of cryptographic expertise and is not short of computing power! Two and a half hours computing on a PC allowed Burnett et al. to find boxes with better properties than those proposed by IBM. This paper is fairly recent but has aroused interest. Heuristic search for cryptological applications have rarely caused surprises. This work is an exception.

2.6.5 Successes Achieved

The Brisbane work has equalled the best achieved nonlinearity values for balanced Boolean functions for $n = 5, 6, 7, 8$. At higher n the effectiveness of the techniques appears to drop. It would seem fairly easy also to achieve correlation immune functions of order 1 with reasonable nonlinearity. No functions of correlation immunity degree 2 have been achieved. The MARS S-box work is significant as indicated above.

2.6.6 General Commentary

The Security Research Centre in Brisbane have been responsible for virtually all progress on the use of guided search for Boolean function and S-box design. The work has attacked a range of desirable properties (nonlinearity, low autocorrelation, correlation immunity and propagation characteristics). It has also been extended to S-box design. The superiority of basic guided search over techniques such as random generation has been repeatedly demonstrated. The most significant outputs are the efficient hill-climbing algorithms for nonlinearity and low autocorrelation.

The techniques have shown promise but there have been few surprises. The MARS S-box design work is an exception. The body of work remains an excellent basis on which to build. To cause some surprises it is necessary to consider what has prevented the techniques from performing better. Directness of the cost functions has been identified as a potential restriction and also, perhaps, a reluctance to embrace more sophisticated local search techniques.

2.7 Cryptanalysis of Systems Based on NP-Hard Problems

Several cryptosystems have been proposed whose security relies on the difficulty of solving instances of NP-hard problem types. Two areas of particular interest are encryption schemes and identification schemes. These should be ‘home ground’ for techniques such as simulated annealing and genetic algorithms. Analysing the application of metaheuristic search to such problems, however, would seem to have few highlights, as is argued below.

2.7.1 Cryptanalysis of Knapsack Encryption Schemes

Knapsack encryption schemes have been the subject of great controversy over the years. The first published public key cryptosystems were based on knapsacks, and so knapsacks are of great historical importance. There have been many variants and the history of their development makes for exciting reading. Odlyzko charts the ‘rise and fall’ of knapsacks [88] as does Moore [85]. They are not used much nowadays. Their security just seems too fragile; using them would be too risky. Cryptographic knapsacks are based on the Subset Problem:

> Given a finite set $W = \{w_1, \ldots, w_n\}$ of positive integers and a positive integer C, does there exist a subset $W' \subseteq W$ such that the sum of all elements in W' is equal to C?

The relationship with encryption is straightforward.

> Let the message space be the set of binary strings $b_1, \ldots, b_n$. Let $W = \{w_1, \ldots, w_n\}$ be a set of positive integers as defined above. Encrypt message $M = b_1, \ldots, b_n$ as $C = \sum_{i=1}^{n} b_i w_i$.

If the subsets of A are arranged to have unique sums then this encryption is reversible and so given a sum S there will be a unique plaintext message. Encryption is clearly fast but decryption is generally very hard indeed unless there is a secret trapdoor available to the receiver. The original Merkle-Hellman knapsacks have been attacked, as have multiplicative knapsacks and the Chor-Rivest knapsack. The details of how the receiver's trapdoor works is of no concern here. Only the enemy's cryptanalysis search problem is of concern: given the sum C and a knapsack of elements, can the actual plaintext be recovered?

2.7.2 All Knapsacks Seem Very Small

It would appear that all metaheuristic search work in this area is based on extraordinarily small problems. The initial work by Spillman [119] dealt with knapsacks of size 8 and 15. The work used a rather odd cost function with an unclear rationale (Clark and Dawson [18] also seem unclear about the precise nature of the cost function and offer a cost function that is more intuitive and demonstrably better anyhow). In 1997, Kolodziejczyk [67] demonstrated that Spillman's results were distinctly suboptimal and that variation of the genetic algorithm parameters could readily improve results. She concludes 'the genetic algorithm offers a powerful tool for the cryptanalysis of knapsack ciphers.' Since the experiments were limited to knapsacks of size 8 and for 5 specific 'messages' (the ASCII encodings of the letters of the word "MACRO" formed the five messages) this seems without foundation. In 1998, Lebedko and Topchy [87] noted 'it is unclear how capabilities of these techniques scale up with dimension of the problem because cryptographically strong applications require at least 10^{20} times bigger search space'. I agree entirely.

The work of Lebedko and Topchy [87] and similarly that of Clark and Dawson [18] is of particular interest in that both challenge the actual usefulness of the cost functions chosen by previous researchers. If C is the target sum and $b = b_1 \ldots b_n$ is a proposed solution, the fitness of that solution was generally of the form

$$f_1(b) = g\left(\frac{|C - \sum_{i=1}^{n} w_i b_i|}{\sum_{i=1}^{n} w_i}\right), \tag{2.28}$$

where $g()$ is some monotone function. Lebedko and Topchy note that neighboring points of the search space have significantly different values of fitness and that the intuitive notion of neighborhood based on Hamming distance is very hard to hill-climb. Amusingly they proposed a counter-intuitive cost function based on actual Hamming distances,

$$f_2(b) = H\left(C, \sum_{i=1}^{n} w_i b_i\right), \tag{2.29}$$

with similar performance. They report the results for knapsacks of sizes 15 and 20 and provide commentary on the role of fine-tuning via local search. The need for fine tuning is well-established in the genetic algorithms community. Clark and Dawson take this notion further giving an improved fitness function. They measured the fitness value for all 2^{30} possible solutions of a knapsack of size 30 and a randomly chosen secret. The Hamming distances from the actual secret were measured for high fitness solutions. The results are reproduced below in Table 2.1. One is left with the inescapable conclusion that the fitness function is far from ideal! Indeed, Clark and Dawson comment on the table results stating 'the correlation between the fitness and Hamming distance is essentially random.' This statement is not true! Look closely at the table. Consider the numbers of solutions with fitnesses greater than 0.95. For a Hamming Distance HD of 3 there 48 solutions, for HD of 27 there are 31. For HD of 4 there are 336, for HD of 26 there are 223, for HD of 5 there are 1692 and for HD of 25 there are 1208. It is clear that there is a very stark statistical bias. Highly fit solutions are more likely to be closer to the actual solution. A one-shot approach to optimisation is unlikely to solve the problem but what would happen if there were many thousands of attempts to solve the problem via metaheuristic search? Could the results obtained not be combined to exploit the statistical bias indicated?

In 1998 and 1999 Yaseen and Sahasrabuddhe attacked multiplicative knapsack ciphers [126] and the Chor-Rivest cipher [127]. Their work is a little more sophisticated in that they allow themselves many target sums (those in the neighborhood of the current sum). There is no known successful attack on the Chor-Rivest scheme but again the sizes of knapsack used cast doubt over the ability of the techniques to scale.

The author has included the knapsack work mostly for reasons of completeness. If one point emerges from the critique above it is the chasm between real cryptographic knapsacks and those addressed by heuristic search researchers. There would appear to be a severe lack of ambition. As stated above, many knapsack schemes have been broken by other means.

2.7.3 Identification Protocols based on NP-Complete Problems

A zero knowledge protocol allows a principal to demonstrate he holds a secret without actually revealing that secret. Originally proposed by Goldwasser et al. in 1985 [42], Fiat and Shamir give impetus to the topic by showing how such protocols might be used to prove user identities [37]. Their first scheme was considered impractical and the second revolved around public key cryptography (and so used large numbers). Subsequent attempts have been made to obtain zero-knowledge protocols by appealing to known NP-hard problems from the literature, since the problem can be formulated much more efficiently (in terms of memory

Hamming	Fitness Value			
Distance	> 0.95	> 0.99	> 0.999	> 0.9999
30	0	0	0	0
29	0	0	0	0
28	7	0	0	0
27	31	2	0	0
26	223	8	0	0
25	1208	51	0	0
24	5080	188	3	0
23	17289	654	6	0
22	49186	1985	16	0
21	119098	4720	50	1
20	248696	9789	82	1
19	451327	18176	182	2
18	714445	28630	275	2
17	989898	39301	395	3
16	1207311	47986	500	5
15	1298561	52285	515	3
14	1230811	49383	513	4
13	1027101	40613	395	7
12	755006	30014	295	1
11	487647	19785	203	0
10	275545	10988	113	2
9	135305	5214	58	2
8	57498	2298	26	1
7	20973	911	14	0
6	6446	247	0	0
5	1692	57	2	0
4	366	14	0	0
3	48	3	0	0
2	6	0	0	0
1	0	0	0	0
0	1	1	1	1

Table 2.1: Hamming Distance Distribution for High Fitness

storage and computation needed). Some of these are given below.

Syndrome Decoding. Syndrome decoding, due to Stern, appeals to a known problem from coding theory [121]. Suppose H is an n by k binary matrix (arithmetic will be $\mod 2$) and let s be an m by 1 secret with a fixed number p of 1's in it. Compute $i = Hs$, the image of s under H. A cryptanalyst must recover s given H and i. Typical values suggested are $(n, k, p) = (512, 256, 56)$ and $(1024, 512, 110)$ derived by appealing to theoretical results.

Permuted Kernel Problem. Adi Shamir published another scheme based on the Permuted Kernel Problem (PKP) [111]. Given a prime number p, an m by n matrix $A = (a_{ij})$ over $\mathbf{Z}_p$, and an n by 1 vector $V = (V_j)$ over $\mathbf{Z}_p$, find a permutation vector V_π (i.e. an n by 1 vector with elements obtained by shuffling the elements of V) such that $AV_\pi = 0$. Shamir suggested several scheme sizes for (m, n, p): (16,32,251), (37,64,251). Patarin and Chauvaud [96] showed how the ideas of previous researchers [2, 58] could be combined and augmented. They give various algorithms making different space-time tradeoffs. For the smallest size their algorithms can break the system in a time of order $2^{52} - 2^{54}$ (and maximum memory of 2^{24} n-tuples). For the (37,64,251) problem the running time is 2^{116} with memory requirement of order 2^{65}. The memory requirement can be reduced to 2^{17} but only at the expense of increasing the running time to order 2^{123}. Note that this improves on previous results.

Permuted Perceptron Problem. In 1995 David Pointcheval proposed a scheme based on the Permuted Perceptrons Problem (PPP) — a more difficult version of the Perceptrons Problem (PP) [98]. The PP is as follows: given an (n, m) matrix A comprising elements from the set $\{1, -1\}$, find an m-vector x such that all elements $w_i = (Ax)_i \geq 0$. Feasible instances can be generated randomly (negating all the elements in matrix row i if $(Ax)_i < 0$ initially). This is a known hard problem. To make it harder (and hence reduce the size of matrices and secrets needed) Pointcheval suggests forming the histogram of (positive) values of the image vector w. Since every solution to the PPP is also a solution to the PP, it would appear that this is harder. It would appear much harder. Pointcheval provides a comparison of the efficiency of this scheme against others.

Unusually, Pointcheval (laudably) provides an assessment of attacks by simulated annealing. In this he may be unique. The cryptanalysis of the PPP by Knudsen and Meier [65] is perhaps the most subtle annealing paper to date. The most subtle element of the work described is the choice to carry out multiple runs of annealing and look for commonality. Typically multiple runs are carried out and common elements of the results determined. These are then fixed and the whole process repeated. Unfortunately, some elements will get fixed wrong early in this procedure. This causes subsequent elements to be fixed at wrong values (because annealing will try faithfully to minimise the cost function having had an element erroneously fixed — there may not actually be a solution).

By profiling this technique in operation, it is possible to determine roughly where the technique first sets a bit wrongly. This allows an enumerative search to be carried out. For example, in the (101,117) problem case if we assume that the technique fixes the first 70 bits correctly there are only 2^{47} possible values for the remaining bits. More sophisticated variants are given but this captures the basic idea.

What is very different here is the notion of repeated runs being used and the monitoring of the technique in action. The technique uses the *distributional properties* of local optima attained using annealing. These are important ideas with considerable potential, as is shown in Chapter 5.

A general comment may be made on the ways schemes based on NP-complete problems are presented. NP-completeness is a statement about the complexity of the *worst-case* computation required to solve an instance. This is pretty much irrelevant to cryptography. It seems that NP-completeness is used informally as a badge meaning just 'hard' generally. The criterion of most relevance to a cryptanalyst is the computational complexity of *this case*. This is not a trite observation. In Chapter 5 it is shown how the specific means of generating instances may seriously compromise the security of a scheme.

2.8 Miscellaneous

Most applications of metaheuristic search techniques to cryptological problems of relevance in this thesis have been considered. There are occasional instances of more sophisticated problems being attacked. Clark [19] has applied annealing techniques to LFSR reconstruction (a fairly sophisticated application). In addition, the author has uncovered web pages hinting at even more sophisticated applications, e.g. the use of genetic algorithms for linear and differential cryptanalysis [128] (though papers have not yet appeared in print). David Boney, a PhD candidate at Georgetown University had posted a term paper entitled 'Block Cipher Cryptanalysis with Genetic Programming' but this is no longer posted (and nothing seems to have been published on this score).

2.9 General Commentary

There would appear to be few applications of metaheuristic search to modern-day cryptological applications. It would also be fair to say that there have been few surprises. The techniques seem generally to have 'promise' but other techniques have typically predominated. If the work in this thesis is to have any impact on the cryptological community something different must be done. The question is 'But what?' The author's research technique in this respect is brutally simple. The prevalent patterns of usage of cryptological metaheuristic search are identified below. In the following chapters the aim will be to deviate as far as possible from them. Though a little perverse, the application of heuristic search in cryptology does give the impression of being a little 'sleepy'; any mechanism that generates new ideas is welcome.

The author believes that the following patterns predominate in the literature.

Directness Assumption In all the literature surveyed the cost functions used were 'direct' or 'obvious'. In carrying out the survey the author never came across a cost or fitness function whose motivation was not obvious. It is generally accepted that choice of cost function is crucial but there appears to be little experimentation with unusual cost functions. Cryptology would seem to provide ample opportunity to embrace indirect approaches since subtle relationships abound (e.g. between nonlinearity and autocorrelation, between algebraic degree and correlation immunity).

Black Box Assumption Every cryptological application surveyed applied a search technique to some problem instance and obtained a 'result'. There is no attempt to use how a search arrived at such a result as a source of information.

One-shot Assumption Every cryptological application surveyed applied a search technique to some problem instance and obtained a 'result'. There would appear to be only a single paper that uses information from multiple runs to attack a modern-day cryptological problem [65].

Concrete Assumption There is something that links all the *problems* to which metaheuristic search has been applied in cryptology. They are all simple and very low-level problems. There are no applications to abstract problems. This seems to indicate a considerable lack of confidence or ambition.

The subsequent chapters give examples when deviation from each of these patterns can be used to excellent effect. The aim is to show that the toolkit of heuristic search approaches can be significantly widened. The identification and challenging of major assumptions is just a means to an end.

Chapter 3

Evolving Boolean Functions

Boolean function design has been the subject of a great deal of theoretical research. In this chapter simulated annealing techniques are used to derive functions with particular desirable cryptographic properties. For small numbers of input variables, functions with properties as good as (and sometimes better than) any seen so far are demonstrated. Some open conjectures in the literature are disproved. Several new aspects of the conceptual 'toolkit' are introduced.

3.1 Introduction

In Chapter 2 a variety of desirable criteria for functions with cryptographic application were identified (balance, high nonlinearity, low autocorrelation, correlation immunity of reasonably high order, high algebraic degree etc.) The tradeoffs between these criteria are improperly understood and have been the subject of recent research, e.g. [13, 70, 72, 95, 116, 122, 123, 124]. The more criteria that have to be taken into account, the more difficult the problem. Generating artifacts that possess several excellent properties simultaneously seems very hard. For some individual properties, it is unclear whether the best theoretical bounds are tight even for small numbers of input variables. For example, the best upper bound for nonlinearity for balanced functions on eight input variables is 118 but the best value demonstrated is 116. Upper bounds on achievable nonlinearity for balanced functions on even numbers of input variables have been the subject of conjecture [32]. Lower bounds on achievable autocorrelation for balanced functions have also been the subject of conjecture [70, 129].

The research reported in this chapter concentrates on four criteria and investigates whether simulated annealing can be applied to good effect. These criteria are:

- balance;

- high nonlinearity;
- low autocorrelation; and
- high algebraic degree.

These criteria, in various combinations, have proven of interest to cryptological researchers (from both theoretical and optimisation perspectives). They form a plausible platform on which to test the efficacy of an annealing-based approach. Extension to correlation immunity (and other properties) is carried out in Chapter 4.

3.2 Motivation for a New Cost Function

In Chapter 2 it was noted that existing optimisation-based work aimed at producing highly nonlinear functions has generally used nonlinearity itself as the fitness function, i.e. the fitness of a function f on n input variables is given by

$$\mathit{fitness}(f) = N_f = \frac{1}{2}(2^n - \max_{\omega} |\hat{F}(\omega)|), \tag{3.1}$$

or, when viewed as a minimisation problem, the cost function is given by

$$\mathit{cost}(f) = \mathit{WH}_{max}(f) = \max_{\omega} |\hat{F}(\omega)|. \tag{3.2}$$

Similarly, with low autocorrelation as the target, the autocorrelation itself has been used as the cost function, i.e. the cost function is given by

$$\mathit{cost}(f) = \mathit{AC}(f) = \max_{s \neq 0} |\sum_{x} \hat{f}(x)\hat{f}(x \oplus s)| = \max_{s \neq 0} |\hat{r}(s)|. \tag{3.3}$$

A typical optimisation approach to multi-criteria problems is to take a weighted sum of the individual cost functions. For the target criteria, this would lead to consideration of cost functions like

$$\mathit{cost}(f) = \alpha WH_{\max}(f) + \beta AC(f) + \gamma(n - \mathit{Degree}(f)) + \delta \mathit{Imbalance}(f), \tag{3.4}$$

where $Imbalance(f)$ measures deviation from balance, for example the absolute difference between 2^{n-1} and the Hamming weight of f. If further criteria were of interest cost function components for these would typically be added. Increasing the number of components will generally entail a great deal of experimentation to determine optimal settings of the component weights. In addition, although optimisation attempts using cost function components such as those indicated have shown promise, rarely have they caused real surprise. This leads one to ask 'Is there a simpler, more effective way forward?' An equation from Chapter 2 provides an opportunity for experimentation. Parseval's equation below

$$\sum_{\omega}(\hat{F}(\omega))^2 = 2^{2n} \tag{3.5}$$

constrains $WH_{max}(f) = \max_{\omega}|\hat{F}(\omega)|$ to be at least $2^{\frac{n}{2}}$. It achieves this bound when, for each ω, $|\hat{F}(\omega)| = 2^{\frac{n}{2}}$. When some $|\hat{F}(\omega)|$ are less than this ideal bound, Parseval's theorem ensures that some $|\hat{F}(\omega)|$ must be greater than it. Thus, attempting to restrict the *spread* of absolute Walsh values achieved would seem to be a *possible* means of achieving high nonlinearity. Some functions achieve this ideal bound. They are generally known as *bent* functions, discovered by Rothaus [105]. They are identical to what Meier and Staffelbach term 'perfect nonlinear functions' [77]. These functions exist only for even numbers of input variables. As well as having the highest possible nonlinearity, such functions also have *zero* autocorrelation. Thus a cost function similar to

$$cost(f) = \sum_{\omega}||\hat{F}(\omega)| - 2^{\frac{n}{2}}| \tag{3.6}$$

would seem a simple candidate for attacking nonlinearity *and* autocorrelation. Functions achieving the ideal bound must have $|\hat{F}(0)| = 2^{\frac{n}{2}}$ and so are not balanced (balanced functions have $\hat{F}(0) = 0$). Even if this particular cost function is unsuitable for evolving desirable balanced functions, might a similar one be appropriate? Cost functions of the form

$$cost(f) = \sum_{\omega}||\hat{F}(\omega)| - X|^R \tag{3.7}$$

would seem plausibly well-motivated. Equation 3.7 generalises Equation 3.6 above. The X and R parameters provide freedom to experiment for the problem at hand. It is far from clear what the effect of imposing a balance requirement will be and what the effect of an odd number of input variables will be. It is difficult to predict what the best parameter values should be. Some parametric flexibility is justified.

Even assuming that the cost function family of Equation 3.7 can handle nonlinearity and autocorrelation, balance and degree must still be considered. These will be handled in different ways — the search will be constrained to move only between balanced functions and algebraic degree will simply be ignored for the purposes of providing guidance to the search via the cost function. It would be possible to allow the search space to include unbalanced functions but this would require an additional cost function component to 'price out' imbalance. It seems easiest to avoid it. Ignoring algebraic degree is a conscious choice. The resulting functions will have some algebraic degree. It may turn out to be high, it may not. Random search typically produces functions with high algebraic degree and there is nothing obvious in the proposed cost function family to drive the search towards low algebraic degree. It will be possible to consider algebraic degree explicitly if necessary.

3.3 The General Approach

A balanced function will be represented using polar form, i.e. as a vector $\hat{f}$ in $\mathbf{R}^{2^n}$ with 2^{n-1} elements equal to 1 and 2^{n-1} elements equal to -1. Local search will be used throughout. A search starts with a balanced (but otherwise random) function in polar form. A valid move simply swaps two dissimilar vector elements and so preserves balance — the (equal) numbers of 1s and -1s are maintained. In formal terms, we can define the neighbourhood of a function $\hat{f}$ as follows. The function $\hat{g}$ is in the neighbourhood of $\hat{f}$ if

$$\begin{array}{lll} \exists\, x, y \in \mathbf{Z}_2^n & \bullet & \hat{f}(x) \neq \hat{f}(y) \wedge \\ & & \hat{g}(x) = \hat{f}(y) \wedge \\ & & \hat{g}(y) = \hat{f}(x) \wedge \\ & & \forall z \in \mathbf{Z}_2^n \setminus \{x, y\} : \hat{g}(z) = \hat{f}(z). \end{array}$$

The approach is as follows:

1. Use an annealing-based search to minimise the value of the new cost function (suitably parametrised) given in Equation 3.7. Let the best solution produced during the search be f_{sa}.
2. Hill-climb from f_{sa} with respect to nonlinearity (or autocorrelation) to produce the final solution f_{sahc}
3. Measure the nonlinearity, autocorrelation and algebraic degree of f_{sahc}.

Although nonlinearity, autocorrelation and algebraic degree are all of interest, the approach is somewhat unusual in that Stage 1 targets none of the criteria directly, Stage 2 considers only one of the first two, and algebraic degree is never considered at all (it is simply measured at the end). The motivation for Stage 1 is very approximate. Its possible use for evolving *balanced* functions with desirable properties is largely based on *analogy* with bent function characterisations, not theoretical analysis. Though the motivation is plausible, there remains the question of whether the idea has any real merit.

3.4 Experiments Performed

Two approaches have been used in experiments. In the first, the second-stage hill-climbing is with respect to nonlinearity. We shall refer to this approach as the NLT (Non-Linearity Targeted) approach. In the second, the second-stage hill-climbing is with respect to autocorrelation. We shall refer to this as the ACT (Auto-Correlation Targeted) approach. For each approach, attempts were made to evolve functions with 5–12 input variables. The target properties are now considered individually and then in combination.

3.4.1 Experimental Results for Nonlinearity

The NLT and ACT approaches were applied over a range of X and R values for the parameters of the cost function of Equation 3.7. Table 3.1 shows the X and R values used together with the parameters of the annealing algorithm (described in Section 2.3.2). α is the geometric cooling parameter for the annealing algorithm, MIL is the number of moves attempted in each inner loop, $MaxIL$ is the maximum number of inner loops for the search. For all runs the maximum number of consecutive unproductive (without any move being accepted) inner loops (MUL) before the search ends was 50. 100 runs of the algorithm were carried out for each parameter set.

Table 3.2 summarises the results obtained. The best values obtained by theoretical construction are shown together with best theoretical upper bounds (based partly on a similar table in [84]). Dobertin's well-known conjecture (stating that for balanced functions on an even number n of inputs the highest achievable nonlinearity $NL(n)$ satisfies $NL(n) = 2^{n-1} - 2^{\frac{n}{2}} + NL(\frac{n}{2})$ is taken from [32].

For eight input variables or fewer the technique can rapidly achieve the indicated theoretical bounds, often requiring only a few seconds on a 1.4 GHz PC. Typical times (for illustration the average times for $X = 0$ and $R = 3.0$) per run are shown in Table 3.3. The interesting cases are for nine to twelve variables. The annealing techniques begin to out-perform previous optimisation techniques. (The genetic algorithms results of Millan et al. [84] are the best results for optimisation-based approaches to date.) This is most dramatic for n=12, the largest size considered here. Indeed, the ACT approach also gives rise to examples with nonlinearity values equal to or in excess of previous results. The technique produces results that are competitive with a well-known construction (the concatenation of bent functions). However, as n increases the best constructions are still significantly better.

The improvement over previous optimisation-based research results would appear primarily due to the new cost function family of Equation 3.7. To confirm this, for each n , 100 annealing runs were carried out with the standard direct cost

n	2nd Stage	X Range (min:max:step)	R Values	α	MIL	$MaxIL$
5	NLT:ACT	(-10: 10: 2)	2.5, 3.0	95	400	400
6	NLT:ACT	(-10: 10: 2)	2.5, 3.0	95	400	400
7	NLT:ACT	(-6: 18: 2)	2.5, 3.0	95	400	400
8	NLT	(-16: 16: 2)	2.0,2.5, 3.0	95	400	400
8	ACT	(-8: 16: 2)	2.5, 3.0	97	500	500
9	NLT	(-8: 20: 2)	2.5, 2.75, 3.0	95	400	400
9	ACT	(-8: 20: 2)	2.5, 3.0	97	500	500
10	NLT	(-8: 20: 2)	2.5, 3.0	95	400	400
10	ACT	(-8: 20: 2)	2.5, 3.0	97	500	500
11	NLT	(-8: 30: 2)	2.5, 3.0	95	400	400
11	ACT	(-8: 16: 2)	2.5, 3.0	97	500	500
12	NLT:ACT	(-8: 30: 2)	2.5	98	1000	1000

Table 3.1: Search Parameters Used

function given by Equation 3.2. A cooling rate of 0.98 was used together with MIL=1000, MaxIL=1000 and $MUL = 50$. Thus, the traditional cost function was given a far greater computational chance to work. The performance of annealing using this direct measure of nonlinearity followed by hill-climbing with respect to nonlinearity (shown in Table 3.2 as Direct NL) is markedly worse than the results of both NLT and ACT.

It is also very clear that the number of moves in a loop MIL is generally very low especially for the larger n. The approximate nature of Stage 1 enables some short cuts to be taken in this respect. [1] However, it seems prudent to revisit this issue and carry out some runs with considerably higher MIL. Accordingly, additional experiments were also carried out using $MIL = 40000$ for $n = 9$ and $n = 10$ with $R = 3.0$ and X ranging over $0, 4, 8, 12, 16, 20$. Despite the hundred-fold increase in MIL no improvements on currently achieved values were obtained.

3.4.2 Experimental Results for Autocorrelation

Work on lower bounds for autocorrelation is less well-established and recent years have seen researchers make conjectures as well as providing constructions for highly nonlinear functions with low autocorrelation. The work of Zhang and

[1] Early results published in [23] indicated that the purpose of the annealing stage was to get the search into the 'right area' from which hill-climbing could give good nonlinearity. Actually finding a global optimum for Equation 3.7 was somewhat secondary.

Method	5	6	7	8	9	10	11	12
Lowest Upper Bound	12	26	56	118	244	494	1000	2014
Best Known Example [53, 52]	12	26	56	116	240	492	992	2010
Dobertin's Conjecture [32]		26		116		492		2010
Bent Concatenation	12	24	56	112	240	480	992	1984
Random	-	-	-	112	230	472	962	1954
Random Plus Hill-Climb	-	-	-	114	236	476	968	1961
Genetic Algorithms [84]	12	26	56	116	236	484	980	1976
Direct NL	12	26	56	114	236	480	974	1972
NLT	12	26	56	116	238	486	984	1992
ACT	12	26	56	116	238	484	982	1986

Table 3.2: Comparing the Nonlinearity of Balanced Functions

n	Time Per Run (seconds)	n	Time Per Run (seconds)
5	0.9	9	16.5
6	3.1	10	38.8
7	3.0	11	83.8
8	6.6	12	1030 ($R = 2.5$)

Table 3.3: Typical Times Per NLT Run ($R = 3$ and $X = 0$)

	5	6	7	8	9	10	11	12
Zhang and Zheng	8	16	16	24	32	48	64	96
Maitra Construction	8	16	16	24	32	40	64	80
Maitra Conjecture		16		24		40		80
Direct AC	8	16	16	32	56	80	128	200
NLT	8	16	16	16	40	64	96	144
ACT	8	16	16	16	40	56	88	128

Table 3.4: Conjectured Bounds and Attained Values for Autocorrelation of Balanced Functions

Zheng [129] is widely referenced and recent (2000) work by Maitra [70] has considerably improved on this. Zhang and Zheng provided constructions for functions f_n on $n = 2k$ and $n = 2k + 1$ input variables such that

$$AC(f_n) \leq 2^{k+1} \tag{3.8}$$

and conjectured that balanced functions g on n variables with algebraic degree at least 3 satisfied

$$AC(g) \geq 2^{\frac{1}{2}(n+1)}. \tag{3.9}$$

Since autocorrelation values for balanced functions are multiples of 8, we can round up to the next available value. Maitra conjectured [70] that, for even n, autocorrelation bounds $ACB(n)$ for balanced functions are provided by the relationship

$$ACB(n) = 2^{\frac{n}{2}} + ACB(\frac{n}{2}). \tag{3.10}$$

The values $ACB(3) = ACB(4) = ACB(5) = 8$ have been reported as having been obtained by researchers using enumerative search (the author has obtained each of these values in under a second with annealing-based approaches). Table 3.4 records the best autocorrelation values obtained by recent theoretical constructions and also by the NLT and ACT approaches together with the bounds from Maitra's conjecture. For nine variables or more the annealing approach would not appear to be able to match the conjectured or achieved bounds (Maitra has demonstrated highly nonlinear functions at these bounds). However, for eight variables the technique has generated a counterexample to Maitra's conjecture. In addition, if any of the generated functions with an autocorrelation of 16 has degree greater than 2, it is a counter-example to the conjecture by Zhang and Zheng also. This turns out to be the case. Indeed, almost all examples generated with this autocorrelation had algebraic degree of 6. Maitra has independently formed a counter-example to Zheng and Zhang's conjecture for functions on 15 inputs (based on a modification of Pedersen-Wiedermann functions). Maitra's conjecture was brought to the author's attention by Millan (Security Research Centre,

(5,3,12,8)	(6,5,26,16)	(7,6,56,16)	(8,7,116,24)
(5,4,12,16)			(8,5,112,16)
(9,8,238,40)	(10,9,486,72)	(11,9,984,96)	(12,10,1992,156)
	(10,9, 484, 64)	(11,10,982, 96)	(12,10,1990,144)

Table 3.5: Best Values (n , d , nl , ac) Obtained Using NLT

(5,3,12,8)	(6,5,26,16)	(7,6,56,16)	(8,7,116,24)
(5,4,12,16)			(8,5,112,16)
(9,8,238,40)	(10,9,484,56)	(11,10,982,88)	(12,11,1986,128)

Table 3.6: Best Values (n , d , nl , ac) Obtained Using ACT

Brisbane). The published NLT work [23] clearly contained counterexamples for n equal to eight. These have been verified. [2]

With ACT autocorrelation has been deliberately targeted but with NLT this was not the case. Here, previously unwitnessed autocorrelation values (indeed counter-examples to conjectures) have been generated by both techniques. The area is clearly very subtle. Interestingly, the technique has generated counter-examples for quite a small value of n. Having broken these conjectures pretty much by accident, it seems appropriate to try to break some conjectures deliberately. This is attempted in Section 3.5. For the time being it may be noted that the techniques, in a small way, have already provided something new.

3.4.3 Nonlinearity, Autocorrelation and Algebraic Degree

Only single attribute results have been presented so far. It is instructive now to examine the *joint* values of nonlinearity and autocorrelation achieved (and note the algebraic degrees). Tables 3.5 and 3.6 record the best functions obtained by *any* run of the NLT and ACT approaches. The quadruples in the tables record the number of inputs n, the algebraic degree d, the nonlinearity nl and the autocorrelation ac. Thus (5, 3, 12, 8) means 5 inputs, algebraic degree of 3, nonlinearity of 12 and autocorrelation of 8 etc. This notation has been chosen for compatibility with standard notation for describing correlation immune function properties (in Chapter 4 an additional component representing order of immunity with be added). An immediate observation is that both NLT and ACT appear capable of generating functions with very high algebraic degree. The highest algebraic

[2] The author is grateful to Dr Subhamoy Maitra for independently confirming the properties of these counter-examples and of several other functions reported in this thesis. On being informed (and provided with) counter-examples Maitra responded with a list of open problems in Boolean function design.

degree for a balanced function on n variables is n-1. Thus, functions of maximal algebraic degree have been generated. This may be regarded as a bonus since degree was ignored as part of the search. However, attaining high algebraic degree is very much the general trend of the annealing approaches taken.

