

Invited Paper. Nature-Inspired Cryptography: Past, Present and Future

John A Clark

Department of Computer Science

University of York

York, YO10 5DD, UK

{jac}@cs.york.ac.uk

1 Introduction

Cryptography is an indispensable component of much modern-day system security. It has also been an attractive application domain for researchers in non-standard computation. In this paper I shall identify what I believe to be important themes and pieces of work and explain why they matter. I shall not provide a full survey, my principal aim is to interest you in the subject.

2 On Home Ground I: The Cryptanalysis of Classical Ciphers

2.1 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. Substitution ciphers replace plaintext characters with ciphertext characters. For example, if the letters of the alphabet A..Z are indexed by $0 \dots 25$, then a Caesar cipher might replace a letter with index k by the letter with index $(k + 3) \bmod 26$. 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 “PYRCRGO-TYHPA”.

Modern crypto-systems have now supplanted the classical ciphers but cryptanalysis of classical ciphers is the most popular cryptological application for meta-heuristic 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 meta-heuristic solution as argued below.

Consider a simple substitution cipher on the letters $A \dots Z$ indexed by $0 \dots 25$ as above. The keyspace for this type of system is the set of bijective functions $f : 0 \dots 25 \rightarrow 0 \dots 25$. 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 correct-

ness 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 natural language 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 a terrible 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 closer the underlying decryption key is assumed to be to the actual key.

2.2 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. [33] showed how simple substitution ciphers could be attacked using genetic algorithms and a cost function based on expected character and bi-gram frequencies. Spillman emphasises the importance of experimentation with genetic algorithm parameters and suggests possibilities of more sophisticated cost functions involving trigrams, i.e. three letter strings such as “THE.”

Independently, Matthews [20] 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 classical one. A key is some permutation of $1 \dots N$, 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 “NOWISTHETIMEFORALL-GOODMEN” is written as shown in Figure 1 and hence enciphered as “WTROIADOE OOTELESMLMNHFGN”. This was one of the earliest papers and exhibits considerable originality and sophistication. Firstly, the standard frequency based cost was replaced with a points scor-

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 1: Transposition Cipher

ing 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

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

where P_i is the percentage frequency of the i th bi- or tri-gram 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 also 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 3).

Giddy and Safavi-Naini [17] use simulated annealing to attack simple transposition ciphers where sections of n letters are each shuffled according to a key permutation. 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

$$C(s) = \sum_{\alpha \in \Lambda} \sum_{\beta \in \Lambda} \left| \frac{p_{\alpha\beta} - c_{\alpha\beta}}{\epsilon + p_{\alpha\beta}} \right| \quad (2)$$

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 [16] attacks simple and polyalphabetic substitution ciphers (assuming that the number of alphabets used is obtained by standard means, see [29]). 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 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 [4] provides a comparison of simulated annealing, genetic algorithms and tabu search attacks on simple substitution ciphers.

As Bagnall et al. note [1] the ciphers attacked are generally simple ones. The Enigma variants attacked by Bagnall et al. are amongst the most sophisticated classical ciphers attacked (both odometer and other rotor rotation strategies are considered) using meta-heuristic search. In much the same way as modern-day linear cryptanalysis works by guessing elements of the final round key and judging the quality of the guess by reference to approximation measurements over $n - 1$ rounds, Bagnall et al. invoke a similar procedure with respect to the last rotor.

Parallelism is little exploited in heuristic cryptanalysis research. Clark and Dawson [3] use a paralised GA to attack a polyalphabetic substitution cipher. This combines several individual substitution ciphers. This paper seems a useful contribution.

2.3 General Commentary

All the work described above has served a useful purpose. Classical cipher cryptanalysis provides a simple test-bed 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).

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 expect to 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 won’t work when applied to modern cryptanalysis problems. I believe that moving away from this one-shot view has considerable potential.

Although my personal interest lies largely in breaking modern-day crypto-algorithms, I recognise the usefulness

of classical cryptanalysis as an entry point for new researchers and as case-study material for applying new techniques. More recent work discussed later in this paper owes a great deal to the published work described above. **I would ask that our community continue to find a place in its sessions for such research.** Forwarding the subject is primary; competing with the cryptographers is secondary.

