NAVAL
POSTGRADUATE
SCHOOL

# Correlation Immune and Resilient Generalized Boolean Functions

Thor Martinsen, PhD
Commander, US Navy
Assistant Professor
Naval Postgraduate School

3rd International Workshop on Boolean Functions and their Applications

June 19, 2018
Loen, Norway

- Boolean functions $f : \mathbb{V}_n \to \mathbb{F}_2$; $\mathbb{V}_n$ – vector space $\mathbb{F}_2^n$.
- Generalized Boolean function $f : \mathbb{V}_n \to \mathbb{Z}_q$, $q \geq 2$.
- For any function $f \in \mathcal{GB}_n^q$ and $2^{k-1} < q \leq 2^k$, we associate a unique sequence of Boolean functions $a_i \in \mathcal{B}_n$ $(i = 0, 1, \ldots, k-1)$ such that

$$f(\mathbf{x}) = a_0(\mathbf{x}) + 2a_1(\mathbf{x}) + \cdots + 2^{k-1}a_{k-1}(\mathbf{x}), \text{ for all } \mathbf{x} \in \mathbb{V}_n.$$

- The derivative of $f$ with respect to a vector $\mathbf{a}$ is denoted $D_{\mathbf{a}}f$ and defined as

$$D_{\mathbf{a}}f(\mathbf{x}) = f(\mathbf{x} \oplus \mathbf{a}) - f(\mathbf{x}) \text{ for all } \mathbf{x} \in \mathbb{V}_n.$$

- A vector $\mathbf{a} \in \mathbb{V}_n$ is said to be a *linear structure* of a generalized Boolean function, if the derivative of the function with respect to $\mathbf{a}$ remains constant for all $\mathbf{x} \in \mathbb{V}_n$.
- The (generalized) *Walsh–Hadamard transform* of $f \in \mathcal{GB}_n^q$ at any point $\mathbf{u} \in \mathbb{V}_n$ is the complex valued function

$$\mathcal{H}_f(\mathbf{u}) = 2^{-\frac{n}{2}} \sum_{\mathbf{x} \in \mathbb{V}_n} \zeta^{f(\mathbf{x})} (-1)^{\mathbf{u} \cdot \mathbf{x}},$$

where $\zeta = e^{2\pi \imath / q}$ is the complex $q$-primitive root of unity. If $q = 2$, we obtain the (normalized) *Walsh–Hadamard transform* of $f \in \mathcal{B}_n$, which will be denoted by $\mathcal{W}_f$.

- Siegenthaler first described the correlation attack in 1984.
- Correlation attacks analyze input vectors and associated functional outputs to determine if a single bit, or a specific subsets of bits, exert greater influence over the output than others.
- There are many Correlation Immune constructions for Boolean functions.
- We will use one of the most basic CI Boolean functions constructions along with two approaches (linear structures and orthogonal arrays) to create correlation immune generalized Boolean functions.

# Correlation Immunity Example

$$f(\mathbf{x}) = 1 \oplus x_2x_3 \oplus x_1 \oplus x_1x_3 \oplus x_1x_2$$

| Input | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| Output | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

| Conditional Prob. Given $f(\mathbf{x}) = 0$ | Conditional Prob. Given $f(\mathbf{x}) = 1$ |
|---|---|
| $Pr(x_1 = 0 \mid f(\mathbf{x}) = 0) = 1/2$<br>$Pr(x_1 = 1 \mid f(\mathbf{x}) = 0) = 1/2$ | $Pr(x_1 = 0 \mid f(\mathbf{x}) = 1) = 1/2$<br>$Pr(x_1 = 1 \mid f(\mathbf{x}) = 1) = 1/2$ |
| $Pr(x_2 = 0 \mid f(\mathbf{x}) = 0) = 1/2$<br>$Pr(x_2 = 1 \mid f(\mathbf{x}) = 0) = 1/2$ | $Pr(x_2 = 0 \mid f(\mathbf{x}) = 1) = 1/2$<br>$Pr(x_2 = 1 \mid f(\mathbf{x}) = 1) = 1/2$ |
| $Pr(x_3 = 0 \mid f(\mathbf{x}) = 0) = 1/2$<br>$Pr(x_3 = 1 \mid f(\mathbf{x}) = 0) = 1/2$ | $Pr(x_3 = 0 \mid f(\mathbf{x}) = 1) = 1/2$<br>$Pr(x_3 = 1 \mid f(\mathbf{x}) = 1) = 1/2$ |