For n less than or equal to 8, there is no difference in the properties of the best functions achieved. As n increases it would appear that NLT has an edge with respect to nonlinearity and ACT an edge with respect to autocorrelation (but this seems marginal in both cases, and pretty much to be expected). There would appear to be some interesting *potential* tradeoffs being made, e.g. for $n = 5$ relaxing the autocorrelation requirement (from 8 to 16) would appear to raise the achieved algebraic degree (from 3 to 4). Similarly for $n = 8$, there would appear to be a potential tradeoff between nonlinearity and autocorrelation. It may simply be the case that our particular search techniques are incapable of finding functions with profiles of (8,6,112,16) etc.

Tables 3.5 and 3.6 record the extremes that were generated but do not indicate how easily the functions were generated (i.e. how often). Tables 3.7, 3.8 and 3.9 show how the value of the parameter X may radically affect the sorts of functions produced. (n, d, nl, ac) indicates for functions of n inputs an algebraic degree ***at least*** **d**, nonlinearity ***at least*** **nl** and autocorrelation ***at most*** **ac**. A '$-$' indicates no restriction. Recall from Section 3.4.1 that the number of runs in all cases was 100. Thus the first column of Table 3.7 indicates that 76 runs at $X = -10$ produced functions with nonlinearity of 12 (which is actually the highest achievable), ten had the (lowest possible) autocorrelation value of 8. The following three entries indicate that all 10 with autocorrelation of 8 actually had nonlinearity of 12 and degree of 3.

The effect of the X parameter value is enormous. For $n = 5$ there are clear differences between NLT and ACT. For the ACT results, the profile of production of $(5, 4, 12, 16)$ contrasts starkly with those involving autocorrelation of 8 above it. Perhaps the most interesting results here are those for 8 input variables given in Table 3.10. Here, the effect of the R parameter is seen to have significant effect. For $R = 2$ few functions of interest are derived. This simply emphasises how crucial experimentation is for these sorts of problems. For $R = 3.0$ and $R = 2.5$ it would appear that the ranges of X for which $(8, -, 116, 24)$ and $(8, -, 112, 16)$ functions are generated are disjoint. It is interesting to note that for eight inputs performance using the 'ideal' bound ($X = 16$) is actually pretty poor. Allowing X to vary considerably is clearly a good idea.

Some features emerge when one considers the average nonlinearity and autocorrelation values attained for each (X, R) pair. For eight and nine input variables these are shown in Tables 3.11 and 3.12. For $n = 8$ we can see that the lowest average autocorrelation and highest average nonlinearity do seem in conflict. This simply reflects the ability to obtain $(8, -, 116, 24)$ and $(8, -, 112, 16)$ but never

	X										
	-10	-8	-6	-4	-2	0	2	4	6	8	10
(5,-,12,-)	76	92	95	100	100	100	100	100	100	100	100
(5,-,-,8)	10	36	69	80	0	0	0	0	0	0	0
(5,-,12,8)	10	36	69	80	0	0	0	0	0	0	0
(5,3,12,8)	10	36	69	80	0	0	0	0	0	0	0
(5,4,12,8)	0	0	0	0	0	0	0	0	0	0	0
(5,4,12,16)	0	4	0	0	100	100	100	100	100	100	100
(5,-,12,-)	6	4	14	100	100	100	100	100	100	100	44
(5,-,-,8)	14	18	12	74	0	0	0	0	68	67	31
(5,-,12,8)	2	3	10	74	0	0	0	0	68	67	30
(5,3,12,8)	2	3	10	74	0	0	0	0	68	67	30
(5,4,12,8)	0	0	0	0	0	0	0	0	0	0	0
(5,4,12,16)	0	0	0	0	100	100	100	100	0	0	0

Table 3.7: R=3.0. Five Variables Results (n, ad, nl, ac) for NLT (upper) and ACT (lower)

$(8, -, 116, 16)$. It is not known whether $(8, -, 116, 16)$ functions exist. (None has ever been demonstrated.) For $n = 9$ the two desirable properties seem broadly in harmony. Indeed, for $n = 9$ and $R = 3.0$ the 236.72 and 51.44 (for $X = -4$) are the highest nonlinearity and second lowest autocorrelation averages attained. For n=9 most parameter choices give rise to nonlinearity averages better than the best result achieved by random, hill-climbing or genetic algorithms with a direct cost function (of which the best for nonlinearity is 236).

3.5 The Intentional Generation of Counter-examples

Zhang and Zheng [129] offered a sum-of-squares measure as a desirable characteristic. For a Boolean function f the new measure σ_f is simply the sum-of-squares of the propagation characteristics:

$$\sigma_f = \sum_{s=0}^{2^n-1} \hat{r}^2(s). \tag{3.11}$$

The sum-of-squares treats all characteristic values $\hat{r}(s)$ equally. In contrast, criteria such as the Strict Avalanche Criterion (SAC), or Propagation Criteria of various orders k ($PC(k)$) were deemed to have a ‘local’ flavour. For example, SAC requires $\hat{r}(s) = 0$ only for vectors s of Hamming weight 1 and places no constraints on the values of other vectors. The sum-of-squares was offered as one of

	X										
	-10	-8	-6	-4	-2	0	2	4	6	8	10
(6,-,26,-)	0	0	0	90	88	89	90	99	100	98	6
(6,-,-,16)	1	3	3	94	100	100	100	100	100	39	10
(6,-,26,16)	0	0	0	84	88	89	90	99	100	37	1
(6,5,26,16)	0	0	0	9	10	39	41	59	51	11	0
(6,4,26,16)	0	0	0	80	84	83	84	95	97	33	1
(6,-,26,-)	0	0	0	0	0	76	91	99	100	30	0
(6,-,-,16)	72	80	82	100	100	100	100	100	100	100	83
(6,-,26,16)	0	0	0	0	0	76	91	99	100	30	0
(6,5,26,16)	0	0	0	0	0	44	46	49	52	12	0
(6,4,26,16)	0	0	0	0	0	67	88	96	93	25	0

Table 3.8: R=3.0. Six Variables Results (n, ad, nl, ac) for NLT (upper) and ACT (lower)

	X												
	-6	-4	-2	0	2	4	6	8	10	12	14	16	18
(7,-,56,-)	35	60	50	51	57	55	47	53	9	0	1	0	0
(7,-,-,16)	2	20	26	27	26	24	31	29	0	0	0	0	0
(7,-,56,16)	2	4	6	6	7	4	3	3	0	0	0	0	0
(7,6,56,16)	0	2	1	2	0	0	2	2	0	0	0	0	0
(7,5,56,16)	2	4	6	6	7	4	3	3	0	0	0	0	0
(7,-,56,-)	11	8	1	6	6	9	10	8	4	0	0	0	0
(7,-,-,16)	13	87	82	87	82	82	78	76	5	0	0	0	0
(7,-,56,16)	0	1	0	1	2	0	2	3	0	0	0	0	0
(7,6,56,16)	0	1	0	1	1	0	2	2	0	0	0	0	0

Table 3.9: R=3.0. Seven Variables Results (n, ad, nl, ac) for NLT (upper) and ACT (lower)

	X													
	-10	-8	-6	-4	-2	0	2	4	6	8	10	12	14	16
(8,-,116,-)	25	22	8	3	1	1	3	1	2	1	57	59	28	11
(8,-,-,16)	0	11	13	21	16	13	11	15	18	22	0	0	0	0
(8,-,116,24)	0	0	0	0	0	0	0	0	0	0	8	1	0	0
(8,-,112,16)	0	11	13	21	16	13	11	15	18	22	0	0	0	0
(8,5,112,16)	0	11	13	21	16	13	11	15	18	22	0	0	0	0
(8,7,116,24)	0	0	0	0	0	0	0	0	0	0	8	0	0	0
(8,-,116,-)	0	0	0	22	2	1	4	3	0	0	52	34	28	13
(8,-,-,16)	0	0	0	8	10	15	13	11	10	7	0	0	0	0
(8,-,116,24)	0	0	0	0	0	0	0	0	0	0	10	1	1	0
(8,-,112,16)	0	0	0	8	10	15	13	11	10	7	0	0	0	0
(8,5,112,16)	0	0	0	8	10	15	13	11	10	7	0	0	0	0
(8,7,116,24)	0	0	0	0	0	0	0	0	0	0	8	1	0	0
(8,-,116,-)	0	0	0	0	0	0	19	11	18	15	11	7	17	10
(8,-,-,16)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(8,-,116,24)	0	0	0	0	0	0	1	0	0	0	1	0	1	0
(8,-,112,16)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(8,5,112,16)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(8,7,116,24)	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.10: Eight Variables Results (n, ad, nl, ac) for NLT R=3.0 (upper), R=2.5 (middle), and R=2.0 (lower)

X	nl	ac	nl	ac	nl	ac
-16	106.22	84.72	108.32	73.68	111.56	59.28
-14	106.24	86.64	108.54	71.68	111.88	56.72
-12	106.08	84.88	109.0	67.12	112.68	49.6
-10	105.86	85.12	109.9	65.92	114.46	41.04
-8	106.08	86.0	111.26	59.84	113.26	28.48
-6	105.78	85.92	112.08	51.84	112.48	25.84
-4	106.28	84.4	113.44	32.08	112.16	24.48
-2	106.02	88.16	112.28	27.12	112.18	24.72
0	110.12	61.36	112.42	27.2	112.16	25.52
2	113.1	36.32	112.28	26.24	112.22	26.08
4	113.0	34.8	112.26	27.92	112.12	24.56
6	113.28	36.48	112.28	27.68	112.18	23.68
8	113.22	36.08	112.12	27.2	112.38	23.84
10	113.08	36.56	114.74	33.12	115.0	33.6
12	112.72	36.0	113.98	35.2	114.9	34.96
14	113.24	35.2	113.78	36.16	114.02	36.96
16	112.94	36.16	113.28	37.92	113.38	38.16
	R=2.0		R=2.5		R=3.0	

Table 3.11: Eight Variables Average Nonlinearity and Autocorrelation Results

X	nl	ac	nl	ac	nl	ac
-8	233.56	73.2	236.5	52.8	236.6	50.56
-6	236.08	55.76	236.68	52.56	236.62	52.16
-4	236.46	52.48	236.44	51.68	236.72	51.44
-2	236.3	51.28	236.3	51.92	236.64	51.84
0	236.12	51.76	236.32	52.0	236.36	52.24
2	236.06	51.68	236.34	51.12	236.46	51.92
4	236.1	51.92	236.4	50.96	236.54	51.52
6	236.22	52.64	236.4	50.88	236.7	51.6
8	236.14	52.08	236.42	51.44	236.56	52.88
10	236.04	53.52	236.38	51.84	236.72	52.0
12	236.06	52.96	236.46	52.96	236.66	52.96
14	235.9	52.72	236.16	52.4	236.46	52.88
16	235.8	54.08	235.92	53.68	236.2	53.44
18	235.66	56.8	235.9	56.8	235.86	55.6
20	235.36	57.2	235.52	56.72	235.54	57.68
	R=2.5		R=2.75		R=3.0	

Table 3.12: Nine Variables Average NL and AC Results

two 'global' avalanche characteristics (the other being what has been referred to so far as autocorrelation). Constructions were proposed for balanced functions on even and odd numbers of input variables and the sum-of-squares values provided. For $n = 2k$ the sum of squares indicator for the indicated construction is given by

$$\sigma_f = 2^{4k} + 2^{3k+3} - 2^{3k+1}. \tag{3.12}$$

The authors note that the lower bound of 2^{4k} is met only when f is a bent function (i.e. $\hat{r}^2(0) = 2^{4k}$) , and conjecture

> "that the function f defined by (5) with $\sigma_f = 2^{4k} + 2^{3k+3} - 2^{3k+1}$ achieves nearly optimal sum-of-squares avalanche characteristic of balanced functions on V_{2k}."

Similarly for $n = 2k + 1$ the sum of squares indicator for the indicated construction is given by

$$\sigma_f = 2^{4k+3}. \tag{3.13}$$

The authors state:

> "the sum-of-squares avalanche characteristic of the function is extremely good. Again we conjecture that it achieves the lowest possible value for balanced functions on V_{2k+1}."

The statement 'nearly optimal' for the $n = 2k$ case is a little unclear. The statement for $n = 2k + 1$ is unequivocal.

3.5.1 Experiments with Sum-of-Squares as the Cost Function

It is possible to generate functions for even and odd n with lower σ_f values than those conjectured. Here σ_f itself was used as the cost function for f. The search was restricted to move over the space of balanced functions with the same move strategy as before. A cooling parameter $\alpha = 0.95$ was used together with $MIL = 200$, $MaxIL = 400$ and $MUL = 50$.

For 5–10 input variables 100 runs of the annealing algorithm were carried out followed by hill-climbing (with the same cost function). The results are given in Table 3.13 and show the GAC conjectured bounds together with the minimum, average and maximum values achieved over all runs. As can be seen, many runs of the algorithm generated counter-examples to the conjectures. For

$n = \mathit{10}$ no counter-example was generated. In some cases the conjectured values are markedly sub-optimal. Average time per run is also shown, indicating the speed with which conjectured bounds were broken (e.g. for $n = 7$ all 100 runs produced a counter-example taking on average 1.25 seconds for each run).

This is clearly a very simple task to carry out. Yet optimisation is not yet established in professional cryptography. Optimisation has the potential to provide confidence in or counter-examples to conjectures like the above. It can do so very efficiently. Furthermore, this is not just an exercise in counter-example generation. If low sum-of-squares really is desirable then heuristic optimisation is obviously a good tool to derive better functions.

Global avalanche characteristics are beginning to receive more attention from researchers. Son et al. have published lower bounds on σ_f for balanced functions [116]. They show that $\sigma_f \geq 2^{2n} + 2^{n+3}$ (and also give upper bounds on nonlinearity of balanced functions in terms of σ_f). The bounds on σ_f are also shown in Table 3.13. It can be seen that there is still considerable distance between the obtained values and the provided bounds but no functions or methods of construction were actually exhibited by Son et al. and the results presented here are the best demonstrated. Sung et al. have improved the lower bound for functions satisfying a propagation criterion for a number of vectors [122]. In a forthcoming paper Maitra will address the bounds on global avalanche criteria for correlation immune functions [71].

Metaheuristic searches are well-known for handling vast search spaces where other techniques break down. Here they have generated counter-examples at small values of n. The practical importance of the results shown here is that counter-examples to conjectures were demonstrated with considerable ease. Only for functions of ten input variables did all runs fail to produce a function achieving or bettering the GAC-conjectured bounds.

3.5.2 Revisiting the Past

The work reported in earlier sections generated very many functions, each of which has some GAC sum-of-squares value. Curiosity suggests that these functions should be revisited to determine whether there are any surprises. The functions generated during the NLT and ACT experiments of Section 3.4 were revisited and their sums-of-squares measured. For 5–10 inputs, functions had been generated with sums-of-squares as low as the minima generated by the direct experiments in this section. Additionally, for $n = 9$, a function with a sum-of-squares value of 376832 had been generated. For $n = 10$, a function with a sum-of-squares value of 1534720 had also been produced. Each is lower than the corresponding result obtained by the direct use of sum-of-squares as a cost function. Since the functions generated earlier had very low autocorrelation for lower

n	Son et al.	GAC-σ_f	Annealing + Hill-climbing			Average
	Bound	Bound	Minimum	Average	Maximum	Time (secs)
5	1280	2048	1664	1664	1664	0.4
6	4608	7168	6784	6784	6784	1.2
7	17408	32768	23936	24550.4	24704	1.25
8	67584	90112	86656	89931.5	101248	2.9
9	266240	524288	379904	389273.6	404864	13.5
10	1056768	1245184	1535488	1550272	1566592	137

Table 3.13: Sum of Squares Bounds and Results

n, a moderately low sum-of-squares might be expected at the very least. Given a suitable histogram of spectral values, an excellent value might be attained. For example, some functions with 7 inputs and autocorrelation of 16 satisfied $|\hat{r}(s)| = 0$ for up to 66 non-zero values of s. This alone is sufficient to break the conjectured value (32768).

3.6 What are these Results Telling Us?

The work reported so far actually belies its origins. The initial work was largely targeted at nonlinearity; low autocorrelation was a secondary concern. The ACT technique was adopted only after it was noticed that the NLT approach generated functions with low autocorrelation. However, the breaking of conjectured autocorrelation bounds and the ease with which the sum-of-squares bounds were broken suggests that a more autocorrelation-focussed effort might well pay dividends. The sum-of-squares cost function is the first to use the autocorrelation spectral values $\hat{r}(s)$, implicitly targeting the 'ideal' value $\hat{r}(s) = 0$ (for non-zero s). As before, only bent functions (on even numbers of variables) achieve this and the focus of this chapter is balanced functions (of both even and odd numbers of inputs). By analogy with the cost function of Equation 3.7, a cost function family of the following form suggests itself

$$cost(\hat{f}) = \sum_{s} ||\hat{r}(s)| - X|^R. \tag{3.14}$$

This seems a very natural cost function. Indeed with $X = 0$ and $R = 2$ it reduces to the sum-of-squares cost function used above. As R increases large values of $\hat{r}(s)$ will clearly be discouraged. Experiments were carried out using the parameter values given in Table 3.14. Fifty runs were carried out for each parameter setting (except for $n = 12$ where only ten runs were attempted). Table 3.15

n	X Range (min:max:step)	R Values	α	MIL	Max IL	No. Runs
5	(-4: 4: 1)	3.0	90	400	400	50
6	(-4: 4: 1)	3.0	90	400	400	50
7	(-4: 4: 1)	3.0	90	400	400	50
8	(-4: 4: 1)	3.0	95	400	400	50
9	(-4: 4: 1)	3.0	95	400	400	50
10	(-8: 8: 1)	3.0	95	400	400	50
11	(-8: 20: 4)	3.0	95	400	400	50
12	(-8: 20: 4)	2.5	95	800	800	10

Table 3.14: Search Parameters Used

Best Functions	Number of Runs	Total Number of Runs Giving Best Value
(5,3,12,8)	450	150
(5,4,12,16)		300
(6,5,26,16)	450	450
(7,6,56,16)	450	2
(8,7,116,24)	450	1
(9,8,236,32)	450	4
(10,9,484,56)	850	18
(11,10,984,80)	400	1
(12,11,1988,120)	80	2

Table 3.15: AC-Cube Results: (n, d, nl, ac)

shows the best results obtained by this method. The small amount of experimentation has already led to improved results. In particular, for the first time an autocorrelation of 32 has appeared for $n = 9$. For $n = 11$ the (11,10,984,80) is the best profile achieved to date (see Tables 3.5 and 3.6). Similarly, (12,11,1988,120) has the best autocorrelation achieved to date for $n = 12$.

3.7 Where are these Results Leading Us?

The preceding sections have proposed plausibly well-motivated cost functions and the results have shown that they are capable of providing highly nonlinear balanced Boolean functions with low autocorrelation and high algebraic degree (with different emphases depending on the cost function used).

That the approach generates functions with high algebraic degree is perhaps not so surprising. Functions of low algebraic degree are actually extremely rare. The search is guided by the cost surface (problem structure). Unless the properties sought actually force the search to move towards low algebraic degree there is little chance that it would.

It is also fairly clear that the cost functions used do not characterise highly desirable functions (judged by our criteria), or even characterise what it means to be 'close' to such functions (or even, for that matter, close to some particular 'family' of such functions — there may well be other functions with excellent properties that are never reached by the technique, even for the smaller n). If they did so, better results should have been obtained for higher numbers n of input variables. (Recall that much computing power was expended to gain optimal values for $n = 9$ and $n = 10$.) So what should be drawn from the work so far? To make further progress it is useful to take a step back and ask 'What has actually happened here?' This is now stated in stark terms.

A parametric family of cost functions has been demonstrated. The importance of parametric flexibility has been stressed and shown. Although the cost function is often cited as playing a crucial role, the reader will recall from Chapter 2 that the great majority of cost functions are very direct. Although, experimentation with metaheuristic search parameters is normal, experimentation with the cost functions seems far less so. For smaller n it is possible to find parameters so that minima of the new cost functions sometimes (or often) are good places from which to hill-climb to desirable functions. In addition, the cost surfaces (landscapes) are sufficiently navigable to allow these extrema to be reached via guided search. The landscapes of the new cost functions used here and the landscapes of the direct cost functions based on the target criteria (e.g. Equation 3.2) may be very different. We only need some local minima of the new cost functions to enable us to reach desirable Boolean functions. This indirectness seems highly unusual.

The use of polynomials as a means of approximation is well-established (ranging from simple Spline curves to more sophisticated Tchebyschev polynomials). Linear and higher-order polynomials have been used for cryptanalytic approximations too. However, the use of polynomials of Walsh values seems very unusual indeed. The author knows of no similar application.

So what prevents the approach getting better results? Consider now the current family and its possible limitations. The principal cost functions for nonlinearity were of form

$$cost(\hat{f}) = \sum_{\omega} \left| |\hat{F}(\omega)| - X \right|^{R}. \qquad (3.15)$$

Assume, for explanatory purposes, that the value of R is now 3. For each ω the cost function contribution is given by

$$|G^3 - 3XG^2 + 3X^2G - X^3|, \qquad (3.16)$$

where $G = |\hat{F}(\omega)|$. But this is restrictive. A more general cost function is given by:

$$cost(\hat{f}) = \sum_{\omega} |p\,(G)\,|, \qquad (3.17)$$

where

$$p(G) = \sum_{i=0}^{m} b_i G^i. \qquad (3.18)$$

That is, adopt the absolute value of some polynomial in $|\hat{F}(\omega)|$. Thus we allow arbitrary order and arbitrary coefficients. The degree of the polynomial is fixed at the beginning of the run by the user. This model is more flexible than the cost functions chosen so far. This flexibility comes at a price. There is no obvious relationship between the coefficients of a quintic polynomial in $\hat{F}(\omega)$ whose minima are reached by functions $\hat{f}$ with desirable properties! What is important is that there should exist some appropriate values of the coefficients for which this is the case and that we should be able to find them. A means of achieving this is given below.

3.7.1 Hill-climbing on Cost Function Parameters

The approach uses higher level optimisation on the polynomial coefficients. For any particular set of coefficients, ten runs of annealing were carried out minimising the the cost function defined by those coefficients. This was followed by a second-stage hill-climb with respect to nonlinearity. The average nonlinearity of the functions resulting from those runs was taken as a fitness measure for the set of coefficients. With this fitness measure a hill-climb was carried out on the set of coefficients. In the results shown quartic polynomials (i.e. of degree four) were used.

A random set of coefficients was used to initialise the cost function. Each coefficient from b_0 to b_4 was increased or decreased (by some specified amount) in turn ($b_4 = 1$ always). Only moves that improved the average value obtained were accepted (thus a form of hill-climbing has been used). Evaluating 10 runs of annealing is very costly in computational terms for a single fitness evaluation of the coefficients. Accordingly, a rapid cooling schedule was used ($\alpha = 0.9$).

n	Runs	Max NL	Runs Max NL	Best NL Average	Runs
8	50	116	50	116.0	24
9	50	238	50	237.6	1
10	30	486	12	484.2	4

Table 3.16: Results for High Level Optimisation Runs

n	Runs	Min AC	Runs Min AC	Best AC Average	Runs
8	50	16	50	20.8	7
9	50	32	1	40.0	11
10	50	56	50	63.2	4

Table 3.17: Results for High Level Optimisation Runs

A feature of this approach is that the fitness of the coefficients is actually stochastic (since the annealing algorithm itself is stochastic). This was catered for by aborting the search only after three consecutive cycles through all the coefficients failed to give an improvement on the current best average obtained. In addition, after a full failing cycle the STEP distance by which coefficients were altered was halved.

Table 3.16 gives the results for $n = 8, 9$ when the target is high nonlinearity. The results show marked improvements on the results achieved so far in terms of efficiency. Thus, for $n = 8$ our higher level optimisation has produced final values for cost function coefficients that achieved a nonlinearity of 116 in all ten runs (24 of the 50 runs of the higher level optimisation produced coefficients with this property). This contrasts with the results presented in Table 3.11 where the highest achieved average was 115 (for $X = 10$ and $R = 3.0$). Similarly, for $n = 9$ the results in Table 3.12 the best average nonlinearity wsa 36.72 (for $R = 3.0$ and $X = -4, 10$). The highest average for $n = 10$ (general table omitted) was 483.84 (for $R = 3.0$ and $X = -6, 2$). Thus, for all values considered higher level optimisation leads to more efficient cost functions. However, no improvements on the best values achieved were recorded.

3.7.2 Commentary

The idea of higher level optimisation does not appear to have been applied to any modern-day cryptological problem and can obviously be made more sophisticated. This is not at all new to the optimisation world. It clearly has a promising place. Earlier work has demonstrated the value of parametric cost functions.

These come at a price — it is far from clear what are the best parameter values to use and a search over them will typically be required. The more cost functions are used as an *indirect* means of characterising desired points the more necessary will become search over the parameter space.

3.8 Increasing the Output: Generalising to S-boxes

Previous optimisation approaches to evolving Boolean functions with desirable cryptographic properties have been generalised to the multiple output case. Millan has compared random generation and hill-climbing as means of evolving highly nonlinear bijective S-boxes [79]. Burnett et al. have investigated the use of genetic algorithms and hill-climbing to evolve regular S-boxes [80]. Both high nonlinearity and low autocorrelation were targets. The fitness and cost measures for an S-box were the nonlinearity and autocorrelation values of that S-box. These measures are clearly 'direct'. Since spectrum based approaches generated interesting results for single-output case an obvious question to pose is 'Can the spectrum-based approaches be generalised to allow S-boxes to be evolved with desirable properties?' This is investigated below.

3.8.1 Spectrum-based Cost Functions for S-boxes

The work so far is easily generalised. If $f(x) : \mathbf{Z}_2^n \rightarrow \mathbf{Z}_2^m$ is an n input m output S-box then each $\beta \in \mathbf{Z}_2^m$ defines a function that is a linear combination $f_\beta(x)$ of the m outputs of f. This is given by

$$f_\beta(x) = \beta_1 f_1(x) \oplus \cdots \oplus \beta_m f_m(x). \tag{3.19}$$

For each such function f_β the Walsh-Hadamard values $\hat{F}_\beta(\omega)$ and autocorrelation values $\hat{r}_\beta(s)$ are defined in the usual way. Two cost functions can now be defined for use in S-box evolution. A cost function based on Walsh-Hadamard spectra is given by

$$cost(f) = \sum_{\beta \in \mathbf{Z}_2^m} \sum_{\omega \in \mathbf{Z}_2^n} ||\hat{F}_\beta(\omega)| - X|^R \tag{3.20}$$

and a similar cost function based on autocorrelation spectra is given by

$$cost(f) = \sum_{\beta \in \mathbf{Z}_2^m} \sum_{s \in \mathbf{Z}_2^n} ||\hat{r}_\beta(s)| - X|^R. \tag{3.21}$$

The single output cost functions have been applied to each function defined as a linear combination of the outputs and the results summed over all such combinations.

	Millan [79]		Annealing	
n	Rnd	HC	SA	AC SA
5	8	10	10	16
6	20	20	22	32
7	44	46	48	48
8	98	100	102	80

Table 3.18: Summary Results for Bijective n

3.8.2 Experiments and Results

Table 3.18 records the best nonlinearity values achieved in Millan's experiments comparing the ability of random search and hill-climbing to evolve 5 by 5, 6 by 6, 7 by 7 and 8 by 8 bijective S-boxes. The cost functions defined by Equations 3.20 and 3.21 have been used to evolve S-boxes of similar dimensions. At the end of each run hill-climbing was carried out with respect to nonlinearity and autocorrelation respectively. The approaches thus mirror those of Section 3.4. 50 runs were carried out for each value of X in the set $-4,-3,-2,-1,0,1,2,3,4$. $R=3.0$ was used throughout. Table 3.18 records the best *joint* values of nonlinearity and autocorrelation achieved by either technique (i.e. functions were generated which possessed both the indicated nonlinearity value and the indicated autocorrelation value).

The results for the bijective S-boxes are not optimal. 6 by 6 boxes with nonlinearity of 24 have been provided by construction but they seem quite rare (Millan [79] attempted one million random generation and hill-climbing attempts and found only a nonlinearity of 20). Deriving bijective S-boxes is not an easy task for annealing. As m increases the number of derived linear combinations to check doubles. An 8 by 8 bijective S-box with the parameter values shown takes about 20 minutes on 1.4 GHz Pentium PC. However, again this is not easy. Only one (104,80) function was generated from 200 runs. Similarly, for n=7 only one (48,48) function was generated. Does this matter? We shall address this issue below.

Burnett et al. applied genetic algorithms followed by hill-climbing to evolve 8 by m regular S-boxes (for $m=2,\ldots 8$). Table 3.19 records the best nonlinearity and autocorrelation values achieved (individually). The new cost functions were again used to evolve regular S-boxes of similar dimensions (with $R=3.0$ and the same range of X as before). Table 3.18 records the best *joint* values of nonlinearity and autocorrelation achieved by each technique. Burnett et al. presented their results as their 'current conjectures for the achievable bounds'. The results of applying the annealing-based approaches with the new cost functions is fairly dramatic (the hill-climbing second stage with respect to nonlinearity or autocor-

		Burnett et al. [80]				Spectrum Based	
		Nonlinearity		Autocorrelation		Joint (d,nl,ac)	
n	m	Rnd	GAs	Rnd	GAs	SNLT	SACT
8	2	108	110	56	48	(7,114,32)	(7,114,32)
8	3	106	108	64	56	(7,112,40)	(7,112,40)
8	4	104	106	72	64	(7,110,56)	(7,110,48)
8	5	102	104	72	72	(7,108,64)	(7,108,56)
8	6	100	104	80	80	(7,106,64)	(7,106,64)
8	7	98	102	80	80	(7,104,80)	(7,104,72)

Table 3.19: Nonlinearity and Autocorrelation Values Achieved for 8 by m S-boxes

relation rarely improves matters). As m increases the same general patterns of declining nonlinearity and increasing autocorelation are witnessed as by Burnett et al. However, the new cost functions and annealing-based searches have found functions that simultaneously improve nonlinearity and autocorrelation. Most typically, for the best functions, nonlinearity is 4 higher and autocorrelation is 16 lower.

Comparison with theoretical approaches is difficult. On specific criteria it is clear that the derived S-boxes are not optimal. Nyberg, for example, has demonstrated 8 by 8 S-boxes with nonlinearity 112. For present purposes we note that spectrum-based cost functions have promise and have provided improvements on previous optimisation-based work.

There appears to be a growing interest in injective S-boxes where m is greater than n. For example, Youssef and Tavares have proposed constructions for 8 by 32 S-boxes. This would seem an interesting avenue to pursue.

3.9 Implementation Considerations

This section documents briefly salient implementation features.

3.9.1 Implementing $\hat{L}_\omega(x)$

The values of $\hat{L}_\omega(x)$ could be stored in a 2-D array (indexed by ω and x). However, this is a highly wasteful approach. A more efficient approach, implemented in the developed toolsets, uses a pre-computed array. Let an index y in the range $0 \ldots (2^n - 1)$ have the natural n-bit representation $y_1 \ldots y_n$. Calculate the corresponding array element $L[y]$ using

$$L[y] = (-1)^{y_1 \oplus \cdots \oplus y_n}, \mathit{for}\, y = 0..(2^N - 1). \tag{3.22}$$

Let $\omega = \omega_1 \ldots \omega_n$ and $x = x_1 \ldots x_n$. Let $y = y_1 \ldots y_n = (\omega_1 x_1 \ldots \omega_n x_n)$. Then $L[y] = \hat{L}_\omega(x)$. The bit-wise logical and of x and ω is calculated on the fly (a single machine instruction) and used to index the appropriate element L[y] of the pre-computed array whenever $\hat{L}_\omega(x)$ is required.