3 On Home Ground II: The Cryptanalysis of Schemes Based on Standard ‘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. As we see below, life in this domain has not been too easy.

3.1 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 [28] as does Moore [26]. They are not used much nowadays. It would appear that their security just seems too fragile for users to have confidence in them. Cryptographic knapsacks are based on the Subset Problem:

Given a finite set $W = \{w_1, \dots, w_n\}$ 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, \dots, b_n . Let $W = \{w_1, \dots, w_n\}$ be a set of positive integers as defined above. Encrypt message $M = b_1, \dots, b_n$ as $C = \sum_{i=1}^n b_i w_i$

In practice arithmetic is carried out in some finite field. If the subsets of W 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 should generally be 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?

It would appear that all work in this area is based on very small problems. The initial work by Spillman [32] dealt

with knapsacks of size 8 and 15. The work used a rather odd cost function with an unclear rationale (Clark and Dawson [4] 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 [19] 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 highly optimistic. In 1998, Lebedko and Topchy [27] 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’.

The work of Lebedko and Topchy [27] and similarly that of Clark and Dawson [4] 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 $b = b_1 \dots b_n$ is a proposed solution, the fitness of that solution was generally of the form

$$f_1(b) = g\left(\frac{|C - \sum_{i=1}^n w_i b_i|}{\sum_{i=1}^n w_i}\right) \quad (3)$$

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

$$f_2(b) = H\left(C, \sum_{i=1}^n w_i b_i\right) \quad (4)$$

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 evolutionary computation 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 size 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 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.’

In 1998 and 1999 Yaseen and Sahasrabudde attacked multiplicative knapsack ciphers [36] and the Chor-Rivest cipher [37]. 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.

Hamming Distance	Fitness Value			
	> 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 1: Hamming Distance Distribution for High Fitness

I have included the knapsack work mostly for reasons of their historical importance. If one point emerges from the above it is the chasm between real cryptographic knapsacks and those addressed by heuristic search security researchers. It is clear, that even on what I have described as ‘home ground’ there are significant challenges for our field. Many knapsack schemes have been broken by professional cryptographers. Will heuristic search techniques ever be able to compete?

3.2 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 [13], Fiat and Shamir give impetus to the topic by showing how they might be used to prove user identities [12]. 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 computationally hard problems from the literature, since the problem can be formulated much more efficiently (in terms of memory storage and computation needed). Some of these are given below.

In 1995 David Pointcheval proposed a scheme based on the Permuted Perceptrons Problem (PPP) — a more difficult version of the Perceptrons Problem (PP) [30]. 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 of A and x can be generated randomly (negating all the elements in matrix

row i if $(Ax)_i < 0$ initially).

Solving the PP is known to be a 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 and requiring an attacker to derivate a secret x with the same histogram of values. Since every solution to the PPP is also a solution to the PP, it would appear that this is harder. Actually, it would appear much harder. Pointcheval gives several protocols that allow a user to show that he possesses such a secret x , without actually revealing x itself. He compares the efficiency of this scheme with that of other schemes.

Unusually, Pointcheval (laudably) provides an assessment of attacks by simulated annealing. In this he may be unique. The schemes were later attacked with some subtlety by Knudsen and Meier in 1999 [18]. They chose to carry out multiple runs of annealing and look for commonality. Typically multiple runs are carried out and common elements of the results determined. (Recall here the work of Matthews on ‘toy’ classical ciphers!) 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) 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 techniques use the *distributional properties* of local optima attained using annealing. These are important ideas with considerable potential. These and related ideas were further exploited by Clark and Jacob (see section 5.4).¹

An important issue arises. Was Pointcheval’s analysis wrong? No, his results suggested that the scheme was secure enough against his particular annealing-based attack. Knudsen and Meier simply used another annealing-based attack. Clark and Jacob later launched further types of attack. It is simply impossible to enumerate all possible forms of search based attacks. For this reason, I believe that much of the cryptographic community’s implicit assumption that modern day cryptographic algorithms are immune to non-standard computational attacks to be a matter of faith and not science. I shall return to this point in section 6.

A general comment may be made on the ways schemes based on NP-complete problems are often presented. NP-completeness is a statement about the complexity of the *worst-case* computation required to solve an instance. It seems that NP-completeness is used informally as a badge

¹I believe that the distributional properties of local optima may well prove highly useful for attacking modern day block ciphers and public key algorithms).

