

Almost Boolean Functions: the Design of Boolean Functions by Spectral Inversion

John A Clark and Jeremy L Jacob

Department of Computer Science
University of York
York, YO10 5DD, UK
{jac, jeremy}@cs.york.ac.uk

Subhamoy Maitra

Indian Statistical Institute
203 B.T. Road, Calcutta
Pin 700 108, India
subho@isical.ac.in

Pantelimon Stanica

Mathematics Department
Auburn University Montgomery
Montgomery, AL 36124-4023
pstanica@mail.aum.edu

Abstract- The design of Boolean functions with properties of cryptographic significance is a hard task. In this paper, we adopt an unorthodox approach to the design of such functions. Our search space is the set of functions that possess the required properties. It is ‘Booleanness’ that is evolved.

1 Introduction

Boolean functions form crucial components in cryptographic systems. The design of suitable functions has received significant attention from cryptographers for decades. Recently, meta-heuristic search (particularly hill-climbing, genetic algorithms and simulated annealing) has emerged as a potentially very powerful tool for the design of such functions [13, 14, 15, 2]. Most recently, meta-heuristic search in combination with well-established theory has found functions with properties unattained by any other means [3].

All the optimisation work so far has searched the space of Boolean functions for those with particular properties. In this paper, we invert this notion and search the space of artifacts with the required properties and seek one which is actually a Boolean function. Combining this general approach with very recent theory allows functions with hitherto unobtainable properties to be found.

In this paper, our primary aim is to report new research results that compete with or improve on the best results of available theoretical constructions. A full motivation or explanation of why what we wish to do is of interest to the cryptographic community is beyond the scope of this paper. However, we have included sufficient background material to make the paper accessible to the non-cryptographer. We hope also that this material may encourage researchers whose primary expertise lies in meta-heuristic search to work in the area.

2 Preliminaries

2.1 Motivation

Boolean functions play a critical role in cryptography. A standard block cipher such as the Data Encryption Standard (DES) or its more recent replacement, the Advanced Encryption Standard (AES), can be thought of as functions that take vectors of Boolean inputs (e.g. 64 or 80-bit plaintext data blocks) and output similar vectors of Booleans (i.e. the ciphertext blocks). The exact functional transformation

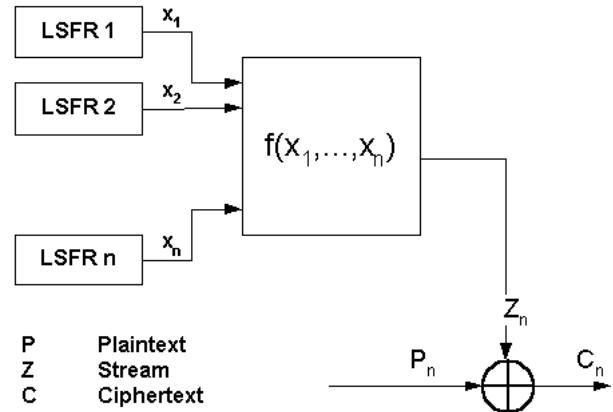


Figure 1: Combining bit streams

is generally defined by a secret key K . Within a block cipher various smaller functions may be deployed. For example, the ‘substitution boxes’ of DES take six-bit input vectors and give four-bit output vectors. Much of the security of the algorithm rests in the design of these boxes. Vector-output functions can be thought of as combinations of single-output functions, but single-output functions have serious applications in their own right. In this paper we shall restrict our attention to single-output functions. Hereafter, the term ‘Boolean functions’ will refer to such functions.

In some stream ciphers Boolean functions may be used to combine the outputs of several bit streams. Figure 1 illustrates a classic stream cipher model. The plaintext stream $\{P_n\}$ of bits is XOR-ed with a pseudo-random bit stream $\{Z_n\}$ to give a cipher text stream $\{C_n\}$. The plaintext is recovered by the receiver by XOR-ing the cipherstream with the same pseudo-random stream. The pseudo-random stream is formed from several bit streams generated by Linear Feedback Shift Registers (LFSRs) using a suitable combining function f . At each step the rightmost bit of register LFSR i becomes the i th input x_i to the combining function f , the register is shifted right by 1 and the new value for the leftmost bit is calculated by XOR-ing a number of the other bits of the register (before shifting). The initial state of the registers forms the *secret key*. The function f must be sufficiently complex that cryptanalysts will not be able to determine the initial state of the registers, even when they know what the plaintext is (and so can recover the pseudo-random stream $\{Z_n\}$).

