A Parallel Approach in Computing Correlation Immunity up to Six Variables

Carole J. Etherington[1], Matthew W. Anderson[2],
Eric Bach[3], Jon T. Butler[1], Pantelimon Stănică[4]

[1]ECE Department and [4]Applied Mathematics Department,
Naval Postgraduate School, Monterey, CA 93943, U.S.A.
Email: carole.etherington@navy.mil,{jbutler,pstanica}@nps.edu
[2]Computer Science Department, Union College
807 Union Street, Schenectady, NY 12308; Email: andersm2@union.edu
[3]Computer Sciences Department, University of Wisconsin
Madison, WI 53706, U.S.A.; Email: bach@cs.wisc.edu

March 10, 2015

Abstract

We show the use of a reconfigurable computer in computing the correlation immunity of Boolean functions of up to 6 variables. Boolean functions with high correlation immunity (in addition to other cryptographic properties) are desired in cryptographic systems because they are immune to correlation attacks. The SRC-6 reconfigurable computer was programmed in Verilog to compute the correlation immunity of functions. This computation is performed at a rate that is 190 times faster than a conventional computer.

Our analysis of the correlation immunity is across all n-variable Boolean functions, for $2 \leq n \leq 6$, thus obtaining, for the first time, a complete distribution of such functions. We also compare correlation immunity with two other cryptographic properties, nonlinearity and degree.

Keywords. reconfigurable computer, configurable computing, cryptographic Boolean functions, correlation immunity, rotation symmetric Boolean functions

1 Introduction

The correlation immunity of a Boolean function measures the extent the variable values can be guessed given the function value. When Boolean functions are used in encryption, functions with high correlation immunity (along with other cryptographic properties) are preferred, since they are less susceptible to an attack than functions with low correlation immunity. Interest in this topic developed because Siegenthaler [28] in 1984 showed how an attack can be effectively applied to encryption systems using functions with low correlation immunity.

Correlation immune (CI) functions are also used in machine learning (see [1, 22]): a "greedy" method to obtain a decision tree representation of a Boolean function given just a set of input-output pairs proceeds by choosing (recursively) a node of the tree to maximize a cost indicator

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
10 MAR 2015		**00-00-2015 to 00-00-2015**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Parallel Approach in Computing Correlation Immunity up to Six Variables	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,ECE Department ,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We show the use of a recon gurable computer in computing the correlation immunity of Boolean functions of up to 6 variables. Boolean functions with high correlation immunity (in addition to other cryptographic properties) are desired in cryptographic systems because they are immune to correlation attacks. The SRC-6 recon gurable computer was programmed in Verilog to compute the correlation immunity of functions. This computation is performed at a rate that is 190 times faster than a conventional computer. Our analysis of the correlation immunity is across all n-variable Boolean functions, for 2 n 6, thus obtaining, for the rst time, a complete distribution of such functions. We also compare correlation immunity with two other cryptographic properties, nonlinearity and degree.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**17**	

(information gain). However, if the function happens to have non-zero correlation immunity, this cost function is zero and thus not useful; i.e., a decision tree representation cannot be obtained in the case of a function that has non-zero correlation immunity. Since most n-variable functions have zero correlation immunity for $n \geq 2$, the greedy method works for most functions.

We show that a reconfigurable computer is effective in enumerating Boolean functions according to their correlation immunity. Especially, we can compare Boolean functions with respect to various cryptographic properties, including nonlinearity and degree due to prior use of a reconfigurable computer in computing these cryptographic properties [27]. Since Rothaus' original paper on bent functions in 1976 [23], there has been much work on the cryptographic properties of Boolean functions [11]. Such properties include strict avalanche criterion [14, 30], propagation criteria [20], and algebraic immunity [9, 10]. We have previously shown a 60,000× speed-up in using a reconfigurable computer to compute bent functions [27].

2 Some definitions

In this paper, we use the Landau symbol O with its usual meaning. Specifically, $f = O(g)$ means $|f(x)| < c|g(x)|$ holds with some constant c, for x sufficiently large. Also, we write $f \sim g$ if $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$.

Let $\mathbb{F}_2$ be the prime field of characteristic 2. For any positive integer n, the set $[n] := \{1, \ldots, n\}$. Let $\mathbb{F}_2^n = \{\mathbf{x} = (x_1, \ldots, x_n) : x_i \in \mathbb{F}_2, \text{ for all } i \in [n]\}$ be the vector space of dimension n over $\mathbb{F}_2$. Any function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ is said to be a *Boolean function* on n variables, whose set is denoted by $\mathfrak{B}_n$. Addition over $\mathbb{F}_2$ and $\mathbb{F}_2^n$ are both denoted by $\oplus$ whereas addition over integers is denoted by $+$. For any $\mathbf{x} \in \mathbb{F}_2^n$, the weight of $\mathbf{x}$ is $\mathrm{wt}(\mathbf{x}) = \sum_{i=1}^n x_i$. The algebraic normal form (ANF) of a Boolean function $f \in \mathfrak{B}_n$ is

$$f(x_1, \ldots, x_n) = \bigoplus_{\mathbf{a}=(a_1,\ldots,a_n)\in\mathbb{F}_2^n} \mu_{\mathbf{a}} x_1^{a_1} \ldots x_n^{a_n},$$

where $\mu_{\mathbf{a}} \in \mathbb{F}_2^n$, for all $\mathbf{a} \in \mathbb{F}_2^n$, and where $x_i^{a_i} = 1$ if $a_i = 0$ and $x_i^{a_i} = x_i$ if $a_i = 1$. The algebraic degree of f is $\deg(f) = \max_{\mathbf{a}\in\mathbb{F}_2^n}\{\mathrm{wt}(\mathbf{a}) : \mu_{\mathbf{a}} \neq 0\}$. The *Fourier transform* or the *Fourier coefficient* of $f \in \mathfrak{B}_n$ at $\mathbf{u} \in \mathbb{F}_2^n$ is

$$\widehat{f}(\mathbf{u}) = 2^{-n} \sum_{\mathbf{x}\in\mathbb{F}_2^n} (-1)^{f(\mathbf{x})} (-1)^{\mathbf{u}\cdot\mathbf{x}},$$

where $\mathbf{u} \cdot \mathbf{x} = \bigoplus_{i=1}^n u_i x_i$ is the inner product of $\mathbf{u} = (u_1, \ldots, u_n)$ and $\mathbf{x} = (x_1, \ldots, x_n)$. The *Walsh–Hadamard transform* of $f \in \mathfrak{B}_n$ at $\mathbf{u} \in \mathbb{F}_2^n$ is $W_f(\mathbf{u}) = 2^n \widehat{f}(\mathbf{u})$.