This function was created using the "folklore" construction.

$$f(\mathbf{x} \oplus \mathbf{1}) = f(\mathbf{x}), \forall \mathbf{x} \in \mathbb{V}_n$$

- A generalized Boolean function $f \in \mathcal{GB}_n^q$ is said to be *correlation immune of order* $t$, with notation $CI(t)$, $1 \le t \le n$, if for any fixed subset of $t$ variables the probability that, given the value of $f(\mathbf{x})$, the $t$ variables have any fixed set of values, is always $2^{-t}$, no matter what the choice of the fixed set of $t$ values is.

**Theorem**

*If $f \in \mathcal{GB}_n^q$ is a $CI(1)$ generalized Boolean function, then the number of occurrences of each output value $c \in \mathbb{Z}_q$ that $f$ achieves is even.*

**Corollary**

*Let $f \in \mathcal{GB}_n^q$ be a correlation immune (order 1) generalized Boolean function. Then the image of $f$ has cardinality $|f(\mathbb{V}_n)| \le 2^{n-1}$.*

# CI(1) Generalized Boolean Function Construction Example

Suppose we wish to construct a CI(1) generalized Boolean function, $f \in \mathcal{GB}_4^q$, where $1 \leq q \leq 4$.

- Select for example the vector $\mathbf{a} = 1010$. $(\kappa = 2)$
- For each $\mathbf{x} \in \mathbb{V}_4$, we pair $\mathbf{x}$ with $\mathbf{x}' = \mathbf{x} \oplus \mathbf{a}$, producing the following partition:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0000 | 0010 | 0100 | 0110 | 0001 | 0011 | 0101 | 0111 |
| 1010 | 1000 | 1110 | 1100 | 1001 | 1001 | 1111 | 1101 |

- The vector $\mathbf{a}$ has 2 zeros (located at index 1 and 3).
- The partition therefore has $2^2$ bit combinations located at index 1 and 3.

# CI(1) Generalized Boolean Function Construction Ex. Cont.

- Combine each pair of vectors with a corresponding pair which disagrees with respect to the bits at index 1 and 3.
- There are $2^{n-1-\kappa} = 2^{4-1-2} = 2$ of each of there possible two-bit combinations, so there are $2^{n-1-\kappa}! = 2^{4-1-2}! = 2!$ possible pairings.
- To all vectors within each of the 4 subsets, we assign the same output value from $\mathbb{Z}_4$.
- There are therefore $4^4 = 256$ possible CI(1) generalized functions, where $1 \leq q \leq 4$, which we can construct using **a**.

Table: A CI(1) generalized Boolean function, $f \in \mathcal{GB}_4^4$

| Input | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 0 | 3 | 2 | 1 | 1 | 2 | 3 | 0 | 2 | 1 | 0 | 3 | 3 | 0 | 1 | 2 |

Revisiting the "folklore" construction example that we began with, observe that

$$\begin{matrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$$

is a linear orthogonal array. We shall use this perspective to construct higher order correlation immune generalized Boolean functions.

- There is a close connection between orthogonal arrays and correlation immune functions. Camion et al. first wrote about this in 1992.

- An $m \times n$ array with entries from a set of $s$ elements is called an *orthogonal array* of size $m$ with $n$ constraints, $s$ levels, strength $t$, and index $r$, if any set of $t$ columns of the array contain all $s^t$ possible row vectors exactly $r$ times.

- We denote orthogonal arrays by $OA(m, n, s, t)$.

# An Orthogonal Array Example

Consider the following $4 \times 3$ binary array, along with all possible combinations of two of its columns:

| $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| $\mathbf{x}_1$ | $\mathbf{x}_2$ |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

| $\mathbf{x}_1$ | $\mathbf{x}_3$ |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 1 |
| 1 | 0 |

| $\mathbf{x}_2$ | $\mathbf{x}_3$ |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 0 | 1 |
| 1 | 0 |

For every possible combination of 2 columns of the array, the row vectors 00, 01, 10, and 11 all occur with frequency 1. Consequently, this is a $OA(4, 3, 2, 2)$ orthogonal array of index 1.