The possession (or otherwise) of various structural prop-

erties by f may facilitate cryptanalysis. Suppose our stream cipher had two 64-bit registers and a combining function f . An enumerative attack on the cipher would involve trying 2^{128} possible keys. However, suppose that $f(x_1, x_2) = x_1 \oplus x_2$. If we know the output from f we know that there are only two possible combinations of x_1, x_2 that would give rise to that output. And so if we observe the first 64 Z_j outputs we have effectively reduced the size of the key space to be searched to 2^{64} . This illustrates the dangers of *linearity* (formally defined below). A dreadful choice of f would be $f(x_1, x_2) = x_1$; observing the first 64 output bits would reveal all the relevant secret key bits (i.e. for LFSR 1). In practice, more sophisticated attacks exploiting linearity of f are possible (e.g. the Best Affine Approximation, see [5], and techniques derived from Matsui's linear cryptanalysis [9]).

It would seem incompetent to use a truly linear function, and in practice the best a cryptanalyst can hope for is a small statistical bias, for example $f(x_1, x_2) = x_1 \oplus x_2$ might be true (or false) for marginally greater than 50 per cent of input combinations.

Exploiting linear approximations involving only one or two bits is generally easier than exploiting linear approximations of several bits. In a larger stream cipher, with say ten register input streams, we might require that there is no effective linear bias involving m or fewer input registers. This leads to the notion of *correlation immunity of order m* (see below).

Autocorrelation is another structural property of f that might cause problems. Essentially, autocorrelation is a measure of a form of periodicity in a function. If autocorrelation is too high then new forms of attack become possible, such as differential cryptanalysis. The details need not concern us here.

Highly nonlinear Boolean functions on small (up to 11) number of variables can be used efficiently in symmetric cipher systems like the COS cipher family (Crossing Over Systems), invented by Eric Filliol et al. [6]. The acronym COS refers to how each cipher internally works. COS ciphers are block ciphers, built only from stream cipher primitives (nonlinear feedback shift registers and Boolean functions). COS ciphers use Boolean functions of 11 variables for key setup and of 9 variables for the encryption procedure itself. These functions must achieve best possible tradeoffs between the important cryptographic properties (balancedness, correlation-immunity, high nonlinearity etc.)

The above text is for motivation only. There is no such thing as a 'secure Boolean function'; there are only functions with properties that prove useful in particular system contexts. We wish only to convey the idea that Boolean functions are an important research topic with practical significance to modern day cryptography. We are not addressing 'toy' problems.

2.2 Single-output Boolean Functions

We denote the binary truth table of a Boolean function by $f : Z_2^n \rightarrow Z_2$ mapping each combination of n binary values to some binary value. If the number of combinations map-

ping to 0 is the same as the number mapping to 1 then the function is said to be *balanced*.

The *polarity truth table* is a particularly useful representation for our purposes. It is defined by $\hat{f}(x) = (-1)^{f(x)}$. Two functions f and g are said to be uncorrelated when $\sum_{x \in Z_2^n} \hat{f}(x)\hat{g}(x) = 0$. If so, if you try to approximate f by using g , you will be right half the time and wrong half the time.

For convenience we shall often refer to elements of Z_2^n by their natural decimal interpretation. Thus 11111111 is interpreted as 255, 00000011 as 3 etc. We now give formal definitions of terms used in this paper. Summations will generally be over $0..(2^n - 1)$ unless otherwise stated.

Linear Boolean Function. A linear Boolean function, selected by $\omega \in Z_2^n$, is denoted by

$$L_\omega(x) = \omega_1 x_1 \oplus \omega_2 x_2 \cdots \oplus \omega_n x_n. \quad (1)$$

where $w_i x_i$ denotes the bitwise AND of the i th bits of ω and x , and \oplus denotes bitwise XOR.

Affine Function. The set of affine functions is the set of linear functions and their complements

$$A_{\omega,c}(x) = L_\omega(x) \oplus c, \text{ where } c \in Z_2 \quad (2)$$

Algebraic Normal Form and Degree. A Boolean function can be expressed as a minimal (XOR) sum of (AND) products:

$$\begin{aligned} f(x_1, \dots, x_n) = & a_0 \oplus a_1 x_1 \cdots \oplus a_n x_n \\ & \oplus a_{1,2} x_1 x_2 \cdots \oplus a_{n-1,n} x_{n-1} x_n \cdots \\ & \oplus a_{1,2,\dots,n} x_1 x_2 \cdots x_n. \end{aligned} \quad (3)$$

This is the *algebraic normal form* of the function. The *algebraic degree* of a function f is the largest number of inputs appearing in any product in its algebraic normal form. Thus $x_1 \oplus x_2$ has degree 1 (i.e. is linear), $x_1 \oplus x_1 x_2 x_3$ has degree 3 etc.

