A Parallel Approach in Computing Correlation Immunity up to Six Variables

Carole J. Etherington
ECE Department, Naval Postgraduate School,
Monterey, CA 93943, U.S.A.
carole.etherington@navy.mil

Matthew W. Anderson
Computer Science Department, Union College
807 Union Street, Schenectady, NY 12308, U.S.A.
andersm2@union.edu

Eric Bach
Computer Sciences Department, University of Wisconsin,
Madison, WI 53706, U.S.A.;
bach@cs.wisc.edu

Jon T. Butler
ECE Department, Naval Postgraduate School,
Monterey, CA 93943, U.S.A.
jbutler@nps.edu

Pantelimon Stănică
Applied Mathematics Department, Naval Postgraduate School,
Monterey, CA 93943, U.S.A.
pstanica@nps.edu

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxxx)

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 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 ≤ n ≤ 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 computing, cryptographic Boolean functions, correlation immunity, symmetric functions, rotation symmetric 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 (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 $|f(x)| < cg(x)$ 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 $\mathbb{F}_2^n = \{x = (x_1, \ldots, x_n) : x_i \in \mathbb{F}_2, \text{ for all } i \in [n]\}$ 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 $\mathcal{B}_n$. Addition over $\mathbb{F}_2$ and $\mathbb{F}_2^n$ are both denoted by $\oplus$, whereas addition over integers is denoted by $+$. For any $x \in \mathbb{F}_2^n$, the weight of $x$ is $\text{wt}(x) = \sum_{i=1}^{n} x_i$. The algebraic normal form (ANF) of a Boolean function $f \in \mathcal{B}_n$ is

$$f(x_1, \ldots, x_n) = \bigoplus_{a=(a_1, \ldots, a_n) \in \mathbb{F}_2^n} \mu_a x_1^{a_1} \cdots x_n^{a_n},$$
where \( \mu_a \in \mathbb{F}_2^n \), for all \( a \in \mathbb{F}_2^n \), and where \( x_1^{a_1} = 1 \) if \( a_1 = 0 \) and \( x_1^{a_1} = x_1 \) if \( a_1 = 1 \). The algebraic degree of \( f \) is \( \deg(f) = \max \{ \text{wt}(a) : \mu_a \neq 0 \} \). The Fourier transform or the Fourier coefficient of \( f \in \mathcal{B}_n \) at \( u \in \mathbb{F}_2^n \) is
\[
\hat{f}(u) = 2^{-n} \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)}(-1)^{u \cdot x},
\]
where \( u \cdot x = \bigoplus_{i=1}^{n} u_i x_i \) is the inner product of \( u = (u_1, \ldots, u_n) \) and \( x = (x_1, \ldots, x_n) \). The Walsh–Hadamard transform of \( f \in \mathcal{B}_n \) at \( u \in \mathbb{F}_2^n \) is \( W_f(u) = 2^n \hat{f}(u) \).

The set of Fourier coefficients \( \{ \hat{f}(u) : u \in \mathbb{F}_2^n \} \) is said to be the Fourier spectrum of \( f \). The set of Walsh–Hadamard coefficients \( \{ W_f(u) : u \in \mathbb{F}_2^n \} \) is said to be the Walsh–Hadamard spectrum of \( f \). These transforms are invertible, since, for all \( x \in \mathbb{F}_2^n \),
\[
(-1)^{f(x)} = 2^{-n} \sum_{u \in \mathbb{F}_2^n} W_f(u)(-1)^{u \cdot x} = \sum_{u \in \mathbb{F}_2^n} \hat{f}(u)(-1)^{u \cdot x}.
\]
The Walsh-Hadamard spectrum of any Boolean function \( f \in \mathcal{B}_n \) is constrained by Parseval's identity
\[
\sum_{u \in \mathbb{F}_2^n} W_f(u)^2 = 2^n. \tag{1}
\]