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. Phrasing my point somewhat bluntly:

What’s NP-completeness got to do with security?

There are various identification schemes based on hard problems, such as syndrome decoding [34] and the permuted kernel problem [31]. I encourage the evolutionary computing community to attack them.

4 In the Cryptographer’s Den I. Twentieth Century Design: The Work of Brisbane ISRC

4.1 Attacking Problems of Real Cryptological Significance

The design of Boolean functions and S-boxes with desirable cryptographic properties (high nonlinearity, low autocorrelation, high algebraic degree, reasonable order of correlation immunity etc.) is an important area of cryptological research. ² The application of heuristic search techniques to these tasks was promoted almost exclusively in the latter half of the 1990s by the Information Security Research Center at the Queensland University of Technology in Brisbane.

The work on efficient derivation of Boolean functions via hill-climbing is, to my mind, of greatest significance. Millan et al. first showed that small changes to Boolean functions do not radically alter their non-linearity (and may not alter it at all) and so some form of guided local search is worth consideration [24]. Consider the polar form $\hat{f}(x)$ of a Boolean function defined by $\hat{f}(x) = (-1)^{f(x)}$. For any index x , flipping the value of $\hat{f}(x)$ from -1 to 1 (i.e from true to false) or vice-versa causes each Walsh-Hadamard value $\hat{F}(\omega)$ to change by $+2$ or -2 . Similarly, if $\hat{f}(x) = 1$ and $\hat{f}(y) = -1$ 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 dissimilar bits in this way preserves the balance of a Boolean function (assuming it starts as a balanced function). Since non-linearity is defined in terms of the Walsh-Hadamard spectrum, this provides for an efficient delta-cost or delta-fitness function for local searches for highly nonlinear functions. Similar considerations apply to autocorrelation. The most high profile of the hill-climbing papers [23] documents efficient hill-climbing for non-linearity and auto-correlation and investigates a variety of joint property hill-climbing strategies. Later work [25] 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).

²I shall assume some familiarity with basic terminology. Detailed definitions can be found in “Almost Boolean Functions: the Design of Boolean functions by Spectral Inversion”, also in this journal issue.

The work on balanced functions was generalised to encompass bijective [21] and regular [22] 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 inputs and m outputs each output occurs precisely 2^{n-m} times. (Regularity is the vector-valued output version of balance.) Despite this early work S-box design via heuristic search seems under-researched.

4.2 Representation Matters Matter I

The choice of solution representation is crucial in many application domains. Cryptology is no exception. However, the issue rears its head in some unusual ways.

First, a ‘truth table’ representation (i.e. a vector of 2^n elements, each with value $+1$ or -1 , or equivalent) lends itself ideally to highly efficient local search (via the smart delta-functions underpinning the hill-climbing approaches described above).

The use of genetic algorithms seems more problematic. With truth table representations it is entirely possible to mate two excellent functions to get awful children. Indeed, two optimally non-linear functions may be isomorphic (under linear transformation of basis) and yet mate to give children with low non-linearity. Recombination must also preserve balance and Millan et al. [25] show how to achieve balance-preserving crossovers. The overall approach occasionally incorporates elements of intermediate hill-climbing.

Although genetic algorithms have shown themselves to be better than hill-climbing, they are not *obviously* natural candidates for Boolean function design. The simplicity of the truth table representation and the availability of suitable delta-functions, coupled with doubts as to how far non-standard genetic algorithm approaches could be pushed, led me to investigate *local* search further. However, the representation issue is not clear-cut and will rear its head again later in this paper.