Walsh Hadamard Transform. For a Boolean function f the Walsh Hadamard Transform W_f is defined by

$$W_f(\omega) = \sum_{x \in Z_2^n} \hat{f}(x) \hat{L}_\omega(x). \quad (4)$$

Thus, each Walsh Hadamard Transform $W_f(\omega)$ is the vector dot product of the polar forms of f and the linear function L_ω . We denote the maximum absolute value taken by the transform by $WH_{max}(f) = \max_{\omega \in Z_2^n} |W_f(\omega)|$. It is related to the nonlinearity of f . We shall refer to the vector $(W_f(0), \dots, W_f(2^n - 1))$ as the *Walsh Spectrum* of the function f .

Nonlinearity. The nonlinearity N_f of a Boolean function f is its minimum distance to any affine function. It is given by:

$$N_f = \frac{1}{2}(2^n - WH_{max}(f)). \quad (5)$$

Correlation Immunity. A function f is correlation immune of order m if and only if

$$|W_f(\omega)| = 0; 1 \leq |\omega| \leq m. \quad (6)$$

where $|\omega|$ denotes the Hamming weight of ω .

Resilience. A function f that is correlation immune of order m is *resilient* if and only if it is also balanced:

$$|W_f(\omega)| = 0; 0 \leq |\omega| \leq m. \quad (7)$$

Parseval's Theorem. This states that

$$\sum_{\omega \in Z_2^n} (W_f(\omega))^2 = 2^{2n}. \quad (8)$$

In a sense this can be thought of as a form of generalised Pythagoras' theorem (see below).

Bent functions. A function on an even number n of inputs is a bent function if and only if

$$|W_f(\omega)| = 2^{\frac{n}{2}}, \forall \omega \in 0..2^n - 1 \quad (9)$$

Bent functions achieve the maximum possible nonlinearity but are unbalanced (since $W_f(0) \neq 0$). They are often used as components in theoretical constructions of larger Boolean functions.

Autocorrelation Transform. The autocorrelation transform of a Boolean function f is given by:

$$\hat{r}_f(s) = \sum_x \hat{f}(x) \hat{f}(x \oplus s). \quad (10)$$

We shall refer to the vector $(\text{hatr}_f(0), \dots, \text{hatr}_f(2^n - 1))$ as the *Autocorrelation Spectrum* of the function f . We denote the maximum absolute value in the autocorrelation spectrum of a function f by Δ_f , i.e., $\Delta_f = \max_{s \neq 0} \left| \sum_x \hat{f}(x) \hat{f}(x \oplus s) \right|$. Here x and s range over Z_2^n and so produces a result in Z_2^n . The Δ_f was one of the 'global avalanche characteristics' suggested by Zhang and Zheng [24].

Boolean functions in polar form are represented here as vectors in \mathbf{R}^{2^n} where each component is either +1 or -1. Thus, Boolean functions are particular points on the surface of a hypersphere of radius $2^{\frac{n}{2}}$. Not all points on this surface are Boolean functions; there are uncountably more points with some elements that are not +1 or -1.

The \hat{L}_ω , where ω ranges over $0..(2^n - 1)$, form a set of orthogonal Boolean functions that span \mathbf{R}^{2^n} (i.e. they form a basis set). A function vector \hat{f} can be expressed as:

$$\hat{f} = \sum_{\omega} \frac{1}{2^n} (\hat{f} \cdot \hat{L}_\omega) \hat{L}_\omega = \sum_{\omega} \frac{W_f(\omega)}{2^{\frac{n}{2}}} \frac{\hat{L}(\omega)}{2^{\frac{n}{2}}} \quad (11)$$

The $\frac{W_f(\omega)}{2^{\frac{n}{2}}}$ values are essentially the coordinates of \hat{f} with respect to the *orthonormal* basis provided by the various

$\frac{\hat{L}(\omega)}{2^{\frac{n}{2}}}$. The sum of the squares of these coordinates gives the square of the magnitude of \hat{f} (i.e 2^n). This leads to Parseval's theorem of Eqn. 8. With the function viewed as the hypotenuse, we can now understand why this theorem can be thought of as an instance of Pythagoras' Theorem in 2^n dimensions.

The magnitude of $\frac{1}{2^n} W_f(\omega)$ gives the extent to which \hat{f} is approximated by the linear function \hat{L}_ω . The vector of Walsh Hadamard values (the Walsh Hadamard *spectrum*) provides an alternative representation for \hat{f} .

A function is *balanced* if $W_f(0) = 0$ indicating that \hat{f} has equal numbers of +1 and -1 elements. 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 [7]. This is the form given in Eqn. 6. It simply states that no linear function of m or fewer variables gives any degree of approximation to f . In general, low order linear approximations are easier to exploit than higher order ones. Similarly low algebraic degree may significantly facilitate cryptanalysis.