3.9.2 Implementing the Cost Functions

The major cost functions used in the research were typically of the form

$$\sum_{\omega=0}^{2^N-1} g(|\hat{F}(\omega)|). \tag{3.23}$$

Computing this from scratch every time would be highly computationally intensive, more so since g involves exponentiation. However, $|\hat{F}(\omega)|$ is a non-negative integer (for balanced functions it is in the set 0,4,8,..., for unbalanced functions it is in the set 0,2,4,..) and so can act as an index into an array containing the corresponding pre-computed value of $g(|\hat{F}(\omega)|)$. The evaluation of the cost functions above is implemented as the sum of pre-computed values. The cost functions using the autocorrelation spectra $\hat{r}(s)$ were implemented similarly.

3.10 Evidence for the Thesis

The work in this chapter is intended to set the ball rolling. Nevertheless, it should contribute in some way to the overall thesis proposition.

3.10.1 A Significant Increase in Power?

The following lend credence to the claim that, within the domain of application, the power of the techniques is significantly greater than evidenced in publicly available literature:

- The techniques have generated counter-examples to conjectures by theoreticians. As far as the author is aware, counter-examples to cryptological conjectures by theoreticians have not previously been demonstrated using optimisation techniques. Counter-examples were occasionally generated in a few seconds. Thus, metaheuristic search can provide a very efficient means of gaining confidence in conjectures or else disproving them.

- The nonlinearity and autocorrelation values attained using the methods described in this chapter match or improve on those documented existing optimisation-based literature. By adopting a somewhat indirect approach, it has proved possible to obtain high nonlinearity and low autocorrelation via a single cost function family (and with high algebraic degree). The approach is also very efficient in terms of execution time, due to the ability to carry out a significant amount of precomputation.

- The ability to generate functions with high algebraic degree pretty much as a consequence of the way search works is a feature that could prove of considerable use. It would be interesting to determine whether more sophisticated criteria for complexity are also attained.

- The nonlinearity results of Millan for bijective S-boxes [79] have been improved with a natural extension of the NCT and ACT Boolean function approaches. The nonlinearity and autocorrelation results by Burnett et al. for 8 by m S-boxes have been significantly improved (indeed S-boxes were demonstrated with properties that simultaneously exceeded the best previous values for each). In the latter case the results were stated as the authors' current conjectures on achievable properties. These have been exceeded.

3.10.2 Toolkit Contributions

This section starts the conceptual toolkit promised in Chapter 1. The contributions in this respect are:

- Establishing the potential for indirect approaches and exploiting problem structure in unusual ways. Recall that the initial motivation was geared to the derivation of highly nonlinear functions. Measurements of other properties were a secondary concern. Breaking autocorrelation conjectures with the NLT approach could accurately be described as ‘accidental’. The decision to record characteristics lead to a much more thorough examination of multiple properties.

- Establishing optimisation as a test device for proposed conjectures.
- Indicating the potential for unusual cost function families that essentially act as approximations to the actual cost surfaces of interest.
- Higher-level optimisation has been shown to have potential for use in cryptological problems.

3.11 Open Problems

Here is a list of hopefully interesting questions, prompted by the research presented so far, to which I do not know the answer:

1. Histogram approaches. Can cost functions based on spectral distribution histograms improve the results presented here? Thus, one could start with a desired spectral histogram (e.g. 4 $\hat{F}(\omega)$ with value zero, 24 with value 4 etc.) A cost function could be created that punished deviation from the desired histogram. Ideas along these lines have emerged recently at the SRC (Millan) as well as to the author (work on histogram-based cost functions for a different problem appears in the next chapter). Histogram approaches will have to cope with problems of discontinuity. However, smoothing methods of some form could be deployed.

2. Optimisation sophistication. How far can the results be improved by adopting more sophisticated optimisation techniques? The results have been obtained with what might accurately be described as ‘vanilla’ simulated annealing. What might happen if the metaheuristic search community brought its expertise (over thirty years in the making) to bear on these problems?

3. Can theory and optimisation be used more harmoniously? For example, the following theorem has recently been proved:

   > Let f be a (8,0,-,118) function (if such exists). Then the degree of f must be 7 and it is possible to write $f = (1 \oplus X_8)f_1 \oplus X_8 f_2$ where f_1 and f_2 are 7-variable functions, each having nonlinearity 55 and degree 7.

   Can optimisation be used to co-evolve appropriate components to achieve an (8,0,-,118) function? The demonstration of such a function would be a significant result.

4. Can optimisation be used to plant trapdoors in Boolean functions and S-boxes?

3.11.1 Summary

This is the first technical research chapter of the thesis and aims to get the ball rolling. There are clearly limitations to what the method proposed here can achieve and further work is needed. However, the techniques have been shown to be capable of demonstrating results of interest. They do so very simply. As it happens, the experimentation performed so far has more surprises in store.

Chapter 4

Correlation Immunity

For small numbers of input variables, annealing-based approaches can be used to evolve Siegenthaler optimal functions of all orders with the highest possible nonlinearity. The autocorrelation of such evolved functions is often extremely low. This is achieved with a minor modification to the approaches of the previous chapter. It is also shown that the research of the last chapter has achieved more than has so far been appreciated. Correlation immune functions are also evolved via a highly unusual approach based on inversion of the Walsh-Hadamard spectrum. The approach starts with a spectrum with appropriate properties and attempts to evolve a permutation which gives rise to a Boolean function under inversion. It is Boolean structure that is evolved. The work is generalised to show how the approaches taken extend naturally to the evolution of bent functions and the evolution of functions satisfying particular propagation criteria.

4.1 Introduction

Siegenthaler was the first to demonstrate how correlation between values of small numbers of inputs to a combining function and the value of its output could form the basis of an effective cryptanalytic attack on a standard stream cipher model — the divide and conquer attack described in Chapter 2 [113]. An attempt to characterise resilience to such attacks lead to the notion of the order of correlation immunity of Boolean functions [114]. A function f is correlation immune of order m ($CI(m)$ for short) if all non-empty subsets of inputs of size m are statistically independent of the output of that function. It is simpler, however, to work with Zhen and Massey's characterisation in terms of the Walsh-Hadamard values [47]. A function f is correlation immune of order m if and only if

$$|\hat{F}(\omega)| = 0; 1 \leq |\omega| \leq m. \tag{4.1}$$

Balance, high nonlinearity and high algebraic degree are typical requirements for functions used in stream cipher designs and were the targets of the work of the previous chapter. An obvious question to ask is ‘Can the techniques developed so far be extended to encompass correlation immunity requirements?’ Addressing this question forms the basis of the research in this chapter. There has been a considerable amount of theoretical work in the derivation of balanced, highly nonlinear correlation immune functions with high algebraic degree. This provides an obvious opportunity to determine how competitive optimisation techniques can be — there is plenty of competition. That the topic should continue to be an active area of research is testament to the fact it is not a ‘solved’ problem, or even an easy one.

4.2 Constructing Correlation Immune Functions

Production of correlation immune functions is most typically carried out using theoretical construction. Designers often construct functions for some number of input variables from smaller ones (i.e. on fewer input variables) with particular properties. A simple example was given by Siegenthaler in his classic paper [113]. From the following two $CI(2)$ functions on four variables:

$$f_1(x_1, \ldots, x_4) = x_1 \oplus x_2 \oplus x_3; \tag{4.2}$$

$$f_2(x_1, \ldots, x_4) = x_1 \oplus x_2 \oplus x_4, \tag{4.3}$$

the following function on five input variables is constructed

$$f(x_1, \ldots, x_5) = x_5 f_1(x_1, \ldots, x_4) \oplus (1 \oplus x_5) f_2(x_1, \ldots, x_4). \tag{4.4}$$

This function is also $CI(2)$. The construction proceeds in a similar vein to give a $CI(2)$ function on seven variables. The top half of the truth table of the function of Equation 4.4 is that of f_2 and the bottom half is that of f_1 — the function is a *concatenation* of the two smaller functions. Some researchers have given general recursive methods of constructing functions with desirable cryptographic properties from smaller ones. Such techniques provide an infinite series of functions. This is the power of mathematics and it is a power optimisation techniques have little prospect, at present, of challenging. These recursive techniques require, however, small functions to get started and so there is value in generating instances even for small numbers of input variables. Additionally, even functions on small numbers of inputs may find direct application. It is with small numbers of input variables that the research reported below is concerned.

(5,1,3,12)	(5,2,2,8)	(5,3,1,0)		
(6,1,4,24)	(6,2,3,24)	(6,3,2,16)	(6,4,1,0)	
(7,1,5,56)	(7,2,4,56)	(7,3,3,48)	(7,4,2,32)	(7,5,1,0)
(8,1,6,116)	(8,2,5,112)	(8,3,4,112)	(8,4,3,96)	(8,5,2,64)
(9,1,7,244)*	(9,2,6,240)*	(9,3,5,240)*	(9,4,4,224)	(9,5,3,192)
(10,1,8,492)*	(10,2,7,480)	(10,3,6,480)	(10,4,5,480)*	(10,5,4,448)

Table 4.1: Upper Bounds on Achievable Properties (n,m,d,nl): '*' indicates not yet demonstrated

4.3 Tradeoffs Between Criteria and Setting Targets

Relationships between the various desirable criteria are the subject of much current research (e.g. [13, 70, 72, 94, 95, 107, 108, 109, 123, 124]) but the most fundamental result is that given by Siegenthaler [113] — a balanced Boolean function of n variables with correlation immunity order m and algebraic degree d must satisfy:

$$m + d \leq n - 1. \tag{4.5}$$

This immediately bounds what can be expected. A function satisfying $m + d = n - 1$ is said to be *Siegenthaler optimal.* Having chosen m one could aim to maximise nonlinearity subject to maximal algebraic degree. A good deal of very recent research has been carried out to determine bounds on what nonlinearity values can be achieved, e.g. [94, 107, 108, 109, 123, 124]. Table 4.1 provides the best theoretical bounds known for optimal tradeoffs for balanced functions and is formed using information in [109]. Each entry is of the form (n, m, d, nl) where n is the number of input variables, m is the order of correlation immunity, d is the algebraic degree and nl is the nonlinearity. A '*' indicates that the indicated bound has not yet been demonstrated by any method.

It is stressed that 'small numbers of input variables' (referred to at the end of Section 4.2) does not mean 'easy', nor does it mean unimportant. Commenting in 1998, Filiol and Fontaine [38] state "achieving the best tradeoffs is the main goal, which remains a difficult problem...exhibiting some of them is very difficult, as soon as $n > 7$" and, subsequently, "until now, only existence results were known." Their paper gave a method based on the notion of idempotents to obtain bent and balanced functions with high nonlinearity. Correlation immunity was achieved by a recursive construction. In fact their (9,2,5,224) functions are clearly not optimal and the area has been the subject of a good deal of work since. It is only very recently that theoreticians have been able to demonstrate functions attaining the theoretical bounds for all functions of seven and eight variables. For seven variables (7,2,4,56) was demonstrated in 2000 by Pasalic et al. [95]. For eight variables (8,3,4,112) was demonstrated by Sarkar et al. at Crypto 2000 [109] and

(8,1,6,116) was demonstrated by Maitra and Pasalic at SETA' 01 [72]. (9,4,4,224) and (10,3,6,480) were demonstrated at Crypto 2000 by Sarkar and Maitra [109]. Open questions remain for functions of nine and ten variables. Only the very leading edge is shown here. The reader interested in comparisons with other authors' work is referred to [108] for details.

All manner of theoretical constructions have been brought to bear to close the case for eight or fewer variables. Here, an opportunity presents itself. *If metaheuristic search could provide examples to meet each of the bounds this would lend substantial credence to the claim that the techniques are significantly more powerful than currently evidenced in the public literature.* This will be the primary goal of the work reported here. However, in keeping with the spirit of the previous chapter, attempts will also be made to evolve larger functions which are beyond the capabilities of the current techniques — the techniques are shown beginning to fail.

4.4 Motivation and Method - the First Pass

Millan et al. [84] were the first to use metaheuristic search (genetic algorithms) to derive correlation immune balanced functions with high nonlinearity. That work provided the initial motivation for the research that follows and also indicated how functions satisfying various propagation orders might be evolved. Deviation from desired correlation immunity and propagation orders were defined as below:

$$cidev_f(m) = \max(|\hat{F}(\omega)|; 1 \leq |\omega| \leq m); \tag{4.6}$$

$$pcdev_f(k) = \max(|\hat{r}(s)|; 1 \leq |s| \leq k). \tag{4.7}$$

A cost function based on these two factors, termed *normed deviation*, was proposed too:

$$normdev_f(m,k) = \max\{\frac{cidev_f(m)}{2}, \frac{pcdev_f(k)}{4}\}. \tag{4.8}$$

Millan et al. successfully derived derived $CI(1)$ functions with the nonlinearity values shown in Table 4.2. They also attempted to generate $CI(2)$ functions. Although low $cidev(2)$ values were attained, no $CI(2)$ functions were successfully evolved. The small deviations from the desired immunity are also given in Table 4.2. Although such deviations may look small, interpreting them needs care. A small $cidev(2)$ deviation from zero may actually be due to one or more $\hat{F}(\omega)$ with $|\omega| = 1$, i.e. the functions derived may not actually be $CI(1)$. More generally, the same deviation may be generated by a single errant $\hat{F}(\omega)$ or by many. A more discriminating cost function would seem appropriate. A cost function influenced by the notions of deviation of Millan et al. but which draws more on the

n	Nonlinearity	$cidev(1)$	Nonlinearity	$cidev(2)$
8	112	0	112	4
9	232	0	232	8
10	476	0	480	8
11	976	0	976	8
12	1972	0	1972	8

Table 4.2: Results from Correlation Immunity Experiments of Millan et al. [84]

experience of Chapter 3 is:

$$cost(f) = \left(\sum_{|\omega| \leq m} |\hat{F}(\omega)|^R \right) + A \times \max_{\omega} |\hat{F}(\omega)|. \tag{4.9}$$

This enables correlation immunity and nonlinearity to be taken into account. For correlation immunity, the values of all relevant $|\hat{F}(\omega)|$ rather than just the most extreme value are considered. The correlation immunity component is simply the cost function of Equation 3.7 with $X = 0$ and restricted to the relevant ω. The search will be restricted to balanced functions and so $\hat{F}(0) = 0$. Here A is a weighting constant for the nonlinearity component. Following the pattern of Chapter 3 it seems prudent (for a first pass at least) to ignore algebraic degree and autocorrelation during the search and just record the values attained by the resulting functions.

Experiments were carried out for 5-10 input variables. The value of A in Equation 4.9 was 10 throughout. An exponent parameter $R = 2.0$ was used throughout. The annealing parameters are given in Appendix A.2.2 as are the numbers of successes achieved for each order of immunity. Table 4.3 records the best values attained. A fifth component has been added to the table entries to record the best autocorrelation achieved for functions satisfying the other indicated criteria. The values marked with an asterisk are known to be suboptimal (from Table 4.1). The symbol $\Longleftarrow$ indicates that direct attempts failed but the values have been inherited from a higher order success (e.g. the technique successfully evolved a $(9, 5, 3, 192, 512)$ function; this is more extreme than a $(9, 4, 3, 192, 512)$ function and so we inherit the more extreme function).

The direct technique would appear to have achieved a fair amount of success. Indeed, $(7, 1, 5, 52)$, $(7, 2, 4, 48)$ and $(8, 1, 6, 112)$ were the best that had been achieved prior to 2000. For n=9 and 10 the case is more difficult. It seems unlikely that optimisation *as currently employed* will match constructive techniques. Constructive techniques have achieved higher nonlinearity values than any functions (correlation immune or not) attained by optimisation techniques deployed in this thesis. At this point some useful advice from a leading researcher

(5,1,3,12,8)	(5,2,2,8,32)	(5,3,1,0,32)		
(6,1,4,24,16)	(6,2,3,24,32)	(6,3,2,16,64)	(6,4,1,0,64)	
(7,1,5,52,32)*	(7,2,4,48,40)*	(7,3,3,48,128)	(7,4,2,32,128)	(7,5,1,0,128)
(8,1,6,112,40)*	(8,2,5,112,56)	(8,3,3,96,256)*	(8,4,3,96,256)	(8,5,2,64,256)
(9,1,7,232,72)*	(9,2,6,232,88)*	⟸*	⟸*	(9,5,3,192,512)
(10,1,8,476,104)*				

Table 4.3: Best Results (n,m,d,nl,ac) Obtained by the Direct Method

in Boolean functions helped the author considerably.

4.5 The Second Pass: Revisiting the Past

The above results were obtained using a cost function that was obviously aimed at achieving some specific order correlation immunity and high nonlinearity. However, low autocorrelation was often achieved essentially without trying in Chapter 3. The functions generated had equaled the best values for nonlinearity with optimal high algebraic degree and some of the lowest autocorrelation values ever obtained. It was suggested to the author [1] that the very same batch of generated functions might contain some correlation immune ones. In addition, the author's attention was brought to how change of basis had been successfully exploited in the construction of correlation immune functions (e.g. [72, 94].) This proves of considerable worth as indicated below.

Consider functions on n input variables. Now consider the set of Walsh zeroes

$$\mathrm{WZ} = \{\omega : \hat{F}(\omega) = \mathit{0}\}. \tag{4.10}$$

If WZ contains a linearly independent subset of dimension n, then there is a linear change of basis that gives rise to a $CI(1)$ function. Addition is logical XORing on the bitwise representation (arithmetic is in $G(2)^n$). Now 10011+11010=01001 and so '19' + '26'='9'. We see that '9', '19' and '26' are linearly dependent. If $\{\omega_1, \ldots, \omega_n\}$ is a linearly independent subset of WZ then an appropriate change of basis is defined by

$$(y_1, \ldots, y_n)^T = \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_n \end{pmatrix} (x_1, \ldots, x_n)^T, \tag{4.11}$$

with the new transformed function given by $g(y) = f(x)$. Here x_1 is the most significant bit of x and x_n the least significant bit. Similarly for y. That this gives

[1] Personal communication with Dr Subhamoy Maitra.

Maitra and	Pasalic and	Sarkar and	Best	Optimisation
Sarkar	Johannson	Maitra	Known	and Change
(1999) [107]	(1999) [94]	(1999)[108]		of Basis
(8,1,6,108)	(8,1,6,112)	(8,1,6,112)	(8,1,6,116)[72]	(8,1,6,116,24)
(9,1,7,220)	(9,1,7,236)	(9,1,7,240)	(9,1,7,240)	(9,1,7,236,40)
(10,1,8,476)	(10,1,8,480)	(10,1,8,484)	(10,1,8,488)[72]	(10,1,8,484,64)
(11,1,9,956)	(11,1,9,976)	(11,1,9,992)	(11,1,9,992)	(11,1,9,984,96)
(12,1,10,1980)		(12,1,10,1996)	(12,1,10,1996)	(12,1,10,1992,160)

Table 4.4: Properties of First Order Correlation Immune Functions

rise to a $CI(1)$ function is easily shown. Let $\gamma_i = 2^{n-i}$ for $1 \leq i \leq n$. Clearly $|\gamma_i| = 1$ since only one bit is set. Let $\hat{G}(\gamma_i)$ be the corresponding Walsh value for the function $g(y)$. Then we have

$$\begin{aligned} \hat{G}(\gamma_i) &= \sum_y \hat{g}(y)\hat{L}_{\gamma_i}(y) = \sum_y \hat{g}(y)(-1)^{\gamma_i \cdot y} = \sum_y \hat{g}(y)(-1)^{y_i}, && (4.12) \\ &= \sum_x \hat{f}(x)(-1)^{\omega_i \cdot x} = \hat{F}(\omega_i) = 0. && (4.13) \end{aligned}$$

In the new basis $0, 1, 2, 4, 8, ..2^{n-1}$ are Walsh zeroes and so $g(y)$ is balanced and $CI(1)$. Such a change of basis preserves algebraic degree, nonlinearity and autocorrelation (proofs omitted, see [77]). The functions obtained in Section 3.4 contained many first order immune functions under change of basis. Table 4.4 records the properties of the best functions obtained for n=8 to 12 and provides the results of three recent publications for comparison. In addition to the results shown, change of basis also provided $(5, 1, 3, 12, 8)$, $(6, 1, 4, 24, 16)$ (obtained also using the direct method) and $(7, 1, 5, 56, 16)$. As well as achieving upper bounds on nonlinearity and being Siegenthaler optimal, these functions also possess the lowest autocorrelations demonstrated for balanced functions of 5, 6 and 7 input variables respectively. The batches of functions generated earlier had contained examples of what at the time of generation would have been new results. Since correlation immunity was not a consideration at the time, no checks were carried out on the rank of the set of Walsh zeroes. For the primary goal (to equal all results for eight or fewer variables) there remain only $(7, 2, 4, 56)$ and $(8, 3, 4, 112)$ as targets. If $(7, 2, 4, 56)$ can be obtained, a simple construction can be used to obtain $(8, 3, 4, 112)$. Thus, the primary target should now be $(7, 2, 4, 56)$. This seems a very hard task for the technique and so some very special treatment (reported in Section 4.7) was adopted.

(5,1,3,12,8)	(5,2,2,8,32)	(5,3,1,0,32)		
(6,1,4,24,16)	(6,2,3,24,32)	(6,3,2,16,64)	(6,4,1,0,64)	
(7,1,5,56,16)	(7,2,4,56,32)	(7,3,3,48,128)	(7,4,2,32,128)	(7,5,1,0,128)
(8,1,6,116,24)	(8,2,5,112,56)	(8,3,3,96,256)	(8,4,3,96,256)	(8,5,2,64,256)
(9,1,7,236,40)	(9,2,6,232,88)	(9,3,3,192,512)	(9,4,3,192,512)	(9,5,3,192,512)
(10,1,8,484,64)				
(11,1,9,984,96)				
(12,1,10,1992,160)				

Table 4.5: Best Achieved Properties (n,m,d,nl,ac) by Any Optimisation Method

4.6 Comments on Autocorrelation

With the bounds of Table 4.1 being progressively achieved, it would appear that consideration of global avalanche characteristics (e.g. sum-of-squares or autocorrelation) are set to be next for consideration. In a forthcoming paper [71] Maitra analyses two recent recursive constructions. The first recursive construction, discussed in [11, 73], is shown to give rise to functions with linear structures (and so have the worst possible autocorrelation). Another recursive construction, used in [123], is also shown to give rise to functions with very poor autocorrelation. The constructions did not have low autocorrelation as a goal.

To provide targets for future research Table 4.5 records the full set of properties achieved by any optimisation technique (including the approach of the next section). It is interesting to note that for the very recently proposed function with profile $(8, 1, 6, 116)$ the best autocorrelation values were 80. [2] In this respect the function with profile $(8, 1, 6, 116, 24)$ obtained by optimisation and change of basis is significantly better (and, as indicated in Chapter 3, no balanced function on eight variables with nonlinearity 116 and autocorrelation of 16 has ever been demonstrated). Various correlation immune functions obtained using annealing have been made available to other researchers for further analysis.

4.7 Third Pass: Time for a Change

The work reported above has enhanced previous optimisation-based work on Boolean functions. However, it is still clear that the techniques *as employed by the author* are reaching their limits. More advanced optimisation technqiues could easily be brought to bear, though it is by no means clear how much this would improve effectiveness. A fair amount of further informal experimentation failed

[2] Personal communication with Dr Subhamoy Maitra indicated that all such generated functions had this value.

to produce $(7, 2, 4, 56, -)$. A decision was made at this point to try a radical departure. This turned out to have unexpected results and further applications.

4.7.1 Almost Boolean Functions 1 (ABF-1)

What is the assumption that seems to bind all current approaches to Boolean function design (by theoretical construction or optimisation-based approaches)? It is, perhaps, that researchers feel obliged to work with Boolean functions. All research reported so far in this thesis has also followed this route. The annealing searches start with balanced Boolean functions and move around the search space of balanced Boolean functions in an attempt to optimise properties such as high nonlinearity and specific order of correlation immunity. Boolean structure is preserved by the move operation of the search. This approach is a choice, it is not essential. Let us turn the problem at hand on its head and ask ‘Can we “fix” nonlinearity and correlation immunity properties and evolve Boolean structure?’

Suppose the Walsh-Hadamard spectrum, $\hat{F}$, of a balanced Boolean function on n input variables with specified nonlinearity nl and order of correlation immunity m, is given by

$$\hat{F} = (\hat{F}(0), \ldots, \hat{F}(2^n - 1)). \tag{4.14}$$

Let $\hat{P} = (\hat{P}(0), \ldots, \hat{P}(2^n - 1))$ be a permutation of the values of $\hat{F}$. Now consider the set of all permutations $\hat{P}$ with $\hat{P}(\omega) = 0$ for all $0 \leq |\omega| \leq m$. This is the set of permutations that maintain the properties required for balance and correlation immunity of order m. Furthermore, the nonlinearity value is maintained too (it is defined in terms of the largest absolute value amongst the spectral values).

For a Boolean function $\hat{f}$, the $\hat{F}(\omega)$ are effectively (uniformly scaled) projections of that function onto the basis vectors $\{\hat{L}_\omega\}$. If $\hat{L}_\omega = (\hat{L}_\omega(0), \ldots, \hat{L}_\omega(2^n - 1))$ and $\hat{f} = (\hat{f}(0), \ldots, \hat{f}(2^n - 1))$ then it is the case that

$$\hat{f} = \sum_\omega \left(\frac{\hat{f} \cdot \hat{L}_\omega}{2^{\frac{n}{2}}} \right) \frac{\hat{L}_\omega}{2^{\frac{n}{2}}} = \frac{1}{2^n} \sum_\omega \hat{F}(\omega) \hat{L}_\omega. \tag{4.15}$$

A Boolean function is a point $\hat{f}$ on the surface of the hypersphere of radius $2^{\frac{n}{2}}$. Permuting the values of the projections $\hat{F}(\omega)$ to give $\hat{P}$ gives rise therefore to another point $\hat{p}$ on this surface. Given a spectrum $\hat{P}$, the corresponding point can be obtained using the inverse Walsh transform

$$2^n \hat{p}(x) = \sum_\omega \hat{P}(\omega) \hat{L}_\omega(x). \tag{4.16}$$

Alternatively this can be viewed as following from Equation 4.15 (substituting p for f) by considering the xth element. Some permutations $\hat{P}$ will correspond to Boolean functions, most will not. Thus, the values of the various

$\hat{p}(x)$ obtained by inversion may not actually be $+1$ or -1. But some points obtained by inversion will be more like Boolean functions than others. For example, $(1.1, -1.1, 1.06, \ldots)$ 'looks more like' a Boolean Function than $(15.2, 12.1, 0.1, \ldots)$ since its elements are quite close to the allowable values of $+1$ or -1. A point $\hat{p}$ on the hypersphere surface can be associated with a Boolean function $b(\hat{p}) = (b(0), \ldots, b(2^n - 1))$ defined by:

$$\begin{array}{lcl} b(i) & = & 1 \text{ if } \hat{p}(i) > 0; \\ b(i) & = & -1 \text{ if } \hat{p}(i) < 0; and \\ b(i) & \in \{+1, -1\} & \text{otherwise.} \end{array} \tag{4.17}$$

This will not necessarily be a balanced Boolean function. This mapping allows us to determine how 'Boolean' a point $\hat{p}$ is and so punish with high costs those points that are not close to some Boolean function. A suitable cost function is given below

$$\text{cost}(\hat{p}) = \sum_{i=0}^{2^n-1} (\hat{p}(i) - b(\hat{p})(i))^2. \tag{4.18}$$

If a permutation of $\hat{P}$ gives rise under inversion to the point $\hat{p}$, then we associate $cost(\hat{p})$ with that permutation. If a permutation $\hat{P}$ gives rise to zero cost, then it is the spectrum of a Boolean function.

4.7.2 Evolving to a Boolean Function

A new approach to obtaining balanced, highly nonlinear correlation immune functions of order m is now possible. Start with a Walsh spectrum with the corresponding properties and search the space that maintains these properties until a zero cost solution is found. However, for this approach to be implemented a suitable initial spectrum satisfying the properties is needed. A move strategy that maintains those properties is also required.

The initial spectrum will be a permutation of the spectrum of a desired Boolean function. In general, it may be rather hard to generate such spectra, but in some cases it is easy. In particular, for the $(7, 4, 2, 56, -)$ case the maximum absolute value of elements in the Walsh spectrum is 16. Theory by Sarkar and Maitra [109] has shown that if $n \geq 3$ and $m \leq n - 3$ then the Walsh values $|\hat{F}(\omega)|$ of an m-th order resilient (balanced $CI(m)$) function on n variables must satisfy $|\hat{F}(\omega)| \equiv 0 \bmod 2^{m+2}$. Thus, the Walsh values for $(7, 2, 4, 56, -)$ must be 0, 16 or -16 (a Walsh value of 32 or above would give rise to a nonlinearity of 48 or less). The formula

$$2^n \times \hat{f}(0) = \sum_{\omega=0}^{2^n-1} \hat{F}(\omega) \tag{4.19}$$

defines the value of $\hat{f}(0)$. Arbitrarily fixing this to be 1 and using Parseval's equation allows us to determine that the spectrum must contain 36 1s, 28 -1s and 64 0s. Places in the starting spectrum corresponding to $0 \leq |\omega| \leq 2$ are fixed at 0 and the rest are arbitrarily allocated. Those elements fixed at 0 remain fixed throughout the search. The local search moves between spectra by making pairwise swaps between the remaining elements. Since the problem is highly nonlinear, once again an annealing approach is applied with the cost function given in Equation 4.18. The annealing parameters were: $\alpha = 0.99$, $MIL = 800$, $MaxIL = 1000$ and $MUL = 50$.

Five hundred runs were carried out resulting in five successes. Though this is hardly efficient it has succeeded in generating some example desired functions. Indeed, $(7, 2, 4, 56, -)$ functions had escaped demonstration until 2000. In [109] a new recursive construction was presented which required a $(7, 2, 4, 56, -)$ as a starting function (it noted that it was not known whether there existed such a function).[3]

It is noted that the best autocorrelation attained is 32, i.e. $(7, 2, 4, 56, 32)$ functions have been obtained in this way. This is lower than previously generated functions (such functions had autocorrelation at least 40. [4]) An example function is in Section A.3. The attempts also gave rise to six $CI(2)$ functions with nonlinearity 48. For the primary goal this leaves only $(8, 3, 4, 112, -)$ to be obtained. Unfortunately all attempts to use the above technique failed to evolve a function with this profile. This is simply obtained in Section 4.11 but we shall now pause for thought before closing the Boolean functions research.

4.7.3 The Importance of Keeping Alert

It might be concluded that the first application of the Almost Boolean Functions 1 (ABF-1) technique has met with success. This is not the case. Indeed, the method was originally intended as a means of attaining $(8, 0, -, 118, -)$ functions, i.e. the aim was to attain the best known upper bound on nonlinearity for balanced functions of eight variables. It is not known whether such functions actually exist. Demonstrating one such function would have been a highly significant result. If they exist, it is not clear what their spectra would be. This is (or seemed) a significant problem with the technique. Needless to say, all attempts failed.

[3] Other constructions were also presented that required instances of known functions as starting functions.

[4] Personal communication with Dr Subhamoy Maitra.

However, the recent theory indicating the existence of certain functions with three valued Walsh spectra (e.g. -16, 0 , 16) allowed the histogram of spectral values to be calculated. This was all that was needed to make progress. The technique had almost been forgotten. Had it not been for the difficulty experienced with the attempts to generate $(7, 2, 4, 56, -)$ functions using the cost function of Equation 4.9 it would have remained so. A significant result has been obtained using what might otherwise accurately be described as a failed technique. Earlier some excellent (and occasionally unequalled) results for $CI(1)$ functions were obtained via change of basis (using functions for which correlation immunity was not a consideration at the time of generation). The same batch of functions had contained counter-examples to autocorrelation conjectures and also to the sum-of-squares conjectures (though the latter was also sought deliberately).

It is prudent at this point to reflect. There has certainly been an element of good fortune here and there is no reason to hide this fact. Rather than wait for another fortuitous accident it is now prudent to ask ‘Can more be made of what has been generated or developed so far?’ Addressing this allows the conceptual toolkit promised in Chapter 1 to be expanded in some rather interesting ways, as is shown below.