The set of Fourier coefficients $\{\widehat{f}(\mathbf{u}) : \mathbf{u} \in \mathbb{F}_2^n\}$ is said to be the *Fourier spectrum* of f. The set of Walsh–Hadamard coefficients $\{W_f(\mathbf{u}) : \mathbf{u} \in \mathbb{F}_2^n\}$ is said to be the *Walsh–Hadamard spectrum* of f. These transforms are invertible, that is, for all $\mathbf{x} \in \mathbb{F}_2^n$,

$$(-1)^{f(\mathbf{x})} = 2^{-n} \sum_{\mathbf{u}\in\mathbb{F}_2^n} W_f(\mathbf{u})(-1)^{\mathbf{u}\cdot\mathbf{x}} = \sum_{\mathbf{u}\in\mathbb{F}_2^n} \widehat{f}(\mathbf{u})(-1)^{\mathbf{u}\cdot\mathbf{x}}.$$

The Walsh-Hadamard spectrum of any Boolean function $f \in \mathfrak{B}_n$ is constrained by Parseval's identity

$$\sum_{\mathbf{u}\in\mathbb{F}_2^n} W_f(\mathbf{u})^2 = 2^{2n}. \tag{1}$$

Since, it will be used later, we introduce below the $2^n \times 2^n$ Walsh-Hadamard matrix $H_n = \left((-1)^{b(i)\cdot b(j)}\right)_{0\le i,j\le 2^n-1}$ ($b(i)$ is the binary expansion of i written as a vector with n components), or inductively, $H_1 = (1)$, $H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and, in general, $H_n = H_1 \otimes H_{n-1} = \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix}$ ('$\otimes$' is the Kronecker product). It is known that $W_f = H_n(-1)^f$.

The weight of a Boolean function is the weight of its truth (output) table. An n-variable function f is *balanced* if its truth table has as many 0's as 1's, that is, its weight is exactly 2^{n-1}.

Example 1. *The weight of the AND function $f(\mathbf{x}) = x_1x_2\cdots x_n$ is 1. The weight of the exclusive OR function $f(\mathbf{x}) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$ is 2^{n-1}. The exclusive OR function is balanced.*

An n-variable function f has *correlation immunity of order* $0 \le k \le n$ if and only if, for every fixed set S of k variables, and for every assignment of values to the variables in S, the weights of all subfunctions are the same. An n-variable function f *is resilient of order* k if it is balanced and has correlation immunity of order k.

An n-variable function f is *correlation immune* if and only if its correlation immunity k is 1 or more. An n-variable function f is *resilient* if and only if it is balanced and correlation immune. When a function has correlation immunity (resiliency) k but not $k+1$, we will describe such a function as *exact correlation immune (resilient) of order* k.

The following result characterizes correlation immunity (resiliency) (see [11]).

Theorem 1. *The following are true.*

(*i*) *An n-variable function f has correlation immunity (respectively, resiliency) of at least order k if and only if $W_f(\mathbf{a}) = 0$, for all $\mathbf{a} \in \mathbb{F}_2^n$ with $1 \le \mathrm{wt}(\mathbf{a}) \le k$ (respectively, $0 \le \mathrm{wt}(\mathbf{a}) \le k$).*

(*ii*) *An n-variable function f has correlation immunity of at least order k if and only if $f \oplus x_{i_1} \oplus x_{i_2} \oplus \cdots \oplus x_{i_k}$ is balanced for all $1 \le i_1 < \ldots < i_k \le n$.*

Correlation immunity describes the extent to which the variable values can be guessed, given the function value. A function that has low correlation immunity is the AND function on $n > 1$ variables. For example, if this function's output value is 1, then the input variable values are $(x_1, x_2, \ldots, x_n) = (1, 1, \ldots, 1)$ with probability 100%. On the other hand, when this function's output value is 0, there is a large uncertainty as to which variable values caused this (from among $2^n - 1$ values). In a function with high correlation immunity, knowing the function's value yields an equal uncertainty as to the variables' values that produced that function's value. For example, in the exclusive OR function $f = x_1 \oplus x_2 \oplus \cdots \oplus x_n$, half of the assignments of values to the variables yield $f = 0$ and half yield $f = 1$. Therefore, knowing that $f = 0$ or $f = 1$ yields the same uncertainty regarding the assignment of values to the variables. If we choose *any pair* of variables and any of the four assignments of values to the pair (00,01,10,11), we also have an equal uncertainty as to which of the remaining assignments yield $f = 0$ and $f = 1$.

Consider the exclusive OR function $f = x_1 \oplus x_2 \oplus \cdots \oplus x_n$, and consider a subset S of $1 \le k < n$ variables. By symmetry, these might as well be the last k. When they are fixed, the resulting subfunction is either $x_1 \oplus \cdots \oplus x_{n-k}$, or its complement. Either choice has weight 2^{n-k-1}. Therefore, f has correlation immunity of order k. Note, however that it does not have order n correlation immunity, because, by fixing the (unique) set of variables of cardinality n, the function becomes constant (either $0, 1$), which are evidently, not balanced. It follows that

this function has exact correlation immunity of $n-1$. We shall see later that the algebraic degree and immunity are constrained.

We recall here that a *barbell function* is the function $\bar{x}_1\bar{x}_2\cdots\bar{x}_n \oplus x_1x_2\cdots x_n$ or its complement. A *threshold function* is a function $f_{\mathbf{w},T}$ such that $f_{\mathbf{w},T}(\mathbf{x})$ is 1 if the weighted sum $\sum_{i=1}^{n} w_i x_i \geq T$, where x_i is viewed as an integer equal to its logic value and w_i and T are real numbers. The *Achilles heel* function, $f = x_1x_2 \oplus x_3x_4 \oplus \cdots \oplus x_{n/2-1}x_{n/2}$, for n even, is known to have a binary decision diagram whose number of nodes is especially sensitive to the ordering of its variables [3].

Table 1 shows the correlation immunity of several example functions, including the barbell, threshold, and Achilles heel functions. Since functions with odd 1's all have correlation immunity 0, more than one-half of the Boolean functions have correlation immunity 0.

Table 1: Correlation immunity of some example n-variable Boolean functions.