2.3 Conflicts, Tradeoffs and Profiles

We might typically wish to obtain balanced functions with high nonlinearity, high algebraic degree, high order of correlation immunity and low autocorrelation. However, it is apparent that some of these criteria are in conflict. Consider bent functions defined in Eqn. 9. These achieve the maximum possible nonlinearity (they minimise the maximum magnitude of Walsh values) but are unbalanced. Now if we require a function to be balanced ($W_f(0) = 0$), then some other $W_f(\omega)$ must have magnitude greater than $2^{\frac{n}{2}}$ to compensate (by Parseval's theorem given in Eqn. 8). Similarly, increasing order of immunity can never lead to an increase in achievable nonlinearity.

This conflict means that tradeoffs have to be made. Considerable research has been carried out to derive bounds on achievable profiles (combinations) of properties and to demonstrate functions achieving those bounds [12, 10, 11, 16, 1, 18, 19, 22, 23]. We denote the profile of a function by a quadruplet: (n, m, d, nl) is the profile of a balanced function on n inputs, correlation immune of order m , algebraic degree d and nonlinearity nl . Where we do not impose the requirement for balancedness, we will use profiles with square brackets $[n, m, d, nl]$. It has been known for a long time that $m + d \leq n$ and that $m + d \leq (n + 1)$ for balanced functions. Much of the above research has concerned with deriving bounds for the nonlinearity components. A summary of current profile bounds can be found in [3] (which also records the most extreme profiles demonstrated by theoretical or optimisation techniques so far).

2.4 Observations on Spectral Properties

We can see that various important cryptographic criteria (balance, nonlinearity, correlation immunity, resilience) are defined in terms of the Walsh Hadamard values of that func-

tion. Given a spectrum of Walsh Hadamard values we can see easily whether the various criteria are met. This motivates the new approach described below. A Walsh spectrum satisfying the criteria forms the starting point for the search and we move amongst spectra that preserve the desired properties, in search of a spectrum that is also that of a Boolean function.

3 The Guided Search based on Simulated Annealing

3.1 Casting the Problem as One of Guided Search

Let us consider that the (currently unknown) Walsh spectrum W_f of a balanced Boolean function f on n input variables with desired nonlinearity nl and order of correlation immunity m , is given by $\{W_f(0), \dots, W_f(2^n - 1)\}$. Let $P = \{P(0), \dots, P(2^n - 1)\}$ be a permutation of the values of W_f . Now consider the set of all such permutations P with $P(\omega) = 0$ for all $0 \leq |\omega| \leq m$. This is the set of permutations that maintain the properties required for balancedness, nonlinearity and correlation immunity in the permuted Walsh spectra. However, it is not guaranteed that a permuted spectrum will be the Walsh spectrum for some Boolean function. We can obtain a function $\hat{p}(x)$ by applying the Inverse Walsh Transform to the spectrum P as shown below

$$\hat{p}(x) = 2^{-n} \sum_{\omega} P(\omega) (-1)^{\omega \cdot x}. \quad (12)$$

While a few permutations P , after Inverse Walsh Transform, will correspond to polar forms of Boolean functions, most will not. Thus, the values of the various $\hat{p}(x)$ obtained by inversion may not actually be $+1$ or -1 . However, the values of \hat{p} can be associated heuristically with a Boolean function \hat{b} (in polar form), by choosing,

$$\begin{aligned} \hat{b}(i) &= +1 \text{ if } \hat{p}(i) > 0; \\ \hat{b}(i) &= -1 \text{ if } \hat{p}(i) < 0; \\ \hat{b}(i) &= +1 \text{ or } -1 \text{ (chosen randomly) if } \hat{p}(i) = 0. \end{aligned}$$

Thus, any inverted spectrum can be ‘collapsed’ to a ‘nearest’ Boolean function. This makes heuristic search over the space of Walsh permutations easier. We need only guide the search to a permutation that *collapses* to a desired function. Many permuted spectra may collapse to the same desired function. With each permutation P we can associate a cost that indicates how far P is from the spectrum of a valid Boolean function. Our initial cost function is given below

$$\text{cost}(P) = \sum_{i=0}^{2^n-1} (\hat{p}(i) - \hat{b}(i))^2. \quad (13)$$

where \hat{p} is the function arising from P under the Inverse Walsh Transform. If a permutation P gives rise to zero cost, then it is the spectrum of a Boolean function satisfying the same desired criteria of f . In practice, if the spectrum P is sufficiently ‘close’ to that of a valid Boolean function, it will ‘collapse’ to that function. Thus, the sought spectra are those corresponding to zero or very low values of the cost function in Eqn. 13. The problem is now one of cost

minimization over the space of permutations P indicated above.