4.8 Learning Lessons — Change of Basis II

Linear change of basis has proved to be an effective means of transforming functions to obtain first order correlation immunity. This raises an interesting question: ‘Can a similar transformation be found to produce second order correlation immunity?’ That is, from the set of Walsh zeroes WZ (defined in Equation 4.10) can a linearly independent subset

$$Z = \{\omega_1, \ldots, \omega_n\} \tag{4.20}$$

be found such that

$$\forall\{\omega_i, \omega_j\} \subseteq Z \cdot \omega_i \oplus \omega_j \in WZ. \tag{4.21}$$

Obtaining a linearly independent subset is an easily-solved problem of linear algebra (start with an empty set and add to the set only vectors that increase the dimension of the space spanned). There would appear to be no known method for obtaining a basis with the indicated second order characteristics. The problem is clearly hard but is of obvious relevance. It can also be couched as a nonlinear search problem. Let

$$pwz = \langle\omega_1, \ldots, \omega_r\rangle \tag{4.22}$$

be a permutation of the Walsh zeroes WZ. For each such permutation, let the first n elements form a candidate basis. Thus,

$$candBasis(pwz) = \{\omega_1, \ldots, \omega_n\}. \tag{4.23}$$

To be a suitable basis the set $\{\omega_1, \ldots, \omega_n\}$ must have rank n and the second order combinations $\omega_i \oplus \omega_j$ of its elements must also be in the set WZ. A permutation not meeting these requirements should be punished. For a candidate basis $candBasis(pwz)$ define the number of *misses* as the number of xor combinations of two candidate basis elements that are themselves not in WZ:

$$misses(candBasis(pwz)) = \#\{i, j : 1..n \cdot i < j \wedge w_i \oplus w_j \notin WZ\}. \tag{4.24}$$

A cost function that seeks to punish deviation from required properties is given by:

$$cost(pwz) = K * (n - rank(candBasis(pwz))) + misses(candBasis(pwz)). \tag{4.25}$$

With K=20 this cost function was used as part of an annealing search over the sets of Walsh zeroes with dimension 7 that did not give rise to correlation immune functions for the $(7, 2, 4, 56, -)$ problems. There were 23 such functions and so the ABF technique had generated 23 functions that could be transformed under simple linear change of basis to be $CI(1)$. The annealing-based search for bases giving second order immunity was successful in the case of 4 of these functions. A sample of ten balanced functions with nonlinearity of 48 with Walsh zeroes of rank 7 was subjected to similar change of basis attempts. There were nine successes out of ten, i.e. nine of the ten functions could be transformed to $CI(2)$. Oddly, the successful cases had 67 Walsh zeroes and the unsuccessful case had 64. Similar experiments were repeated for the functions generated by the ABF technique when targeted at the $(8, 3, 4, 112)$ case but no successes were encountered. A search for second order characteristics usually takes less than a minute.

4.9 Exploiting Invariance — Change of Basis III

4.9.1 Transforming to Gain Propagation Orders

Change of basis is also of use when attempting to obtain functions satisfying particular propagation criteria. Recall that a function f is said to be satisfy the *propagation characteristic of order* m (is $PC(m)$) if

$$|\hat{r}(s)| = 0; 1 \leq |s| \leq m. \tag{4.26}$$

Define the set of autocorrelation zeroes

$$ACZ = \{\omega : \hat{r}(\omega) = \mathit{0}\}. \tag{4.27}$$

If $\{\omega_1, \ldots, \omega_n\}$ is a linearly independent subset of ACZ then the change of basis defined by

$$(y_1, \ldots, y_n) \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_n \end{pmatrix} = (x_1, \ldots, x_n) \tag{4.28}$$

(or $y\Omega = x$), with a new function g defined by $g(y) = f(x)$, provides a transformation of the function to obtain a function that is $PC(1)$. Note that this transformation differs from the one used for correlation immunity. As before, let $\gamma_i = 2^{n-i}$ for $1 \leq i \leq n$. Clearly $|\gamma_i| = 1$ since only one bit is set. Then we have

$$\begin{aligned} \hat{r}(\gamma_i) &= \sum_y \hat{g}(y)\hat{g}(y \oplus \gamma_i) = \sum_x \hat{f}(x)\hat{f}(x \oplus \gamma_i\Omega) && (4.29) \\ &= \sum_x \hat{f}(x)\hat{f}(x \oplus \omega_i) = 0. && (4.30) \end{aligned}$$

In the new basis $1, 2, 4, .., 2^{n-1}$ are autocorrelation zeroes and the function is $PC(1)$. In the same way as before, if all pairwise combinations $w_i \oplus w_j$ from the basis subset are also in ACZ then the function transformed function is $PC(2)$. Very little experimentation has been carried out but this has already provided new information.

4.9.2 Algebraic Degree of $PC(2)$ Functions

Prior to 1997 the highest algebraic degree exhibited for a $PC(2)$ function was $\frac{n}{2}$ (for bent functions, which are actually $PC(n)$ — they have zero autocorrelation). Honda et al. [51] showed how this bound was very weak and demonstrated how to construct functions on $n = l + 2^l - 1$ input bits with algebraic degree $n - l - 1$ and showed also how to construct similar balanced functions. They note that the degree of their constructed functions is 'much larger than the best degree so far'. This is clearly true. They also comment

> Now suppose $f(x_1, \ldots, x_n)$ satisfies $PC(2)$. Then since f satisfies SAC *[Strict Avalanche Criterion]* we obtain a *trivial* [5] upper bound on $deg(f)$ such that $deg(f) \leq n - 1$.

[5] This author's italics.

Learning from past experience, it seems prudent to revisit the batches of functions generated in Chapter 3. For functions of six input variables, application of annealing based searches for second order charcateristics enabled balanced $PC(2)$ functions of algebraic degree 5 to be found. As far as the author is aware, no balanced $PC(2)$ function has ever been demonstrated at the 'trivial' bound of $n-1$. Balanced functions can have degree at most $n-1$ and so this cannot be bettered. An example function obtained is given in Appendix A.4.

Once again for low numbers of input variables optimisation is able to generate examples with optimal properties that have hitherto escaped theoretical construction. Honda et al. make no claim to optimality, merely that the previous best bound can be surpassed. Whether or not $PC(2)$ functions exist with degree $n-1$ for $n > 6$ is left as an open question (though preliminary experimentation has come very close — for $n = 7$ and 8 change of bases have been found that give rise to only a single element $w_i \oplus w_j$ not being in the set of AC zeroes). Thorough investigation of the application of the optimisation techniques to propagation characteristics (and other propagation criteria) is left as future work.

The generation of a single example meeting the 'trivial' bound with ease shows once again that optimisation techniques have some potential to check conjectures or to attack current bounds for relationships between the various criteria.

4.10 Back to Where I Started — ABF-2

The research started in Chapter 3 and continued in this chapter drew initial motivation from Parseval's equation and an observation that only bent functions attained the ideal bound $|\hat{F}(\omega)| = 2^{\frac{n}{2}}$ uniformly. Thus, bent functions were responsible in part for all the work on balanced functions this far. To round off the Boolean function research, it would be fitting to generate bent functions inspired by the experience of the research reported in this chapter.

Bent functions are clearly functions with two-valued Walsh spectra ($2^{\frac{n}{2}}$,$-2^{\frac{n}{2}}$). This means that techniques such as ABF-1 can easily be provided with an initial spectrum whose permutations can be searched. The ABF-1 technique used one particular approach to defining Boolean structure. This means was rather direct. It is useful to ask 'Is there another way?'

The well-known text by Ding et al. [31] quotes a theorem, due to Titsworth, that could be a way forward. This states that $\hat{F}(\omega)$ is the Walsh-Hadamard Transform of a binary Boolean function if and only if

$$\sum_{\omega} \hat{F}(\omega)\hat{F}(s \oplus \omega) = 2^{2n}\delta(s), \tag{4.31}$$

where $\delta(s) = 1$ if $s = 0$ and $\delta(s) = 0$ otherwise. This immediately suggests a cost function that punishes deviation from this:

$$cost(\hat{F}) = \sum_{s}(|\sum_{\omega} \hat{F}(\omega)\hat{F}(s \oplus \omega)|)^{R}. \tag{4.32}$$

When $s = 0$ the inner sum should be non-zero (and is constant for all permutations of the spectrum $\hat{F}$). For $s > 0$ the inner terms should be zero. This cost function punishes deviation from zero for these terms. Two sets of experiments have been performed with $R = 2$ aimed at evolving bent functions. For both sets, the cooling parameter $\alpha = 0.95$, the number of moves within an inner loop $MIL = 400$, the maximum number of inner loops $MaxIL = 800$ and the number of unproductive loops being 50, as usual.

The general strategy is much as for ABF-1. The search starts with the spectrum of a desired function (here a bent function) and attempts to evolve a permutation of it that corresponds to a Boolean function (i.e. one minimising the cost defined in Equation 4.32). Obtaining spectra for bent functions is easy since it must be the case that $|\hat{F}(\omega)| = 2^{\frac{n}{2}}$ for all ω. Equation 4.19 was used again to determine the numbers of positive and negative Walsh values. For six input variables, the spectra used had 36 elements at 8 and 28 elements at -8 (or vice-versa). For eight input variables, the spectra used had 136 elements at 4 and 120 elements at -4; although these element values should in fact be $+16$ and -16, it is convenient to scale them to avoid numerical overflow. The reasoning for adopting the cost function of Equation 4.32 still holds.

Fifty runs were carried out in each case. For six input variables the technique generated a bent function with (maximal) nonlinearity 28 and (maximal) algebraic degree 3 every time. The average time per run was 20.6s. For eight input variables the results are shown in Table 4.6. The average time per run was 4m 58s. Subsequent attempts to evolve bent functions on 10 inputs failed, despite significantly increasing computational resources for the search: $\alpha = 0.99$, $MIL = 1000$, $MaxIL = 3000$. Attempts to derive $(9, 3, 5, 240, -)$ functions failed similarly. Example six and eight variable bent functions are given in Appendix A.5 and A.6. As far as the author is aware, optimisation has not been applied to generate bent functions before. The technique above, which now will be referred to as ABF-2, has shown there may be many ways to attack the same problem. There is much scope for novel approaches.

4.11 And Finally - Resort to ‘Theory’

A rather odd approach to function evolution has been adopted. Attempts have been made to evolve functions without reference to any previous results for smaller or different functions. As noted earlier this is not the way theoreticians work.

Functions	Number
(8,-,4,120,0)	14
(8,-,4,105,12)	35
(8,-,4,103,12)	2

Table 4.6: Results of 50 Attempts to Generate Bent Functions on 8 Input Variables

Attempting to generate all functions from scratch is a bit like building a house without allowing the concept of a brick. There are many simple but effective constructions that can be brought to bear using the building blocks created so far. Having created a $(7, 2, 4, 56, -)$ function $f(x_1, \ldots, x_7)$ it can be shown that $g = x_8 \oplus f(x_1, \ldots, x_7)$ is $(8, 3, 4, 112, -)$. Similarly we can now construct $(9, 4, 4, 224, -)$, $(10, 5, 4, 448, -)$ and so on. The attempts to generate all functions achieving all bounds directly is, however, a good testing ground for the optimisation techniques. However, perhaps a more practical approach would be to use theoretical construction and heuristic search together.

4.12 Evidence for the Thesis

4.12.1 A Significant Increase in Power?

The following lend credence to the claim that, within the domain of application, the power of the techniques is significantly greater than evidenced in publicly available literature:

- The techniques have been able to generate functions demonstrated for the first time by theoretical construction as recently as 1999, 2000 and 2001. The techniques have shown themselves capable of providing cryptological researchers with useful information on current research problems. Though the primary criteria of concern in this chapter were balance, nonlinearity, algebraic degree and correlation immunity, some functions generated also had exceptionally low autocorrelation (much lower than those of some recently demonstrated functions). The generated functions are already under scrutiny by the research community. It is hoped that analysis of these functions will lead to further theoretical insights and eventually to more powerful theoretical constructions.

- As far as the author is aware, the annealing approach to finding change of basis is entirely novel and is capable of producing interesting results (both for correlation immunity and for propagation characteristics). The author knows of no mathematical technique to achieve similar results.

- Optimisation has provided examples exceeding previously met bounds for propagation characteristics. Once again, it is revealing new information that can inform theoretical reasoning about achievable bounds (similar points have been made in the previous chapter).

- Previous optimisation-based work on correlation immunity produced only $CI(1)$ functions. None possessed optimal nonlinearity. The techniques in this chapter has shown how all orders of correlation immunity can be generated (for small n at least) with Siegenthaler optimality and achieving at times the best possible theoretical bounds on nonlinearity with lower autocorrelations than any previously constructed functions.

- All results have been achieved with simply stated means. There is considerably conceptual economy demonstrated in the work reported here.

4.12.2 Toolkit Contributions

The contributions to the conceptual toolkit demonstrated in this chapter are:

- A simple conceptual framework for obtaining Siegenthaler-optimal and highly nonlinear $CI(m)$ functions with low autocorrelation.

- The evolution of Boolean structure using the ABF1 and ABF2 spectrum based methods.

- Annealing-assisted change of basis.

4.13 Commentary

Recent achievements in correlation immune functions design (2000 and 2001) have closed the case for functions of eight or fewer variables (in the sense that functions achieving optimal tradeoffs have been demonstrated). For nine or more input variables there remain unachieved bounds. These are likely to attract research attention since their attainment will provide whole series of larger functions by recursive constructions. Even when bounds have already been met by theoretical constructions the techniques of this chapter provide new ways of achieving these bounds.

Thus, the achievements of this chapter are significant. We have equalled the capabilities of all work to date in this respect. The autocorrelation of the best functions obtained in this chapter is low.

The $CI(1)$ cases deserve special mention. $(7, 1, 5, 56, -)$ was demonstrated only in 2000 and $(8, 1, 6, 116, -)$ was demonstrated only in 2001. Yet it would appear that the method of Chapter 3 contained such functions under change of basis. Furthermore some of the functions demonstrated, $(7, 1, 5, 56, 16)$ and $(8, 1, 6, 116, 24)$, equalled the best nonlinearity and autocorrelation values ever achieved by other researchers (even when a correlation immunity constraint was not required). Thus, it would appear that a cost function whose initial motivation was entirely driven by nonlinearity has proved sufficient to generate optimal or extreme values across the remaining criteria (algebraic degree, correlation immunity order and autocorrelation).

The concept of Almost Boolean Functions is a significant novelty in its own right. Ideally, this would have been suggested as an interesting avenue for 'Further Work.' It was the failure to achieve $(7, 2, 4, 56, -)$ by other means that forced it into service. As far as the author is aware only one paper has appeared using search and Walsh inversion (and then only for six or fewer variables). However, the authors searched for spectral distributions that corresponded precisely to Boolean functions. The concept of Almost Boolean Functions, although originally intended for a different purpose, has been shown capable of equaling one of the bounds only recently achieved. An enhancement allowed bent functions to be generated with maximal algebraic degree. Both are significant challenges to the Directness Assumption of Chapter 1. Again the autocorrelation of the heuristic method functions were favorable.

The simple change of basis idea exploited for $CI(1)$ gave rise to the idea of obtaining $CI(2)$ via a similar approach. The use of annealing to achieve a suitable change of basis shows that even in an area which is fundamental linear algebra (change of basis) there is perhaps room to consider the use of metaheuristic search.

Overall the conceptual toolkit promised in Chapter 1 has been considerably enhanced.

4.14 Issues Arising

4.14.1 Economy of Effort

Once again very lightweight cost functions have been adopted. In the more direct approach only correlation immunity and nonlinearity were targeted. Algebraic degree and autocorrelation were simply measured at the end. In addition, the use of change of basis on the functions generated in Chapter 3 allowed even correlation immunity to be effectively ignored as part of the search.

4.14.2 Breakdown at Higher n

The techniques clearly cannot match the nonlinearity values attained for n=9 and above. This is perhaps a little unusual since metaheuristic searches are usually seen at their best when the search space increases. What should one conclude? It would be simple to conclude that increasing sophistication of the optimisation techniques would improve matters. Though increasing optimisation sophistication can hardly do any harm there may be other factors at work. One should consider seriously that the models used (cost functions) are simply inappropriate for higher n. For higher numbers of input variables we simply have a different problem to address and this will require new methods.

4.14.3 Exploiting Invariance

Change of basis preserves certain desirable properties (nonlinearity, autocorrelation, algebraic degree) but not others, e.g. correlation immunity. The method of change of basis gave excellent results for $CI(1)$ functions but was seen at its best with the transformation of functions to give $(7, 2, 4, 56, -)$ functions. In the $CI(1)$ case no attempt was made to achieve correlation immunity by the original search. Here, simple linear algebra was used to achieve the desired result.

To achieve $(7, 2, 4, 56)$ with regularity simple linear algebra was unavailable but the invariance of nonlinearity, autocorrelation and algebraic degree permitted an additional search to roam over the space of bases to optimise with respect to the desired Walsh zero property. More generally if a set of desirable properties, I, are invariant under change of basis and another set, J, are not, there would appear to be an obvious approach to take — generate functions which are good with respect to I and then search over the space of bases to optimise with respect to J. As the $(7, 2, 4, 56)$ change of basis examples show this may well be a nonlinear search.

4.15 Open Problems

Below we shall state several research questions motivated by the work presented in this chapter:

1. Can Walsh Inversion and the concept of Almost Boolean Functions be exploited to derive further correlation immune Boolean functions?

2. What further properties not invariant under change of basis can usefully and effectively be optimised as indicated above?

3. What additional criteria are desirable and how do the functions generated by metheuristic search and traditional construction methods compare?

4. Why do the techniques not equal the achievements of construction as the number of input variables increases? Will different cost functions give better practical results?

5. Can theory and heuristic search be blended more effectively?

Chapter 5

Side Channels on Analysis

This chapter proposes two non-standard approaches to the cryptanalysis of identification schemes based on an NP-complete problem — the Permuted Perceptron Problem. These approaches are based loosely around the cryptanalysis notions of fault injection and timing analysis. Obvious cost functions are 'warped' in various ways. The solutions obtained using these warped functions are shown to be better than those obtained using the obvious ones. Warping is, in a sense, fault injection on the problem definition. The computational dynamics of a search for a secret are also shown to reveal much more information about the secret than the final result of the search itself. As the search progresses, solution elements eventually assume their final values and do not change for the rest of the search (i.e. they get stuck). The order *in which solution elements get stuck acts as a form of 'timing channel' that leaks huge amounts of information on the sought secret.*

5.1 Introduction

The previous chapters have shown optimisation techniques being applied to problems that have largely been the domain of theoretical construction. In this chapter the research moves on to consider a problem that might reasonably described as 'home ground' for optimisation — the analysis of crypto-systems based on NP-complete problem instances. Some very novel attacks are shown. The results show that, even on 'home ground', the power of metaheuristic search may be significantly underestimated.

5.2 Preliminaries: Perceptron Problems

In 1995 David Pointcheval suggested promising identification schemes based on the *Perceptron Problem* (PP) [98]. In fact, he chose a variant of this problem that is much harder to solve, known as the *Permuted Perceptron Problem* (PPP). The schemes rely for their security on the computational difficulty of finding highly constrained binary-valued solutions to a system of linear equations. If instances can be solved then the identification schemes are broken. The protocols used to implement the identification schemes are not described here (the reader is referred to [98] for details). This chapter concentrates on attacking the underlying PP and PPP problem instances.

A column vector whose entries have value +1 or -1 is termed an ϵ-vector. Similarly, a matrix whose entries have value +1 or -1 is termed an ϵ-matrix. The Perceptron Problem and Permuted Perceptron Problem are stated below:

- **Perceptron Problem: PP**
  Input: An m by n ϵ-matrix A.
  Problem: Find an ϵ-vector v of size n such that
  $(Av)_i \geq 0$ for all $i = 1, ..., m$.

- **Permuted Perceptron Problem: PPP**
  Input: An m by n ϵ-matrix A and a multiset S of non-negative numbers of size m.
  Problem: Find an ϵ-vector v of size n such that
  $\{\!\{ (Av)_i | i = \{1, ..., m\} \}\!\} = S$.

If v is a solution to the PP then all product elements $(Av)_i$ must be non-negative. The PPP multiset constraint requires the element values $(Av)_i$ to have a particular profile (i.e. a particular histogram). It specifies what set of values are to be taken by elements of the product Av and how many elements take each such value, but does not specify the particular value taken by any particular product element. For example consider the following matrix product Av:

$$Av = \begin{pmatrix} 1 & -1 & -1 & -1 & 1 \\ -1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 1 \\ 5 \end{pmatrix}.$$

All the product elements $(Av)_i$ are non-negative and so $v = (1, -1, 1, -1, 1)^T$ is a solution to the PP problem with the indicated 4 by 5 matrix A . The histogram of the values of the product elements is given by

$$H = \langle < 1, 1 >, < 3, 2 >, < 5, 1 > \rangle,$$

where $< n, k >$ denotes k elements taking value n.

It is always possible to generate feasible instances of these problems as indicated by Pointcheval. The matrix A and column vector v are generated randomly. If $(Av)_i < 0$ then the elements a_{ij} of the ith row are negated. This method of generation introduces significant structure into the problem. In particular, the elements of the column majority vector [1] are correlated with the corresponding elements of the generating secret v [98, 65]. The security of the schemes relies on the computational intractability of exploiting such structure.

Any PPP solution is obviously a solution to the corresponding PP since the PPP simply imposes an extra histogram (multiset) constraint. There may be many solutions to a particular PP instance and their corresponding histograms will usually be different. Thus, given a PPP instance, solving the underlying PP instance will usually not solve the PPP instance. Pointcheval investigated the complexity of generating PPP solutions by the repeated generation of PP solutions. His work lead him to recommended the use of (m, n) (i.e. m by n) matrices with $n = m+16$, indicating that these gave best practical security against a PP-solution based attack. Three particular sizes were identified as good candidates:

1. $(101, 117)$;
2. $(131, 147)$; and
3. $(151, 167)$.

5.3 Problem Warping

Pointcheval [98] and Knudsen and Meier [65] attack the PP using simulated annealing and a cost function of the form

$$cost(x) = \sum_{i=1}^{m} (max\{-(Ax)_i, 0\}), \tag{5.1}$$

where x is a candidate solution. The neighbourhood of the current candidate solution is simply the set of all vectors x' obtained from x by flipping the value of a single element. Pointcheval uses the annealing process directly to obtain solutions to the PP and reports

> "We have carried out many tests on square matrices ($m = n$), and on some other sizes, and during a day, we can find a solution of any instance of PP which [sic] size is less than about 200."

[1] The jth element of the column majority vector has value $+1$ if the sum of elements in the jth column is positive, and has value -1 otherwise.

Knudsen and Meier use an iterative procedure, each stage using multiple runs of the annealing algorithm. At each stage element values common across the various candidate solution vectors produced by the annealing runs are determined and then fixed for the remaining stages. They report solving PP-instances for various sizes including $(m, n) = (151, 167)$ far quicker (a factor of 180) than those reported earlier by Pointcheval.

The cost function of Equation 5.1 is very *direct*. It is an obvious characterisation of what the search process is required to achieve. Direct cost functions are, however, not always the most effective, as shown in Chapters 3 and 4. Examination of the way problem instances are generated reveals that small values of $(Av)_i$ are more likely than larger values. The initial distribution of the $(Av)_i$ is (essentially) binomial, with values potentially ranging from $-n$ to n. The negation of particular matrix rows simply folds the distribution at 0. This causes difficulties for the search process since attempting to cause negative $(Ax)_i$ to become positive by flipping the value of some element x_j is likely to cause various small but positive $(Ax)_i$ to become negative. It is just too easy for the search to get stuck in local optima.

One solution is to encourage the $(Ax)_i$ to assume values far from 0. This is easily effected: rather than punish when a product element $(Ax)_i$ is negative, punish when $(Ax)_i < K$ for some positive value K. A cost function of the following form suggests itself

$$cost(x) = g \sum_{i=1}^{m} (max\{K - (Ax)_i, 0\})^R. \tag{5.2}$$

The exponent parameter R once again allows for appropriate experimentation. Here, g magnifies the effect of changes when the current solution is changed and is really intended as a weighting factor when the cost function is extended for the PPP problem (see Section 5.4). For current purposes it is held constant. The annealing search is now posed with a different problem to solve. For larger values of K it is highly likely that no zero-cost solution to Equation 5.2 exists. This should be no deterrent to the use of this cost function. A cost function is *a means to an end* and a good cost function is one that works, i.e. one that guides the search to obtain desired results. Following the style of previous chapters, using this plausibly-motivated cost function family and simply seeing what happens is the first course of action.

By varying the parameters of this new cost function quite radical improvements can be brought in effectiveness. There would seem no obvious reason to restrict R to 1 (the value used by previous researchers). Experiments were carried out for PP instances of the following sizes and using the indicated values of K and R:

- $(201, 127)$: $K \in \{10, 15, 20\}$, $R = 2.0$;
- $(401, 427)$: $K \in \{10, 20, 25, 30\}$, $R = 2.0$;
- $(501, 517)$: $K = 25$, $R = 3.0$; and
- $(601, 617)$: $K = 25$, $R = 3.0$.

For each size ten problem instances were attacked. For each pair (K, R) of parameter values ten runs were carried out for each problem instance. A weighting value $g = 20$ was used throughout (this particular value is also used in some PPP problem runs in Section 5.4 though it does not pay a critical role in the current PP problem runs). For all runs the number MIL of moves in an inner loop was 400, the maximum number MUL of consecutive unproductive loops was 50 and the maximum number $MaxIL$ of inner loops was 400, except for the $(601, 617)$ instances, where 600 was used. The particular values were adopted after a little informal experimentation on instances on each problem size. No claim to optimality is made but the indicated values are sufficient to solve some instances for each indicated problem size. The results are shown in Tables 5.1 and 5.2. It was found that there were few direct simulated annealing solutions of the largest PP instances. However, it was often found that flipping a small number of annealing solution bits (e.g. 1, 2 or 3) provided a solution to the PP instance. Thus, each simulated annealing solution was subject to a 1, 2 and 3-bit enumerative search (where necessary). The aim was to find some solution to each of the problem instances.

All $(201, 217)$ problem instances gave rise to some solution, with Problem 0 being the most resilient (only three out of thirty annealing solutions gave rise to a PP solution and then only after three-bit enumerative search). Four of the ten $(401, 417)$-problems produced direct (0-bit search) simulated annealing solutions. All problems were solved by some run followed by at most an enumerative 2-bit search. For the $(501, 517)$ problems seven produced a solution (with up to 3-bit search used). Half the $(601, 617)$ problems gave rise to a solution. No claim to optimality is made here. For the larger problem sizes only one cost function has been used with only ten runs for each problem.

The results serve as a simple demonstration of how small changes may matter greatly. Mutated or warped cost functions have been used to excellent effect, easily out-performing the standard one. The solutions obtained by the warping are highly correlated with the actual defining solution of the problem. For the $(201, 217)$ problems, the best solution over the 30 runs for each problem ranged from 79.2% – 87.1 % correct. For the $(401, 417)$, $(501, 517)$ and $(601, 617)$ problems, the ranges were 83.4 % – 87.5%, 80.6% – 86.4% and 77.5% – 86.1%. This

Pr	0	1	2	3	Pr	0	1	2	3
0	0	0	0	3	5	0	4	6	5
1	3	6	2	11	6	3	6	12	5
2	1	11	6	8	7	4	7	14	2
3	8	12	6	3	8	3	14	2	9
4	0	4	5	4	9	1	1	5	4
PP(201,217):30 Runs									

Pr	0	1	2	Pr	0	1	2
0	0	0	1	5	0	1	0
1	0	0	2	6	1	2	6
2	0	0	1	7	0	11	6
3	1	4	14	8	0	2	9
4	1	3	6	9	3	12	11
PP(401,417):40 Runs							

Table 5.1: Number of Successes after Simulated Annealing Plus N-bit Hill Climbing for (201,217) and (401,417)

Pr	0	1	2	3	Pr	0	1	2	3
0	0	0	0	0	5	0	0	0	2
1	0	0	1	1	6	0	0	0	1
2	0	2	2	4	7	0	0	0	1
3	0	1	1	3	8	0	0	0	0
4	0	0	0	0	9	0	1	3	4
PP(501,517):10 Runs									

Pr	0	1	2	3	Pr	0	1	2	3
0	0	0	0	1	5	0	0	2	2
1	0	0	0	1	6	0	2	1	1
2	0	0	0	0	7	0	0	0	0
3	0	0	0	2	8	0	0	0	0
4	0	0	0	0	9	0	0	0	0
PP(601,617):10 Runs									

Table 5.2: Number of Successes after Simulated Annealing Plus N-bit Hill Climbing for (501,517) and (601,617)

is generally much better than solutions obtained using the standard cost function ($K = 0$ and $R = 1$).

That an enumerative search should be required to obtain PP solutions is not surprising. The cost functions used do not define what it means to be a solution to the PP and the annealing runs have attempted to solve the problems they were posed. However, the results show that the cost functions used do characterise in some way what it means to be 'close' to some PP solution. The enumerative search can be considered as a second stage optimisation with respect to the traditional cost function (i.e. with $K = 0$). Such two-stage optimisation mirrors the basic NLT and ACT approaches of Chapters 3 and 4.

Application of warped cost functions has allowed instances of the Perceptron Problem to be handled far larger than any successfully attacked in the literature so far — previous attacks sought vectors of length at most 200. Here vectors of more than 600 elements have been successfully found. This is a huge increase in power and stresses how fragile is current understanding of the power of heuristic optimisation for cryptanalysis. Warping the problem is, of course, a challenge to the directness assumption. More stringent problems have been attacked with the result that the original (weaker) problems are actually solved. Warping the cost functions is, in a sense, a form of fault injection on the problem definition.

5.4 Attacking the Permuted Perceptron Problem

In 1999 Knudsen and Meier showed that the smallest size $(m, n) = (101, 117)$ recommended by Pointcheval for the PPP was susceptible to the same sort of iterative attack described earlier but with a modified cost function given by

$$Cost(x) = g \sum_{i=1}^{m} (max\{K - (Ax)_i, 0\})^R + \sum_{i=1}^{n} (|H_x(i) - H_v(i)|)^R . \qquad (5.3)$$

H_x is the histogram for the current solution x. Thus $H_x(k) = \#\{j : (Ax)_j = k\}$, i.e. $H_x(k)$ is the number of the $w_i = (Ax)_i$ that have value k. Similarly H_v is the reference histogram for the target solution v. The histograms apply only to positive $(Ax)_i$ elements. In all experiments reported by Knudsen and Meier $R = 1$ and $K = 0$. If all runs of the technique agree on certain secret element values the assumption is that there is a good chance that the agreed value is the correct one. Agreed bits are fixed and the process is carried out repeatedly until all bits are agreed by all runs. Unfortunately some (small number of) bits unanimously agreed at some stage in this way are actually wrong and an enumerative search is made to find them. Such a search makes use of the fact that a good number of bits fixed by this technique are fixed correctly before any bits are fixed incorrectly.

(m, n)	Values of g	Values of K	Values of R
(101, 117)	20, 10, 5	1, 3, 5, 7, 9, 11, 13, 15	2, 1.5, 1
(131, 147)	20, 10	7, 10, 13, 16	2,1
(151, 167)	20, 15, 10, 5	5, 10, 15, 20	2, 1

Table 5.3: Cost Function Parameter Values for the PPP Problems

5.4.1 ClearBox Cryptanalysis - Looking Inside the Box

All applications of optimisation techniques in cryptography seem to view optimisation as a black-box technique. A problem is served as input, the optimisation algorithm is applied, and some output is obtained (a candidate secret in the PP and PPP examples). However, in moving from the starting solution to the eventual solution an heuristic algorithm will have evaluated a cost function at many thousands of points. Each such evaluation is *a source of information* for the guidance process. In the black-box approach this information is simply *thrown away* after its immediate use. For the PPP, the information loss is huge.

As the temperature cools in an application of simulated annealing it becomes more difficult to accept worsening moves. At some stage an element will assume the value of 1 (or -1) and then never change for the rest of the search, i.e. it gets stuck at that value. This observation is at the root of Chardaire et al.'s thermostatistical persistency [16]. It is found that some bits have a considerable tendency to get stuck earlier than others when annealing is applied. One could ask 'Why?' The answer is that the structure of the problem instance defined by the matrix and reference histogram exerts such influence as to cause this. The bits that get stuck early *tend to get stuck at the correct values*. Once a bit has got stuck at the wrong value it is inevitable that other bits will subsequently get stuck at wrong values too. However, it is unclear how many bits will get stuck at the right value before a wrong value is fixed. Could this be significantly high?

This has been investigated for the three suggested problem sizes and a variety of cost functions. For each problem size a particular cost function is defined by a value of g, a value of K and a value of R. Thirty problem instances were created for each problem size. For each problem and each cost function ten runs of the annealing process were carried out. The runs were assessed on two criteria: the number of bits set correctly in the final candidate solution and the number of bits initially stuck correctly before a bit became stuck at an incorrect value. The annealing parameters for the PPP experiments are: $MIL = 400$, $MaxIL = 400$, $MUL = 50$ and $\alpha = 0.95$. For $(101, 117)$ instances $3 \times 8 \times 3 = 72$ cost functions were used and so there were 720 runs in total for each problem instance. Similarly, for the $(131, 147)$ and $(151, 167)$ instances the numbers of runs carried out were 160 and 320 respectively. The results are shown in Table 5.4. For each problem the maximum number of correctly set bits in some final output of an annealing run is recorded together with the maximum number of bits fixed correctly during some run before a bit was set incorrectly (usually these will not be simultaneously achieved by one particular run).