4.3 Objective and Fitness

For both nonlinearity and autocorrelation the objective function itself was used as the fitness or cost function. Thus, when high non-linearity or low autocorrelation was the goal the fitness (cost) functions were:

$$\text{fitness}(f) = N_f = \frac{1}{2} \left(2^n - \max_{\omega} |\hat{F}(\omega)| \right) \quad (5)$$

$$\text{cost}(f) = AC_f = \max_{s \neq 0} |\hat{r}(s)| \quad (6)$$

In the case of correlation and propagation immunity notions of correlation deviation and propagation deviation were defined:

$$\text{cdev}_m(f) = \max_{|\omega| \leq m} |\hat{F}(\omega)| \quad (7)$$

$$\text{pcdev}_m(f) = \max_{1 \leq |s| \leq m} |\hat{r}(s)| \quad (8)$$

The cost functions used are direct and perfectly natural ones. Optimal values correspond directly to sought solution

properties. What is not so clear is whether these measures are best for providing *guidance*. For example, certain Walsh or autocorrelation values may be at the extremes at the moment, but the ability to navigate to better solutions depends on the current distribution of less extreme values. Perhaps functions which recognise whole spectra (rather than just subsets) might improve matters? For the time being we simply note that there is potentially an uneasy relationship between objective and fitness functions. We shall return to this issue in the next section.

4.4 Observations

The work of the ISRC in the 1990s laid the foundations for the use of heuristic search techniques for the design of cryptological components. The work above was also responsible for interesting me in the field. It seems surprising that the wider cryptographic community did not readily take up the ideas. My personal belief is that the cryptography community places too great an emphasis on what John Hooker has described as ‘competitive testing’ [14, 15], e.g. a need to show that your technique surpasses the abilities of others. In an academically competitive world, ‘potential’ may be difficult to sell. The cryptographers will take note when results appear that are unattained by other means. Potential, however, there certainly was.

5 In the Cryptographer’s Den II. Twenty-first Century Cryptology

As noted above, the cryptographers will take an interest when the techniques can do things they currently cannot by other means. Since 2000, there have been several examples of heuristic search achieving leading-edge or highly interesting results.

5.1 S-box Improvement

Recent years have seen an international competition to develop a replacement for the aging Data Encryption Standard (DES), which can now be broken in around two hours on specially configured FPGA boards. One such entry (from IBM) was the MARS block cipher. On analysing the S-boxes chosen by the IBM team, Burnett et al. [2] decided to apply a hill-climbing technique to improve their performance against stated criteria. The results were successful. Lesson - hill-climbing works, and traditional cryptographers do not currently reach for guided search as a tool.

5.2 Strange Cost Functions and Conjecture Breaking

In 2000 Clark and Jacob applied simulated annealing with an unusual two-stage approach to achieve some results hitherto unattained by any means [5]. An unusual feature was that the cost functions used embraced all elements of the Walsh spectrum. (Further evidence of the importance and utility of Walsh spectrum distribution properties can be found in [11].) Furthermore, the relationship between the cost function and the actual objective function, though theoretically *inspired* was somewhat obscure (and required sig-

nificant experimentation to determine the best parameters). Embracing the whole Walsh spectrum was an important and successful deviation.

Buried in the results was a counter-example to an extant conjecture on autocorrelation. Only when informed by Millan on best reported results in the literature was it realised that the conjecture had been broken already. As it happens, the results from [5] also contained counter-examples to further conjectures.³ A more informed account of conjecture breaking appears in [7].

5.3 Representation Matters Matter II

Recent work has seen the emergence of some very encouraging and, in some cases, leading results. Millan et al. [10] show how searching over the (restricted) space of algebraic normal forms of Boolean functions (of degree less than or equal to $n/2$) can be used to provide Bent functions of highest possible degree. Clark et al. [8] search over the space of Walsh spectra for a permutation of an initial spectrum that corresponds to a Boolean function. It is thus ‘Boolean-ness’ that is evolved. Of interest in both papers is that the form of representation used greatly facilitates the *restriction of the search space*. In an earlier paper [7] linear transformation of basis is used to provide new functions that possess desired properties. In three different ways, representation is seen to be part of the *solution*. In addition, some representations may greatly facilitate evolutionary approaches (e.g. our manipulation of the Walsh spectrum [8] is essentially a permutation problem, and evolutionary approaches are very strong in this area).