## Lemma

*Let $O$ be an $OA(m, n, 2, t)$ binary orthogonal array. Complementing any column, $i$, $1 \leq i \leq n$, of $O$ produces another $OA(m, n, 2, t)$ binary orthogonal array.*

There is also a close connection between orthogonal arrays and error correcting codes.

- An *error correcting code* $C$ of length $n$, size $m$, minimum pairwise Hamming distance between distinct codewords of $d$, and which is defined over an alphabet $s$, is denoted $(n, m, d)_s$.
- To any such code we associate the $m \times n$ array whose rows are the codewords of $C$. This array is an orthogonal array $OA(m, n, s, t)$ for some $t$.
- A code $C$ of length $n$ is said to be *linear* if the codewords are distinct and $C$ is a vector subspace of $\mathbb{F}_s^n$, thus $C$ has size $m = s^\ell$ for some non negative integer $0 \leq \ell \leq n$.
- The orthogonal array associated with a code is *linear* if and only if the code is linear.

# Higher Order CI Gen. Boolean Function Const. Example

Suppose we wish to construct a higher order ($t > 1$) correlation immune generalized Boolean function, $f \in \mathcal{GB}_5^4$. We begin by finding a suitable linear orthogonal array. For example, the following $OA(8, 5, 2, 2)$ linear orthogonal array.

$$O_0 = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0. \end{matrix}$$

# Higher Order CI Gen. Boolean Function Const. Ex. Cont.

Since $OA(8,5,2,2)$ is a linear orthogonal array, $O_0$'s row vectors form a subgroup of $\mathbb{V}_5$. We can therefore cover $\mathbb{V}_5$ by forming the 3 cosets of $O_0$.

$$O_1 = \begin{matrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1, \end{matrix} \quad O_2 = \begin{matrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0, \end{matrix} \quad O_3 = \begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0. \end{matrix}$$

Lemma 3 ensures that these newly formed cosets are all $OA(8,5,2,2)$ orthogonal arrays in their own right.

# Higher Order CI Gen. Boolean Function Const. Ex. Cont.

We now select a permutation, $p$ of the set $\{1, 2, \ldots, 5\}$, say for example $p = \{2, 1, 3, 5, 4\}$. For each of the orthogonal arrays, $O_i$, $i = 0$ to 3, we rearrange the columns of $O_i$ such that
$O_i^{(p)} = [\mathbf{c}_{p(1)}, \mathbf{c}_{p(2)}, \mathbf{c}_{p(3)}, \mathbf{c}_{p(4)}, \mathbf{c}_{p(5)}] = [\mathbf{c}_2, \mathbf{c}_1, \mathbf{c}_3, \mathbf{c}_5, \mathbf{c}_4]$.

$$O_0^{(p)} = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0, \end{matrix} \qquad O_1^{(p)} = \begin{matrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0, \end{matrix}$$

$$O_2^{(p)} = \begin{matrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1, \end{matrix} \qquad O_3^{(p)} = \begin{matrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0. \end{matrix}$$

# Higher Order CI Gen. Boolean Function Const. Ex. Cont.

By assigning the same output value from $\mathbb{Z}_4$ to all vectors within each orthogonal array, say for example $\{O_0^{(p)} \to 0, O_1^{(p)} \to 1, O_2^{(p)} \to 2, O_3^{(p)} \to 3\}$, we create the following $CI(2)$ generalized Boolean function:

| $\mathbb{V}_5$ | $a_0$ | $a_0$ | $a_0 \oplus a_1$ | $f$ |
|---|---|---|---|---|
| 00000 | 0 | 0 | 0 | 0 |
| 00001 | 0 | 1 | 1 | 2 |
| 00010 | 1 | 0 | 1 | 1 |
| 00011 | 1 | 1 | 0 | 3 |
| 00100 | 1 | 0 | 1 | 1 |
| 00101 | 1 | 1 | 0 | 3 |
| 00110 | 0 | 0 | 0 | 0 |
| 00111 | 0 | 1 | 1 | 2 |
| 01000 | 1 | 1 | 0 | 3 |
| 01001 | 1 | 0 | 1 | 1 |
| 01010 | 0 | 1 | 1 | 2 |
| 01011 | 0 | 0 | 0 | 0 |
| 01100 | 0 | 1 | 1 | 2 |
| 01101 | 0 | 0 | 0 | 0 |
| 01110 | 1 | 1 | 0 | 3 |
| 01111 | 1 | 0 | 1 | 1 |
| 10000 | 0 | 1 | 1 | 2 |
| 10001 | 0 | 0 | 0 | 0 |
| 10010 | 1 | 1 | 0 | 3 |
| 10011 | 1 | 0 | 1 | 1 |
| 10100 | 1 | 1 | 0 | 3 |
| 10101 | 1 | 0 | 1 | 1 |
| 10110 | 0 | 1 | 1 | 2 |
| 10111 | 0 | 0 | 0 | 0 |
| 11000 | 1 | 0 | 1 | 1 |
| 11001 | 1 | 1 | 0 | 3 |
| 11010 | 0 | 0 | 0 | 0 |
| 11011 | 0 | 1 | 1 | 2 |
| 11100 | 0 | 0 | 0 | 0 |
| 11101 | 0 | 1 | 1 | 2 |
| 11110 | 1 | 0 | 1 | 1 |
| 11111 | 1 | 1 | 0 | 3 |

# Some Orthogonal Arrays and Associated $\mathcal{GB}_n^q$ function parameters

| $n$ | $q \leq$ | $CI(t)$ | $OA$ |
|---|---|---|---|
| 5 | 4 | 2 | $OA(8, 5, 2, 2)$ |
| 6 | 4 | 3 | $OA(16, 6, 2, 3)$ |
| 7 | 16 | 2 | $OA(8, 7, 2, 2)$ |
| 7 | 8 | 3 | $OA(16, 7, 2, 3)$ |
| 8 | 16 | 3 | $OA(16, 8, 2, 3)$ |
| 9 | 4 | 5 | $OA(2^7, 9, 2, 5)$ |
| 12 | 4 | 7 | $OA(2^{10}, 12, 2, 7)$ |
| 15 | $2^{11}$ | 2 | $OA(16, 15, 2, 2)$ |
| 15 | $2^8$ | 3 | $OA(2^7, 15, 2, 3)$ |
| 15 | $2^7$ | 4 | $OA(2^8, 15, 2, 4)$ |
| 16 | $2^{11}$ | 3 | $OA(32, 16, 2, 3)$ |
| 16 | 32 | 7 | $OA(2^{11}, 16, 2, 7)$ |
| 18 | 8 | 9 | $OA(2^{15}, 18, 2, 9)$ |
| 20 | $2^{11}$ | 5 | $OA(2^9, 20, 2, 5)$ |
| 24 | $2^{14}$ | 5 | $OA(2^{10}, 24, 2, 5)$ |
| 24 | $2^{12}$ | 7 | $OA(2^{12}, 24, 2, 7)$ |
| 31 | $2^{26}$ | 2 | $OA(32, 31, 2, 2)$ |
| 32 | $2^{26}$ | 3 | $OA(64, 32, 2, 3)$ |
| 32 | $2^{21}$ | 5 | $OA(2^{11}, 32, 2, 5)$ |
| 32 | $2^6$ | 15 | $OA(2^{26}, 32, 2, 15)$ |

# Rotation Symmetric CI Generalized Boolean Functions

- We can use the linear orthogonal array construction technique (sans permutations) to also create Rotation Symmetric (RotS) generalized Boolean functions.
- Rotation symmetric Boolean functions, were introduced by Pieprzyk and Qu in 1999 (although they appeared in the work of Filiol and Fontaine as idempotents, the preceding year).
- RotS functions remain invariant under cyclic rotations of their input vectors.