It is interesting also to see how average results vary between problems. For each (151,167) problem Table 5.5 records the average behaviour over all strategies. For a particular problem each cost function has been exercised ten times. From the set of 10 results the minimum number of bits correct, the average number of bits correct and the maximum number of bits correct can be calculated. The averages of these results can be calculated over the 32 different cost functions. The results are shown in the columns 2–4 of Table 5.5. Similar results for the initial bits correctly set are shown in columns 5–7. The volatility of the approach can easily be seen. The average maximum number of bits correct over all strategies for Problem 3 was 147.19 (i.e. on average the best result from the 10 annealing runs using a cost function was about 20). For Problem 20 the corresponding figure is 131.56. For Problem 18, the number of initial bits set correctly for each set of 10 runs for a cost function averages 73, whereas for Problem 19 it averages 28.03. The average amount of information leaked varies hugely.

Similarly, for each cost function Table 5.6 records similar information but calculated over all problems. There are some subtle interactions between parameters choices. In general it can be seen that K values of 15 and 10 seem better for maximum final and initial bits correct than values of 20 and 5. For $K = 20$, $R = 2$ always gives better results than $R = 1$. This is clearly not always the case when $K = 5$. It would seem possible to use such inter-dependencies to make attacks more efficient. Here however, a rather more blunt approach will be taken.

5.4.2 Making Best Use of Available Information

To make most efficient use of the results in Table 5.4 some particular features of the PPP problem may be exploited. These are now presented.

It is generally possible to tell whether the number of incorrectly set bits in a vector x is even or odd. Consider Ax for any candidate solution vector x. Flipping any single element of x causes the components $(Ax)_i$ to change by ± 2. Similarly, flipping any two bits of x causes the components to change by ± 4, or else stay the same. Flipping three bits causes the components to change by ± 2 or ± 6. Generalising, if x may be transformed into the secret generating solution v by changing an even number of bits, then $(Ax)_i = (Av)_i \pm 4k$ for some integer k. Similarly, if an odd number of bit changes are needed then $(Ax)_i = (Av)_i \pm 4k + 2$. For any x let

$$SUM1(x) = \#\{i : (Ax)_i = 4k + 1, \text{for some k}\}; \text{ and let} \quad (5.4)$$

Pr 0	102	50	Pr 15	102	56
Pr 1	100	45	Pr 16	101	39
Pr 2	103	45	Pr 17	103	51
Pr 3	99	53	Pr 18	103	40
Pr 4	101	46	Pr 19	103	50
Pr 5	108	72	Pr 20	105	62
Pr 6	99	39	Pr 21	107	68
Pr 7	101	56	Pr 22	106	58
Pr 8	104	55	Pr 23	103	62
Pr 9	106	56	Pr 24	103	53
Pr 10	102	56	Pr 25	100	56
Pr 11	107	56	Pr 26	104	51
Pr 12	101	58	Pr 27	98	53
Pr 13	104	42	Pr 28	105	57
Pr 14	102	47	Pr 29	103	56
Size (101,117) 720 runs					

Pr 0	126	42	Pr 15	122	59
Pr 1	135	68	Pr 16	124	41
Pr 2	128	64	Pr 17	121	42
Pr 3	126	67	Pr 18	130	62
Pr 4	130	39	Pr 19	129	53
Pr 5	131	70	Pr 20	132	67
Pr 6	126	47	Pr 21	128	59
Pr 7	128	56	Pr 22	129	97
Pr 8	123	52	Pr 23	127	61
Pr 9	139	75	Pr 24	126	43
Pr 10	129	51	Pr 25	127	72
Pr 11	123	48	Pr 26	132	44
Pr 12	134	57	Pr 27	125	68
Pr 13	132	62	Pr 28	126	38
Pr 14	124	37	Pr 29	123	50
Size (131,147) 160 runs					

Pr 0	148	72	Pr 15	141	63
Pr 1	142	64	Pr 16	151	56
Pr 2	145	66	Pr 17	144	82
Pr 3	157	88	Pr 18	147	98
Pr 4	147	58	Pr 19	137	47
Pr 5	140	67	Pr 20	136	69
Pr 6	151	86	Pr 21	140	59
Pr 7	135	48	Pr 22	142	55
Pr 8	143	55	Pr 23	146	67
Pr 9	150	95	Pr 24	138	69
Pr 10	149	61	Pr 25	147	69
Pr 11	145	70	Pr 26	145	61
Pr 12	143	49	Pr 27	146	68
Pr 13	138	63	Pr 28	141	64
Pr 14	147	58	Pr 29	143	80
Size (151,167) 320 runs					

Table 5.4: Maximum Bits Correct and Maximum Initial Bits Correct Over All Runs

Problem	Final Bits Correct			Initial Bits Correct		
	Av.Min	Over.Av	Av.Max	Av.Min	Over.Av	Av.Max
Prob 0	130.84	135.82	140.34	6.84	22.06	43.81
Prob 1	125.66	130.25	136.12	7.41	23.68	43.28
Prob 2	129.31	135.24	140.94	8.78	25.78	47.59
Prob 3	133.16	140.88	147.19	23.72	44.34	66.81
Prob 4	127.53	132.96	138.44	9	24.3	41.25
Prob 5	128.22	132	135.72	9.66	27.13	47.22
Prob 6	136.06	140.78	145.22	12.91	31.31	53.69
Prob 7	116.94	123.18	129.22	8.28	21.33	35.56
Prob 8	129.56	133.32	137.22	5.16	16.9	33.09
Prob 9	135.34	139.48	143.84	25.03	49.25	73
Prob 10	127.34	132.09	136.94	3.84	16.82	34.78
Prob 11	127.91	135.78	141.53	23.66	39.82	57.62
Prob 12	124.69	131.02	137.34	6.19	20.69	35.25
Prob 13	122.69	127.63	132.44	10.66	25.44	42.16
Prob 14	127.91	132.17	137.12	9.91	25.47	43.5
Prob 15	125.25	130.67	134.91	5.59	18.6	35.62
Prob 16	132.75	139.54	145.41	3.84	17.9	38.12
Prob 17	125.91	131.04	136.38	24.12	42.57	61.72
Prob 18	133.78	138.31	143.12	30.06	53.65	72.81
Prob 19	122.94	127.63	132.25	1.09	11.93	28.03
Prob 20	122.16	126.53	131.56	10.25	26.03	43.94
Prob 21	126.62	131.68	136.69	8.72	24.37	43.34
Prob 22	123.28	129.99	135.91	4.06	17.48	36.09
Prob 23	127.34	133.53	139.31	7.25	21.68	40.88
Prob 24	120.28	126.09	131.94	16.78	35.12	54.59
Prob 25	130.16	136.05	140.38	8	25.65	46.06
Prob 26	134.09	138.25	141.94	6.84	23.44	45.97
Prob 27	131.22	136.85	142.19	5.97	23.45	46.22
Prob 28	119.03	125.63	133.38	6.19	20.47	37.62
Prob 29	129.12	134.59	139.47	18.25	37.94	59.44

Table 5.5: Summary over all strategies

Strategy	Final Bits Correct			Initial Bits Correct		
(K,g,R)	Av.Min	Over.Av	Av.Max	Av.Min	Over.Av	Av.Max
(20,20,2)	132.47	136.02	139.23	12.9	29.25	49.97
(20,20,1)	129.93	132.71	135.07	8.47	24.13	40.73
(20,15,2)	132.17	135.82	138.9	14.8	30.57	49.27
(20,15,1)	130.03	132.49	135	9.33	24.86	43.27
(20,10,2)	132.6	135.74	138.57	12.77	29.54	49.1
(20,10,1)	129.97	132.37	134.53	10.3	24.3	41.8
(20,5,2)	132.13	135.8	138.9	13.7	30.71	49.33
(20,5,1)	129.73	132.29	135.07	8.47	24.44	41.97
(15,20,2)	129.7	135.13	139.63	12.23	29.67	48.1
(15,20,1)	129.57	133.67	137.6	10.57	26.64	45.63
(15,15,2)	130.87	135.32	140.1	12.37	30.43	50.7
(15,15,1)	130.1	133.76	137.33	12.1	28.09	46.3
(15,10,2)	129.63	135.15	140.1	12.53	30.85	49.83
(15,10,1)	129.23	133.57	137.3	12.83	29.77	47.9
(15,5,2)	130.5	135.49	140.13	14.9	31.16	51.67
(15,5,1)	129.63	133.67	137.3	10.6	27.61	47.07
(10,20,2)	126.67	133.01	140.23	10.53	28.56	48.87
(10,20,1)	128	134.06	139.47	11.27	28.89	50.6
(10,15,2)	126.6	133.39	140.1	11.53	29.25	47.23
(10,15,1)	128.87	134.17	138.9	12.4	29.28	49.47
(10,10,2)	126.93	133.54	139.73	10.67	28.06	49.3
(10,10,1)	128.57	133.91	139.13	14.27	29.95	47.03
(10,5,2)	126.6	133.43	140.53	12.63	28.5	47.73
(10,5,1)	128.8	134.2	139.73	12.77	29.48	49.27
(5,20,2)	120.4	128.16	135.97	7.23	21.24	38.93
(5,20,1)	122	130.1	138.77	8.87	23.74	43.23
(5,15,2)	120.27	129.18	137	7.97	23.81	44
(5,15,1)	122.5	130.04	137.53	7.47	23.73	43.27
(5,10,2)	121.37	129.17	137.03	7.57	22.54	42.1
(5,10,1)	122.8	130.13	137.2	8.9	24.15	44.6
(5,5,2)	121.23	129.19	136.87	8.13	21.95	40.67
(5,5,1)	122.37	130.22	137.77	8.87	23.75	42.77

Table 5.6: Summary of Each Strategy over All Problems

$$SUM2(x) = \#\{i : (Ax)_i = 4k+3, \text{for some k}\}. \tag{5.5}$$

If x is the secret v then

$$SUM1(v) = H_v(1) + H_v(5) + H_v(9) + \dots \tag{5.6}$$

and

$$SUM2(v) = H_v(3) + H_v(7) + H_v(11) + \dots, \tag{5.7}$$

where H_v is the publicly available reference histogram. If x is obtained from v by an even number of bit changes, then

$$SUM1(v) = SUM1(x) \tag{5.8}$$

and

$$SUM2(v) = SUM2(x). \tag{5.9}$$

If x is obtained from v by an odd number of bit changes, then

$$SUM1(v) = SUM2(x) \tag{5.10}$$

and

$$SUM2(v) = SUM1(x). \tag{5.11}$$

Only one of $SUM1(v)$ and $SUM2(v)$ can be odd (since $n = SUM1(v) + SUM2(v)$ is odd for the proposed PPP sizes). Thus, for any vector x it is possible to determine whether it differs from the (unknown) secret v by an even or odd number of bits using the (known) respective values of $SUM1(x)$ and $SUM1(v)$.

Suppose that v is the actual secret and x is a solution obtained by annealing. If x is a high performing solution (with few bits wrong) then $(Ax)_i$ will often be very close to $(Av)_i$. Indeed it is generally the case that the values of $(AV)_i$ and $(Ax)_i$ are very close. For the (151,167) problem instances, if $(Ax)_i = 1$ then the average actual value of $(Av)_i$ was 6.46. For (131,147) and (101,117) instances the averages were 6.23 and 6.02.

Suppose that $(Ax)_i = 1$ and ten bits are wrong. Typically it will be the case that $(Av)_i \in \{1, 5, 9, 13\}$. This observation has a big impact on enumerative search. For the sake of argument suppose that $(Ax)_i = (Av)_i = 1$. Then the ten bit changes must have no effect on the resulting value of $(Ax)_i$. This means that for five bits $a_{ij}x_j = 1$ for the other five $a_{ij}x_j = -1$. This considerably reduces any enumerative search. For example, searching over 117 bits would usually require C_{10}^{117} (around 4.4×10^{15}) but now requires a search of order around $C_5^{58} \times C_5^{57}$ (around 2.1×10^{13}). This assumes that for bits x_j $\#\{a_{ij}x_j = 1\} = 58$ and $\#\{a_{ij}x_j = -1\} = 57$ (or vice versa). In practice, this may not be the case but any skew actually reduces the complexity of the search. In this respect, it may be computationally advantageous to consider some $(Ax)_i < 0$. For example, if $(Ax)_i = -7$ and there are 10 bits wrong then $(Av)_i$ must be in the range $1..13$ with the smaller values much more likely. If $(Av)_i = 1$ then there must be seven wrong bits currently with $a_{ij}x_j = -1$ and three with $a_{ij}x_j = 1$. This is a powerful mechanism that will be used repeatedly.

To use this mechanism one has to guess the relationship of $(Ax)_i$ to $(Av)_i$. As indicated above, this will generally add only a factor of four to the search (and often less). One has also to determine how many bits are actually wrong too. One can start by assuming that the solution vector has the minimum number of bits wrong yet witnessed and engage in enumerative searches. If these fail, simply increment the number of bits assumed incorrect by 2 and repeat the search processes (only even numbers or odd numbers of wrong bits need be considered). The complexity of the search is dominated by the actual number of wrong bits (searches assuming fewer numbers of wrong bits are trivial by comparison). The complexities reported in this chapter therefore assume knowledge of the number of wrong bits in the current solution.

5.4.3 The Direct Attack

It is obvious that ‘warping’ the cost function produces results that are indeed better than those obtained under standard cost functions. Thus, in the $(101, 117)$ problems three have given rise to solutions with 10 bits or fewer wrong. Once the highest performing solution has been selected (a factor of 720) an enumerative search of order $C_5^{58} \times C_5^{57}$ will find the solution in these cases.

For the (131,147) and (151,167) instances extreme results are also occasionally produced. (131,147), Problem 9, gave rise to one solution with only 8 bits wrong. (151,167), Problem 3, similarly gave rise to a solution with only 10 bits wrong. This would require a total search of approximately $320 \times C_5^{84} \times C_5^{83}$ which is less than 2^{60}. Thus, even a fairly brutal search will suffice on occasion, even for the biggest sizes. This is not the most efficient way of solving the problem however.

5.4.4 Timing Supported Attack

The largest number of initially settled bits can clearly leak a huge amount of information. For (151,167) problems 18, 9 and 3 some solution was obtained whose first 98, 95,88 initially stuck bits were correct. The respective complexities of brute force search over the remaining entries would be of order $2^{69}, 2^{72}, 2^{79}$. Although not within the traditional 2^{64} distance they are sufficiently close to render use of the PPP scheme impossible. For (131,147) there would appear to be an outlier problem 22 with 97 initial bits correct. This leaves a search of order 2^{50}.

66	71	65	69	69	47	75	65	60	56
63	54	64	64	67	63	71	64	67	64
56	51	56	59	63	66	62	70	58	62

Table 5.7: Complexity (log 2) of Timing Supported Attacks on (101,117)

85	59	74	76	77	67	84	77	87	47
76	88	64	67	91	85	89	94	71	76
65	76	57	77	85	72	71	77	86	88

Table 5.8: Complexity (log 2) of Timing Supported Attacks on (131,147)

Another approach would be to consider in turn all possible pairs of solutions obtained. One pair contains a solution VMAX with the maximum number of bits correct and a solution VINIT with the maximum number of initial bits correct. This pair could form the basis for the subsequent search and we can calculate the computational complexity of finding the exact solution. Obtaining this pair requires a search factor equal to the number of runs squared.

Assume that at least the first I bits initially fixed in VINIT are correct. Change the corresponding bits in VMAX to agree with those in VINIT. These bits are now excluded from the subsequent search — the search will be over the remaining $n-I$ bits of the modified VMAX. For example, suppose in the $(101, 117)$ case that the best initial solution provides us with at least 37 bits (from Table 5.4 this applies to all 30 problems). This leaves us with 80 bits over which to conduct the remaining search. Suppose ten wrong bits remain. The total complexity of the whole search is now approximately

$$720 \times 720 \times C_5^{40} \times C_5^{40}. \tag{5.12}$$

Enumerative searches can be performed under optimistic assumptions and these can be progressively relaxed leading to more complex searches. Assuming the number of initially set bits is known, and the number of bits wrong in the best final solution is known, the complexities of the searches are given in Tables 5.7, 5.8 and 5.9.

80	94	88	56	86	95	70	111	95	68
81	86	97	100	86	96	78	83	72	108
99	99	97	86	97	83	89	85	95	85

Table 5.9: Complexity (log 2) of Timing Supported Attacks on (151,167)

Probs 0–9	0	41	0	78	0	30	0	0	0	87
Probs 10–19	0	41	0	0	0	39	0	62	88	0
Probs 20–29	0	36	0	0	36	32	33	36	28	46

Table 5.10: Top N Agreed Correct

5.4.5 Other Attacks

Other attacks are possible. For example taking the majority vector over all solution runs (whatever the strategy) can on occasion leak a great deal of information. Commonality of solution elements of repeated runs is at the heart of Knudsen and Meier's technique. This strategy can be adopted here. If runs agree under widespread problem deformation (i.e. under multiple parameter strategies) then there is often good cause to believe they agree correctly. Rather than insist on absolute agreement, we can rank the bits according to the degree of agreement. Frequently the top ranked bits are correct though the method is somewhat sporadic. Table 5.10 shows the number of top ranked bits correct for each (151,167) problem. Thus, for problem 1 the 41 bits that gave rise to most agreement over all runs were actually correct. We can see that for problems 3,9 and 18 the most agreed 78, 87 and 88 bits were correct (in the sense that the majority vector is right). This is very significant since around half of all bits are revealed with very little computational cost. It is also possible to add up the sticking times of components over all runs. When these are ranked (the highest being the one that took least aggregate time to get stuck) the results can also leak information. In some cases only 1 bit of the first ranked 100 for the (151,167) gave rise to a majority vector component that was incorrect.

5.5 Evidence for the Thesis

5.5.1 A Significant Increase in Power?

The following lend credence to the claim that, within the domain of application, the power of the techniques is significantly greater than evidenced in publicly available literature:

- Annealing techniques have been shown to be capable of bringing within established computational range problems that have not been attacked successfully by any other technique.

- The problem warping technique (with hill-climbing) has been shown to be capable solving instances of the Perceptron Problem with secret vectors approximately three times as long as previous attempts.

- A timing channel on the search process has rendered instances of the largest suggested PPP sizes vulnerable. As far as the author is aware, no schemes based on NP-hard problem instances have been derived to be resistant to such an attack.

5.5.2 Toolkit Contributions

The principal contributions to the conceptual toolkit demonstrated in this chapter are:

- The notion of problem warping — fault injection on mathematics. Every cost function achieves something. This issue is really understanding what.

- The timing channel attack motivated by thermo-statistical persistency [16].

Neither concept has been seen in the context of cryptological application of metaheuristic search before.

5.6 Issues Arising

5.6.1 Choice of Cost Function I - Directness

It is generally agreed that the success of heuristic searches may depend crucially on the cost function used. The results of this chapter (both raw problem warping and the timing channel work) lend further credence to this view. However, virtually every cost function encountered during the wide-ranging review of Chapter 2 could be described as 'direct' — a zero cost solution would for most purposes define the solution to the actual problem. The cost functions used have been entirely natural characterisations of the problem at hand. All the cost functions used previously for the PP and PPP problems fall into this category. The author has looked at many optimisation papers from other application domains. It would appear that cost function choice is also generally direct. For the PP and PPP problems, the natural and most direct cost functions (e.g. with K=0) seem to be *the worst possible* choices from the identified family. Once again, problem formulation is crucial much as argued in Chapter 3. The cost functions used here are unusual in that minimal achievable cost solutions will generally not be solutions to the original problem. However, the solutions obtained may be *very highly correlated* with actual solutions.

5.6.2 Choice of Cost Function II — Black Box Assumption

To date, every application of heuristic techniques for cryptanalysis seems concerned only with the final outputs from the searches — the search trajectories are ignored. For the PPP instances it has been shown that the search trajectory itself conveys huge amounts of information. Techniques such as simulated annealing are *guided* search techniques. The guidance is provided by the cost surface as the search moves from state to state. The guidance is information about the problem instance and its solution. Each cost function evaluation provides (a very small amount of) information on the cost surface. Working only with the final solutions throws away vast amounts of information.

The timing channel is a highly unusual means of exploiting the way search works. As far as the author is aware this is the first time monitoring the search process in action has been used to attack cryptographic schemes. The strong structure in problem instances reveals itself in the early trajectory. This is a radical shift in thinking for cryptography. Yet the idea has clearly been of use in optimisation. In particular, thermo-statistical persistency fixes bits at particular values if a bit is tending very strongly to adopt a particular value. The approach adopted here is motivated by the thermo-statistical persistency work but is simpler — just watch the process cool. The work has shown only one way in which analysis of search trajectories can be exploited. There may be many others.

5.6.3 Profiling and the One-Shot Fallacy

Knudsen and Meier's work and the work described here do not seek to find a solution via a single application of an heuristic search technique. Rather, they acknowledge that the results of such applications will be stochastic (the quality of the result will vary between runs) and that no single application is likely to produce an totally correct result. Consequently, they carry out multiple runs and seek to exploit the *distributional properties* of the results obtained. This is a major shift in thinking. The author believes that the principal reason why the techniques have seen little application to cryptanalysis (even to schemes based on standard NP-complete problems — the most natural domain of application) is because it has generally been accepted that the techniques just 'will not work'. For most serious cryptanalysis problems, the author believes that this is entirely true, as long as a *single* run is assumed.

Let us examine further what lies behind the success of Knudsen and Meier's work and the work reported in this chapter. Knudsen and Meier started with the idea of exploiting commonality in solutions obtained by runs of an annealing algorithm. By *observing* where the process started to 'go wrong' they were able to identify suitable points for enumerative search to begin. Experimentation and observation of the results were crucial to the development of the technique. This is seen to even greater effect in this chapter. Large scale experimentation has been carried out and features of the searches identified to produce two major attacks. Not only have we observed the properties of the final outcomes of the searches, we have also observed and utilised the trajectories of such searches. What links all this work is the notion of *profiling*.

Profiling is the most important notion used in this thesis. It underpins both the fault injection and the timing work. Let us state very clearly: for most problems of interest the application of annealing (or similar) techniques will most likely not produce a solution directly. Carrying out multiple runs will also generally fail to produce an answer. Furthermore, given a particular cost function it is likely to be very difficult to predict a priori what the quality of the results will be.

However, it is apparent that some cost functions give radically better results than others. Furthermore by running the techniques with various parameter settings over many instances of randomly generated problems we were able to understand better what the quality and distribution of correctness was likely to be. By experimentation we were able to see how solutions could be combined and what information could reasonably be expected. This demonstrates the importance of profiling.

5.6.4 Side Channels — Retrospective Analogies

In the 1990s three new (or at any rate publicly low profile) and dangerous types of attacks were seen. These were fault injection attacks [5], timing attacks [66] and power analysis attacks [57]. De Milo, Lipton and Boneh [5] showed how failures in a cryptographic public key algorithm could leak information in a way that could cause the secret keys to be found. Subsequent papers have extended this sort of attack to work on block cipher schemes. Hardware failures can be induced by a variety of means and since this form of attack is highly potent, measures must be taken by crypto-system designers. Paul Kocher's attack on exponentiation based algorithms in 1996 [66] was another significant result. Essentially the data dependent execution times of exponentiation operations are used to compromise a variety of systems including RSA. Further results revealed that the power consumed by crypto-systems (e.g. smart cards) could leak information about internal state and operations to compromise secret key data. The term 'side channel' is now used to refer to these forms of leakage.

There is a factor that links all these attacks — they are attacks on an *implementation* not an attack on the mathematical algorithm itself. The bulk of cryptographic literature deals with mathematical entities. Issues such as fault injection, timing and power consumption are outside the remit of the models considered. The three attacks above coupled with the rise of smart cards means such attacks are now considered as a matter of course.

There may be a variety of other physical observables that could be used to leak information, e.g. heat or electromagnetic fields. We shall not address them here. We shall simply observe that the timing channel attack on the PPP is an obvious analogue of Kocher's real-time timing attacks. Similarly, problem warping could be viewed as an analogue of Boneh et al.'s fault injection attacks. We have simply injected a 'fault' into the 'natural' problem definition. The results would appear to be similar. It is these side channels that leak far more information than can be obtained by direct cryptanalytical techniques (i.e. by attacks on the algorithm). Furthermore, these are side channels on an *analysis* technique (here annealing-based search).

These analogies were drawn after the work had been carried out. The principal motivation for the work was Chardaire et al.'s thermo-statistical persistency [16]. However, once such links have been made, there is an obvious desire to seek out further analogies. Might other forms of physical side channel find interpretation in annealing-based cryptanalysis? What for example, would the analogue of power analysis be? Various optimisation techniques adapt the search dynamically based on monitoring of the trajectory (tabu search may take into account various statistics of the history of the search in deciding which move to take). What would the cryptanalysis analogues be?

5.7 Open Problems

Prompted by the work reported so far, a number of questions of obvious importance are presented, to which the author does not know the answer.

1. Suppose a 'naïve' attempt is made to evolve a key for a modern block cipher using annealing. For example, consider a plaintext P and corresponding ciphertext C obtained using some secret key K_s. A simple attempt to evolve a key might use a cost function of the form

$$cost(K) = HammingDistance(C, E(K : P)),$$

   i.e. encrypt the plaintext P with K and measure the Hamming distance of the resulting ciphertext with the reference ciphertext. Such attempts will undoubtedly fail — the search will get stuck in a local optimum every time.

However, the fact that a key K is a local optimum is a *source of information* on the secret key K_s. Is there any way of exploiting this? If such attempts are repeated using a large number of plaintext-ciphertext pairs can distributional properties of the results obtained be exploited to reveal the key (or components of it)?

2. Timing attacks on smart cards have proven to be very powerful. Little regard is paid to *extreme* timing values. Can optimisation be used to generate many plaintexts with extreme encryption times? The fact that a particular plaintext gives rise to extreme execution time is a *source of information* on the secret key K_s. How could this be exploited?

3. Can profiling techniques be applied successfully to other identification schemes based on NP-complete problems? Consider the permuted kernel problem [111]. Let all arithmetic be modulo some prime p. Consider an m by n matrix A and an n by 1 column vector v. Can a permutation vector v' be found (with the elements of v simply reordered) such that $Av' = 0$? It seems likely that cost functions can be found that give rise to biases in the solutions obtained. The real issue is somewhat more difficult — how can such patterns be exploited to break the system? Can genetic programming techniques be used to discover exploitable patterns?

4. Can specific induced distributions of properties be used to identify useful extremes? For example, by examination of the Hamming distances between the solutions obtained over many runs can those solutions that are of highest quality (i.e. with most correct bits) be identified?

5. The cost functions giving best results vary between problems. Can characteristics of specific problem instances be identified to identify cost functions (or combinations) with best chances of high performance for those specific instances?

6. Differential power analysis is one of the major physical attacks in recent years on smart cards. Can a cryptanalysis analogy with power analysis be found for heuristic searches?

7. In the PPP warped cost functions were adopted to produce various leakage channels but the very form of the cost functions impeded a true solution to the problem being gained. Can the cost functions be dynamically altered so that at the end of the search they reduce to the traditional ones? Can the value of K be progressively lowered so that all bits are revealed by the timing channel? There is a conflict here to be resolved. The timing channel relies on bits getting ‘stuck’ and lowering K makes accepting moves easier.

Chapter 6

The Heuristic Evolution of Security Protocols

Tradeoffs are an important part of engineering security. Protocol security is important. So are efficiency and cost. This chapter provides a framework for handling such aspects in a uniform way using heuristic search techniques. A belief logic, due to Burrows, Abadi and Needham (and universally called BAN logic) is viewed as both a specification language and proof system and also as a 'protocol programming language'. The work shows how simulated annealing and genetic algorithms can be used to evolve efficient and provably correct protocols.

6.1 Introduction

Security protocol development and analysis is probably the most active research area in computer security. This is due partly to its practical importance, but also to the fact that protocols are often small enough to be meaningfully analysed by formal means. Abstractions of protocol implementations are typically expressed in some formal notation and proofs of security properties are typically carried out using theorem provers or model checkers. The topic is generally considered a challenging one. This presents an opportunity to show that problems of significant abstraction in cryptology can be attacked using metaheuristic search. A method is described to refine automatically a formal protocol specification to a more concrete protocol in a manner that ensures it is provably correct. The work is intended to counter the almost universal tendency in extant cryptological applications of metaheuristic search to address what are essentially low-level problems. Also, the reader will recall that work of Chapter 5 was described (rightly) as 'home ground' for metaheuristic search. This research reported in this chapter is an attempt to 'play away from home'.

6.2 Automated Support for Protocol Development

Providing convincing and practical support for secure co-operation between distributed parties is one of the major tasks facing the security community. Over a decade ago Burrows, Abadi and Needham recognised this and developed a belief logic (almost invariably called 'BAN Logic') that could be used to reason about the security of protocol abstractions [10]. The work created considerable debate and gave fresh impetus to protocol security research and to formal approaches in particular. These formal approaches can be seen as complementing the more heuristic techniques such as the Interrogator and the NRL Protocol Analyser [63].

Various formalisms and tools have been brought to bear on the problem. Some researchers, for example, have encoded theories of protocol security in the logics of powerful theorem provers such as HOL [115]. Brackin has also developed BAN-like belief logic approaches to verification and has produced significant HOL-based support [6, 8, 7]). Some have provided process-algebraic definitions of security and used model-checking to demonstrate an implementation's conformance. Of particular note here is the flexible CSP framework described in [106] (which includes a chapter on the use of theorem proving). Others have sought to use theorem proving and model checking harmoniously [48]. There has been a great deal of research in the field.

It would seem that automated support in the area is largely limited to the analysis of existing protocols (or abstractions of them) with respect to a definition of security. There is virtually no work at all in automated secure refinement (i.e. design synthesis). The first results on metaheuristic search for secure protocol development (based on the work reported in this chapter) were presented in [22]. The only other work using search as a design technique is the recent model checking work of Song and Perrig [117, 118].

It must be remembered that correctness is only one design goal and designers often wish to find an *efficient* way of implementing a specification. In this chapter a framework is described for automated protocol synthesis that seeks to handle issues such as correctness and efficiency requirements in a uniform way.

6.3 Protocols and Belief Logic Representations

The notational conventions used in this chapter are now introduced together with some background on security protocols and how they can be represented in a belief logic. A subset of BAN logic will be used. This subset is powerful enough to allow the evolution of meaningful protocols and acts as a vehicle for proof of concept.

6.3.1 Notational Conventions

General Protocol Description. The parties that participate in security protocols are generally termed *principals*. A protocol run consists of a sequence of messages between principals and will be described using the standard notation. Principals are generally denoted by capitals such as A, B and S (for a key server). The sequence of messages

$$\begin{array}{l}(1)\ A \rightarrow B : F\\ (2)\ B \rightarrow S : G\\ (3)\ S \rightarrow B : H\end{array}$$

denotes a protocol in which A sends F to B, B then sends G to S, who then sends H to B.

Keys and Encryption. All messages are encrypted using symmetric (conventional) key encryption, where both principals share the same key. Messages sent over a network unencrypted might simply be spoofed or altered by a malicious adversary. In practice unencrypted concrete messages may be sent to implement a protocol but these do not carry security significance. They will generally be messages to achieve synchronisation between principals, e.g. to cause secure encrypted components to be sent. A symmetric key intended for communication between A and B is denoted by Kab etc. The message X encrypted using key K is denoted by $\{X\}_K$. A message may have several components, separated by commas. Thus

$$(1)\ A \rightarrow B : \{Y, Z\}_{Kab}$$

denotes that in the first message of the protocol A sends to B the message with two components Y and Z encrypted using key Kab.

Nonces. Principals often generate and use elements to enable them to determine that messages they receive really have been created as part of the current run and are not replays of previously issued messages. These elements have values specific to the current run and are intended to be used in at most one protocol run. A stream of nonces could typically be obtained using a secure pseudo-random number generator. A nonce generated by A is denoted by Na etc. If a principal generates a nonce for the current protocol run and receives messages that contain it, those messages are deemed to have been created after the nonce was generated.