Function Description	Expression	Correlation Immunity
Constant Functions	$f = 0$, $f = 1$	n
Parity Functions	$f = x_1 \oplus x_2 \oplus \cdots \oplus x_n$, $f \oplus 1$	$n-1$
Barbell Functions	$f = x_1x_2\cdots x_n \oplus \bar{x}_1\bar{x}_2\cdots\bar{x}_n$, $f \oplus 1$	1
Functions With Odd Weight	e.g., $f = x_1x_2\cdots x_n$, $f = x_1 \vee x_2 \vee \cdots \vee x_n$	0
Threshold Functions	e.g., $f = x_1x_2 \vee x_1x_3 \vee x_2x_3$	0
Achilles Heel Function	$f = x_1x_2 \oplus x_3x_4 \oplus \cdots \oplus x_{n/2-1}x_{n/2}$, n even, $f \oplus 1$	0

In applying the definition of the correlation immunity to determine the correlation immunity of a given function, it was convenient to specify a two-step process. In the first step, we specify that a condition hold for all k'-subsets of variables. In the second step, we specify that a condition hold across all assignments of values to the variables chosen in the first step. For pedagogical reasons, we could view this as a one-step process. That is, we could think of *simultaneously* choosing a k'-subset and some assignment of values to the variables. In this way, we produce one of the $\binom{n}{k'}2^{k'}$ subfunctions of f. The definition of correlation immunity then requires that *all* of these $\binom{n}{k'}2^{k'}$ subfunctions have the *same* weight. The maximum k' for which this is true is the exact correlation immunity of the function. This viewpoint will be useful in the description of the circuit to compute correlation immunity in Section 4.

3 Some results on the number of correlation immune and resilient functions

Various results are known about the tradeoff among cryptographic properties involving correlation immunity. For example, if f is a Boolean function in n variables that has correlation immunity of order k, then 2^k divides the Hamming weight of f. Also, if f is a Boolean function in n variables, that has correlation immunity of order k, then the degree d of f is at most $n-k$. If further, f is balanced and correlation immune of order 1 or more (hence, resilient) and $k < n-1$, then the degree of f is at most $n-k-1$. Other more esoteric results exist, like the fact that

if f has correlation immunity k, then the algebraic normal form (positive polarity Reed-Muller form) of f either has no terms of degree $n-k$ or has all possible terms of degree $n-k$.

Camion et al. [4] attempted a nice recursive approach for the construction of a resilient function f on $n+1$ variables. It is based on the Shannon decomposition of a Boolean function f, where $f = \bar{x}_n f_0 \vee x_n f_1$, such that $f_0 = f|_{0\to x_n}$ and $f_1 = f|_{1\to x_n}$. A Boolean function is $(k+1)$-resilient if and only if the following two conditions hold:

(*i*) f_0 and f_1 are resilient functions of order k;

(*ii*) for all n component vectors $\mathbf{v}$ of weight $k+1$, the Walsh–Hadamard transform equation $W_{f_0}(\mathbf{v}) + W_{f_1}(\mathbf{v}) = 0$ holds.

Also, if the degrees of f, f_0 and f_1 are equal (so, $\deg(f_0 \oplus f_1) < \deg(f)$, because otherwise, we would have $\deg(f_0 + f_1) = \deg(f)$, but that is impossible since $f = f_0 + x_n(f_0 + f_1)$ and so, f would increase its degree), then f has its maximum degree $n+1-(k+2)$ if and only if f_0 and f_1 have their maximum degree $n-(k+1)$.

Improving upon an interesting result obtained by Sarkar and Maitra [24], Carlet [6] showed the following theorem.

Theorem 2 ([6, 24])**.** *If a degree $d \geq 1$, n-variable function f has correlation immunity, respectively, resiliency of order k, then its Walsh–Hadamard coefficients are divisible by $2^{k+1+\lfloor \frac{n-k-1}{d} \rfloor}$, respectively, $2^{k+2+\lfloor \frac{n-k-2}{d} \rfloor}$. Moreover, the nonlinearity N_f of an order k correlation immune, respectively, resilient function f satisfies*

$$2^{k+\lfloor \frac{n-k-1}{d} \rfloor} \mid N_f \leq 2^{n-1} - 2^k, \text{ respectively,}$$
$$2^{k+1+\lfloor \frac{n-k-2}{d} \rfloor} \mid N_f \leq 2^{n-1} - 2^{k+1}.$$

This easily implies that functions whose correlation immunity is at least 1 have even nonlinearity, and if, in addition, they are balanced, their nonlinearity is divisible by 4.

Let $CI(n,k)$ (respectively, $BCI(n,k)$) be the number of exact order k correlation immune, (respectively, further balanced) n-variable Boolean functions. The notations $CI(n,k,d)$, $BCI(n,k,d)$ restricts the previous count to degree d Boolean functions.

Theorem 3. *The following are true:*

(*i*) $BCI(n,n,0) = 0, CI(n,n,0) = 2$, $CI(n,k,1) = BCI(n,k,1) = 2\binom{n}{k+1}$, $0 \leq k \leq n-1$.

(*ii*) $BCI(n,n-2) = 2\binom{n}{n-1} = 2n$.

(*iii*) $BCI(n,n-3) = \frac{n(n-1)(3n-2)(n+1)}{3} + 2\binom{n}{n-2}$.

(*iv*) $BCI(n,k,d) = 0$, *if* $n > (n-k-1)2^{d-1}$*; in particular,* $BCI(n,k,2) = 0$, *for all* $k \geq \frac{n}{2}$.

(*v*) $CI(n,n-\ell,2) = 0$, *if* $n > 4k-5$*; in particular,* $CI(n,n-2,2) = 0$, *if* $n > 3$.

Proof. We first show (*i*). Let f be an affine function, $f(\mathbf{x}) = \bigoplus_{i=1}^{n} c_i x_i \oplus c$, $c_i, c \in \mathbb{F}_2$. Certainly, correlation immunity is preserved by complementation. So, from here on, we always assume that the constant $c = 0$. We next take K to be the exact number of nonzero coefficients, say $c_{i_j} = 1, 1 \leq j \leq K$. If $K = 0$, then f is constant (hence, non-balanced) and we see that the constant functions $\mathbf{1}, \mathbf{0}$ are correlation immune of order n.

If $K > 0$, then f is balanced. So, the number of correlation immune functions of whatever order will match the number of resilient functions of the same order. Using Theorem 1, we see that f is not correlation immune (resilient) of order K, since $f \oplus x_{i_1} \oplus \cdots \oplus x_{i_K} = 0$. Hence, f is not balanced. However, f is correlation immune (resilient) of order $k := K - 1$, since then $f \oplus x_{i_1} \oplus \cdots \oplus x_{i_j} = 0$, $j \leq K - 1$ will be nonzero and linear, hence balanced. There are $\binom{n}{k+1}$ ways of choosing the nonzero coefficients, and (i) follows. Certainly, (ii) follows by the same argument, since we know that a function that is resilient of order $n - 2$ must have degree ≤ 1.

Next, the first term from claim (iii) was found in [4] and counts the number of resilient of order $n - 3$ *quadratic* Boolean functions. The second term corresponds to affine resilient functions of order $n - 3$, and follows from (i).