3.2 Getting Started and Traversing the Space

A standard form of simulated annealing has been used as the search heuristic. A description is given in the Appendix A. For present purposes the details are irrelevant. We note only that it is a local search technique with an excellent ability to escape from local optima.

Our local search moves by swapping two elements $P(\alpha)$ and $P(\beta)$ of the current spectrum with $|\alpha| > m$, $|\beta| > m$ (i.e. the indices α and β must have Hamming weight greater than m) and $P(\alpha) \neq P(\beta)$. If the initial spectrum has the required properties, this move strategy preserves them. We now need to find an appropriate initial spectrum for the search.

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 relatively easier. Consider the spectrum of a function with profile $(7, 2, 4, 56)$. The maximum absolute value of elements in the Walsh spectrum is 16. Theory has shown that if $n \geq 3$ and $m \leq n - 3$ then the Walsh values of an m -th order resilient function f on n variables must satisfy $|W_f(\omega)| \equiv 0 \pmod{2^{m+2}}$ [19]. 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

$$2^n \times \hat{f}(0) = \sum_{\omega=0}^{2^n-1} W_f(\omega) \quad (14)$$

defines the value of $\hat{f}(0)$. Arbitrarily fixing this to be 1 and using Parseval’s equation allows us to determine that the Walsh spectrum must contain 36 ‘+16’s, 28 ‘-16’s and 64 ‘0’s. Zeroes are placed in the positions in the starting spectrum corresponding to $0 \leq |\omega| \leq 2$ (these elements remain fixed throughout the search) and the remaining Walsh values (i.e. 36 ‘+16’s, 28 ‘-16’s and 35 ‘0’s) are arbitrarily allocated to the remaining positions. Using the cost function given in Eqn. 13, five hundred runs were carried out resulting in five successes. Although this is not efficient, it did generate some example desired functions.¹ A function with this profile was first demonstrated in 2001 [16] and then more recently in [20].

3.3 Refining the Approach

Attempts to generate bent functions on 8 variables (with 136 ‘16’s and 120 ‘-16’s) proved very inefficient and so a second cost function was developed using Titsworth’s theorem [5]. This states that $W_f(\omega)$ is the Walsh spectrum of a binary Boolean function if and only if

$$\sum_{\omega} W_f(\omega) W_f(s \oplus \omega) = 2^{2n} \delta(s), \quad (15)$$

¹Also, we were able to apply a change of basis to transformed some of the ‘failed’ solutions into successful examples. However, finding a suitable change of basis is a highly nonlinear problem, and one which we addressed via annealing in [3]

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

$$\text{cost}(W_f) = \sum_s (|\sum_{\omega} W_f(\omega)W_f(s \oplus \omega)|)^R. \quad (16)$$

When $s = 0$ the inner sum should be non-zero (and is constant for all permutations of the spectrum W_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 on 6 and 8 variables. Obtaining spectra for bent functions is easy since it must be the case that $|W_f(\omega)| = 2^{\frac{n}{2}}$ for all ω . Equation 14 was used again to determine the numbers of positive and negative Walsh values.

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 average time per run was 4m 58s. Out of 50 experiments, we got bent functions (nonlinearity 120) 14 times, and all of them are of degree 4, which is the maximum possible. Note that for all the experiments related to simulated annealing, we used a 1GHz Pentium processor with 512 MByte RAM. Subsequent attempts to evolve bent functions on 10 inputs failed, despite significantly increasing computational resources for the search. Attempts to derive (9, 3, 5, 240) functions failed similarly. The experiments were performed with $R = 2$, a 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.

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.

3.4 ROTS Enabled Guided Search

With the technique described above we could achieve some important resilient functions on 7-variables and bent functions on 8-variables. However, due to the superexponential search space of Boolean functions, the method showed its limitations from 9-variables onwards. Restricting the search space seems the order of the day. One class of function that has attracted attention recently is that of Rotation Symmetric Boolean Functions (RSBFs) [4, 17, 20]. Here, all indices which are rotationally equivalent have the same value for f . Thus, for a 5 variable Boolean function f , we would have $f(10001) = f(11000) = f(01100) = f(00110) = f(00011)$. We used the same strategy, but only in the space of RSBFs introduced earlier, not the total space of Boolean functions and the result of such an effort produced results which had not previously been demonstrated by any technique.

Rotational equivalence partitions the elements $0..(2^n - 1)$. For RSBFs rotationally equivalent elements have the

same Walsh value. Rather than swapping dissimilar pairs of Walsh values in the full spectrum, we now swap dissimilar Walsh values of two equivalence classes. We used the cost function of Equation 16 in exactly the same way as before. This enabled us to evolve bent RSBFs on 10 variables (for truth table see Appendix B).