Since, it will be used later, we introduce below the \( 2^n \times 2^n \) Walsh-Hadamard matrix \( H_n = ((-1)^{b(i) \cdot b(j)})_{0 \leq i, j \leq 2^n - 1} \) (\( b(i) \) 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,
\[
H_n = H_1 \oplus H_{n-1} = \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix} \quad ('\oplus' \text{ is the Kronecker product}). \text{ It is known that } W_f = H_n(-1)^f.
\]

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

An \( n \)-variable function \( f \) has correlation immunity of order \( 0 \leq k \leq n \) 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
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 \( W_f(a) = 0 \), for all \( a \in \mathbb{F}_2^n \) with \( 1 \leq \text{wt}(a) \leq k \) (respectively, \( 0 \leq \text{wt}(a) \leq k \)).

(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 \( 1 \leq i_1 < \ldots < i_k \leq n \).

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 \( 1 \leq k < n \) 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 \( \bar{x}_1 \bar{x}_2 \cdots \bar{x}_n \oplus x_1 x_2 \cdots x_n \) or its complement. A threshold function is a function \( f_{w,T} \) such that \( f_{w,T}(x) = 1 \) 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, \( f = x_1 x_2 \oplus x_3 x_4 \oplus \cdots \oplus x_{n/2-1} x_{n/2} \) 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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Expression</th>
<th>Correlation Immunity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>$f = 0$, $f = 1$</td>
<td>$n$</td>
</tr>
<tr>
<td>Parity</td>
<td>$f = x_1 \oplus x_2 \oplus \cdots \oplus x_n$, $f \oplus 1$</td>
<td>$n - 1$</td>
</tr>
<tr>
<td>Barbell</td>
<td>$f = x_1x_2 \cdots x_n \oplus \bar{x}_1\bar{x}_2\cdots\bar{x}_n$, $f \oplus 1$</td>
<td>1</td>
</tr>
<tr>
<td>Odd Weight</td>
<td>e.g., $f = x_1x_2\cdots x_n$, $f = x_1 \lor x_2 \lor \cdots \lor x_n$</td>
<td>0</td>
</tr>
<tr>
<td>Threshold</td>
<td>e.g., $f = x_1x_2 \lor x_1x_3 \lor x_2x_3$</td>
<td>0</td>
</tr>
<tr>
<td>Achilles Heel</td>
<td>$f = x_1x_2 \oplus x_3x_4 \oplus \cdots \oplus x_{n/2-1}x_{n/2}$, $n$ even, $f \oplus 1$</td>
<td>0</td>
</tr>
</tbody>
</table>

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 \(f = \bar{x}_nf_0 \lor x_nf_1\), such that \(f_0 = f|_{0 \rightarrow x_n}\) and \(f_1 = f|_{1 \rightarrow x_n}\). 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 \(\deg(f_0 \oplus f_1) = \deg(f)\), but that is impossible since \(f = f_0 \oplus x_nf_0 \oplus f_1\) and \(f\) 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 \(2^{k+1}|\frac{n-k}{2}+1|\), respectively, \(2^{k+2}|\frac{n-k}{2}+2|\). Moreover, the nonlinearity \(N_f\) of an order \(k\) correlation immune, respectively, resilient function \(f\) satisfies

\[
2^{k+1}|\frac{n-k}{2}+1| |N_f| \leq 2^{n-1} - 2^k, \text{ respectively,}
\]

\[
2^{k+2}|\frac{n-k}{2}+2| |N_f| \leq 2^{n-1} - 2^{k+1}.
\]

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

(iv) \(BCI(n,k,d) = 0\), if \(n > (n-k-1)2^{l-1}\); 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 \(c_i = 1, 1 \leq j \leq K\). If \(K = 0\), then \(f\) is constant
(hence, non-balanced) and we see that the constant functions $1, 0$ 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 $f \oplus x_i \oplus \cdots \oplus x_K = 0$. Hence, $f$ is not balanced. However, $f$ is correlation immune (resilient) of order $k := K - 1$, since then $f \oplus x_j \oplus \cdots \oplus x_{j+1} = 0$, $j \leq K - 1$ 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 $\leq 1$.