6.3.2 Belief Logic Representations

Motivation and Basic Notation

At the concrete level a protocol is a sequence of messages between principals. Initially the principals hold sets of data items. The protocol messages are used to communicate such data items. At the end of the protocol the principals should hold some identified sets of data. For example, in a key distribution protocol, the aim might be that a particular key Kab, held initially by a key server S, should be held at the end by both A and B (and no-one else).

In belief logic representations, the initial data items, the message components and the eventual desired data state elements are replaced by assertions of belief. Rather than holding a key Kab the key server now holds the *belief* that the key Kab is good for communication between A and B. Similarly, a message does not contain the key Kab but the same belief about key goodness. Rather than A and B both holding the key at the end they are now expected to hold the belief in the goodness of key Kab. In some cases the relationship between beliefs held and corresponding concrete data is not so clear. For example, it might be required that A believe that B believes that the key Kab is good for communication (second order belief). This may occur, for example, when A receives a message from B encrypted using the key Kab without there being any explicit data item to represent this.

Thus, a belief logic can be used to provide a semantics to an existing concrete protocol (a process known as *idealisation*). BAN logic provides a language for expressing belief assertions. These assertions are found as assumptions, statements made in messages and in the final goals of the protocols. This has been the way BAN logic has been used so far, i.e. to prove properties of, or discover flaws in, existing protocols. The techniques reported below will work directly with the abstract belief logic representations (with the eventual aim of refining to a concrete representation). Only a subset of the full logic is used. This is for proof of concept only.

Key Goodness. The assertion

$$P \overset{K}{\longleftrightarrow} Q$$

means K is a good key for communication between P and Q. Implicit in this is that the key K has not been revealed to any principal other than P or Q (and possibly a highly trusted third party).

Nonceness and Freshness. The assertion

$$Na$$

means that Na has the appropriate form for a nonce, i.e. it satisfies any formatting conventions. The assertion

$$\sharp(Na)$$

means that the candidate nonce is *fresh* (♯); it has not been used before and the presence of Na in any message means that that message is part of the current protocol run.

Once Said. The assertion

$$P \mid\!\sim X$$

means P once sent ($\mid\!\sim$) a message containing assertion X. More informally it is customary to say that P *once said* X. For example

$$S \mid\!\sim A \overset{Kab}{\longleftrightarrow} B$$

means that principal S (a key server) once said that Kab is a good key for A and B to use.

Believes. The assertion

$$P \mid\!\equiv X$$

means that P *believes* ($\mid\!\equiv$) assertion X. For example

$$S \mid\!\equiv A \overset{Kab}{\longleftrightarrow} B$$

means that principal S (a key server) believes that Kab is a good key for A and B to use to communicate.

Jurisdiction. The assertion

$$S \mid\!\Rightarrow X$$

means that S *has jurisdiction* ($\mid\!\Rightarrow$) over statement X. This captures the notion that some principals are trusted to carry out certain tasks and make particular judgements and statements. Key servers, for example, are highly trusted and developed to very rigorous standards. They might legitimately be trusted far more than normal principals. They should have ‘jurisdiction’ over statements about whether a key is good. A principal P who accepts S’s jurisdiction over such a statement will ‘take S’s word for it’. This notion is formalised below by means of an inference rule.

Inference Rules

When a message is received by a principal who possesses the appropriate key to decrypt it, the logic provides inference rules that dictate what new beliefs he may infer from the message contents. The major inference rules are given below (with P and Q representing arbitrary principals and X an arbitrary assertion).

Message Meaning Rule If principal P sees ($\triangleleft$) a message encrypted under a key K it believes it shares only with principal Q, P may conclude that it was originally created by Q, who ‘once said’ its contents. In formal terms

$$\frac{P \triangleleft \{X\}_K, P \mid\!\equiv P \overset{K}{\longleftrightarrow} Q}{P \mid\!\equiv Q \mid\!\sim X}$$

Nonce Verification Rule If P believes that Q once said X and believes X to be recent (fresh), then P may conclude (believe) that Q currently believes X. In formal terms

$$\frac{P \mid\equiv Q \mid\sim X, P \mid\equiv \sharp(X)}{P \mid\equiv Q \mid\equiv X}$$

Jurisdiction Rule If P believes that Q believes X and$P$ also believes that Q is an authority on (has *jurisdiction* ($\mid\Rightarrow$) over) the matter, then P should believe X too. In formal terms

$$\frac{P \mid\equiv Q \mid\equiv X, P \mid\equiv Q \mid\Rightarrow X}{P \mid\equiv X}$$

These rules are well-motivated. The Message Meaning Rule captures the notion that only P and Q are able to create the encrypted message $\{X\}_K$ and so if P did not create it (it is implicit in the logic that P can recognise messages he has created himself) it must have been created by Q, and so he may deduce that Q once said its contents X.

The Nonce Verification Rule is a way of 'promoting' once said assertions to actual belief. Even if P believes that Q once said X he cannot be sure Q believes X now. Q may have actually uttered a message yesterday (containing X) and a malicious observer may simply have recorded it and may now be replaying it today. Consider for example a message sent by Q containing the assertion $\sharp(Nq)$. Q may well have believed Nq to be a fresh (never used before) nonce yesterday, but clearly would not make such an utterance today (because Nq has been used already). On receiving a malicious replay of the message P would still legitimately conclude that Q once said $\sharp(Na)$ but not that Q actually believes it. If however, there is something about the message that indicates it has been created very recently (typically a nonce connected to the current run of the protocol) then P may legitimately infer that Q actually believes it now.

The Jurisdiction Rule provides a formal means by which a belief held by a trusted principal can become held by another principal. The rules above provide means by which a principal P may legitimately deduce (i.e. believe) that Q believes X. P need not, in general, believe X himself. It is acceptance of Q's jurisdiction over X that allows P to believe X himself.

Some lesser rules are also needed, such as the ability to deduce $A \mid\sim Na$ from $A \mid\sim (Na, Nb)$ etc. but these are omitted here (they are implemented as part of the tool).

An important feature of BAN logic is that principals are *honest*. They do not lie; the sender of a message communicates only sincere beliefs, i.e. ones that it holds at the time of message issue. These beliefs may have been held initially or else derived via BAN-inference rule applications when previous messages were received. *Any series of honest exchanges between two principals defines a feasible (with respect to the logic) protocol.* It is this set of feasible protocols that are considered as the design space.

Illustrative Example

Figure 6.1 gives a set of initial assumptions and a feasible protocol. In this protocol fragment key server S believes that the key Kab is a good key for A and B to use. Both A and S believe that the key Kas is good for communication between them. A believes that S has jurisdiction over key Kab's goodness. A also believes that a particular number Na is a well-formed nonce and that it is actually fresh. Let us assume that the single goal of this protocol is for A to believe the key Kab is good for communicating with B (i.e. $A \mid\equiv A \overset{Kab}{\longleftrightarrow} B$), and so the protocol is a fragment of some key distribution protocol.

$$\begin{array}{ll} S \mid\equiv A \overset{Kab}{\longleftrightarrow} B & S \mid\equiv A \overset{Kas}{\longleftrightarrow} S \\ A \mid\equiv A \overset{Kas}{\longleftrightarrow} S & A \mid\equiv S \mid\Rightarrow A \overset{Kab}{\longleftrightarrow} B \\ A \mid\equiv Na & A \mid\equiv \sharp(Na) \end{array}$$

$$\begin{array}{l} (1)\ A \rightarrow S : \{Na\}_{Kas} \\ (2)\ S \rightarrow A : \{A \overset{Kab}{\longleftrightarrow} B, A \mid\sim Na\}_{Kas} \end{array}$$

Figure 6.1: Initial Assumptions and An Example Feasible Protocol

A believes Na is well-formed and so may legitimately include it in message (1) to S. This is encrypted with a key A believes is good for communicating with S. When S receives (sees) this encrypted message, it can apply the key Kas to decrypt it and deduce (using the Message Meaning Rule) that $A \mid\sim Na$. There is nothing about the message that should convince S that the message is fresh. S may now reply with message (2) which contains two of its currently held beliefs (the first is an initial assumption, the second is the newly derived once 'said in the past' belief). Now when A receives message (2) he may decrypt it to reveal its contents. Using the Message Meaning Rule he may conclude that S once uttered its contents. In detail, he may conclude $S \mid\sim A \overset{Kab}{\longleftrightarrow} B$ and $S \mid\sim A \mid\sim Na$. But this message contains an assertion involving Na, a number A believes to be fresh and A can conclude (using the Nonce Verification Rule) that S actually believes what he has uttered. Thus, A deduces that $S \mid\equiv A \mid\sim Na$ and also $S \mid\equiv A \overset{Kab}{\longleftrightarrow} B$. Since A believes that S has jurisdiction over key goodness assertions, A may now conclude $A \overset{Kab}{\longleftrightarrow} B$ using the jurisdiction rule.

General Protocols

Starting with a set of assumptions, a number of choices can be made for the direction and contents of the first message. Once the direction of a message is decided, the contents of each such message can be chosen. In the protocol above A could send any of $2^4 - 1$ non-empty combinations of beliefs (however odd they may seem). Once the contents are chosen the message can then be 'sent' and the receiver's belief state updated accordingly. A protocol is a sequence of feasible choices about which messages to send at each stage. Many feasible protocols are generated and 'executed' to find out what they actually achieve. Indeed, the author views BAN logic not only as a specification notation and proof system but also as a *protocol programming language*. This view seems close to that of the original BAN logic's authors. They start with designed protocols and 'execute' them (i.e. make the appropriate BAN inferences) and check whether the protocols meet their goals. The approach described here starts with arbitrary feasible protocols executes them to see what they achieve.

The abstract *execution* of any series of *feasible* exchanges defines a constructive proof that a protocol achieves what it does (since the execution requires actual updating of the belief states). The aim is to find protocols that achieve what they do but *also achieve what is wanted*. Thus, we wish to search the space of feasible protocols for ones satisfying a specification. Candidate protocols will be generated in a way that ensures they are feasible.

6.4 Solutions and Fitness

6.4.1 Solutions as Integer Sequences

Sequences of non-negative integers can be interpreted as valid protocols. Each message has a sender, a receiver and a sequence of C belief components. C is chosen by the user. Consider an arbitrary sequence of $C + 2$ non-negative integers $vs, vr, vb_1, \ldots, vb_c$. Number the principals $0..(N-1)$. $vs \bmod N$ gives the sender, $vr \bmod N$ gives the receiver. If the sender and receiver turn out to be the same then take $(vr+1) \bmod N$ as the receiver. $vb_1, \ldots, vb_c$ are interpreted as indices into the vector of beliefs currently held by the sender (i.e. held by the sender at the time the message is issued). Thus, if there are T beliefs currently held by the sender (indexed by $0..(T-1)$) then the index of the first belief component of the message to be sent is given by $vb_1 \bmod T$ etc. In this way, arbitrary positive integers can be interpreted as validly held beliefs. In fact, only certain simple types

2	$A \overset{Kab}{\longleftrightarrow} B$
1	$A \overset{Kas}{\longleftrightarrow} S$
0	null
Index	Belief

Initially

3	$A \mid\!\sim Na$
2	$A \overset{Kab}{\longleftrightarrow} B$
1	$A \overset{Kas}{\longleftrightarrow} S$
0	null
Index	Belief

After Message

Figure 6.2: Belief States of S

of messages may be sent and so the implementation actually deals with indices of *sendable* beliefs (see Section 6.4.2).

Initially each principal maintains an ordered sequence of beliefs $X, Y, \ldots$. The first belief is a special “null belief” that is used to denote absence of a belief in a message component. When a principal derives a new belief Z, i.e. when it receives a message, this is added to the sequence at the tail. For the protocol given in Figure 6.1 the belief states of principal S, initially and after receiving the first message

$$(1)A \rightarrow S : \{Na\}_{Kas}$$

are given in Figure 6.2.

The toolset implementation is in Java with principals’ beliefs stored in vectors with index values starting at 0. Beliefs are indexed by their current position in the sequence. Thus, the null belief has index 0 for each principal.

After receiving the first message the contents of the second message

$$(2)S \rightarrow A : \{A \overset{Kab}{\longleftrightarrow} B, A \mid\!\sim Na\}_{Kas}$$

would be represented by a sequence of appropriate indices for the sender S, e.g. $(2, 3, 0, 0)$, $(0, 3, 2, 0)$ or even (with redundancy) by $(0, 2, 2, 3)$ etc. Thus, only beliefs actually held are included. The message is honest. Non-null initial beliefs are read in from a file. The initial belief ordering may be considered arbitrary.

6.4.2 Initial and Sendable Beliefs

Initial beliefs may involve 1, 2 or 3 operators from the set $\{\mid\!\sim, \mid\!\equiv, \mid\!\Rightarrow\}$. Let SIMPLE be the set of atomic assertions about nonces (e.g. Na), the goodness of keys (e.g. $A \overset{Kab}{\longleftrightarrow} B$), or the freshness of such assertions (e.g. $\sharp(Na), \sharp(A \overset{Kab}{\longleftrightarrow} B)$). Then initial beliefs may take the following forms:

- $P \mid\equiv X$, where P is a principal and $X \in$ SIMPLE.
  For example, $A \mid\equiv Na, S \mid\equiv A \overset{Kab}{\longleftrightarrow} B$.

- $P \mid\equiv Q \mid\Rightarrow X$, where P and Q are principals and $X \in$ SIMPLE.
  For example, $A \mid\equiv S \mid\Rightarrow A \overset{Kab}{\longleftrightarrow} B$

- $P \mid\equiv Q \mid\Rightarrow R\ op\ X$
  where $op \in \{\mid\equiv, \mid\sim\ , \mid\Rightarrow\}$, P, Q and R are principals and $X \in$ SIMPLE.
  For example, $A \mid\equiv S \mid\Rightarrow B \mid\sim Nb$.

The above do not seem unduly restrictive; and reasonable assumptions tend to be immediate to a principal or else about jurisdiction; examination of [10] seems to confirm this.

Sendable beliefs are assertions in SIMPLE, the null belief and assertions involving a single operator in the set $\{\mid\equiv,\ \mid\sim,\ \mid\Rightarrow\}$ to be included in messages. Thus, Na, $S \mid\Rightarrow A \overset{Kab}{\longleftrightarrow} B$ are both sendable but $A \mid\sim B \mid\sim Nb$ is not. *This is to simplify initial investigation only*.

6.4.3 Goals

Goals are read in from a file. They may have one, two or three operators from the set $\{\mid\equiv, \mid\sim\ , \mid\Rightarrow\}$. The first such operator must be $\mid\equiv$. For example
$A \mid\equiv Na,\ \ A \mid\equiv B \mid\equiv A \overset{K}{\longleftrightarrow} B, A \mid\equiv S \mid\Rightarrow B \mid\sim Nb$

6.4.4 Interpreting a Solution

An integer sequence representation of M messages is decoded and executed as a protocol as follows.

- Install the initial belief states of the relevant principals (from file).

- For $m = 1$ to M (i.e. for each message do the following)

  1. determine $sender_m$ and $receiver_m$ (as indicated in section 6.4.1). If they share a key for communication then proceed else ignore the rest of the message and go to 5.

  2. decode each of the C beliefs corresponding to message m. Each belief is represented by an integer V say. If the sender currently holds T sendable (see Section 6.4.2) beliefs then this is interpreted as the $jth = (V\ mod\, T\ + 1)th$ sendable belief.

3. Examine the set of received beliefs. Note whether any of the beliefs contains a component that the receiver believes to be fresh, e.g. if A receives a message containing the belief $A \mid\!\sim Na$ and A believes that the nonce is fresh, $\sharp(Na)$, then the whole message is regarded as fresh.
4. Update the receiver's belief sequence to reflect receipt of these assertions. If an assertion X was received then $sender \mid\!\sim X$ is added to the receiver's belief vector (this represents $receiver \models sender \mid\!\sim X$). This together with (1) above implements the Message Meaning Rule (see Section 6.3.2). If the message is fresh then the Nonce Verification Rule is applied to add $sender \models X$ to the receiver's current belief vector. Thus, it is now the case that $receiver \models sender \models X$. This implements the Nonce Verification Rule of Section 6.3.2. Similarly the Jurisdiction Rule is now applied to create further beliefs until no further beliefs can be added.
5. Record the number of goals achieved after this message has been processed.

For convenience belief states are not updated with conjunctions. Thus, on receiving (Na, Nb) say, the receiver's belief state would be updated with $sender \mid\!\sim Na$, $sender \mid\!\sim Nb$ but not with $sender \mid\!\sim (Na, Nb)$.

Once a protocol has been executed in the above way and the intermediate results recorded, a fitness can be calculated for the protocol as described below.

6.4.5 The Fitness Function

The fitness function for each putative protocol generated needs to guide the search to a solution. There must be some notion of goodness of a putative protocol reflecting how close it comes to satisfying the goals of the problem. Fitness functions for a protocol of the following form have been used:

$$\sum_{i=1}^{M} w_i * g_i \tag{6.1}$$

The w_i are weightings and g_i is the number of goals achieved after message i (including goals achieved after previous messages). Since the nature of the fitness function influences the heuristic search, a number of strategies for setting the weights w_i are investigated below. These are detailed in Table 6.1 and described below. Other general forms of fitness function might usefully be investigated. The form shown is for illustrative purposes but the cumulative nature of the reward for satisfying a goal early has some interesting consequences. The weighting strategies are:

weight	*Strategy*					
	EC	UC	DG	ADG	UDG	DJ
w_1	2000	500	50	0	0	0
w_2	1000	500	100	0	0	0
w_3	500	500	200	200	1000	0
w_4	200	500	500	500	1000	0
w_5	100	500	1000	1000	1000	0
w_6	50	500	2000	2000	1000	1000
w_7	25	500	4000	4000	1000	1000
w_8	10	500	8000	8000	1000	1000

Table 6.1: Weighting Strategies

early credit (EC) The weights are monotonically decreasing with i. The notion is that satisfying goals early should be rewarded.

uniform credit (UC) All the weights are the same.

delayed gratification (DG) The weights are monotonically increasing. This captures the idea that early satisfaction of goals may not necessarily be a good thing.

advanced delayed gratification (ADG) The weights are monotonically increasing and no credit is given immediately for satisfying goals in the initial exchanges.

uniform delayed gratification (UDG) No credit is given immediately for satisfying goals in the initial exchanges and later weights are equal and positive.

destination judgement(DJ) Here, credit is given only towards the end of the protocol (i.e. after the sixth message).

6.4.6 Protocol Encoding

For the protocol tools the integer sequences defining messages (see section 6.4.1) are encoded either as integer arrays (in simulated annealing) or else as bit substrings of chromosomes (for genetic algorithms). Four bits were used to encode senders and receivers and six bits were used to encode indices of beliefs. It was found to be beneficial to have all senders and receivers at the beginning of the chromosome. [1] Thus, for a five-message protocol the first $5 \times 4 \times 2 = 40$ bits represented the five senders and receivers of the messages. With four beliefs per message the remaining $5 \times 4 \times 6 = 120$ bits represented the 20 belief components that make up the five messages. Other optimisation techniques can easily be incorporated in the framework.

6.5 From Assumptions to Goals

This section reports the results of applying the technique described above to the derivation of a three-party key distribution protocol. A set of initial assumptions is given and the technique is applied to derive abstract protocols meeting the stated goals. The experiments reported below serve as proof of concept. At first a good number of goals are given, including some that might be regarded as of assistance to the technique (but these will be removed later). I also wish to show that the business of protocol synthesis is quite a subtle one and we demonstrate just how important is the choice of cost function.

6.5.1 The Assumptions

Three parties participate in this key distribution protocol A, B and S. A and B both share keys with the server S. They maintain their own nonces that they believe to be fresh. The assumptions are:

$$
\begin{array}{lll}
A \mid\equiv A \overset{Kas}{\longleftrightarrow} S & A \mid\equiv Na & A \mid\equiv \sharp(Na) \\
A \mid\equiv S \mid\Rightarrow A \overset{Kab}{\longleftrightarrow} B & A \mid\equiv S \mid\Rightarrow B \mid\sim Nb & \\
B \mid\equiv B \overset{Kbs}{\longleftrightarrow} S & B \mid\equiv Nb & B \mid\equiv \sharp(Nb) \\
B \mid\equiv S \mid\Rightarrow A \overset{Kab}{\longleftrightarrow} B & B \mid\equiv S \mid\Rightarrow A \mid\sim Na & \\
S \mid\equiv A \overset{Kas}{\longleftrightarrow} S & S \mid\equiv B \overset{Kbs}{\longleftrightarrow} S & \\
S \mid\equiv A \overset{Kab}{\longleftrightarrow} B & &
\end{array}
$$

The assumptions are straightforward except perhaps for the common belief by A and B that S tells the truth about the other's uttering of nonces. This is addressed in Section 6.5.5.

[1] Precisely why this is so is unclear. The experiments reported here do not permit general deductions to be made. The adopted placing of senders and receivers may not be beneficial if alternative crossover regimes are used (two-point crossover is used here). An alternative set of protocol goals might also give different results. Future work might usefully address such issues.

6.5.2 The Goals

The first set of goals requires that at the end of the protocol run A and B must each believe that it possesses a good key Kab for session communication, that the other has made a statement to this effect, and that each believes the other believes the key is good.

$$\begin{array}{ll} A \mid\equiv A \overset{Kab}{\longleftrightarrow} B & B \mid\equiv A \overset{Kab}{\longleftrightarrow} B \\ A \mid\equiv B \mid\sim A \overset{Kab}{\longleftrightarrow} B & B \mid\equiv A \mid\sim A \overset{Kab}{\longleftrightarrow} B \\ A \mid\equiv B \mid\equiv A \overset{Kab}{\longleftrightarrow} B & B \mid\equiv A \mid\equiv A \overset{Kab}{\longleftrightarrow} B \end{array}$$

The search is limited to six messages. The reader can verify that establishing common (to A and B) first order belief in the key requires 4 messages. Furthermore, to establish the remaining four goals in the next two messages requires either A or B to be in possession of information about the other's nonce and this must have come via the server. This explains the inclusion as assumptions of $B \mid\equiv S \mid\!\!\Rightarrow A \mid\sim Na$ and $A \mid\equiv S \mid\!\!\Rightarrow B \mid\sim Nb$. The first, for example, allows B to receive a belief $A \mid\sim Na$ from S that he himself should now believe and hence be able to use in a message to A. Similarly for B.

6.5.3 Experimental Method

Both simulated annealing and genetic algorithms have been applied to this problem. In the descriptions of protocols that follow in the rest of this chapter, belief components that do not actually contribute to the attainment of the overall protocol goals have been excised. Similarly, redundant beliefs have also been removed (i.e. when the same belief is included twice or more in a message). Only the core security relevant protocol is presented. 'Junk beliefs' have been removed manually (though this could easily be automated).

Simulated Annealing Results

For simulated annealing a temperature cooling factor of 0.95, a cutoff of 150 iterations, 400 candidate moves within each inner loop and a maximum without accept of 50 iterations were used. Twenty runs of the algorithm were carried out for each fitness function strategy. The results are presented in Figure 6.2. Significantly, Destination Judgement (the most direct fitness function) is the worst performing fitness function. Uniform Credit performs reliably and efficiently. Since a run takes at most a few minutes, even a success rate of 0.55 is not a practical problem. Perhaps the most interesting thing to note about these results is the apparent practical robustness against choice of fitness function. This contrasts with the results from genetic algorithms which are presented below.

Strategy	Success Fraction	Function Evals (000s) Per Success
EC	1	43.8
UC	1	31
DG	0.85	35.2
ADG	0.75	49.9
UDG	0.95	27.6
DJ	0.55	82.9

Table 6.2: Simulated Annealing on First Problem (6 Goals) with Three Beliefs per Message

Genetic Algorithm Results

The technique was applied to the above problem with various combinations of weighting strategy, crossover and mutation probabilities. Crossover probabilities took values from the set 0.2, 0.4, 0.6, 0.8, 1.0, mutation probabilities took values from the set 0.005, 0.01, 0.015, 0.02, 0.025. Every combination of weighting strategy, crossover probability and mutation probability was tried (making 150 combinations). Each combination was tested by applying the algorithm 20 times to the problem and recording the results. A population size of 200 was used. This value is for illustrative purposes and no claim to optimality should be assumed.

The technique is not guaranteed to succeed and so an upper bound of 200 was imposed on the number of generations allowed before a run is terminated. This choice is motivated in part by the need to carry out very large numbers of runs (3000) to test the technique. A designer might require only a few runs. In practice a run will take only a few minutes (running on a 450MHz Pentium PC).

Table 6.3 shows the fraction of successful runs of the technique at each combination of crossover and mutation probability. The results indicate that Early Credit clearly performs poorly and Uniform Credit seems the most robust. The number of protocol evaluations per success (i.e. finding a protocol meeting all goals) for Uniform Credit is given in Table 6.4.

The results seem plausible. Consider the evolved protocol shown in Figure 6.3. Under the Early Credit scheme if the first two exchanges are between A and S as shown then one of the goals is met, ($A \models A \overset{Kab}{\longleftrightarrow} B$), and so incurs a reward. However, from this point onwards the choices are very restricted. To satisfy all the goals using only six messages the next two exchanges must be between B and S and similar to those shown. The third message must be from B to S and contain something B believes to be fresh. The fourth message must be from S to B and contain the response to the challenge in the third message and also provide the belief $A \overset{Kab}{\longleftrightarrow} B$. Two goals have now been established and A and B are

EC	0.005	0.01	0.015	0.02	0.025
0.2	0.8	1.0	0.8	0.25	0.15
0.4	0.85	0.9	0.65	0.1	0.15
0.6	0.75	0.85	0.4	0.2	0.05
0.8	0.95	0.75	0.3	0	0.05
1.0	0.9	0.7	0.4	0.05	0

UC	0.005	0.01	0.015	0.02	0.025
0.2	0.8	0.9	0.95	0.9	0.95
0.4	0.6	0.95	0.9	0.95	0.9
0.6	0.7	0.85	1	0.95	1
0.8	0.6	0.85	0.95	1	0.95
1	0.7	0.8	0.9	0.85	0.85

DG	0.005	0.01	0.015	0.02	0.025
0.2	0.5	0.8	0.85	1	0.75
0.4	0.75	0.8	0.9	0.95	0.8
0.6	0.65	0.65	0.9	0.9	0.6
0.8	0.45	0.9	0.8	0.85	0.55
1	0.6	0.8	0.8	0.85	0.35

ADG	0.005	0.01	0.015	0.02	0.025
0.2	0.3	0.8	0.85	0.85	0.95
0.4	0.55	0.75	0.9	1	0.7
0.6	0.6	0.65	0.8	0.9	0.6
0.8	0.5	0.8	0.7	0.9	0.7
1	0.6	0.7	0.8	0.75	0.45

UDG	0.005	0.01	0.015	0.02	0.025
0.2	0.45	0.7	1	0.9	0.55
0.4	0.6	0.95	1	0.85	0.55
0.6	0.65	0.65	0.9	0.65	0.5
0.8	0.6	0.75	0.95	0.75	0.35
1	0.6	0.85	0.9	0.4	0.25

DJ	0.005	0.01	0.015	0.02	0.025
0.2	0.25	0.5	0.5	0.6	0.85
0.4	0.25	0.35	0.65	0.7	0.85
0.6	0.25	0.6	0.55	0.85	0.85
0.8	0.2	0.45	0.6	0.45	0.8
1	0.45	0.35	0.4	0.7	0.7

Table 6.3: Success Fractions for Combinations of Crossover and Mutation Probabilities. Rows are indexed by crossover probability, columns indexed by mutation probability.

UC	0.005	0.01	0.015	0.02	0.025
0.2	24.6	$\boxed{13.3}$	16.5	18.1	18.5
0.4	43.4	15.0	17.4	15.5	19.7
0.6	27.6	19.4	12.1	18.9	17.9
0.8	33.1	18.8	20.4	18.1	20.0
1	25.3	25.3	20.2	20.8	26.3

Table 6.4: Protocol Evaluations (000's) per Success for Uniform Credit

$$\begin{array}{l}
(1)\ A \rightarrow S : \{Na\}_{Kas} \\
(2)\ S \rightarrow A : \{A \mid\!\sim Na, A \xleftrightarrow{Kab} B\}_{Kas} \\
(3)\ B \rightarrow S : \{Nb\}_{Kbs} \\
(4)\ S \rightarrow B : \{A \mid\!\sim Na, B \mid\!\sim Nb, A \xleftrightarrow{Kab} B\}_{Kbs} \\
(5)\ B \rightarrow A : \{A \mid\!\sim Na, Nb, A \xleftrightarrow{Kab} B\}_{Kab} \\
(6)\ A \rightarrow B : \{B \mid\!\sim Nb, A \xleftrightarrow{Kab} B\}_{Kab}
\end{array}$$

Figure 6.3: Protocol Generated During Experimentation

now in a position to communicate. If the protocol is now to meet all its goals the next two exchanges must be between A and B and cause the remaining goals to be achieved. This will require at least one of the principals to hold a belief that contains an element believed by the other to be fresh. Thus, the inclusion of $A \mid\!\sim Na$ in the fourth message is essential. The fifth message must come from B and include $A \mid\!\sim Na$. It must also include the statement of goodness of the key $A \xleftrightarrow{Kab} B$ and supply a suitable challenge Nb to A. The 6th message must therefore return the challenge and contain the belief $A \xleftrightarrow{Kab} B$. The point to note here is that after the first two messages the die is largely cast and the search must find messages (3)–(6), or very similar ones, from the decision space — the possibilities are very few. Early Credit favours this sort of initial message sequence.

Consider now the protocol shown in Figure 6.4, also found during experimentation. After two messages *no* goals have been met. However, the symmetry established allows numerous routes to success. For example, the roles of A and B can be swapped in (3)–(6), *mutatis mutandis*. Although the protocol in Figure 6.4 may appear strange it should be remembered that this is an abstract description. Implementationally, the first message could actually be sent from A to B (together with some helpfpul plaintext (invisible in BAN logic) saying who it was from, and B could then simply pass it on to S (together with the encrypted message (2) shown). Asymmetric protocols similar to the original one may be created,

$(1)\ A \rightarrow S : \{Na\}_{Kas}$
$(2)\ B \rightarrow S : \{Nb\}_{Kbs}$
$(3)\ S \rightarrow A : \{A \mid\!\sim Na, B \mid\!\sim Nb, A \overset{Kab}{\longleftrightarrow} B\}_{Kas}$
$(3)\ S \rightarrow B : \{B \mid\!\sim Nb, A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kbs}$
$(5)\ A \rightarrow B : \{B \mid\!\sim Nb, Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kab}$
$(6)\ B \rightarrow A : \{A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kab}$

Figure 6.4: Symmetric Protocol Generated During Experimentation

$(1)\ A \rightarrow S : \{Na\}_{Kas}$
$(2)\ B \rightarrow S : \{Nb\}_{Kbs}$
$(3)\ S \rightarrow A : \{A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kas}$
$(4)\ S \rightarrow B : \{B \mid\!\sim Nb, A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kbs}$
$(5)\ B \rightarrow A : \{A \mid\!\sim Na, Nb, A \overset{Kab}{\longleftrightarrow} B\}_{Kab}$
$(6)\ A \rightarrow B : \{B \mid\!\sim Nb, A \overset{Kab}{\longleftrightarrow} B\}_{Kab}$

Figure 6.5: Asymmetric Protocol Generated During Experimentation

for example the one in Figure 6.5.

Again the roles of A and B can also be swapped (*mutatis mutandis*). The greater design freedom is apparent and yet this freedom comes at the price of foregoing early rewards. Note that making *some* progress initially seems a good idea. For this problem it is essential that $A \mid\equiv A \overset{Kab}{\longleftrightarrow} B$ and $B \mid\equiv A \overset{Kab}{\longleftrightarrow} B$ be established first. A protocol that achieves these two goals in the first four messages is clearly better than one which achieves these only after six messages. The former may well be close to achieving all the goals, the latter certainly is not. This explains in part the poor performance of Destination Judgement which rewards these two protocols equally. Reward is based on achievement and not on potential (judgement of which requires insight).

The conclusion to be drawn is that the evaluation function matters. Designers will have to experiment with a number of different strategies or else a more sophisticated *adaptive* strategy will be needed (this is common in genetic algorithm frameworks). However, all the strategies achieved some success. For specifications with a greater range of possible protocols it would be interesting to note the differences in the protocols that emerge using different strategies. Investigation of the efficacy of various evaluation functions will form a major part of future research.

The genetic algorithm results indicate quite starkly the interaction between fitness function, crossover probabilities, mutation probabilities and success rate. For any fitness strategy it seems possible to derive GA parameter settings that give reasonable results but these settings need to be determined on a per fitness function basis. There is an interesting contrast with the simulated annealing results (which were pretty robust against choice of fitness function). In particular, Early Credit is highly reliable for SA but poor for GA. It is however, not very efficient even for SA. Destination Judgement is poor for both. However, when GAs work they would appear to do so efficiently (as shown in Table 6.4).