The item (iv) is [29, Theorem 5 and Theorem 7]. Next, we show (v). We recall the interesting upper bound of Tarannikov et al. [29] for the correlation immunity order k of an *unbalanced* Boolean function in n variables, namely

$$k \leq \frac{3n - 5}{4}. \tag{2}$$

For $k := n - 2$, $n \geq 4$, this would imply $n - 2 \leq \frac{3n-5}{4}$, which contradicts $n \geq 4$, and this shows that there are no unbalanced quadratic Boolean functions that are correlation immune of order k. If f is balanced and correlation immune of order $n - 2$, then we can apply (iv), or observe that the degree of f cannot exceed 1 and so, f cannot be quadratic. □

Remark 4. *Bierbrauer and Friedman* [2, 15] *found the following bound on the Hamming weight of a function f that is correlation immune of order k*

$$wt(f) \geq 2^n \frac{2(k+1) - n}{2(k+1)},$$

which gives further constraints on the parameters of a correlation immune Boolean function.

Denisov [12] found that the number of n-variable correlation immune functions of order k, say $CI(n, k)$, is asymptotically

$$CI(n, k) \sim 2^{2^n + Q - k} (2^{n-1}\pi)^{-(M-1)/2},$$

where $M = \sum_{j=0}^{k} \binom{n}{j}$ and $Q = \sum_{j=1}^{k} j\binom{n}{j}$. Denisov published a "correction" in 2000 (see [13]), but it turns out that his original result was correct and the latter paper is incorrect, as was shown by Canfield et al. [5]. For $k = 1$, one can get a simpler estimate

$$CI(n, 1) \sim D_n = \frac{1}{2}\left(\frac{8}{\pi}\right)^{n/2} 2^{2^n - n^2/2}, \text{ as } n \to \infty.$$

A more refined version of the approximation for $CI(n, 1)$ was computed by Bach [1]

$$CI(n, 1) = D_n \left(1 - \frac{n^2}{2^{n+2}} + O\left(\frac{n^4}{2^{2n}}\right)\right).$$

Denisov [12] also showed that the number $BCI(n, k)$ of balanced and correlation immune (resilient) functions, satisfies the asymptotic formula

$$BCI(n, k) \sim 2^{2^n - \binom{n}{k}(n-k)/2} (2/\pi)^{M/2}, \text{ as } n \to \infty,$$

where $M = \sum_{j=0}^{k} \binom{n}{j}$.

There are various results concerning bounds on the number of correlation immune and/or resilient functions and the interested reader can find some in the listed references (or elsewhere). Also, Yang and Guo [31] found in 1995 that the number of correlation immune functions of order 1 is upper bounded by

$$CI(n,1) \leq \sum_{k=0}^{2^{n-1}} \sum_{r=0}^{k} \binom{2^{n-2}}{r}^2 \binom{2^{n-2}}{k-r}^2.$$

Le Bars and Viola [16] found a lower bound for the number of resilient Boolean functions of order 1, namely

$$BCI(n,1) \geq 2^{2^n - n^2 + \frac{3}{2}n + 1} e^{\frac{1}{2} - n} (n\pi)^{n/2}.$$

A quick analysis of this lower bound gives us a 'glimpse' at the complexity of completely enumerating all resilient functions for $n = 6, 7$: if $n = 6$ there are more than $2^{42.77}$, and for $n = 7$, there are more than $2^{96.72}$ resilient Boolean functions. By using a construction of a resilient of order k function in n variables from a resilient of order 1 in $n - k + 1$ variables, Le Bars and Viola found (for free) the following lower bound for the number of resilient of order k functions

$$BCI(n,k) \geq 2^{2^{n-k+1} - (n-k+1)^2 + \frac{3}{2}(n-k+1) + 1} e^{k - n - \frac{1}{2}} ((n-k+1)\pi)^{(n-k+1)/2},$$

which can be combined with Schneider's bound [25, 26] for $0 < k < n$,

$$BCI(n,k) \leq \prod_{j=1}^{n-k} \binom{2^j}{2^{j-1}}^{\binom{n-j-1}{k-1}}.$$

The previous bound is rather weak for high order of resiliency, and it was slightly improved by Carlet and Klapper [8], who showed that $BCI(n,k)$ is upper bounded by

$$\begin{cases} \dfrac{2^{\sum_{i=0}^{n-k-1} \binom{n}{i}} - 2^{\sum_{i=0}^{n-k-2} \binom{n}{i}}}{2^{2^{2k+1}-1}} + 2^{\sum_{i=0}^{n-k-2} \binom{n}{i}}, & \text{for } 2 \leq k < n/2 \\ 2^{1+\sum_{i=0}^{n-k-1} \binom{n}{i} - \sum_{i=0}^{n-k-1} \binom{k-1}{i}} (1+\epsilon) + 2^{\sum_{i=0}^{n-k-2} \binom{n}{i}}, \epsilon = 2^{-\Omega((2^n/n)^{1/2})}, & \text{for } n/2 \leq k < n \end{cases}$$

and Carlet and Gouget [7], who showed that $BCI(n,k)$ is upper bounded by

$$2^{\sum_{i=0}^{n-k-2} \binom{n}{i}} + 2^{-\binom{k+1}{n-k-1}-1} \binom{n}{n-k-1} \prod_{j=1}^{n-k} \binom{2^j}{2^{j-1}}^{\binom{n-j-1}{k-1}}.$$

We further point to [18] for an alternative representation of resilient functions.

Unfortunately, all of these bounds, and asymptotics for $n \to \infty$, simply estimate counts of the correlation immune and resilient functions. They say nothing about the size of the aforementioned sets for a small number of variables n. Indeed, there are no known expressions for the *exact* counts $CI(n,k)$ and/or $BCI(n,k)$ for $n \geq 6$ (and all $k > 1$). We show that a reconfigurable computer, combined with the theoretical results can tractably compute the correlation immunity of functions exhaustively along with other cryptographic properties. Thus, we can compare their correlation immune/resilient properties, and compare against other cryptographic properties, such as nonlinearity and degree.

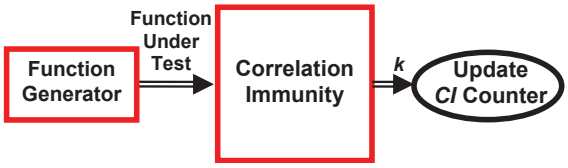

Figure 1: Block diagram of circuit for computing correlation immunity.

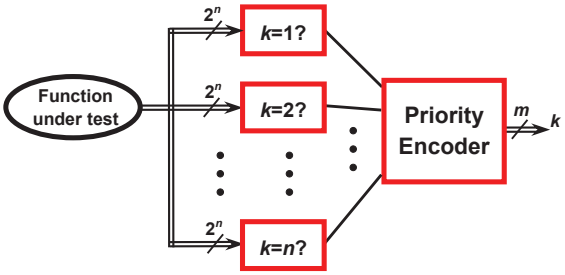

Figure 2: Breakdown of correlation immunity computation circuit.