5.4 Side Channels on Analysis Techniques

Traditional cryptography has taken a very mathematical view of the subject. Cryptographic algorithms are seen as functions that map plaintext to ciphertext in a key-dependent manner. However, functions in the real world have implementations that consume power and take time to execute. Resource usage properties of such implementations have been used to leak secret key information. Such side channels have provided a major shock to the cryptographic community. Similarly, it has also been demonstrated that the injection of a fault into a cryptosystem (e.g. causing a state bit to flip) can similarly be used to break a system. In [6] it is demonstrated that *analysis techniques* themselves also have computational dynamics that can be exploited to reveal much information. Essentially, watching the progress of an annealing search in action is shown to be a richer source of information than the final ‘result’ provided by the search! The trajectory is more important than the destination. Similarly, embracing multiple runs of a search technique, each applied to a perturbation of the original problem proves also to be a rich source of information about a sought secret. Of all the work I have been involved in, it is the exploitation of such ‘side-channel’ analogies that

³This might be described as a multiple-injury accident. Even worse, at ACISP 2000, Linda Burnett commented from the floor that the results were ‘some of the most extreme she had ever seen’. I ought to have taken a hint!

I find most exciting. Computation is in the eye of the beholder.

5.5 Smarter Problem Selection

I have promoted the use of heuristic search in cryptology for several years now — to anyone who will listen. Invariably, when I say what I really want to do is apply them to cryptanalysis of modern ciphers I am told that the search space is ‘just too discontinuous’, and there will be a single point ‘spike’ in the fitness function that will never be found. Invariably, the pessimists have in mind a very blunt approach and fitness function and they are probably right in their assessment.⁴

I believe that the work carried out into cryptanalysis of classical ciphers may well have given the impression that the cost functions would be obvious and that we might expect a search of ten minutes or less to break AES. However, we don’t need to be so direct. We can attempt to evolve better *tools* (e.g. better higher-order approximations) to augment current capabilities. Similarly, it is not necessary to serve up a key on a plate immediately. Distinguishing ciphers is a good start. Hernández et al. have applied search techniques to evolved bit masks that allow 2,3, and 4 round discriminators to be evolved for the TEA cipher [9]. This is an important instance of smart *problem selection*.

6 The Future and Denouncing Wrong Faith

I hold great hope applying non-standard computational techniques in the service of cryptology. Below I outline some reasons why.

In many cases, it may be observed that the optimisation sophistication applied has been slight, almost across the board. Results will inevitably improve when the full armoury of non-standard computational techniques are brought to bear. In many cases the problems are easy to state; a great deal of previous cryptographic knowledge is not essential to get started in the field. (Classical cipher cryptanalysis or Boolean function design is a good place to start.) I recommend the area to the evolutionary computation community. And where is the evolutionary community? Ants, swarms, artificial immune systems...come out, come out, wherever you are! You know you want to do cryptography.

Achieving our potential will not be straightforward but there has been considerable evidence of researchers ‘thinking outside the box’. I think the notion of improving existing cryptological tools (distinguishers, approximations etc.) will bear fruit and the profiling of search trajectories may well prove of wider use than its current application to the PPP.

It is important to realise our current limitations. We have yet to launch a truly competitive attack on a modern-day block cipher (but given the recent flurry of cryptanalytical interest, the new world standard block cipher AES looks a

truly tempting target) and hats off to the first convincing application of our techniques to a serious number theoretic cipher (e.g. RSA or elliptic curve cipher).

Finally, in an invited paper, I feel free to engage in a little religious denunciation. That religion is modern-day cryptology! I know of no proof that modern day crypto-algorithms are immune to attacks by evolutionary and other non-standard computational techniques. Current cryptographic notions of ‘provably secure’ generally mean ‘against things we have thought of’. Most cryptographers rarely think of our techniques at all. Many I have spoken too seem to have what I would describe as ‘faith’ that evolutionary techniques have little chance of real success.⁵ I would imagine many in the evolutionary computing community feel this too! I denounce you all!

If it’s provably secure, it probably isn’t. Lars Knudsen.

A little magic from the evolutionary computing community seems called for.

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] 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.
- [3] Andrew Clark and Ed Dawson. A Parallel Genetic Algorithm for Cryptanalysis of the Polyalphabetic Substitution Cipher. *Cryptologia*, 21(2):129–138, April 1998.
- [4] Andrew Clark and Ed Dawson. Optimisation Heuristics for the Automated Cryptanalysis Classical Ciphers. *JCMCC*, 28:63–86, 1998.
- [5] 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.
- [6] John A Clark and Jeremy L Jacob. Fault Injection and a Timing Channel on an Analysis Technique. In *Advances in Cryptology Eurocrypt 2002*. Springer Verlag LNCS 1592, 2002.
- [7] John A Clark, Jeremy L Jacob, Susan Stepney, Subhamoy Maitra, and William Millan. Evolving Boolean

⁴Patient: ‘Doctor, doctor. It hurts when I bang my head.’ Doctor: ‘Stop banging your head.’