Suppose we wish to construct a *RotS* and $CI(2)$ generalized Boolean function, $f \in \mathcal{GB}_7^4$. We first select the cyclic $\overleftarrow{OA}(8,7,2,2)$ linear array:

$$O_0 = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1. \end{matrix} \qquad O_1 = \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0\,. \end{matrix}$$

$$(7, \{0000001, 1000000, 0100000, 0010000, 0001000, 0000100, 0000010\})$$

Using these vectors, the algorithm in turn constructs and stores the following seven cosets to $V$:

$$O_2 = \begin{array}{ccccccc} 0&0&0&0&0&0&1\\ 1&0&1&1&1&0&1\\ 0&1&0&1&1&1&1\\ 0&0&1&0&1&1&0\\ 1&0&0&1&0&1&0\\ 1&1&0&0&1&0&0\\ 1&1&1&0&0&1&1\\ 0&1&1&1&0&0&0, \end{array} \qquad O_3 = \begin{array}{ccccccc} 1&0&0&0&0&0&0\\ 0&0&1&1&1&0&0\\ 1&1&0&1&1&1&0\\ 1&0&1&0&1&1&1\\ 0&0&0&1&0&1&1\\ 0&1&0&0&1&0&1\\ 0&1&1&0&0&1&0\\ 1&1&1&1&0&0&1, \end{array}$$

$$O_4 = \begin{array}{ccccccc} 0&1&0&0&0&0&0\\ 1&1&1&1&1&0&0\\ 0&0&0&1&1&1&0\\ 0&1&1&0&1&1&1\\ 1&1&0&1&0&1&1\\ 1&0&0&0&1&0&1\\ 1&0&1&0&0&1&0\\ 0&0&1&1&0&0&1, \end{array} \qquad O_5 = \begin{array}{ccccccc} 0&0&1&0&0&0&0\\ 1&0&0&1&1&0&0\\ 0&1&1&1&1&1&0\\ 0&0&0&0&1&1&1\\ 1&0&1&1&0&1&1\\ 1&1&1&0&1&0&1\\ 1&1&0&0&0&1&0\\ 0&1&0&1&0&0&1, \end{array}$$

$$O_6 = \begin{array}{ccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1, \end{array}$$

$$O_7 = \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1, \end{array}$$

$$O_8 = \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1. \end{array}$$

$7, \{0000011, 1000001, 1100000, 0110000, 0011000, 0001100, 0000110\}).$

Using these vectors, the algorithm in turn constructs and stores the following seven cosets to $V$:

$$O_9 = \begin{matrix} 0&0&0&0&0&1&1\\ 1&0&1&1&1&1&1\\ 0&1&0&1&1&0&1\\ 0&0&1&0&1&0&0\\ 1&0&0&1&0&0&0\\ 1&1&0&0&1&1&0\\ 1&1&1&0&0&0&1\\ 0&1&1&1&0&1&0, \end{matrix} \qquad O_{10} = \begin{matrix} 1&0&0&0&0&0&1\\ 0&0&1&1&1&0&1\\ 1&1&0&1&1&1&1\\ 1&0&1&0&1&1&0\\ 0&0&0&1&0&1&0\\ 0&1&0&0&1&0&0\\ 0&1&1&0&0&1&1\\ 1&1&1&1&0&0&0, \end{matrix}$$

$$O_{11} = \begin{matrix} 1&1&0&0&0&0&0\\ 0&1&1&1&1&0&0\\ 1&0&0&1&1&1&0\\ 1&1&1&0&1&1&1\\ 0&1&0&1&0&1&1\\ 0&0&0&0&1&0&1\\ 0&0&1&0&0&1&0\\ 1&0&1&1&0&0&1, \end{matrix} \qquad O_{12} = \begin{matrix} 0&1&1&0&0&0&0\\ 1&1&0&1&1&0&0\\ 0&0&1&1&1&1&0\\ 0&1&0&0&1&1&1\\ 1&1&1&1&0&1&1\\ 1&0&1&0&1&0&1\\ 1&0&0&0&0&1&0\\ 0&0&0&1&0&0&1, \end{matrix}$$