4 Computation of correlation immunity

A Verilog program was written to compute the correlation immunity of Boolean functions on the SRC-6 reconfigurable computer. Because of the large logic resources available, it was possible to implement the correlation immunity computation for one function per clock period. With a clock frequency of 100 MHz, we can compute a function's correlation immunity at a rate of 100,000,000 functions per second. Later, we compare this to a conventional processor.

Fig. 1 shows a block diagram of correlation immunity computation circuit. The block on the left labeled "Function Generator" generates the truth table of the function whose correlation immunity is currently being computed. When this circuit is used in exhaustive enumeration, the Function Generator is an up counter. The block labeled "Correlation Immunity" is a combinatorial logic circuit whose input is the truth table of a function and whose output is the value of its exact correlation immunity. The oval labeled "Update CI Counter" represents that part of the system that records the correlation immunity. It records each contribution to the histogram of the number of functions with various values of correlation immunity.

Fig. 2 shows a block diagram of the combinatorial logic block in Fig. 1 labeled "Correlation Immunity". The truth table of the function under test is applied on the left to n blocks labeled "$k = \alpha$?", where $1 \leq \alpha \leq n$. Each block tests whether the function has correlation immunity α and produces a 1 if and only if the function has correlation immunity α. This output is applied to a priority encoder that produces at its output a value that is the largest α such that the block labeled "$k = \alpha$?" produces a 1. m, the number of lines in the output bus labeled "k" is $\lceil \log_2 n \rceil$ and represents the number of bits needed to represent a number between 0 and n.

Fig. 3 shows the circuit that realizes the "$k = \alpha$?" circuit in Fig. 2. The line of blocks on the left are circuits that separate out the k'-subsets of variables. The blocks near the center extract the truth table of the subfunctions associated with assigning all combinations of values to the variables in each subset. Then, the blocks labeled "Ones Count" to the right compute the weight of each subfunction. Then, the single block on the right produces at is output a 1 if and only if all weights are the same. This drives the "$k = \alpha$?" output.

5 Meet in the middle algorithm

In this section, we describe an algorithm that can count the n-variable k-correlation immune functions, using $2^{2^{n-1}+O(n)}$ time and space. Effectively, n is reduced by 1 but a high price is paid in memory. This makes it unsuitable for a reconfigurable computer, such as the SRC-6. It

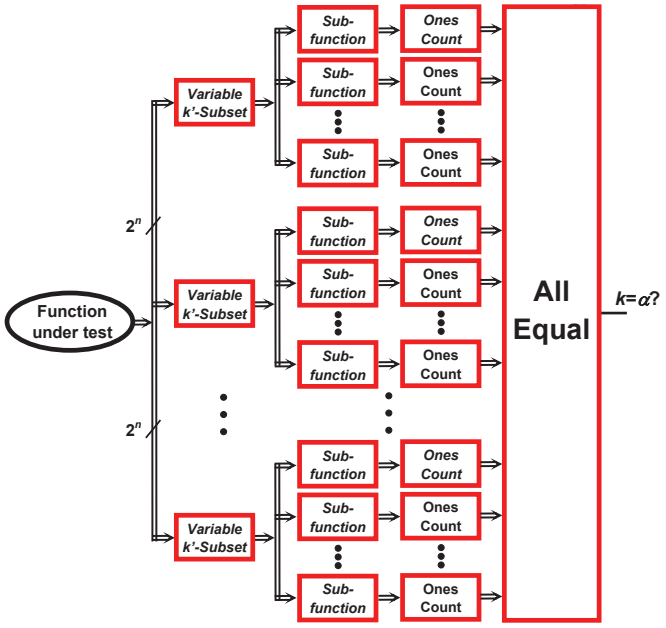

Figure 3: Correlation immunity circuit.

did serve, however, for completing the analysis for $n = 6$. This algorithm is described in [17], but only briefly, so we elaborate here.

Recall that the conditions for k-immunity ((i) of Theorem 1) are linear. Therefore, we can split our truth tables in two, and attempt to find matching left and right halves.

Let $m = n + \binom{n}{2} + \cdots + \binom{n}{k}$. From the Walsh-Hadamard matrix, extract the rows indexed by $\mathbf{u}$ with $1 \le \mathrm{wt}(\mathbf{u}) \le k$, to form $W^{(k)}$. If we write $W^{(k)} = (A\ B)$, the Walsh-Hadamard condition for k-immunity becomes $(A\ B)(\mathbf{x}\ \mathbf{y})^T$, where A, B are $m \times 2^{n-1}$ matrices with ± 1 entries, and the column vector $(\mathbf{x}\ \mathbf{y})^T$ is the truth table of f (in ± 1 form). Equivalently, the two "signatures" $A\mathbf{x}$ and $-B\mathbf{y}$ must match.

Make a list of the 2^{n-1} pairs $(A\mathbf{x}, 0)$ and another list of the 2^{n-1} pairs $(-B\mathbf{y}, 1)$. (The "tag" (0 or 1) indicates which matrix the pair came from.) Sort the combined lists lexicographically. A first component $\mathbf{z}$ that occurs r times with a 0 and s times with a 1 contributes rs to the count of k-correlation immune functions. (If we append $\mathbf{x}$ and $\mathbf{y}$ to the pairs, the actual functions could be produced as well.)

Here is an example. Take $n = 2$ and $k = 1$. Then, $A = \left(\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right)$ and $-B = \left(\begin{smallmatrix} -1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$. Applying these two matrices to the four vectors $(\pm 1, \pm 1)^T$, we get two lists, each with the four vectors $(0, \pm 2)$, $(\pm 2, 0)$. Thus, there are four correlation immune functions of two variables.

To get fast code, we can use the following idea. If $\mathbf{a}, \mathbf{b}$ are ± 1 vectors of length 2^{n-1}, and $\mathbf{u}, \mathbf{v}$ their 0/1 images (under the map that sends $+1$ to 0 and -1 to 1), we have

$$\sum_{i=1}^{2^{n-1}} a_i b_i = 2^{n-1} - 2\ \mathrm{wt}(\mathbf{u} \oplus \mathbf{v}).$$

For the last factor, bitwise XOR can be used, followed by a "1's count" operation. Since our goal is only to find matches, the rest of the operations can be skipped.

Since $0 \le \mathrm{wt}(\mathbf{u} \oplus \mathbf{v}) \le 2^{n-1}$, the information payload in each pair $(A\mathbf{x}, 0)$ and $(-B\mathbf{y}, 1)$ can be stored in a bit string of length $mn + 1$, if encoded in a straightforward way.