For (9, 3, 5, 240) attempts we can use the same divisibility theory [19] as for the (7, 2, 4, 56) attempts to deduce that the Walsh values are 0, 32 or -32. Furthermore, assuming $f(\bar{0}) = 0$, there are 256 '0's, 136 '32's and 120 '-32's. A little elementary mathematics allows us to further fix the values of specific classes (see Appendix B for details). The runs produced no (9, 3, 5, 240) ROTS functions but (unbalanced) 3rd order CI ROTS functions of algebraic degree 5 (i.e., [9, 3, 5, 240]) could be produced reliably (see Appendix B for details). Similarly, additionally allowing classes to take on values of 16 and -16 allowed us to evolve (9, 2, 6, 240) functions. Both [9, 3, 5, 250] and (9, 2, 6, 240) have never been demonstrated previously. For truth tables of these functions, refer to Appendix B. (10, 3, 5, 480) function in the RSBF class were also found. This gives the confidence that with more computational resource and time, it may be possible to get cryptographically significant functions on higher number of variables.²

4 Conclusions and Future Work

The design of Boolean functions with cryptographic significance by search or by theoretical construction has generally considered only the space of Boolean functions. Here, we have observed that many properties of interest are defined in terms of the Walsh spectrum. Using very recent cryptographic theory, we have been able work directly in the space of Walsh spectra, starting with the properties we actually require and evolving to a solution that is a Boolean function. Embracing the concept of 'almost Boolean functions', the cost function measuring just 'how Boolean' a functions is, has allowed us to attain functions with optimal profiles of properties. In some cases, such profiles have never been demonstrated by theoretical construction.

In [3] it was shown how metaheuristic search could equal the best achievements of all theoretical constructions for eight or fewer inputs (and in many cases demonstrate more extreme profiles). In this paper we have shown how a novel technique can extend the achievements (particularly to higher numbers of inputs). We have used a 'vanilla' variant of simulated annealing. More sophisticated search techniques should bring even better results. (Permutation genetic algorithms are currently under investigation.)

The functions evolved in this work are small (modern day cryptographic applications of Boolean functions tends to favour higher numbers of input variables). For functions of 9 or fewer variables, annealing-based approaches have shown themselves more than equal to theoretical approaches. For higher numbers of inputs they have yet to

²It is fairly trivial to use the technique for 7 and 8 input functions. For example, 50 attempts to derive bent functions on 8 variables met rapidly with 100 per cent success (all with maximal degree 4).

make a real impact. This shows very clearly a direction for research — to significantly extend the size of functions that can be evolved.

Cryptanalysis aims to exploit structure in cryptographic system components. Restricting the search space to a structured subset (functions with rotation symmetry) may provide functions that facilitate cryptanalysis (including forms of analysis not currently known). Thus, although we have demonstrated functions with hitherto unattained profiles of properties, crypto-designers may be wary of using such functions in particular contexts. However, once we have a rotation symmetric function with the required properties, it may be possible to transform it to one without rotation symmetric structure. Indeed, current work suggests that if the magnitude of each Walsh spectrum value is known, annealing can be used to evolve the signs of the magnitude (using the same cost functions as used in this paper) to give the Walsh spectrum of a legitimate Boolean function. The absolute values of the spectrum can be obtained from the evolved rotation symmetric boolean functions. This seems a very exciting avenue to pursue.

Our results are made possible only by embracing leading edge theory. We have used results on Walsh divisibility developed as recently as 2000 [18] and improved matters by restricting ourselves to a class of functions investigated as recently as 2002 [4, 17, 20, 21]. The message is clear: a fusing of evolutionary approaches and leading-edge theory has the potential to be a very significant design tool for functions of cryptographic significance. The two communities should talk more often! We recommend the area to specialists in evolutionary computing.³