⁵There are signs this may be changing. See [35]

- Functions Satisfying Multiple Criteria. In *Progress in Cryptology - INDOCRYPT 2002*, pages 246–259. Springer Verlag LNCS 2551, 2002.
- [8] J.A. Clark et al. Almost Boolean Functions: the Design of Boolean Functions by Spectral Inversion. In *Proceedings of Conference on Evolutionary Computation. CEC 2003. Canberra, Australia*, 2003.
- [9] Julio César Hernández et al. Efficient distinguishers for cryptographic mappings, with an application to the block cipher TEA. In *Proceedings of this session.*, 2003.
- [10] William Millan et al. Evolutionary Generation of Bent Functions for Cryptography. In *Proceedings of Conference on Evolutionary Computation. CEC 2003. Canberra, Australia*, 2003.
- [11] William Millan et al. New Concepts in Evolutionary Search for Boolean Functions in Cryptology. In *Proceedings of Conference on Evolutionary Computation. CEC 2003. Canberra, Australia*, 2003.
- [12] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions of Identification and Signature Problems. In Ed Dawson, Andrew Clark, and Colin Boyd, editors, *Advances in Cryptology — Crypto '86*, pages 186–194. Springer Verlag LNCS 263, July 1987.
- [13] 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.
- [14] J.N. Hooker. Needed: An empirical science of algorithms. *Operations Research*, 42:201–212, 1994.
- [15] J.N. Hooker. A Testing heuristics: We have it all wrong. *Journal of Heuristics*, 1:33–42, 1996.
- [16] Thomas Jakobsen. A Fast Method for Cryptanalysis of Substitution Ciphers. *Cryptologia*, XIX(3):265–274, July 1995.
- [17] Giddy J.P. and Safavi-Naini R. Automated Cryptanalysis of Transposition Ciphers. *The Computer Journal*, XVII(4), 1994.
- [18] 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.
- [19] Joanna Kolodziejczyk. The Application of Genetic Algorithm in Cryptanalysis of Knapsack Cipher. In *European School on Genetic Algorithms, Eurogen97*, 1997.
- [20] Robert A J Mathews. The Use of Genetic Algorithms in Cryptanalysis. *Cryptologia*, XVII(2):187–201, April 1993.
- [21] 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.
- [22] W. Millan, L. Burnett, G. Carter, A. Clark, and E. Dawson. Evolutionary Heuristics for Finding Cryptographically Strong S-Boxes. In *ICICS 99*, 1999.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] Judy H. Moore. Protocol Failures in Cryptosystems. *Proceedings of the IEEE*, 76(5), May 1988.
- [27] Levbedko O. and Topchy A. On Efficiency of Genetic Cryptanalysis for Knapsack Ciphers. In *Poster Proceedings of ACDM 98*, 1998.
- [28] 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.
- [29] F. Piper and P. Beker. *Cryptography and Communications Security*. Prentice-Hall International, 1982.
- [30] David Pointcheval. A New Identification Scheme Based on the Perceptron Problem. In *Advances in Cryptology Eurocrypt '95*. Springer Verlag LNCS X, 1995.
- [31] A. Shamir. An Efficient Scheme Based On Permuted Kernels. In *Advances in Cryptology — Crypto '89*, pages 606–609. Springer Verlag LNCS 435, 1997.
- [32] Richard Spillman. Cryptanalysis of Knapsack Ciphers Using Genetic Algorithms. *Cryptologia*, XVII(4):367–377, 1993.
- [33] 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.

- [34] Jaques Stern. A New Identification Scheme Based On Syndrome Decoding. In *Advances in Cryptology — Crypto '93*, pages 13–21. Springer Verlag LNCS 773, 1997.
- [35] STORK. STORK: Strategic Roadmap For Crypto. Open Problems in Cryptology. www.stork.eu.org/documents, 2003.
- [36] 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.
- [37] 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.