We now justify the time and space claims made above. For the $A\mathbf{x}$'s, we need $m2^{2^{n-1}}$ bitwise

XOR's, and $2^{2^{n-1}}$ 1's counts. The same number is needed for the $-B\mathbf{y}$'s. If we assume that there are instructions for XOR and 1's count, the complexity for this phase of the algorithm is $O(m2^{2^{n-1}})$. Then, we sort the combined table and make a final pass to count the matches. With standard in-place sorting algorithms, this costs $O(2^{2^{n-1}+n})$, if we reckon that a comparison is one step. The claimed bounds then follow, since $m \leq 2^n$.

If the machine does not have a 1's count instruction, this can be done in software at a cost of $O(n)$ (remember the word size is 2^n) [21]. This will not affect the result.

In practice, k will be small, since the ℓ-correlation immune functions could be culled from a list of k-immune functions with $k \geq \ell$. Also, since the balance condition is also linear, the same idea works for counting k-resilient functions.

We implemented three variations on this algorithm for $n = 6$ to do specialized counting jobs.

First, to count the 2-correlation immune functions, we spread the work over about 729 processors, using Wisconsin's Condor distributed computing system. Each processor was responsible for a subset of $\mathbf{x}$'s and a subset of $\mathbf{y}$'s, and selected possible matches by hashing the signatures. The $\mathbf{x}$'s and $\mathbf{y}$'s were included in the tuples, making it possible to verify alleged matches. For our partition of the data, $\mathbf{x}$ and $\mathbf{y}$ cannot match if they are on different processors, so the individual counts found by the processors could be summed at the end.

Second, we counted the correlation immune functions with degree ≤ 4. Applying the "esoteric result" at the beginning of Section 2 to $n = 6$, we see that a 2-correlation immune function has degree < 5 if and only if its ANF omits the quintic term $x_1x_2x_3x_4x_5$. It can be shown that this happens if the "combed" truth table (result of bitwise AND with 010101...) has even Hamming weight. This is another linear condition. Rather than add a row to our matrices, however, we just treated the pairs with even and odd combed Hamming weights separately, and summed the results. Conveniently, with base 33 encoding, the signatures fit into 32 bits, and the maximum imaginable signature ($33^6 - 1$) was small enough that we could sort by counting.

Finally, we counted the 2-resilient functions. To do this, we used Camion et al.'s result (see Section 2) that the left and right halves of any 2-resilient truth table are 1-resilient. Using this criterion as a filter, we made a table of 2 $BCI(5, 1)$ (about 1.6×10^6) tagged signatures, and then sorted it to get the desired count. By Theorem 2, all Hamming weights are even, so we stored halved weights (which cannot exceed 16). There were $\binom{6}{2} = 15$ of these, since we only needed weight 2 parities. Using base 17 encoding, each tagged signature fit into 63 bits. (Note that $17^{15} < 2^{62}$.) Therefore, 64 bit integer variables could be used.

6 The computational results

Table 2 shows the distribution of n-variable functions by exact order of correlation immunity, for $2 \leq n \leq 6$. This table clearly shows that the majority of functions have correlation immunity 0. The value of correlation immunity that has the next largest number of functions is 1. Also, Table 2 shows that, for all values of n, there are two functions with correlation immunity n. These are the constant functions $f = 0$ and $f = 1$, appearing in Table 1. Table 2 also shows there are two functions with correlation immunity $n-1$. These are the parity functions $f(\mathbf{x}) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$ and $f(\mathbf{x}) = 1 \oplus x_1 \oplus x_2 \oplus \cdots \oplus x_n$, also shown in Table 1.

The complete data for $n = 6$ is certainly new. We can, however, sum the functions with correlation immunity greater than 0, and thus derive the number of correlation immune functions. The result is shown in Table 3. These values are identical to those computed by Palmer et al. [19]) and Le Bars and Viola [16], which verifies our results. The number of correlation

Table 2: Distribution of n-variable functions by exact correlation immunity, k, for $2 \le n \le 6$.

n / k	0	1	2	3	4	5	6
2	12	2	2	0	0	0	0
3	238	14	2	2	0	0	0
4	64888	636	8	2	2	0	0
5	4291827234	3139004	1044	10	2	2	0
6	18446240589943529428	503483719470800	46549718	1654	12	2	2

immune functions for $n = 7$ is also known; it is 171522187398423323340476473786538 [16].

Table 3: Number of n-variable correlation immune and resilient functions, for $2 \le n \le 6$.

n	2	3	4	5	6
Cor. Imm.	4	18	648	3140062	503483766022188
Resilient	2	8	222	807980	95259103924394

Table 4 shows the distribution of n-variable balanced functions to exact correlation immunity, where $2 \le n \le 6$. This data is similar to that shown in Table 2, except that it applies *only* to balanced functions (whose function values have as many 0's as 1's). Thus, this table shows only the resilient functions. Note that there are no resilient functions with correlation immunity n. The only possible candidates are the constant functions in in Table 4, which are not balanced. However, there are two functions with correlation immunity $n-1$ in these tables. These are the parity functions, which *are* balanced. From the above enumeration, we can sum the functions with correlation immunity greater than 0, and thus compute the number of correlation immune functions that are balanced. These are the resilient functions. The result is shown in the second line in Table 3. The values are identical to those appearing in [19] and in [16]. Le Bars and Viola [16] have also determined that there are 2347801575478885443949762268929 6 1-resilient functions for $n = 7$.

Table 4: Distribution of n-variable balanced functions by exact resiliency, k, for $2 \le n \le 6$.

n / k	0	1	2	3	4	5	6
2	4	2	0	0	0	0	0
3	62	6	2	0	0	0	0
4	12648	212	8	2	0	0	0
5	600272410	807428	540	10	2	0	0
6	1832120657223119734	503483702719940	16749696	1150	12	2	0

A function f is *rotation symmetric* [11] if and only if for any values $(x_1, x_2, \ldots, x_n)$,

$$f(x_1, x_2, \ldots, x_n) = f(x_n, x_1, x_2, \ldots, x_{n-1}), \tag{3}$$

that is, the function is invariant under rotation of indices.