6.5.4 Reduced Goals

It could be argued that two of the required six goals, namely $A \mid\equiv B \mid\sim A \overset{Kab}{\longleftrightarrow} B$ and $B \mid\equiv A \mid\sim A \overset{Kab}{\longleftrightarrow} B$ are somewhat artificial and really serve only to help the search process. Although $A \mid\equiv B \mid\sim A \overset{Kab}{\longleftrightarrow} B$ and $A \mid\equiv B \mid\equiv A \overset{Kab}{\longleftrightarrow} B$ are often satisfied at the same time, the inclusion of the first as a *goal* causes a greater reward to be given. In addition, it would be possible to achieve $A \mid\equiv B \mid\sim A \overset{Kab}{\longleftrightarrow} B$ for example without $A \mid\equiv B \mid\equiv A \overset{Kab}{\longleftrightarrow} B$. The message that caused the first goal to be met could cause the second to be met too if the search process were to augment the message with a suitable freshness indication. It is part way there already and rewarding this is likely to help the technique climb towards a solution. These two intermediate goals are now removed leaving the required set of goals as:

$$\begin{array}{ll} A \mid\equiv A \overset{Kab}{\longleftrightarrow} B & B \mid\equiv A \overset{Kab}{\longleftrightarrow} B \\ A \mid\equiv B \mid\equiv A \overset{Kab}{\longleftrightarrow} B & B \mid\equiv A \mid\equiv A \overset{Kab}{\longleftrightarrow} B \end{array}$$

Simulated Annealing Results

For simulated annealing, with parameters as before, 20 runs of the algorithm were carried out for each fitness function strategy. The results are presented in Table 6.5 Again simulated annealing is robust to changes in the fitness function (though destination judgement is clearly still poor).

Genetic Algorithm Results

For the same population size (200) and the Uniform Credit strategy this new problem was subjected to the technique (20 runs at each combination of crossover and mutation probabilities as before). The results are given in Tables 6.6 and 6.7. As expected, the results degrade but a fair degree of success is still obtained. The intermediate goals present in the previous problem clearly helped. An interest-

Strategy	Success Fraction	Function Evals (000s) Per Success
EC	1	47.5
UC	1	34.3
DG	0.9	38.6
ADG	1	30.3
UDG	1	30.5
DJ	0.55	87.7

Table 6.5: Simulated Annealing on Reduced Goals Problem (4 Goals) with Four Beliefs per Message

UC	0.005	0.01	0.015	0.02	0.025
0.2	0.55	0.7	0.95	0.55	0.25
0.4	0.65	0.95	0.8	0.45	0.3
0.6	0.75	0.85	0.55	0.45	0.1
0.8	0.8	0.7	0.35	0.35	0.2
1.0	0.55	0.5	0.25	0.2	0.15

Table 6.6: Success Fraction for Uniform Strategy on Reduced Goals

ing avenue to investigate would be the tool assisted provision of such hints, given stated goals.

It can be seen that simulated annealing here is far more robust to changes in the fitness function than genetic search (though destination judgement is clearly still very poor). Indeed, it would seem that simulated annealing markedly outperforms the genetic algorithm here unless very particular parametric choices are made.

UC	0.005	0.01	0.015	0.02	0.025
0.2	47	36.9	21.1	51.5	149
0.4	43.4	20.1	29.4	70.8	119.7
0.6	30.2	26.9	55.1	73.2	380.8
0.8	33.2	35.2	102.2	97.6	189.7
1.0	56.3	60.6	145.4	179.6	245.7

Table 6.7: Protocol Evaluations per Success (000's) for Uniform Strategy on Reduced Goals

Strategy	Success Fraction	Function Evals (000s) Per Success
EC	0.9	59.1
UC	0.9	43
DG	0.75	43.5
ADG	0.9	33.2
UDG	1	30.7
DJ	0.60	81.3

Table 6.8: Simulated Annealing on Reduced Assumptions Problem (also 4 Goals) with Four Beliefs per Message

6.5.5 Reduction of Assumptions

There are two rather strange assumptions in the precondition, namely

$$A \mid\equiv S \mid\Rightarrow B \mid\sim Nb$$

$$B \mid\equiv S \mid\Rightarrow A \mid\sim Na$$

These were introduced to allow communication of freshness indicators between A and B via the server S, allowing a simple belief like $A \mid\equiv B \mid\sim Nb$ to be derived for A and so $B \mid\sim Nb$ can be included in messages by A. The need for this arises due to the restriction on the messages that can be communicated in the current tools. If A were able to send a belief $S \mid\sim B \mid\sim Nb$ to B the assumptions would be unnecessary. It is still possible to remove these assumptions but an extra interaction between A and B is required (i.e. the final three interactions will be between A and B). Accordingly, the two assumptions were removed and the problem re-attempted.

Simulated Annealing Results

The results (for 20 runs, parameters as before) are given in Table 6.8. Interestingly, although the problem is logically harder, the efficiency and success rates on some strategies are actually better. This may seem paradoxical but missing out the jurisdiction assumptions may actually cause fewer beliefs to be deduced on receiving a message (and indeed those very assumptions are no longer available for inclusion in a message as beliefs themselves). This may reduce the number of feasible combinations of message contents. Again robustness and efficiency are key features.

$$
\begin{array}{l}
(1)\ B \rightarrow S : \{Nb\}_{Kbs} \\
(2)\ S \rightarrow B : \{B \mid\!\sim Nb, A \overset{Kab}{\longleftrightarrow} B\}_{Kbs} \\
(3)\ A \rightarrow S : \{Na\}_{Kas} \\
(4)\ S \rightarrow A : \{A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kas} \\
(5)\ B \rightarrow A : \{Nb\}_{Kab} \\
(6)\ A \rightarrow B : \{B \mid\!\sim Nb, Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kab} \\
(7)\ A \rightarrow B : \{A \models Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kab}
\end{array}
$$

Figure 6.6: Typical Seven-message Solution Found

Genetic Algorithms Results

The problem was attacked using the same mutation and crossover parameter values, the Uniform Credit strategy and with a population of gene strings representing seven message protocols. The results were disappointing and the technique failed to find a solution more often than not. Furthermore, examination of the evolution process and the best candidate protocols found during the searches showed that the search often got very close to solving the problem and often quite early in the search (e.g. less than thirty generations) but found great difficulty getting the final messages right. Typically the best solution found satisfied three goals but had a final message involving the server. In a sense, the technique was unable to 'hill-climb' its way to a solution.

A modification of the technique was attempted. The population size was increased to 400. In addition, after 60 generations the population was replaced entirely with copies of the best solution found so far. The bits of the strings corresponding to message (7) were randomised (i.e. the bits representing the sending and receiving principals and also the message contents). This reseeding of the population was repeated every subsequent 20 generations. The idea is that the population starts with identical messages 1 through 6 and randomisation is used to help move in the right direction. Of course, all parts of the strings may change by mutation subsequently (and so later by crossover too). Eighteen out of twenty searches found a solution. A typical solution found is given in Figure 6.6: All eighteen solutions found have essentially the same 'shape', though the particular nonces supplied may differ, for example $1.B \rightarrow S : \{\sharp(Nb)\}_{Kbs}$, or the protocols simply reversed the roles of A and B. None corresponded to having the above messages appear in the the order $(1)(3)(2)(4)$; this is not surprising since Uniform Credit does not favour this option. The above rectification seems, however, very *ad hoc*.

That the original approach should be unsuccessful is interesting. The search rapidly established the first two beliefs $A \models A \overset{Kab}{\longleftrightarrow} B$ and $B \models A \overset{Kab}{\longleftrightarrow} B$ after 4 messages but seemed to find difficulty in reliably establishing the two remaining beliefs. A third belief may be established after 6 or 7 messages. To satisfy a third belief seems to leave too much to chance. You need to get messages (5) and (6) right and there is no way to hill-climb to this situation.

Even when a third belief is achieved, the cumulative number of achieved goals during the protocols is given by $(0, 1, 1, 2, 2, 3, 3)$ and $(0, 1, 1, 2, 2, 2, 3)$ giving values of 6000 and 5500 under the Uniform Credit scheme. This is not that much in excess of the 5000 that would arise when only two goals are met $(0, 1, 1, 2, 2, 2, 2)$ and so a protocol with the most helpful achivement profile of $(0, 1, 1, 2, 2, 3, 3)$ may not even survive selection. With cumulative fitness functions later achievement seems inadequately rewarded. When a better candidate arises it must flourish under selection and survive the worst excesses of mutation and crossover. Also, protocols with an achievement profile of $(0, 1, 1, 2, 2, 2, 3)$ tend to cause the search process to be deceived (they look appealing in terms of achievement but the technique has no way of knowing that it is heading for a cul-de-sac).

6.6 Reducing the Number of Beliefs per Component

The experiments described above allowed protocols to include ‘junk’ beliefs, beliefs that did not contribute to the logical attainment of the security goals in the final protocol. That is, it is possible simply to erase certain beliefs from the messages of the final protocol and still achieve the stated security goals. From the point of view of *solution mechanism,* however, all belief components are relevant since the indexing of beliefs in the receiver’s state will be affected by the receipt of ‘irrelevant’ beliefs.

Use of the word ‘irrelevant’ here would seem rather narrow. In practice it may be impossible to determine which beliefs are irrelevant before the end of the protocol. Thus, the irrelevance of beliefs in earlier messages may be determined by the content of subsequent messages. However, the more beliefs included in messages the richer the information receiving principals obtain. Thus, information rich messages actually create greater potential for achieving goals (giving the receiver more information cannot reduce the numbers of ways the goals may be met). However, this is viewing things from the abstract logic point of view. Large messages may carry a cost in terms of the solution mechanism. Goals are often satisfied by having appropriate components in various messages. Larger messages will lead to larger belief states and this may affect the ability of the search process to find appropriate combinations of beliefs to satisfy the goals.

The number of beliefs components per message is now reduced. The above experiments have been repeated but allowing first three beliefs per component and

Strategy	Success Fraction	Function Evals Per Success (000s)	Success Fraction	Function Evals Per Success (000s)	Success Fraction	Function Evals Per Success (000s)
EC	0.95	49.7	0.85	65.3	0.65	87.2
UC	0.95	39	0.95	40	0.95	42.7
DG	0.55	78.2	0.7	56.6	0.75	53.1
ADG	0.55	78.6	0.85	42.9	0.9	39.9
UDG	0.9	35.2	1	34.5	1.0	32.7
DJ	0.75	61.4	0.45	117.1	0.60	82.8
Three Beliefs	Original Problem (6 goals)		Reduced Goals (4)		Reduced Goals (4) and Reduced Assumptions	

Strategy	Success Fraction	Function Evals Per Success (000s)	Success Fraction	Function Evals Per Success (000s)	Success Fraction	Function Evals Per Success (000s)
EC	0.55	131.5	0.35	214.5	0.35	218.4
UC	0.45	144.7	0.55	122.3	0.7	90.5
DG	0.30	255.7	0.5	117.2	0.45	148.4
ADG	0.25	278.9	0.65	90.3	0.20	365.4
UDG	0.5	121.1	0.5	139.6	0.55	117.3
DJ	0.20	373.7	0.20	371.2	0.35	204.1
Two Beliefs	Original Problem (6 goals)		Reduced Goals (4)		Reduced Goals (4) and Reduced Assumptions	

Table 6.9: Reduced Variants Results

then only two beliefs per component. In the case of two beliefs per component it is necessary to extend the number of messages to eight (i.e. this is the least number of messages possible to satisfy the goals).These additional experiments have been carried out these additional experiments for annealing only. In [22] I showed how the reliability of the GAs could be increased by *increasing* the number of belief components per message. This is not without significant computational cost as well as necessitating a rather cumbersome post-processing to extract those fragments of the generated protocol relevant to attaining the specified security goals.

6.6.1 Reducing the Number of Beliefs per Component To Three

The results for three beliefs per message are given in the upper half of Table 6.9. For the original problem, in all cases except destination judgement (DJ), both the success fraction and efficiency are decreased. For the reduced goals problem all fitness functions give rise to poorer results than for four component messages. Rather strangely, the harder reduced assumptions problem does not lead to a significant reduction in performance.

6.6.2 Reducing the Number of Beliefs per Component To Two

Here at most two belief components per message are allowed. This gives rise to a highly constrained problem (with solutions largely being variations on the same

$$
\begin{array}{l}
(1)\ A \rightarrow S : \{Na\}_{Kas} \\
(2)\ S \rightarrow A : \{A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kas} \\
(3)\ B \rightarrow S : \{Nb\}_{Kbs} \\
(4)\ S \rightarrow B : \{B \mid\!\sim Nb, A \overset{Kab}{\longleftrightarrow} B\}_{Kbs} \\
(5)\ B \rightarrow A : \{Nb\}_{Kab} \\
(6)\ A \rightarrow B : \{B \mid\!\sim Nb, A \overset{Kab}{\longleftrightarrow} B\}_{Kab} \\
(7)\ A \rightarrow B : \{Na\}_{Kab} \\
(8)\ B \rightarrow A : \{A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kab}
\end{array}
$$

Figure 6.7: Protocol Generated During Experimentation

$$
\begin{array}{l}
(1)\ A \rightarrow S : \{Na\}_{Kas} \\
(2)\ S \rightarrow A : \{A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kas} \\
(3)\ B \rightarrow S : \{Nb\}_{Kbs} \\
(4)\ S \rightarrow B : \{B \mid\!\sim Nb, A \overset{Kab}{\longleftrightarrow} B\}_{Kbs} \\
(5)\ S \rightarrow A : \{B \mid\!\sim Nb, A \mid\!\sim Na\}_{Kas} \\
(6)\ A \rightarrow B : \{B \mid\!\sim Nb, A \overset{Kab}{\longleftrightarrow} B\}_{Kab} \\
(7)\ A \rightarrow B : \{Na, B \mid\!\sim Nb\}_{Kab} \\
(8)\ B \rightarrow A : \{A \mid\!\sim Na, A \overset{Kab}{\longleftrightarrow} B\}_{Kab}
\end{array}
$$

Figure 6.8: Protocol Generated During Experimentation

theme). First note that to exchange messages A and B must first obtain the key Kab. This will require each of A and B to receive a message that contains the key Kab together with an appropriate freshness indicator (which it must have supplied to S in a previous message). Thus, the first four messages must be pairwise exchanges between A and S and also between B and S. The final four messages are communications between A and B in which each supplies a nonce to the other which is returned together with an assertion that the key Kab is good. A typical protocol is shown in Figure 6.7.

There is another general way in which the goals can be satisfied. A nonce indication is passed between the two communicants in a message from the server S. This is shown in the protocol in Figure 6.8. Variations on this theme were very much in the minority amongst successful protocols evolved.

The results for two beliefs per message are given in the lower half of Table 6.9.

The results seem markedly worse than for two component messages. They show that the relentless pursuit of ‘efficiency’ is not without its costs. Indeed, it would appear that embracing the greater logical potentials of component redundancy and post-processing away irrelevant beliefs in the final protocol is far more efficient than constraining oneself to be more efficient from the start. In a sense, when redundancy is allowed the simulated annealing has features of a genetic algorithm, but without any problems with respect to hill-climbing.

6.7 Efficiency

The efficiency of the protocol has not been addressed *directly* in this chapter, but it has not been ignored. Indeed, a bias towards short protocols has actually been built in to most of the weighting strategies of Table 6.1. The cumulative nature of the reward strategies (except Destination Judgment) means that shorter protocols will be favoured. For a protocol with 6 messages a goal achieved after one message will contribute $\sum_{i=1}^{6} w_i$ to the eventual fitness, a goal achieved with the last message contributes only w_6. The earlier a goal is achieved the more reward it gets. As a consequence a shorter protocol that meets all the goals will generally be favoured over a longer one.

This notion of efficiency is rather crude. A more sophisticated approach would recognise that server interactions are to be reduced if possible (since the server may provide services to many thousands of principals), and that actually implementing the sending of different types of belief will incur different computation expense when implemented. However, a suitably parametrised fitness function would seem an ideal approach to handle these issues.

6.8 Discussion

Automated refinement is a very recent innovation. The work reported here addresses a *modern-day* cryptological research issue of significant practical importance. The results have indicated that the mechanisms of evolutionary search and simulated annealing may plausibly be used to generate abstract protocols from end-to-end specifications. Furthermore, the resulting protocols are provably correct vis à vis the BAN logic — though additional security analysis would be essential, showing that a concrete refinement does not have type flaws, that a sender can recognise its own messages etc. The tools can be used to search the design space and provide input to human designers who would derive a concrete refinement of the chosen abstract protocol. An interesting by-product of the approach is that logging the application of the inference rules during execution provides a proof script of the protocol's correctness. The problems of logical correctness and efficiency *are handled by the same solution mechanism* (optimisation). This framework has the potential to be extended to handle other criteria, as is argued below.

The strengths of the work of Song and Perrig and the strengths of the work presented here would seem complementary. The model checking approaches of Song and Perrig requires significant computation time to carry out a search amongst all possibilities. It also incorporates a sophisticated intruder model. The quasi-enumerative nature of model checking also provides guarantees that certain solutions are optimal (i.e. no shorter protocol exists satisfying the requirements). However, the potential for state space explosion is significant. Many of the protocols generated by the model checking search technique have been very small (3 or 4 messages). The work described in this chapter has demonstrated the ability to generate protocols with 8 messages (and other experiments have in fact generated larger protocols). The metaheuristic approach seems highly scalable. It exchanges guarantees of optimality for computational tractability (the typical story for metaheuristics). It could perhaps be used as a front end to generate candidates for further analysis.

6.9 Evidence for the Thesis

6.9.1 A Significant Increase in Power?

The technique has shown distinct strengths. In particular it is able rapidly (a few minutes at most) to evolve protocols of considerable size. In both respects it easily outperforms the recent design by model checking approaches. It would appear that the technique is highly competitive in some respects, but the alternative approaches are better in others. An obvious way forward would be to use heuristic search to generate candidate protocols and then use model checking to check for any additional security flaws. The topic is very much leading-edge research and the approach described here can achieve things that the major (and, the author believes, sole) competition cannot.

Current use of heuristic search tends to be very limited in scope. Various lines of application have been developed fairly early and subsequent research seems to have followed them. The work reported here is a deliberate attempt to significantly widen the scope of application of heuristic search.

6.9.2 Toolkit Contributions

The principal toolkit contribution is simply a demonstration that automated refinement of security protocols with proofs of correctness is possible with significant potential for development.

6.10 Issues Arising

Specific issues arising from the work reported in this Chapter are:

1. Problems with protocol abstraction. Most papers on formal analysis of protocols talk about abstraction as if it were a good thing! Of course, abstraction is a very useful tool but carries with it certain consequences. As noted in Chapter 5 the implementation is important as well as the mathematical function. This holds for protocols too. It would be possible to refine the protocol to an implementation in a way that compromised security horribly (e.g. by allowing timing or other covert channels via the use of specific cryptographic means etc). These are outside the scope of the formal BAN logic model. This complaint is shared with all other formal analysis methods. True security would require confirmation by other means that the intent of the implemented protocol has not been compromised.

2. Implementation decisions. The precise scheme used to choose the recipient of a message is biased (due to the way a receiver is created when initially the decoding has sender and receiver identical).[2]

3. Limitations of the logic. A limited subset of the BAN belief logic. This limits the sorts of protocols that can be specified and the means by which specifications can be refined. Other elements would need to be incorporated in an industrially useful experimentation tool.

4. Problem formulation. The work has benefited greatly from certain early modeling decisions. The interpretation of arbitrary integer sequences as valid protocols eases the optimisation task significantly.

5. The problem lends itself to heuristic search. A protocol is a sequence of interactions in which information is exchanged to achieve a number of goals. A protocol is essentially a program (whether one considers security or not). Messages provide the receiver with new information which is then available for use. Earlier messages will generally establish some goals to provide the context in which later messages can achieve their goals. This notion that some goals must naturally be achieved before others forms the basis for the guidance given by many of the cost functions used. In the simplest cases hill-climbing of some form will suffice to reach the full set of goals. In other cases, the ability to escape local optima is essential.

[2] An observation made by an anonymous referee for the accepted submission for Information and Software Technology Special Issue [24]

6. The results show that the choice of cost function distinctly affects the results obtained. A family of useful functions has been created and examined. These were motivated in part by the desire to promote efficiency (i.e. short protocols). Other families may well be possible.

7. In all the protocol described in this chapter there were a number of goals. What happens if there is a single goal? The obvious answer is that the method will not work. There needs to be some notion of guidance provided to the search, but a single goal is either met or it is not. There is no notion of 'half a goal' being met. However, it might be possible to generate preliminary goals based on past experience. Searches could be made using a wide range of intermediate goals.

6.11 Open Problems

Several avenues for future work can be identified:

1. Efficiency: integrate efficiency considerations into the search process itself.

2. Avoidance of overloading: encourage reduced interactions with particular principals, most typically key servers.

3. Probabilistic belief inference. There is always some risk associated with statements about the real world and various probabilistic belief logics have been developed, e.g. [12]. The designer may wish to develop a protocol that maximises the probabilities of particular goals being satisfied.

4. Redundancy elimination. The current approach allows a belief to be incorporated in a message more than once. This seems wasteful. A more sophisticated approach might use messages without redundancy and allow different numbers of beliefs (the current use of the 'null' belief to achieve this seems a little clumsy). Alternatively, redundancy could be discouraged by a suitable choice of fitness function.

5. Further Representation Issues (for Genetic Algorithms). The current tool set places the senders and receivers at the beginning on the chromosomes. Further experimentation is needed to determine whether this is a good choice more generally.

6. Increasing design flexibility. The restriction that only simple beliefs and single operator beliefs may be communicated in messages can usefully be dropped allowing a richer set of beliefs to be communicated.

7. Better optimisation. Adopting a much more flexible and advanced genetic algorithms framework — one that works with a natural encoding of the problem rather than a bit string representation. In addition, the use of a Multi-Objective Genetic Algorithm (MOGA) framework would seem ideal for further experimentation.

8. Further testing. The method described should be tested using a much wider range of protocol specifications.

9. Negative testing. Can optimisation-based approaches be used to find *attacks* on protocols?[3]

6.12 Acknowledgements

The work reported in this chapter work was supported in part by the UK Defence Evaluation Research Agency (DERA) as part of a wider investigation into protocol design and analysis techniques. Thanks go to DERA for this support. Thanks also go to Peter Ryan and Irfan Zakiuddin for their support during the investigation. Elements of the review work [20, 21] have been published as have elements of the heuristic search results described in this Chapter [24, 22]. Thanks must go to the anonymous referees for their helpful and very detailed reviews.

[3] A question posed to the author by Jonathan Millen of SRI.

Chapter 7

Evaluation and Conclusions

7.1 The Hypothesis

This research reported in the previous chapters provides evidence in support of the following proposition: The hypothesis is stated below:

> **The power of metaheuristic search as a tool for modern-day cryptological research is significantly greater than currently evidenced in publicly available literature.**

Below the achievements of each technical chapter in this thesis are identified and assessed. The text below also indicates novel aspects of the work performed.

7.2 Evaluation

7.2.1 Evolving Boolean Functions and Correlation Immunity

The work of Chapters 3 and 4 has exhibited considerable originality and achievement. Metaheuristic search has been used to generate functions with hitherto undemonstrated characteristics (counter-examples to conjectures on autocorrelation and sums-of-squares, PC(2) functions meeting the 'trivial' bound on algebraic degree). These results are of immediate interest to researchers in Boolean functions. The approach has considerable potential to act as a rapid and efficient mechanism for gaining increased confidence in private conjectures. The best nonlinearity and autocorrelation values reported by previous optimisation researchers have been exceeded, often simultaneously. The work extends naturally to S-boxes. The results of previous optimisation work on injective and also bijective S-boxes have been improved on, though it seems clear that theoretical construction remains significantly better with respect to nonlinearity.

Previous optimisation-based work on correlation immunity attained only $CI(1)$ functions and produced no functions with optimal profiles. In this thesis Siegenthaler optimal functions with highest possible nonlinearity values have been evolved (for small numbers of inputs). The techniques have been used to evolve several functions that have been demonstrated only very recently by theoreticians.

Change of basis is clearly a powerful tool. The original suggestion to investigate simple linear change of basis came from a leading Boolean functions researcher (Dr Subhamoy Maitra) who had made use of such transformations in his work. The variations on a theme that followed are the author's own. An annealing approach to change of basis to obtain high order properties would seem original and useful.

For theoreticians, working with the Walsh-Hadamard spectrum is pretty much second nature but its manipulation in the manner of the ABF-1 and ABF-2 techniques seems original. The notion of 'almost a boolean function' is a simple concept that enables some very difficult functions to be obtained. As far as the author is aware, no optimisation work has ever generated bent functions before.

The limitations of the techniques become apparent when attempts are made to generate functions with nine variables and above. A limited family of cost functions has been considered. No attempt has been made to *design* cost functions or their particular parameter values — greater theoretical insight should now be brought to bear. There would also seem to be an obvious need to extend the criteria considered. Very little work has been performed on propagation characteristics, although this did produce something new — the PC(2) function on 6 inputs with degree 5. Similarly other S-box criteria could easily be addressed (e.g. criteria more directly related to differential cryptanalysis). Equally, attempting to evolve large S-boxes (e.g. 8 by 32 S-boxes) would seem the obvious next step to take.

Overall, it seems reasonable to claim that considerable novelty has been exhibited in these chapters; there are several new ways of approaching the evolution of desirable Boolean functions. In terms of effectiveness, previous optimisation results have been improved on. The correlation immunity results for functions with eight inputs or fewer, have matched those that theoreticians have been able to demonstrate. In a small number of cases, the properties of functions evolved are better than any demonstrated by other means. The ease with which additional uses were readily found for techniques initially motivated only by achieving high nonlinearity and specific degrees of correlation immunity serves to emphasise the flexibility of the metaheuristic approach. It would seem reasonable to claim that an original and a competitive contribution has been made.

7.2.2 Perceptron and Permuted Perceptron Problems

The results reported in Chapter 5 show that all sizes ($(101, 117)$, $(131, 147)$ and $(151, 167)$) of PPP scheme suggested by Pointcheval are susceptible (on occasion) to annealing-based attacks. Knudsen and Meier [65] have previously shown that (101,117) instances are insecure. Perceptron Problem instances of hugely greater size than anything previously considered feasible are also shown to be susceptible. The power of previous annealing-based attacks has been significantly increased.

More important, however, are the concepts of *problem warping* and *timing channel*. Though similar 'side-channels' are now well-known methods of attack on cryptosystems, no search-based cryptanalysis analogues have been found in the literature (though dynamic profiling of the search is a known metaheuristic concept). The notion of analysis side-channels is novel and potentially very powerful; the author knows of no cryptosystem that has been designed to be secure against such attacks.

Perhaps the most important observation *in the whole thesis* is that cryptosystems would seem ideal candidates for *profiling*. It is this notion that unifies the problem warping and the timing channel ideas. Every annealing run achieves *something* and does so *in some way*. The real issue is understanding how the computational dynamics and final results of search algorithm runs relate to what we actually want to find. This would seem to be an exercise in profiling (though a theoretical approach is not precluded). Finding such relationships may well prove to be very difficult. Interpreting the results of annealing-based searches may begin to look like an exercise in cryptanalysis. Cryptanalysts, however, have a long history of doing cryptanalysis.

The results show that using the schemes is unsafe but fall short of providing a repeatable and reliable means of breaking specific instances. The overriding weakness is that the side-channels have only been demonstrated on a single problem family (Perceptron Problem variants). There would seem to be a pressing need to demonstrate the efficacy of these concepts on schemes based on other NP-complete problem families.

Overall, the notion of analysis side channels is original and previous optimisation based results (which are the best results to date) have been improved on. It seems reasonable to claim that an original and competitive contribution has been made.

7.2.3 The Evolution of Security Protocols

Security protocol engineering is one of the most active areas of security. Automated security protocol synthesis has only recently emerged; there would appear to be no papers on the topic prior to 2000. The only two techniques in the literature (as far as the author is aware) are the metaheuristic search approach of Chapter 6 [22, 24] and the model checking approach of Song and Perrig [117, 118].

The approach reported in this thesis has significant strengths. In particular, it finds protocols satisfying a specification in at most a few minutes and is able to work with very large protocols (some specifications could not be refined to fewer than 8 messages). This contrasts markedly with the hours of computation time required by model checking approaches to generate even quite small protocols (e.g. 3 or 4 messages).

However, the current toolset implements only a small subset of BAN logic (public key encryption is not currently handled, for example) and, consequently, the model checking work of Song and Perrig has a richer design space. Their approach incorporates a sophisticated attack model too. A belief logic approach is only as powerful as the logic it implements. If the logic misses certain flaws, or else makes particular assumptions, the user must augment any automated designs the technique produces with additional checks to ensure adequate security. The approach currently produces only an *abstract* refinement, not an implementation. Adding a code generation stage would enable rapid experimentation and complete the automated design path.

The work reported in Chapter 6 establishes 'proof of concept'. The metaheuristic evolution of security protocols is, perhaps, a surprising idea but it has not produced any *surprising protocols*. The examples reported in this thesis are really staged exploratory tests. The approach has obvious potential for extension but requires significant further development and experimentation before a true assessment of its merits can be made. It also adds a new twist to the long and controversial life of BAN logic.

Overall, the technique shows promise. It clearly out-performs rival approaches with respect to the size of protocols that can be generated and also the speed with which they are generated, but suffers in terms of restrictiveness of the design space and the power of the underlying logic. It would seem reasonable to claim that a 'competitive' contribution has been made in this new area. A claim to originality is easier — there are only two approaches at present and the metaheuristic search one is radically different from Song and Perrig's model checking approach!

7.2.4 A Significant Increase in Power?

The research reported in this thesis has addressed a number of problems. Only problems of modern day cryptology have been addressed — a very deliberate choice. The thesis would appear very unusual in this respect. Boolean function work is a well-established topic in cryptology, crypto-schemes based on instances of NP-complete problems might plausibly be described as 'home ground' and the metaheuristic evolution of security protocols with proofs of their own correctness is a very significant leap in the level of abstraction at which metaheuristic search has been applied in cryptology.

The results reported in this thesis have improved on the published results of search-based cryptological research. In addition, results have been generated of genuine interest to professional cryptological researchers. In some cases, results have been demonstrated that improve on those of any applied techniques. The achievements of the research reported in this thesis have been summarised in Section 7.2. Those achievements allow a reasonable claim that *the power of metaheuristic search as a tool for modern-day cryptological research is significantly greater than currently evidenced in publicly available literature*.

7.3 Optimisation and Sophistication

Most of the research has been carried out using a 'vanilla' simulated annealing — the simplest variant of, perhaps, the simplest metaheuristic search technique. A fairly basic genetic algorithm was used in Chapter 6. It would be fair to say that little sophistication has been deployed with respect to optimisation (though the motivation behind thermo-statistical persistency [16] has been used in Chapter 5). More sophisticated optimisation approaches could and should now be brought to bear. As indicated in Chapter 1 the results here are offered as targets.

7.4 Observations for Adventure

Research should be something of an adventure. There are many 'Open Questions' identified in the specific chapters of this thesis. Below is an attempt to summarise very briefly, and hopefully in an entertaining style, some exhortations on how to engage in exciting metaheuristic search-based work in cryptology. Some of these are justified by the research reported here. Some are simply inspired by it.

- **Watch It!** Exploit the computational dynamics of search. The author knows of no cryptosystem designed to withstand an attack based on analysis of a simulated-annealing run in action.

- **Achieve Less More Often!** There seems to be a general opinion that metaheuristic search will never be able to evolve a secret key for a modern-day crypto-algorithm. This is probably true, but it seems to assume that the only way to break a cryptosystem is to evolve a key. Can metaheuristic search be used to evolve *approximations* that are better than currently used? Can many but diverse approximations be evolved and combined? Being less ambitious very many times may be a powerful idea.

- **Measure Everything!** [1] Structures abound in cryptology and there may be very subtle interactions and relationships between properties. Be prepared to take advantage of these relationships. Optimisation results may indeed suggest unusual relationships. If you do not look you will not find them.

- **Profile it!** Every cost function achieves something. The issue is 'What?' Use of certain cost functions may have unanticipated consequences (as has been seen to considerable effect in this thesis). What patterns are there in the results? Can they be exploited?

- **Describe it!** Patterns or structure in the results may be present but the analyst may be unable to see them. Can techniques such as genetic programming be used to evolve descriptions of the results?