# Rotation Symmetric CI Generalized Boolean Functions

$$O_{13} = \begin{matrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1, \end{matrix} \qquad O_{14} = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1, \end{matrix}$$

$$O_{15} = \begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1. \end{matrix}$$

$$\{O_0 \to c_0, O_1 \to c_1, \{O_2, \ldots, O_8\} \to c_2, \{O_9, \ldots, O_{15}\} \to c_3\}, c_i \in \mathbb{Z}_4.$$

## A few Comments on RotS Generalized Boolean Functions

In general we can partition $\mathbb{V}_n$ into:

$$g_n = \frac{1}{n}\sum_{\tau|n}\phi(\tau)2^{n/\tau},$$

cyclic classes, and there are therefore $g(n)^{g(n)}$ possible RotS generalized Boolean functions.
If $n$ is prime it possible to obtain a simpler expression for $g(n)$, namely

$$g_p = \frac{1}{n}\sum_{\tau|n}\phi(\tau)2^{n/\tau} = 2 + \frac{2^p - 2}{p}.$$

If we use linear orthogonal arrays of the form OA(2,p,2,1), where p is an odd prime, and construct Rots CI(1) generalized Boolean functions, then there are at most

$$\left(1 + \frac{2^{p-1} - 1}{p}\right)^{1+\frac{2^{p-1}-1}{p}}$$

such functions.

# A few Comments on RotS Generalized Boolean Functions

- Although there are no symmetric and balanced generalized Boolean function, with $2^k$ output values, $k > 1$, (Meidl, Pott, Stanica, M), there are RotS and balanced generalized Boolean functions with more than two output vales. For example: $\{\{\langle 0000\rangle, \langle 1111\rangle, \langle 0101\rangle\} \to 0, \langle 0001\rangle \to 1, \langle 0011\rangle \to 2, \langle 0111\rangle \to 3\}$
- There are however no balanced and RotS generalized Boolean functions in $p$ variables where $p$ is an odd prime and $q > 2$.

# New from Old

We generalize the Siegenthaler CI(t) function concatenation construction as follows:

**Theorem**

*Let* $\mathbf{x} = (x_1, \ldots, x_n)$ *and suppose that we have correlation immune (order* $t$*) generalized Boolean functions,* $f_1, f_2 \in \mathcal{GB}_n^q$*, such that* $\forall c \in f_1(\mathbb{V}_n) = f_2(\mathbb{V}_n)$*,* $Pr(f_1(\mathbf{x}) = c) = Pr(f_2(\mathbf{x}) = c) = p$*. Then the function* $f$ *of* $n+1$ *variables defined by*

$$f(\mathbf{x}, x_{n+1}) = x_{n+1} f_1(\mathbf{x}) + (x_{n+1} \oplus 1) f_2(\mathbf{x}) \tag{1}$$

*is also correlation immune of order* $t$ *and satisfies* $Pr(f(\mathbf{x}) = c) = p$*.*

# A Generalized Siegenthaler Construction Example

Table: Siegenthaler constructed CI(1) function, $f \in \mathcal{GB}_4^4$

| $\mathbb{V}_4$ | $a_0$ | $a_1$ | $f$ |
|---|---|---|---|
| 0000 | 0 | 0 | 0 |
| 0001 | 1 | 1 | 3 |
| 0010 | 0 | 1 | 2 |
| 0011 | 1 | 0 | 1 |
| 0100 | 1 | 0 | 1 |
| 0101 | 0 | 1 | 2 |
| 0110 | 1 | 1 | 3 |
| 0111 | 0 | 0 | 0 |
| 1000 | 0 | 1 | 2 |
| 1001 | 1 | 0 | 1 |
| 1010 | 1 | 1 | 3 |
| 1011 | 0 | 0 | 0 |
| 1100 | 0 | 0 | 0 |
| 1101 | 1 | 1 | 3 |
| 1110 | 1 | 0 | 1 |
| 1111 | 0 | 1 | 2 |

# A Cautionary Tale

Note: When performing Siegenthaler construction for generalized Boolean functions, care must be taken to ensure that:

$$\forall c \in f_1(\mathbb{V}_n) = f_2(\mathbb{V}_n),\ Pr(f_1(\mathbf{x}) = c) = Pr(f_2(\mathbf{x}) = c) = p.$$