Example 2. *The four functions* $f(\mathbf{x}) = 0$, $f(\mathbf{x}) = 1$, $f(\mathbf{x}) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$, *and* $f(\mathbf{x}) = 1 \oplus x_1 \oplus x_2 \oplus \cdots \oplus x_n$ *(for all* $\mathbf{x} \in \mathbb{F}_2^n$*) are all rotation symmetric.*

Table 5 shows the distribution of n-variable balanced functions to exact correlation immunity k for rotation symmetric functions. The fraction of all functions that are rotation symmetric

Table 5: Distribution of n-variable rotation symmetric functions versus exact correlation immunity k and n, for $2 \leq n \leq 6$.

n / k	0	1	2	3	4	5	6
2	4	2	2	0	0	0	0
3	10	2	2	2	0	0	0
4	48	12	0	2	2	0	0
5	214	34	4	0	2	2	0
6	14656	1686	36	2	0	2	2

functions is small, and this is seen in the table. It is interesting that, for correlation immunity equal to n and $n-1$, there are two functions for all values of n shown. This is because all functions with these values of correlation immunity are rotation symmetric. Indeed, all of these functions are symmetric, which is a subtype of rotation symmetric functions.

Table 6 shows the distribution of 4-variable functions as a function of both correlation immunity and nonlinearity. The computation of the nonlinearity by reconfigurable computer is described in [27]. So, for each function we compute its correlation immunity, as described in this paper and its nonlinearity, as described in [27]. The functions with largest nonlinearity are the bent functions; for $n = 4$, there are 896 bent functions. As with the distributions discussed earlier, the majority of functions have a correlation immunity of 0. But, in this table, it can be seen how the functions are distributed according to nonlinearity. Most functions have nonlinearity near the middle values, 3 through 5. And, most of these are concentrated along the value of correlation immunity equal to 0. It is interesting that the largest concentration of functions with the highest correlation immunity (1 only) and relatively high nonlinearity occur at nonlinearity 4.

Table 6: Distribution of n-variable functions versus exact correlation immunity k and nonlinearity (N), for $n = 4$.

N / k	0	1	2	3	4
0	8	12	8	2	2
1	512	0	0	0	0
2	3712	128	0	0	0
3	17920	0	0	0	0
4	27504	496	0	0	0
5	14336	0	0	0	0
6	896	0	0	0	0

Table 7 shows data similar to that of Table 6 except that it is for $n = 5$. Interestingly, there are a relatively substantial number of functions, that is 384, with the highest nonlinearity 12 and moderate correlation immunity 2.

Table 8 shows the distribution of 4-variable functions versus exact correlation immunity k and degree. High degree in Boolean functions is a desired cryptographic property. The computation of degree is accomplished using the "transeunt triangle" [27]. This is a circuit consisting entirely of exclusive OR gates that transforms the truth table of a function to its ANF. Additional gates extract from the ANF a binary number that is the degree of the function. So, for each function, we compute its correlation immunity, as described in this paper and its degree as described

Table 7: Distribution of n-variable functions versus exact correlation immunity k and nonlinearity (N), for $n = 5$.

N / k	0	1	2	3	4	5
0	10	20	20	10	2	2
1	2048	0	0	0	0	0
2	31232	512	0	0	0	0
3	317440	0	0	0	0	0
4	2278400	23040	0	0	0	0
5	12888064	0	0	0	0	0
6	57873920	122368	0	0	0	0
7	215414784	0	0	0	0	0
8	645867160	1799080	640	0	0	0
9	1362452480	0	0	0	0	0
10	1411209216	890880	0	0	0	0
11	556408832	0	0	0	0	0
12	27083648	303104	384	0	0	0

here. Table 9 shows a distribution similar to that of Table 8 except that it is for 5-variable functions. There are a relatively substantial number of functions (384) with moderate degree (3) and moderate exact correlation immunity (2).

Table 8: Distribution of n-variable functions versus correlation immunity k and degree (Deg), for $n = 4$.

Deg / k	0	1	2	3	4
0	0	0	0	0	2
1	8	12	8	2	0
2	1712	304	0	0	0
3	30400	320	0	0	0
4	32768	0	0	0	0

In Table 10, the rows for $d \leq 3$ were computed on the SRC-6. This was combined with the count of 2-correlation immune functions to complete the $k = 2$ column. Since the number of 1-correlation immune functions is known, the total for $d \leq 4$ and $k \geq 1$ could be used to complete the $k = 1$ column. The remaining column was determined by subtraction.

Table 11 shows the time it takes to do the exhaustive enumeration across 4 variables. The first row shows that 0.655 msec. is needed to complete the enumeration on the SRC-6 reconfigurable computer; this corresponds to one function per clock cycle of a 100 MHz clock. The second row shows that 1,238.7 msec. is needed when a C program is compiled into Verilog using the SRC-6's compiler and run on the SRC-6's FPGA (Xilinx Virtex-II Series 6000). The third row shows that 190 msec. is needed by the same C program when it is run on a conventional processor (the SRC-6's 2.8 GHz Xeon microprocessor). The small time required in the case of a Verilog program shows a significant advantage in using the large logic resources of an FPGA. Compared to the conventional processor time, the SRC-6 programmed in Verilog has a 190 times speedup.

Table 9: Distribution of n-variable functions versus exact correlation immunity k and degree (Deg), for $n = 5$.

Deg / k	0	1	2	3	4	5
0	0	0	0	0	0	2
1	10	20	20	10	2	0
2	59736	5096	640	0	0	0
3	65478976	1563968	384	0	0	0
4	2078804864	1569920	0	0	0	0
5	2147483648	0	0	0	0	0

Table 10: Distribution of n-variable functions versus exact correlation immunity k and degree (Deg), for $n = 6$.

Deg / k	0	1	2	3	4	5	6
0	0	0	0	0	0	0	2
1	12	30	40	30	12	2	0
2	3760424	417512	15000	1240	0	0	0
3	4388747656096	9270073536	24586784	384	0	0	0
4	143859057441024156	251732566372708	21947904	0	0	0	0
5	9079005106896312932	251741882607004	0	0	0	0	0
6	9223372036854775808	0	0	0	0	0	0
Total	18446240589943529428	503483719470790	46549728	1654	12	2	2

7 Concluding Remarks

Correlation immunity is an important cryptographic property of Boolean functions. We show a fast circuit that allows a computation of correlation immunity of Boolean functions at a rate of 10^8 functions per second on the SRC-6 reconfigurable computer. In the case of 4-variable functions, this results in a 190 times speedup compared to a conventional computer. For the first time ever we are able to find the distribution of 6 variable functions versus the order of correlation immunity. We also can quickly analyze and compare Boolean functions on the basis of their cryptographic properties. Specifically, we compare correlation immunity with two other cryptographic properties, nonlinearity and degree, and obtain for the first time, a complete distribution of such functions for ≤ 6 dimensions.

Acknowledgments. Eric Bach was partially supported by NSF Grant CCF-1420750. Matt Anderson's work was done at the University of Wisconsin, partially supported by NSF Grant CCF-0523680. Thanks also to Barbara Hamilton of IDA for assistance with references.