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).

Item (iv) is from Tarannikov et al [29]. Next, we show (v). We recall the interesting upper bound of [29] for the correlation immunity order $k$ of an unbalanced Boolean function in $n$ variables, namely

$$k \leq \frac{3n - 5}{4}. \quad (2)$$

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 $k$

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

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

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

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]

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

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

$$BCI(n, k) \sim 2^{2^n - \left(\binom{n}{k} - 1\right) \frac{(n-k)/2}{2/\pi} M^{1/2}},$$

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

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

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 $2^{60.72}$ 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

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

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

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

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

$$2^{\sum_{i=0}^{n-k-1} \binom{n}{i}} - 2^{\sum_{i=0}^{n-k-2} \binom{n}{i}} - 2^{\sum_{i=0}^{n-k-1} \binom{k-1}{i}} (1 + \epsilon) + 2^{\sum_{i=0}^{n-k-2} \binom{n}{i}}, \quad \epsilon = 2^{-\Theta((2^n/n)^{1/2})},$$

for $n/2 < k < n$ and Carlet and Gouget [7], who showed that $BCI(n, k)$ is upper bounded by

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

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 \( \alpha \) and produces a 1 if and only if the function has
correlation immunity $\alpha$. This output is applied to a priority encoder that produces at its output a value that is the largest $\alpha$ 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 $2^{2n-1 + O(n)}$ 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 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 $u$ with $1 \leq \text{wt}(u) \leq k$, to form $W^{(k)}$. If we write $W^{(k)} = (A \ B)$, the Walsh-Hadamard condition for $k$-immunity becomes $(A \ B)(x \ y)^T = 0$, where $A, B$ are $m \times 2^n - 1$ matrices with $\pm 1$ entries, and the column vector $(x \ y)^T$ is the truth table of $f$ (in $\pm 1$ form). Equivalently, the two “signatures” $Ax$ and $-By$ must match.

Make a list of the $2^n - 1$ pairs $(Ax, 0)$ and another list of the $2^n - 1$ pairs $(-By, 1)$. (The “tag” (0 or 1) indicates which matrix the pair came from.) Sort the combined lists lexicographically. A first component $z$ 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 $x$ and $y$ to the pairs, the actual functions could be produced as well.)

Here is an example. Take $n = 2$ and $k = 1$. Then, $A = (1 -1 1 1)$ and $-B = (-1 1 1 1)$. 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 $a, b$ are $\pm 1$ vectors of length $2^n - 1$, and $u, v$ their 0/1 images (under the map that sends +1 to 0 and $-1$ to 1), we have

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

For the last term, 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 $0 \leq \text{wt}(u \oplus v) \leq 2^n - 1$, the information payload in each pair $(Ax, 0)$ and $(-By, 1)$ 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 $Ax$’s, we need $m2^{2^n-1}$ bitwise XOR’s, and $m2^{2^n-1}$ 1’s counts. The same number is needed for the $-By$’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 $\ell$-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 729 processors, using Wisconsin’s Condor distributed computing system. Each processor was responsible for a subset of $x$’s and a subset of $y$’s, and selected possible matches by hashing the signatures. The $x$’s and $y$’s were included in the tuples, making it
possible to verify alleged matches. For our partition of the data, \( x \) and \( y \) 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 \( \leq 4 \). 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 \( 1.6 \times 10^6 \)) 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 \( f(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n \) and \( f(x) = 1 \oplus x_1 \oplus x_2 \oplus \cdots \oplus x_n \), also shown in Table 1.