- **Warp it!** The best cost functions are those that get you what you want. Be flexible in your choice of cost function families. Deviate from the standard or obvious cost functions. Can 'warped' functions be used? Can approximation families be used? We have seen the use of polynomial-based approximations. What others are there?

- **Embrace Local Optima!** The world seems grossly prejudiced against local optima! One might be tempted to believe that the only good local optimum is a global optimum. This seems extreme. As argued in Chapter 5 local optima may better be regarded as *sources of information* and not failures. Stop worrying and learn to love local optima! Just how far can this notion be pushed?

[1] This particular phrase was suggested to the author by Dr William Millan.

Appendix A

Supporting Material

A.1 Example (8,0,6,116,24) Function With Walsh and AC Zeroes Rank of 8

Here is the support for a (8,0,6,116,24) derived using the NCT method. It has Walsh and AC zeroes of rank 8. These provide linear transformation bases to give CI(1) and PC(1) transformed functions.

$1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1$
$0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0$
$1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0$
$0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0$
$0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0$
$0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0$
$0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0$
$1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1$

$a53a20176ca6cbd8$
$97f5a8743035cda4$
$7fc5ace26bc8ef4e$
$4030ad66929c0ebb$

$WalshZeroes:$
$0, 7, 8, 21, 24, 39, 44, 55, 61, 9397, 114, 116, 119, 141$
$156, 166, 249$

$WalshTransformationBasis:$
$7, 39, 55, 93, 97, 114, 116, 156$

$ACZeroes:$
$1, 2, 8, 9, 10, 22, 24, 32, 33, 47, 58, 62, 64, 66, 68, 84$
$85, 93, 95, 99, 107, 108, 131, 143, 149, 153, 154, 159$
$162, 166, 169, 177, 178, 179, 186, 190, 192, 193, 195$
$207, 208, 211, 212, 217, 222, 225, 229, 236, 239, 240$
$241, 245, 247, 251$

$ACTransformationBasis: \ 32, 62, 93, 166, 207, 211, 236, 245$

$nonLin: \ 116$
$ACorrel: \ 24$
$AlgebraicDegree: \ 6$

A.1.1 Function Transformed to (8,1,6,116,24)

$1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1$
$0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1$
$1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0$
$1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1$
$1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1$
$1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0$
$0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1$
$0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1$

$c7d185111af4adfd$
$c36666da964280f9$
$c93ab2558d28cd62$
$1fd63a0b6a8fb531$

$nonLin: 116$
$ACorrel: 24$
$AlgebraicDegree: 6$

A.1.2 Function Transformed to (8,0,6,116,24) with PC(1)

$1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1$
$1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1$
$1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0$
$0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1$
$1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0$
$1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0$
$1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1$
$0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1$

$9215f91fa524ff81$
$ab12337e5b7d328d$
$bba8c1b2e0241968$
$9e6cf8e1372742c5$

$nonLin: 116$
$ACorrel: 24$
$AlgebraicDegree: 6$

A.2 CI Direct Method

A.2.1 Successes Achieved for CI Direct Method

	1	2	3	4	5	6	7	8	9
5	100	44	58						
6	100	44	34	96					
7	98	6	11	26	98				
8	100	9	0	2	24	93			
9	100	2	0	0	1	21	100		

Table A.1: Number of Successes From 100 Runs of the Direct CI Method

Note: 500 runs were carried out for $n =$ and $m = 4$.

A.2.2 Annealing Parameters for CI Direct Method

n	α	MIL	$MaxIL$	MUL
5	95	400	200	50
6	95	400	200	50
7	95	400	200	50
8	95	600	200	50

Table A.2: Annealing Parameters for Direct CI Method Runs

m	α	MIL	$MaxIL$	MUL
1	95	800	400	50
2	95	800	400	50
3	99	2000	1600	200
4	00.5	2000	1000	50
5	97	1000	400	50
6	97	1000	400	50
7	97	1000	400	50

Table A.3: Annealing Parameters for Direct CI Method Runs on 9 Input Variables

A.3 Example (7,2,4,56,32) Function

Here is the support for a (7,2,4,56,32) derived using the ABF method:

$0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0$
$1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0$
$1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1$
$0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0$

$7969817cc5893ba6$
$ac326e47619f5ad0$

$WalshZeroes :$
$0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 20, 22, 24, 30, 31, 32, 33, 34$
$36, 37, 40, 44, 46, 48, 51, 55, 57, 58, 63, 64, 65, 66, 68, 70, 72, 76, 77, 80, 83, 84$
$89, 90, 92, 96, 98, 102, 105, 107, 108, 109, 111, 113, 117, 118, 120, 121, 123$

$nonLin :\ 56$
$ACorrel :\ 32$
$AlgebraicDegree :\ 4$

A.4 PC(2) Function on 6 Input Variables with Degree 5

Below is the support for a PC(2) function on 6 Input Variables with highest possible nonlinearity, lowest autocorrelation demonstrated and highest possible algebraic degree.

$1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0$
$0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1$

$c65b4d405ceb91f1$
$ACZeroes :$
$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17$
$18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35$
$36, 37, 38, 40, 44, 45, 48, 49, 53, 56, 57, 59, 60, 61, 62, 63$

$nonLin :\ 26$
$ACorrel :\ 16$
$AlgebraicDegree :\ 5$

A.5 Example Bent Function on 6 Input Variables

Here is the support for a bent function on 6 variables

$$
\begin{array}{l}
0,1,0,0,0,1,0,0,1,0,1,0,0,1,1,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,1,1 \\
0,0,0,1,0,1,1,1,0,1,0,1,0,0,0,0,0,1,0,0,0,0,1,0,1,0,0,1,0,0,1,1 \\
\\
44a678f317504293 \\
\\
nonLin:\ 28 \\
ACorrel:\ 0 \\
AlgebraicDegree:\ 3
\end{array}
$$

A.6 Example Bent Function on 8 Input Variables

Here is the support for a bent function on 8 variables

$$
\begin{array}{l}
0,1,1,1,0,1,1,1,0,0,0,0,1,1,1,0,1,1,1,0,1,1,0,1,0,1,0,0,0,1,0,1 \\
0,1,1,0,1,1,1,0,0,1,0,1,0,1,1,0,1,0,1,0,1,1,1,0,1,1,0,1,0,0,0,1 \\
1,1,1,0,0,0,0,1,1,1,1,1,0,0,0,1,0,1,0,1,1,0,0,1,1,1,0,0,1,1,0,1 \\
0,1,1,0,1,1,0,1,0,1,0,1,1,0,1,0,1,1,1,0,0,1,1,0,1,1,0,0,0,0,1,1 \\
1,1,0,1,0,1,0,1,0,1,1,1,1,1,0,1,0,0,1,0,1,1,0,0,0,0,1,1,0,0,1,1 \\
1,1,1,0,0,1,0,0,1,1,0,0,0,0,0,1,1,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0 \\
0,0,1,1,1,0,1,1,1,0,1,0,0,0,0,0,1,0,0,0,0,1,1,0,1,1,1,1,1,1,1,1 \\
1,1,1,1,1,0,0,1,0,1,1,1,0,1,1,0,0,1,0,0,1,0,1,1,1,0,1,1,0,0,0,0 \\
\\
770eed456e56aed1 \\
e1f159cd6d5ae6c3 \\
d57d2c33e4c18480 \\
3ba086fff9764bb0 \\
\\
nonLin:\ 120 \\
ACorrel:\ 0 \\
AlgebraicDegree:\ 4
\end{array}
$$

Bibliography

[1] Tony Bagnall, G. P. McKeown, and V. J. Rayward-Smith. The cryptanalysis of a three rotor machine using a genetic algorithm. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, San Francisco, CA, 1997. Morgan Kaufmann.

[2] T. Baritaud, M. Campana, P. Chauvaud, and H. Gilbert. On the Security of the Permuted Kernel Identification Scheme. In Ernest F. Brickell, editor, *Advances in Cryptology - Crypto '92*, pages 305–311, Berlin, 1992. Springer-Verlag. Lecture Notes in Computer Science Volume 740.

[3] E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems (Extended Abstract). In Alfred J. Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - Crypto '90*, pages 2–21, Berlin, 1990. Springer-Verlag. Lecture Notes in Computer Science Volume 537.

[4] Christian Blum and Andrea Roli. Metheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. Research Report TR/IRIDIA/2001-13, Institut de Recherches Interdisciplinaires et de Developpements en Intelligence Artificielle (IRIDIA), Université Libre de Bruxelles, 2001.

[5] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract). In Walter Fumy, editor, *Advances in Cryptology - EuroCrypt '97*, pages 37–51, Berlin, 1997. Springer-Verlag. Lecture Notes in Computer Science Volume 1233.

[6] Steve Brackin. Deciding Cryptographic Protocol Adequacy with HOL: The Implementation. In *Theorem Proving in Higher Order Logics*. Springer LNCS 1125, 1996.

[7] Steve Brackin. A Highly Effective Logic for Automatically Analysing Cryptographic Protocols. Technical Report 99023, Arca Systems, Exodus Communications, May 1999.

[8] Steve Brackin. Empirical Tests of the Automatic Authentication Protocol Analyser, 2nd Version (AAPA2). Technical Report 99006, Arca Systems, Exodus Communications, October 1999.

[9] L. Burnett, G. Carter, E. Dawson, and W. Millan. Efficient Methods for Generating MARS-like S-Boxes. In Bruce Schneier, editor, *Fast Software Encryption 2000*, pages 300–313. Springer-Verlag, April 2000. Lecture Notes in Computer Science Volume 1978.

[10] Michael Burrows, Martin Abadi, and Roger Needham. A Logic of Authentication. Technical Report 39, Digital Systems Research Center, February 1989.

[11] P. Camion, C. Carlet, P. Charpin, and N. Sendrier. On correlation-immune functions. In Joan Feigenbaum, editor, *Advances in Cryptology - Crypto '91*, pages 86–100, Berlin, 1991. Springer-Verlag. Lecture Notes in Computer Science Volume 576.

[12] E. A. Campbell, R. Safavi-Naini, and P. A. Pleasants. Partial Belief and Probabilistic Reasoning in the Analysis of Secure Protocols. In *Proceedings 5th IEEE Computer Security Foundations Workshop*, pages 84–91. IEEE Computer Society Press, 1992.

[13] C. Carlet. On the Coset Weight Divisibility and Nonlinearity of Resilient and Correlation-immune Functions. In *SETA 2001*, 2001. Personal copy.

[14] F. Chabaud and S. Vaudenay. Links Between Differential and Linear Cryptanalysis. In Alfredo De Santis, editor, *Advances in Cryptology - EuroCrypt '94*, pages 356–365, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 950.

[15] A. Chams, A. Hertz, and D. de Werra. Some Experiments with Simulated Annealing for Colouring Graphs. *European Journal of Operations Research*, 32:260–266, 1987.

[16] P Chardaire, J C Lutton, and A Sutter. Thermostatistical Persistency: A Powerful Improving Concept for Simulated Annealing. *European Journal of Operations Research*, 86:565–579, 1995.

[17] Andrew Clark and Ed Dawson. A Parallel Genetic Algorithm for Cryptanalysis of the Polyalphabetic Subsitution Cipher. *Cryptologia*, 21(2):129–138, April 1998.

[18] Andrew Clark and Ed Dawson. Optimisation Heuristics for the Automated Cryptanalysis Classical Ciphers. *JCMMCC*, 28:63–86, 1998.

[19] Andrew John Clark. *Optimisation Heuristics for Cryptology*. PhD thesis, Security Research Centre, Faculty of Information Technology, Queensland University of Technology, February 1988.

[20] John A Clark and Jeremy L Jacob. On The Security of Recent Protocols. *Information Processing Letters*, 56(3):151–155, October 1995.

[21] John A Clark and Jeremy L Jacob. Attacking Authentication Protocols. *High Integrity Systems*, 1(5):465–474, August 1996.

[22] John A Clark and Jeremy L Jacob. Searching for a Solution: Engineering Tradeoffs and the Evolution of Provably Secure Protocols. In *Proceedings 2000 IEEE Symposium on Research in Security and Privacy*, pages 82–95. IEEE Computer Society, May 2000.

[23] John A Clark and Jeremy L Jacob. Two Stage Optimisation in the Design of Boolean Functions. In Ed Dawson, Andrew Clark, and Colin Boyd, editors, *5th Australasian Conference on Information Security and Privacy, ACISP 2000*, pages 242–254. Springer Verlag LNCS 1841, july 2000.

[24] John A Clark and Jeremy L Jacob. Protocols are Programs Too: the Metaheuristic Search for Security Protocols. *"Information and Software Technology"*, December 2001. Special issue on Metaheursitic Search for Software Engineering.

[25] Richard Clayton. Brute Force on Cryptographic Keys. *www.cl.cam.uk/ rncl/brute.html*, 2001.

[26] David Corne, Marco Dorigo, and Fred Glover, editors. *New Ideas in Optimizarion*. Advanced Topics in Computer Science. McGraw Hill, 1999.

[27] IBM Corporation. MARS - A Candidate Cipher for AES. *www.research.ibm.com*, 2000.

[28] Y. Desmedt, F. Hoornaert, and J. J. Quisquater. Several Exhaustive Key Search Machines and DES. In Ingemar Ingemarsson, editor, *Abstracts of Papers: EuroCrypt '86*, pages 17–19, Linkoping, Sweden, 1986. Department of Electrical Engineering, University of Linköping.

[29] D. Deutsch. Quantum Theory, the Church-Turing Thesis, and the Universal Computer. *Proceedings of the Royal Society of London. Series A*, 400, 1985.

[30] W. Diffie and M. E. Hellman. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10(6):74–84, June 1977.

[31] C. Ding, G. Xiao, and W. Shan. *The Stability of Stream Ciphers*, volume 561 of *Lecture Notes in Computer Science*. Springer-Verlag, 1991.

[32] H. Dobbertin. Construction of Bent Functions and Balanced Functions with High Nonlinearity. In *Fast Software Encryption, 1994 Leuven Workshop*, pages 61–74, Berlin, 1994. Springer-Verlag. Lecture Notes in Computer Science Volume 1008.

[33] Engineering and Physical Science Research Council. E-Science Web Site. *www.research-councils.ac.uk/escience/*, 2001.

[34] J. H. Evertse. Linear structures in Block Ciphers. In David Chaum and Wyn L. Price, editors, *Advances in Cryptology - EuroCrypt '87*, pages 249–266, Berlin, 1987. Springer-Verlag. Lecture Notes in Computer Science Volume 304.

[35] T. A. Feo and M. G. C. Resende. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[36] R. P. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21, 1982.

[37] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In A. M. Odlyzko, editor, *Advances in Cryptology - Crypto '86*, pages 186–194, Berlin, 1986. Springer-Verlag. Lecture Notes in Computer Science Volume 263.

[38] Eric Filiol and Caroline Fontaine. Highly Nonlinear Balanced Boolean functions with a Good Correlation-Immunity. In *Advances in Cryptology EUROCRYPT'98*, pages 475–488. Springer Verlag LNCS 1403, 1998.

[39] Electronic Frontier Foundation. Electronic Frontier Foundation DES Cracker Web Site. *www.eff.org/descracker/*, 1998.

[40] Altaf Abdul Gaffar. A Survey on the Handel-C Language. *www.iis.ee.ic.ac.uk/ frank/surp99/article1/amag97*, 2001.

[41] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[42] S. Goldwasser, S. Micali, and C. Rackoff. Knowledge Complexity of Identification Proof Schemes. In *17th ACM Symposium on the Theory of Computing STOC,* pages 291–304. SACM, 1985.

[43] J. D. Golic. Fast Low Order Approximation of Cryptographic Functions. In Ueli Maurer, editor, *Advances in Cryptology - EuroCrypt '96,* pages 268–282, Berlin, 1996. Springer-Verlag. Lecture Notes in Computer Science Volume 1070.

[44] J. D. Golic. Linear Statistical Weakness of Alleged RC4 Keystream Generator. In Walter Fumy, editor, *Advances in Cryptology - EuroCrypt '97,* pages 226–238, Berlin, 1997. Springer-Verlag. Lecture Notes in Computer Science Volume 1233.

[45] J. D. Golic and R. Menicocci. Edit Distance Correlation Attack on the Alternating Step Generator. In Burt Kaliski, editor, *Advances in Cryptology - Crypto '97,* pages 499–512, Berlin, 1997. Springer-Verlag. Lecture Notes in Computer Science Volume 1294.

[46] Lov K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In *ACM Symposium on Theory of Computing*, pages 212–219, 1996.

[47] Xiao Guo-Zeng and James L. Massey. A Spectral Characterization of Correlation-Immune Combining Functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.

[48] Klaus Havelund and Natarajan Shankar. Experiments in Theorem Proving and Model Checking for Protocol Verification. In Marie-Claude Gaudel and Jim Woodcock, editors, *FME'96: Industrial Benefit and Advances in Formal Methods,* pages 662–681. Springer-Verlag LNCS 1051, 1996.

[49] C.A.R Hoare. *Communicating Sequential Processes*. Series in Computer Science. Prentice-Hall International, 1985.

[50] T Hogg. Highly Structured Searches with Quantum Computing. *Physical Review Letters*, 80:2473–2473, 1998.

[51] T Honda, T Satoh, T Iwata, and K Kurosawa. Balanced Boolean functions Satisfying PC(2) and Very Large Degree. In *Workshop on Selected Areas in Cryptography (SAC'97)*, 1997.

[52] X.-D. Hou. On the Norm and Covering Radius of First-Order Reed-Muller Codes. *IEEE Transactions on Information Theory*, 43(3):1025–1027, May 1997.

[53] X.-D. Hou. The Covering Radius of the (2^{15},16) Reed-Muller Code is at Least 16276. *IEEE Transactions on Information Theory*, 43(3):1025–1027, May 1997.

[54] IBM. RSA Factoring By Web Project. *www.npac.syr.edu/factoring.html*, 1995.

[55] IBM. Blue Gene Web Site. *www.research.ibm.com/bluegene/*, 2001.

[56] IBM. Deep Blue Web Site. *www.research.ibm.com/deepblue/meet/html/d.1.1.html*, 2001.

[57] Cryptography Research Inc. DPA Web Site. *www.cryptography.com/dpa/technical/*, 2001.

[58] Georgiades J. Some Remarks on the Security of the Identification Schemes Based on Permuted Kernels. *Journal of Cryptology*, 5(2):133–137, 1992.

[59] Thomas Jakobsen. A Fast Method for Cryptanalysis of Substitution Ciphers. *Cryptologia*, XIX(3):265–274, July 1995.

[60] Giddy J.P. and Safavi-Naini R. Automated Cryptanalysis of Transposition Ciphers. *The Computer Journal*, XVII(4), 1994.

[61] B. S. Kaliski and M. J. B. Robshaw. Linear Cryptanalysis using Multiple Approximations. In Yvo Desmedt, editor, *Advances in Cryptology - Crypto '94*, pages 26–39, Berlin, 1994. Springer-Verlag. Lecture Notes in Computer Science Volume 839.

[62] B. S. Kaliski and Y. L. Yin. On Differential and Linear Cryptanalysis of the RC-5 Encryption Algorithm. In Don Coppersmith, editor, *Advances in Cryptology - Crypto '95*, pages 171–184, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 963.

[63] R. Kemmerer, C. Meadows, and Jonathan Millen. Three Systems for Cryptographic Protocol Analysis. *Journal of Cryptology*, 7(2):79–130, 1994.

[64] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 1983.

[65] Lars R. Knudsen and Willi Meier. Cryptanalysis of an Identification Scheme Based on the Permuted Perceptron Problem. In *Advances in Cryptology Eurocrypt '99*, pages 363–374. Springer Verlag LNCS 1592, 1999.

[66] P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *Advances in Cryptology - Crypto '96*, pages 104–113, Berlin, 1996. Springer-Verlag. Lecture Notes in Computer Science Volume 1109.

[67] Joanna Kolodziejczyk. The Application of Genetic Algorithm in Crptoanalysis of Knapsack Cipher. In *European School on Genetic Algorithms, Eurogen97*, 1997.

[68] Chemometrics Research Group. Naval Research Laboratory. Practical Guide to Genetic Algorithms.
chemdiv-www.nrl.navy.mil/6110/sensors/chemometrics/practga.html, 2001.

[69] S. K. Langford and M. E. Hellman. Differential-linear cryptanalysis. In Yvo Desmedt, editor, *Advances in Cryptology - Crypto '94*, pages 17–25, Berlin, 1994. Springer-Verlag. Lecture Notes in Computer Science Volume 839.

[70] S. Maitra. Highly Nonlinear Balanced Boolean Functions with very good Autocorrelation Property. Technical Report 2000/047, Indian Statistical Institute, 203 B.T. Road, Calcutta 700 035, India, 2000. Cryptology ePrint Archive, http://eprint.iacr.org/.

[71] S. Maitra. Autocorrelation Properties of Correlation Immune Boolean Functions. In *To appear: Proceedings of Indocrypt' 01*, 2001.

[72] S. Maitra and E. Pasalic. Further Constructions of Resilient Boolean Functions with Very High Nonlinearity. In *Proceedings of SETA' 01*, 2001. Extended version of SETA' 01 paper. Personal copy.

[73] S. Maitra and P. Sarkar. Cryptographically Significant Boolean Functions with Five-valued Walsh Spectra. *Accepted to appear: Journal of Theoretical Computer Science*, 2001.

[74] Robert A J Mathews. The Use of Genetic Algorithms in Cryptanalysis. *Cryptologia*, XVII(2):187–201, April 1993.

[75] M. Matsui. Linear Cryptanalysis Method for DES Cipher. In Tor Helleseth, editor, *Advances in Cryptology - EuroCrypt '93*, pages 386–397, Berlin, 1993. Springer-Verlag. Lecture Notes in Computer Science Volume 765.

[76] W. Meier and O. Staffelbach. Fast Correlation attacks on Stream Ciphers. In Christof G. Günther, editor, *Advances in Cryptology - EuroCrypt '88*,

pages 301–316, Berlin, 1988. Springer-Verlag. Lecture Notes in Computer Science Volume 330.

[77] W. Meier and O. Staffelbach. Nonlinearity Criteria for Cryptographic Functions. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology - EuroCrypt '89*, pages 549–562, Berlin, 1989. Springer-Verlag. Lecture Notes in Computer Science Volume 434.

[78] Zbigniew Michalewizc. *Genetic Algorithms+Data Structures=Evolution Programs*. Springer, 1996.

[79] W Millan. How to Improve the Non-linearity of Bijective S-boxes. In C. Boyd and E. Dawson, editors, *3rd Australian Conference on Information Security and Privacy*, pages 181–192. Springer-Verlag, April 1998. Lecture Notes in Computer Science Volume 1438.

[80] W. Millan, L. Burnett, G. Carter, A. Clark, and E. Dawson. Evolutionary Heuristics for Finding Cryptographically Strong S-Boxes. In *ICICS 99*, 1999.

[81] W. Millan, A. Clark, and E. Dawson. Boolean Function Design Using Hill Climbing Methods. In Bruce Schneier, editor, *4th Australian Conference on Information Security and Privacy*. Springer-Verlag, april 1999. Lecture Notes in Computer Science Volume 1978.

[82] William Millan, Andrew Clark, and Ed Dawson. An Effective Genetic Algorithm for Finding Highly Non-linear Boolean Functions. In *Proceedings of the First International Conference on Information and Communications Security*, pages 149–158. Springer Verlag LNCS 1334, 1997.

[83] William Millan, Andrew Clark, and Ed Dawson. Smart Hill-climbing Finds Better Boolean Functions. In *Proceedings of the First International Conference on Information and Communications Security*, pages 149–158. Springer Verlag LNCS 1334, 1997.

[84] William Millan, Andrew Clark, and Ed Dawson. Heuristic Design of Cryptographically Strong Balanced Boolean Functions. In *Advances in Cryptology EUROCRYPT'98*, pages 489–499. Springer Verlag LNCS 1403, 1998.

[85] Judy H. Moore. Protocol Failures in Cryptosystems. *Proceedings of the IEEE*, 76(5), May 1988.

[86] K. Nyberg and L. R. Knudsen. Provable Security against Differential Cryptanalysis. In Ernest F. Brickell, editor, *Advances in Cryptology -*

Crypto '92, pages 566–574, Berlin, 1992. Springer-Verlag. Lecture Notes in Computer Science Volume 740.

[87] Levbedko O. and Topchy A. On Efficiency of Genetic Cryptanalysis for Knapsack Ciphers. In *Poster Proceedings of ACDM 98*, 1998.

[88] A. Odlyzko. The Rise and Fall of Knapsack Cryptosystems. In *PSAM: Proceedings of the 42th Symposium in Applied Mathematics, American Mathematical Society*, volume 42, pages 75–88, 1991.

[89] University of California at Berkeley. SETI@home Web Site. *setiathome.ssl.berkeley.edu/*, 1995.

[90] National Bureau of Standards. Data Encryption Standard. *NBS FIPS PUB 46*, 1976.

[91] K. Ohta and K. Aoki. Linear Cryptanalysis of the Fast Data Encipherment Algorithm. In Yvo Desmedt, editor, *Advances in Cryptology - Crypto '94*, pages 12–16, Berlin, 1994. Springer-Verlag. Lecture Notes in Computer Science Volume 839.

[92] K. Ohta, S. Moriai, and K. Aoki. Improving the Search Algorithm for the Best Linear Expression. In Don Coppersmith, editor, *Advances in Cryptology - Crypto '95*, pages 157–170, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 963.

[93] I.H Osman. Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem. *Annals of Operations Research*, 41, 1993.

[94] E. Pasalic and T. Johansson. Further Results on the Relation between Nonlinearity and Resiliency of Boolean Functions. In *IMA Conference on Cryptography and Coding Theory*, pages 35–45, Berlin, 1999. Springer-Verlag. Lecture Notes in Computer Science Volume 1746.

[95] E. Pasalic, S. Maitra, T. Johansson, and P. Sarkar. New Constructions of Resilient and Correlation Immune Boolean Functions Achieving Upper Bound on Nonlinearity. In *Workshop on Coding Theory, Electronic Notes in Discrete Mathematics*. Elsevier, January 2001.

[96] Jaques Patarin and Pascal Chauvaud. Improved Algorithms for the Permuted Kernel Problem. In *Crypto '93*, pages 391–402. Springer Verlag, 1993.

[97] F. Piper and P. Beker. *Cryptography and Communications Security*. Prentice-Hall International, 1982.

[98] David Pointcheval. A New Identification Scheme Based on the Perceptron Problem. In *Advances in Cryptology Eurocrypt '95*. Springer Verlag LNCS X, 1995.

[99] J-J Quisquater and Y.G. Desmedt. Chinese Lotto as an Exhaustive Code-Breaking Machine. *Computer,* 24(11):14–22, Nov 1991.

[100] V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, and G. D. Smith, editors. *Modern Heuristic Search Methods*. Wiley, 1996.

[101] V.J. Rayward-Smith, G.D Smith, and J. C. W. Debuse. Parameter Optimisation for a Discrete Event Simulator. *Journal of Computers and Industrial Engineering*, 37(1–2):181–184, 1999.

[102] Colin R. Reeves, editor. *Modern Heuristic Techniques for Cominatorial Problems*. McGraw Hill, 1995.

[103] M. G. C. Resende. A Bibliography of GRASP.

[104] Eleanor G. Rieffel and Wolfgang Polak. An Introduction to Quantum cComputing for Non-Physicists. *ACM Computing Surveys*, 32(3):300–335, 2000.

[105] O.S. Rothaus. On Bent Functions. *Journal of Combinatorial Theory: Series A*, 20:300–305, 1976.

[106] Peter Ryan, Steve Schneider, Michael Goldsmith, Gavin Lowe, and Bill Roscoe. *Modelling and Analysis of Security Protocols*. Addison-Wesley, 2001.

[107] P. Sarkar and S. Maitra. Highly Non-linear Resilient Functions Optimising Siegenthaler's Inequality. In *Advances in Cryptology - Crypto '99*, pages 198–215, Berlin, 1999. Springer-Verlag. Lecture Notes in Computer Science Volume 1666.

[108] P. Sarkar and S. Maitra. Construction of Nonlinear Boolean Functions with Important Cryptographic Properties. In Preneel B., editor, *Advances in Cryptology - EuroCrypt '2000*, pages 485–506, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science Volume 1807.

[109] P. Sarkar and S. Maitra. Nonlinearity Bounds and Constuction of Resilient Boolean Functions Boolean Functions. In Mihir Bellare, editor, *Advances in Cryptology - Crypto '2000*, pages 515–532, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science Volume 1880.

[110] Bruce Schneier. *Applied Cryptography*. Wiley, 1996.

[111] A. Shamir. An Efficient Scheme Based On Permuted Kernels. In *Advances in Cryptology — Crypto '89*, pages 606–609. Springer Verlag LNCS 435, 1997.

[112] W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *35th IEEE Symposium on the Foundations of Computer Science,* 1994.

[113] T. Siegenthaler. Correlation Immunity of Non-linear Combining Functions for Cryptographic Applications. *IEEE Transactions on Information Theory*, IT-30(5):776–780, Sept. 1984.

[114] T. Siegenthaler. Decrypting a Class of Stream Ciphers Using Ciphertext Only. *IEEE Transactions on Computers*, C-34(1):81–85, Jan. 1985.

[115] Einar Snekkenes. *Formal Specification and Analysis of Cryptographic Protocols*. PhD thesis, Faculty of Mathematics and Natural Sciences, University of Oslo, Norwegian Defence Research Establishment, P.O. Box 25, N-2007, Kjeller, Norway, January 1995.

[116] JJ Son, JI Lim, S Chee, and SH Sung. Global Avalanche Characteristics and Nonlinearity of Balanced Boolean Functions. *Information Processing Letters*, 65(3):139–144, 1998.

[117] Dawn Song and Adrian Perrig. A First Step on Automatic Protocol Generation. In *Proceedings of Network and Distributed System Security 2000,* February 2000.

[118] Dawn Song and Adrian Perrig. Looking for Diamonds in the Dessert — Automatic Security Protocol Generation for Three-party Authentication and Key Agreement. In *Proceedings of the 13th Computer Security Foundations Workshop*. IEEE Computer Society, June 2000.

[119] Richard Spillman. Cryptanalysis of Knapsack Ciphers Using Genetic Algorithms. *Cryptologia*, XVII(4):367–377, 1993.

[120] Richard Spillman, Mark Janssen, Bob Nelson, and Martin Kepner. Use of A Genetic Algorithm in the Cryptanalysis of Simple Substitution Ciphers. *Cryptologia*, XVII(1):187–201, April 1993.

[121] J. Stern. A New Identification Scheme Based on Syndrome Decoding. In Douglas R. Stinson, editor, *Advances in Cryptology - Crypto '93*, pages 13–21, Berlin, 1993. Springer-Verlag. Lecture Notes in Computer Science Volume 773.

[122] SH Sung, S Chee, and C Park. Global Avalanche Characteristics and Propagation Criterion of Balanced Boolean Functions. *Information Processing Letters*, 69(1):21–24, 1999.

[123] Y. Tarannikov. On Resilient Boolean Functions with Maximal Possible Nonlinearity. Technical Report 2000/005, Mech. and Math. Department, Moscow State University, 119899 Moscow, Russia, 2000. Cryptology ePrint Archive, http://eprint.iacr.org/, later appearing Indocrypt 2000.

[124] Y. Tarannikov and D. Kirienko. Spectral Analysis of High Order Correlation Immune Functions. Technical Report 2000/050, Mech. and Math. Department, Moscow State University, 119899 Moscow, Russia, 2000. Cryptology ePrint Archive, http://eprint.iacr.org/.

[125] M. J. Wiener. Efficient DES Key Search. Technical Report TR-244, School of Computer Science, May 1994.

[126] I. F. T. Yaseen and H. V. Sahasrabuddhe. Breaking Multiplicative Knapsack Ciphers Using a Genetic Algorithm. In *International Conference on Knowledge Based Computer Systems*, pages 129–139, 1998.

[127] Imad F.T. Yaseen and H.V. Sahasrabuddhe. A Genetic Algorithm for the Cryptanalysis of the Chor-Rivest Knapsack Public Key Cryptosystem (PKC). In *Third International Conference on Computational Intelligence and Multimedia Applications*. IEEE Computer Society, 1998.

[128] Y.S. Yeh. Web Site. *www.csie.nctu.edu.tw/english/member/faculty/ysyeh.html*, 2001.

[129] X-M. Zhang and Y. Zheng. GAC— the Criterion for Global Avalanche Characteristics of Cryptographic Functions. *Journal of Universal Computer Science*, 1(5):316–333, 1995.