References

[1] E. Bach, "Improved asymptotic formulas for counting correlation-immune Boolean functions", *SIAM J. Discrete Math.* 23 (2009), 1525–1538.

Table 11: Comparing computation time for correlation immunity over all 4-variable functions.

Computer/ Program	Time (msec.)
100 MHz FPGA/Verilog	0.655
100 MHz FPGA/C	1,238.7
2.8 GHz Xeon/C	190

[2] J. Bierbrauer, "Bounds on orthogonal arrays and resilient functions", *J. Combin. Des.* 3 (1995), 179–183.

[3] R. K. Brayton, G. D. Hachtel, C. T. McMullen, A. L. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*, Kluwer Acad. Publ., ISBN 0-89838-164-9, 1984.

[4] P. Camion, C. Carlet, P. Charpin, N. Sendrier, "On correlation immune functions", *Adv. in Crypt. – CRYPTO* '91, Springer–Verlag, 1991, pp. 86–100.

[5] E. R. Canfield, Z. Gao, C. S. Greenhill, B. D. McKay, R. W. Robinson, "Asymptotic enumeration of correlation immune Boolean functions", *Cryptogr. Commun.* 2:1 (2010), 111–126.

[6] C. Carlet, "On the coset weight divisibility and nonlinearity of resilient and correlation-immune functions", Proceedings of SETA'01 (Sequences and their Applications 2001), Discrete Mathematics and Theoretical Computer Science, Springer, 2001, pp. 131–144.

[7] C. Carlet, A. Gouget, "An upper bound on the number of m-resilient Boolean functions", *Adv. in Crypt. – Asiacrypt* 2002, LNCS 2501 (2002), pp. 484–496.

[8] C. Carlet, A. Klapper, "Upper bounds on the numbers of resilient functions and of bent functions", 23rd Symposium on Information Theory in the Benelux, Louvain-La-Neuve, Belgique, May, 2002.

[9] N. Courtois, "Fast algebraic attacks on stream ciphers with linear feedback", *Adv. in Crypt. - CRYPTO* 2003, Berlin, Germany, Springer-Verlag, LNCS 2729, 2003, pp. 176–194.

[10] N. Courtois, W. Meier, "Algebraic attacks on stream ciphers with linear feedback", *Adv. in Crypt. – Eurocrypt* 2003, Berlin, Germany, Springer-Verlag, LNCS 2656, 2003, pp. 345–359.

[11] T. W. Cusick, P. Stănică, *Cryptographic Boolean Functions and Applications*, Academic Press - Elsevier, March 2009.

[12] O. V. Denisov, "An asymptotic formula for the number of correlation immune of order k Boolean functions", *Discrete Math. Appl.* 2 (1992), 407–426.

[13] O. V. Denisov, "A local limit theorem for the distribution of a part of the spectrum of a random binary function", *Discrete Math. Appl.* 10 (2000), 87–101.

[14] R. Forré, "The strict avalanche criterion: Spectral properites of Boolean functions and an extended definition", *Adv. in Crypt. – CRYPTO* '88, Berlin, Germany, Springer-Verlag, LNCS 403, 1990, pp. 450–468.

[15] J. Friedman, "On the bit extraction problem", *Proc. 33rd IEEE Symposium on Foundations of Computer Science*, 1992, pp. 314–319.

[16] J. M. Le Bars, A. Viola, "Equivalence classes of Boolean functions for first-order correlation", *IEEE Trans. Inform. Theory* 56:3 (2010), 1247–1261.

[17] L. Hellerstein, B. Rosell, E. Bach, S. Ray, D. Page, "Exploiting product distributions to identify relevant variables of correlation immune functions", *J. Mach. Learn. Res.* 10 (2009), 2374–2411.

[18] S. Mesnager, "On the number of resilient Boolean functions", Algebraic Geometry and Its Applications, *Proc. of the First SAGA Conference*, Papeete, France, 7–11 May 2007, pp. 419–443.

[19] E. M. Palmer, R. C. Read, R.W. Robinson "Balancing the n-cube: a census of colorings", *J. Algebraic Combin.* 1 (1992), 257–273.

[20] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, J. Vandewalle, "Propagation characteristics of Boolean functions", *Adv. in Crypt. – Eurocrypt* '88, Berlin, Germany, Springer-Verlag, LNCS 473, 1990, pp. 161–173.

[21] E. M. Reingold, J. Nievergelt, N. Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, 1977.

[22] B. Rosell, L. Hellerstein, S. Ray, D. Page, "Why skewing works: learning difficult Boolean functions with greedy tree learners", *Proc. 22nd Intl. Conf. Machine Learning*, 2005, pp. 728–735.

[23] O. S. Rothaus, "On 'bent' functions", *J. Combin. Theory Ser. A* 20 (1976), 300–305 (This is nearly identical to O. S. Rothaus, "On bent functions", IDA CRD W.P. No. 169, 1966).

[24] P. Sarkar, S. Maitra, "Nonlinearity Bounds and Constructions of Resilient Boolean Functions", *Adv. in Crypt. – CRYPTO* 2000, LNCS, vol. 1880, ed. Mihir Bellare, 2000, pp. 515–532.

[25] M. Schneider, "On the construction and upper bounds of balanced and correlation-immune functions", *Proc. Sel. Areas in Crypt.* (SAC 1997) (Carleton Univ., Ottawa), pp. 73–87.

[26] M. Schneider, "A note on the construction and upper bounds of correlation-immune functions", *Cryptography and coding* (Cirencester, 1997), LNCS 1355, Springer, Berlin, 1997, pp. 295–306.

[27] J. L. Shafer, S. Schneider, J. T. Butler, P. Stănică, "Enumeration of bent Boolean functions by reconfigurable computer", *The 18th Annual Internat. IEEE Symp. on Field-Programmable Custom Comput. Machines*, Charlotte, NC, May 2-4, 2010, pp. 265-272.

[28] T. Siegenthaler, "Correlation immunity of nonlinear combining functions for cryptographic applications", *IEEE Trans. Inform. Theory* 30:5 (1984), 776–780.

[29] Y. Tarannikov, P. Korolev, and A. Botev, "Autocorrelation coefficients and correlation immunity of Boolean functions", *Adv. in Crypt. – ASIACRYPT* 2001 (Gold Coast), LNCS 2248, Springer, Berlin, 2001, pp. 460–479.

[30] A. F. Webster and S. E. Tavares, "On the design of S-boxes", *Adv. in Crypt. – CRYPTO* 1985, Springer-Verlag, Berlin, Germany, LNCS 218, 1986, pp. 523–534.

[31] Y. X. Yang, B. Guo, "Further enumerating Boolean functions of cryptographic significance", *J. Cryptology* 8 (1995), 115–122.