Table: Correlation immune generalized Boolean function construction failure

| $\mathbb{V}_3$ | $a_0$ | $a_1$ | $f$ |
|---|---|---|---|
| 000 | 1 | 0 | 1 |
| 001 | 0 | 1 | 2 |
| 010 | 0 | 1 | 2 |
| 011 | 1 | 0 | 1 |
| 100 | 0 | 0 | 0 |
| 101 | 1 | 1 | 3 |
| 110 | 1 | 1 | 3 |
| 111 | 0 | 0 | 0 |

# Necessary and Sufficient Conditions

Recall:

$$f(\mathbf{x}) = a_0(\mathbf{x}) + 2a_1(\mathbf{x}) + \cdots + 2^{k-1}a_{k-1}(\mathbf{x}), \text{ for all } \mathbf{x} \in \mathbb{V}_n.$$

**Theorem**

*If $f$ is a correlation immune (order $t$) generalized Boolean function, then all of its constituent Boolean functions, $a_j \in \mathcal{B}_n$, are also correlation immune (order $t$).*

**Theorem**

*Let $f \in \mathcal{GB}_n^q$ be the generalized Boolean function $f(\mathbf{x}) = \sum_{j=0}^{k-1} 2^j a_j(\mathbf{x})$, where $0 \leq j \leq k-1$, $a_j \in \mathcal{B}_n$ and $\mathbf{x} \in \mathbb{V}_n$. Then $f$ is correlation immune (order $t$) if and only if all Boolean functions $a_j$ are $CI(t)$ and use the same partition $P$ of $\mathbb{V}_n$ consisting of $q$ orthogonal arrays, $O_j$, each of strength $t$.*

# Conclusion

Thank you for your attention!

# Correlation Immunity and the Walsh-Hadamard Transform

- A [Boolean] function $f(\mathbf{x})$ in $n$ variables is correlation immune of order $t$, $1 \leq t \leq n$ if and only if all of the Walsh transforms $\mathcal{W}_f(\mathbf{w}) = 0$, where $1 \leq wt(\mathbf{w}) \leq t$.
- A generalized Boolean function is *generalized correlation immune of order* $t$, denoted $gCI(t)$, if and only if all of the (generalized) Walsh transforms $\mathcal{H}_f(\mathbf{w}) = 0$, where $1 \leq wt(\mathbf{w}) \leq t$.
- Let $f \in \mathcal{GB}_n^q$ be a generalized Boolean function. If $f$ is $CI(1)$, then $f$ is $gCI(1)$.
- The converse is in general not true.

Table: Non-$CI(1)$ function $f \in \mathcal{GB}_4^4$, where $\mathcal{H}_f(\mathbf{w}) = 0$

| Input | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 0 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 | 1 | 3 | 3 | 1 | 0 | 0 |

The $4^{th}$ root of unity is $\zeta_4 = i$. Letting $\mathbf{w} \in \{0001, 0010, 0100, 1000\}$, we compute $\mathcal{H}_f(\mathbf{w})$, which yields the following:

$$\mathcal{H}_f(0001) = i^0 + i^0 + i^0 + i^2 + i^2 + i^1 + i^3 + i^0 - i^0 - i^2 - i^2 - i^0 - i^0 - i^3 - i^1 - i^0 = 0,$$
$$\mathcal{H}_f(0010) = i^0 + i^0 + i^0 + i^2 + i^2 + i^0 + i^3 + i^1 - i^0 - i^2 - i^2 - i^0 - i^1 - i^3 - i^0 - i^0 = 0,$$
$$\mathcal{H}_f(0100) = i^0 + i^0 + i^0 + i^2 + i^2 + i^0 + i^1 + i^3 - i^0 - i^2 - i^2 - i^0 - i^3 - i^1 - i^0 - i^0 = 0,$$
$$\mathcal{H}_f(1000) = i^0 + i^0 + i^0 + i^2 + i^0 + i^2 + i^2 + i^0 - i^2 - i^0 - i^1 - i^3 - i^3 - i^1 - i^0 - i^0 = 0.$$