Bibliography

- [1] C. Carlet and P. Sarkar. Spectral domain analysis of correlation immune and resilient Boolean functions. *Finite Fields and Its Applications*, 8(1):120–130, January 2002.
- [2] J. A. Clark and J. L. Jacob. Two-Stage Optimization in the Design of Boolean Functions. In *ACISP 2000*, number 1841 in Lecture Notes in Computer Science, pages 242–254. Springer-Verlag, 2000.
- [3] J. A. Clark, J. L. Jacob, S. Stepney, S. Maitra and W. Millan. Evolving Boolean Functions Satisfying Multiple Criteria. In *INDOCRYPT 2002*, Volume 2551 in Lecture Notes in Computer Science, pages 246–259, Springer Verlag, 2002.
- [4] T. W. Cusick and P. Stănică. Fast Evaluation, Weights and Nonlinearity of Rotation-Symmetric Functions. *Discrete Mathematics*, pages 289-301, vol 258, no 1-3, 2002.
- [5] C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers*. Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.
- [6] E. Filliol, C. Fontaine and D. Vianne *A New Fast Block Cipher Design: COS Ciphers*. Proceedings of the International Symposium on Information Theory 2001.
- [7] G.Z. Xiao and J.L. Massey. A spectral characterization of correlation immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.
- [8] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.
- [9] M. Matsui. Linear Cryptanalysis Method for DES Cipher. In Tor Hellesest, editor, *Advances in Cryptology - EuroCrypt '93*, pages 386–397, Berlin, 1993. Springer-Verlag. Lecture Notes in Computer Science Volume 765.
- [10] S. Maitra and E. Pasalic. Further constructions of resilient Boolean functions with very high nonlinearity. *IEEE Transactions on Information Theory*, 48(7):1825–1834, July 2002.
- [11] S. Maitra and P. Sarkar. Cryptographically significant Boolean functions with five-valued walsh spectra. *Theoretical Computer Science*, Volume 276, Number 1–2, pages 133-146, 2002.
- [12] S. Maity and T. Johansson. Construction of Cryptographically Important Boolean Functions. In *INDOCRYPT 2002*, Volume 2551 in Lecture Notes in Computer Science, pages 234–245, Springer Verlag, 2002.
- [13] W. Millan, A. Clark and E. Dawson. An effective genetic algorithm for finding highly nonlinear Boolean functions. In *First International Conference on Information and Communications Security*, number 1334 in Lecture Notes in Computer Science, pages 149–158. Springer Verlag, 1997.
- [14] W. Millan, A. Clark and E. Dawson. Heuristic Design of Cryptographically Strong Balanced Boolean Functions. In *Advances in Cryptology EUROCRYPT'98*, pages 489–499. Springer Verlag LNCS 1403, 1998.
- [15] W. Millan, A. Clark and E. Dawson. Boolean function design using hill climbing methods. In *4th Australasian Conference on Information, Security and Privacy*, number 1587 in Lecture Notes in Computer Science, pages 1–11. Springer Verlag, April 1999.
- [16] 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 and Cryptography - WCC 2001*, Paris, January 8–12, 2001. Electronic Notes in Discrete Mathematics, Volume 6, Elsevier Science, 2001.
- [17] J. Pieprzyk and C. X. Qu. Fast Hashing and Rotation-Symmetric Functions. *Journal of Universal Computer Science*, 5(1):20–31, 1999.

³A special page containing many on-line papers concerning optimisation and cryptology can be found at <http://www.cs.york.ac.uk/security/Security/LibraryPages/NatureInspired.html>

- [18] P. Sarkar and S. Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. In *Advances in Cryptology - EUROCRYPT 2000*, number 1807 in Lecture Notes in Computer Science, pages 485–506. Springer Verlag, May 2000.
- [19] P. Sarkar and S. Maitra. Nonlinearity bounds and construction of resilient 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.
- [20] P. Stănică and S. Maitra. Rotation Symmetric Boolean Functions – Count and Cryptographic Properties. In *R. C. Bose Centenary Symposium on Discrete Mathematics and Applications*, December 2002, Indian Statistical Institute, Calcutta and Technical Report of Cryptology Research Group, Indian Statistical Institute, Technical Report Number CRG/2002/10, Nov 18, 2002, http://www.isical.ac.in/~crg/tech_reports.html.
- [21] P. Stănică and S. Maitra and John A. Clark. Results on Rotation Symmetric Bent and Correlation Immune Boolean Functions. In B. Roy and S. Maitra, editors, *Fast Software Encryption 2004*, February 2004, Delhi.
- [22] Y. V. Tarannikov. On resilient Boolean functions with maximum possible nonlinearity. In *Progress in Cryptology - INDOCRYPT 2000*, number 1977 in Lecture Notes in Computer Science, pages 19–30. Springer Verlag, 2000.
- [23] Y. Zheng and X. M. Zhang. Improved upper bound on the nonlinearity of high order correlation immune functions. In *Selected Areas in Cryptography - SAC 2000*, number 2012 in Lecture Notes in Computer Science, pages 264–274. Springer Verlag, 2000.
- [24] 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.