<table>
<thead>
<tr>
<th>( n / k )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>238</td>
<td>14</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>64888</td>
<td>636</td>
<td>8</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4291827234</td>
<td>3139004</td>
<td>1044</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>18446240589943529428</td>
<td>503483719470800</td>
<td>46549718</td>
<td>1654</td>
<td>12</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

The complete data for \( n = 6 \) is certainly new. We can, however, sum the func-
tions 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 immune functions for \( n = 7 \) is also known; it is \( 171522187398423323340476473786538 \) [16].

![Table 3](image)

Table 3. Number of \( n \)-variable correlation immune and resilient functions, for \( 2 \leq n \leq 6 \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>Cor. Imm.</th>
<th>Resilient</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>648</td>
<td>222</td>
</tr>
<tr>
<td>5</td>
<td>3140062</td>
<td>807980</td>
</tr>
<tr>
<td>6</td>
<td>503483766022188</td>
<td>95259103924394</td>
</tr>
</tbody>
</table>

Table 4 shows the distribution of \( n \)-variable balanced functions to exact correlation immunity, where \( 2 \leq n \leq 6 \). 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 \( 23478015754788854439497622689296 \) 1-resilient functions for \( n = 7 \).

![Table 4](image)

Table 4. Distribution of \( n \)-variable balanced functions by exact resiliency, \( k \), for \( 2 \leq n \leq 6 \).

<table>
<thead>
<tr>
<th>( n/k )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>62</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>12648</td>
<td>212</td>
<td>8</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>600272410</td>
<td>807428</td>
<td>540</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>183212065723119734</td>
<td>503483702719940</td>
<td>16749696</td>
<td>1150</td>
<td>12</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

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

\[
f(x_1, x_2, \ldots, x_n) = f(x_n, x_1, x_2, \ldots, x_{n-1});
\]

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

**Example 2.** The four functions

\[
f(x) = 0,
\]

\[
f(x) = 1,
\]

\[
f(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n,
\]

and

\[
f(x) = 1 \oplus x_1 \oplus x_2 \oplus \cdots \oplus x_n \quad (\text{for all } x \in \mathbb{F}_2^n)
\]

are all rotation symmetric.
Table 5. Distribution of \( n \)-variable rotation symmetric functions versus exact correlation immunity \( k \) and \( n \), for \( 2 \leq n \leq 6 \).

<table>
<thead>
<tr>
<th>( n / k )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>214</td>
<td>34</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>14656</td>
<td>1686</td>
<td>36</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

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 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 \( n = 4 \).

<table>
<thead>
<tr>
<th>( N / k )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>12</td>
<td>8</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>512</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3712</td>
<td>128</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>17920</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>27504</td>
<td>496</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>14336</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>896</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

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

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 “transesent 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 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 $n = 4$.

<table>
<thead>
<tr>
<th>Deg / $k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>12</td>
<td>8</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1712</td>
<td>304</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>30400</td>
<td>320</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>32768</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Deg / (k)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>59736</td>
<td>5096</td>
<td>640</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>65478976</td>
<td>1563968</td>
<td>384</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2078804864</td>
<td>1569920</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2147483648</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Deg / (k)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>30</td>
<td>40</td>
<td>30</td>
<td>12</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3760424</td>
<td>417512</td>
<td>15000</td>
<td>1240</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4388747656096</td>
<td>9270073536</td>
<td>24586784</td>
<td>384</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>143859057441024156</td>
<td>251732566372708</td>
<td>21947904</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>9079005106896312932</td>
<td>251741882607004</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>9223372036854775808</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>18446240589943529428</td>
<td>503483719470790</td>
<td>46549728</td>
<td>1654</td>
<td>12</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Computer/ Program</th>
<th>Time (msec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 MHz FPGA/Verilog</td>
<td>0.655</td>
</tr>
<tr>
<td>100 MHz FPGA/C</td>
<td>1,238.7</td>
</tr>
<tr>
<td>2.8 GHz Xeon/C</td>
<td>190</td>
</tr>
</tbody>
</table>
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 $\leq 6$ 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


A Parallel Approach in Computing Correlation Immunity up to Six Variables