Appendix A - Description of Simulated Annealing

In 1983 Kirkpatrick et al. [8] proposed *simulated annealing*, a new search technique inspired by the cooling processes of molten metals. It merges hill-climbing with the probabilistic acceptance of non-improving moves. The basic algorithm is shown in Figure 2. 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 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 $\delta < 0$ 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 paper, 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 (i.e., without a single move having been accepted). Generally the best state achieved so far is also 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 to multiply by a constant cooling factor α in the range $(0..1)$; this is known as *geometric cooling*. The basic simulated annealing algorithm has proven remarkably effective over a range of problems.

```

S := S0
T := T0
repeat
{
  for(int i = 0; i < MIL; i++)
  {
    Select Y ∈ N(S)
    δ = f(Y) – f(S)
    if (δ < 0) then
      S = Y
    else
      Generate U := U(0, 1)
      if (U < exp(–δ/T)) then S := Y
  }
  T = T × α
}
until stopping criterion is met

```

Figure 2: Basic Simulated Annealing for Minimization Problems

Appendix B : Truth table of the functions

We present the truth tables in Hex. For RSBFs on 9 variables there are 2 classes of size 1 (containing respectively, the elements 000000000 and 111111111), 2 of size 3, and 56 of size 9. Simple mathematics reveals that for Walsh value of 32 there are 15 classes of size 9 and 1 of size 1, for Walsh value of –32 there are 13 of size 9 and 1 of size 3, and for Walsh value of 0 there are 28 of size 9, one of size 3 and 1 of size 1. All classes with elements of weight less than or equal to 3 can be fixed to zeroes. However, this search was unsuccessful but by removing the balance constraint,

i.e., $W_f(\bar{0})$, we could easily obtain unbalanced 3rd order CI functions with nonlinearity 240 and algebraic degree 5. This function has never previously been demonstrated. On the other hand, by allowing a spectrum with value of +16 and -16, we obtained a (9, 2, 6, 240) function.

The truth table of one of the evolved (9, 2, 6, 240) RSBF functions is given below. The value of Δ_f is 152. The annealing parameters are as follows $MIL = 1000$, $MaxIL = 800$, $MUL = 50$ and $\alpha = 0.95$. The initial pattern had 14 classes with Walsh value 32, with value -32, and 8 with value 16 and 8 with value -16. All classes with weight ≤ 2 were fixed at 0. The singleton class containing 11111111 was fixed at value 32. The average time for each run was 22 minutes.

```
0576 7a2d 6bcc 0da6 6c8b b1a0 45b7 d87c
79b5 c0db 9a17 9c44 6027 de3a b681 3fa1
3ec3 9a32 a150 b7ce d78d 476a 83e4 6171
6944 1c7a f3b8 1a9d 9a6c d447 4aeb 8917
```

The truth table of one of the evolved [9, 3, 5, 240] RSBF functions is given below. The value of Δ_f is 80. The annealing parameters are as follows $MIL = 1000$, $MaxIL = 1000$, $MUL = 50$ and $\alpha = 0.98$. There were 13 successes from 50 runs. The average time for each run was 22 minutes. Later experiments (with $MIL = 8000$, $MUL = 200$ and $MaxIL = 2000$) produced 8 successes from 10 runs. The average time for each run was 1 hour 10 minutes.

```
177a 2e98 18a9 d2d1 43c5 9c92 b649 a647
640a b467 c7f4 8709 db38 24d6 8839 712e
7c24 41cd 9b64 692b a06b be31 d42e 05d2
e29b 5a91 0965 e23c 85c5 1b86 3b52 5ce8
```

The truth table of an evolved bent RSBF on 10 variables is given below. The annealing parameters are as follows $MIL = 2000$, $MaxIL = 1000$, $MUL = 50$ and $\alpha = 0.98$. The average time for each run was 55 minutes.

```
7ffa ab9d dc9a d6f2 b2a1 82d8 e238 bb5c
cb19 9957 c11c b2c4 b958 0fc4 8fce 63e5
e1ce 0283 8393 277b b006 12f0 de5d b020
df97 76c0 05be e425 c1ef b0a9 285a a932
fc17 e4ec 410c 845e 955f 964a 5c3a 3a9b
9a01 153d 424c ef40 f7e8 36a3 9b41 0c14
e7aa 922e 7f68 e445 4177 8ab8 f861 5926
b042 f8ab 9f04 c9d2 48c1 729c 89c6 1a49
```

The truth table in Hex (lowest index first) of an evolved (10, 3, 5, 480) RSBF is given below having $\Delta_f = 192$. The annealing parameters are as follows $MIL = 1000$, $MaxIL = 1000$, $MUL = 50$ and $\alpha = 0.98$. In 10 runs one success was achieved. The average time for each run was 55 minutes.

```
69c6 a578 9c23 6ad5 92b5 1d4b 7dc8 a227
d31c 8b36 56a7 61ca 2ea3 e4d4 9958 593e
b64b 16e1 91da 4f29 662c dc3a 2957 e1d8
19e8 d81f ac35 b664 c6d3 3385 6696 4ba9
9a3c 618f 4779 f803 d616 e398 21ae 5cd6
6969 5cb1 a2e4 1f9c 58c7 276a ad53 a691
4693 bdc1 e694 12ee 99e5 0a76 cb38 3d25
a579 a64a 5b0e 9473 3d28 d26d 70cb c996
```