Classification of All t-Resilient Boolean Functions with $t + 4$ Variables

Shahram Rasoolzadeh

Radboud University, Nijmegen, The Netherlands, firstname.lastname@ru.nl

Abstract. We apply Siegenthaler's construction, along with several techniques, to classify all $(n-4)$-resilient Boolean functions with n variables, for all values of $n \geq 4$, up to extended variable-permutation equivalence. We show that, up to this equivalence, there are only 761 functions for any n larger than or equal to 10, and for smaller values of n, i.e., for n increasing from 4 to 9, there are 58, 256, 578, 720, 754, and 760 functions, respectively. Furthermore, we classify all 1-resilient 6-variable Boolean functions and show that there are 1 035 596 784 such functions up to extended variable-permutation equivalence.

Keywords: correlation immunity · resilient functions · boolean functions

1 Introduction

Correlation immune boolean functions were first introduced by Siegenthaler in [Sie84] as a countermeasure against correlation attacks on the combiner model of stream ciphers. Soon after, the balanced correlation immune functions been called *resilient* functions in [CGH+85] and were used to resist the bit extraction problem.

Siegenthaler found all the t-resilient functions with n variables for $t \geq n-2$ in [Sie84] and showed that all of them are affine functions. Later in [CCCS91], Camion, Carlet, Charpin, and Sendrier used Siegenthaler's construction, introduced in [Sie84], to find all the t-resilient functions with $(t+3)$ variables. However, the number of t-resilient functions with $(t+4)$ variables were not investigated before this paper.

In [TK00], Tarannikov and Kirienko showed that for every positive integer m, there exists a number $p(m)$ such that for $n > p(m)$, any $(n-m)$-resilient n-variable function $f(x_0, \ldots, x_{n-1})$ is equivalent, up to a permutation of its input variables, to a function of the form $g(x_0, \ldots, x_{p(m)-1}) \oplus x_{p(m)} \oplus \ldots \oplus x_{n-1}$ and proved that $p(4) = 10$.

Moreover, in [CC05], Carlet and Charpin classified all the *cubic* t-resilient functions with $(t+4)$ variables up to the spectrum of their Walsh transform and showed that, up to this equivalence, there are only four type of such functions.

This paper presents a comprehensive classification of all t-resilient boolean functions with $(t+4)$ variables, for all values of $n \geq 4$, up to extended variable-permutation equivalence. We first establish that the resilience behavior of boolean functions remains unchanged under extended variable-permutation equivalence. Next, we use Siegenthaler's construction to generate all boolean functions of $(t+1)$-resilience and $(n+1)$ variables from two t-resilient boolean functions in n variables. To efficiently search for all such resilient boolean functions, we develop several techniques to present an efficient algorithm for searching all such resilient boolean functions. Compared to a naive approach, in this paper, we introduce the concept of representative pairs and by applying the developed techniques, we reduce the number of potential representative pairs to construct higher order resilient functions.

We show that up to this equivalence, for any n larger than or equal to 10, there are only 761 functions, and for smaller values of n, there are 58, 256, 578, 720, 754, and 760 functions, for $n = 4$ increasing to $n = 9$, respectively. That means, up to the extended variable-permutation equivalence, there are only 747 t-resilient $(t+4)$-variable functions with all variables non-linearly involved in the function, excluding the 14 boolean functions with 3 variables; 44 with 4, 198 with 5, 322 with 6, 142 with 7, 34 with 8, 6 with 9, and 1 with 10 variables.

Furthermore, we classify all 1-resilient 6-variable boolean functions up to extended variable-permutation equivalence and show that there are 1 035 596 784 such functions.

2 Preliminaries

In this section, we explain the notations used in this paper, along with the necessary basics related to boolean and resilient functions.

We use $\mathbb{F}_2$ to denote the finite field of two elements, $\{\mathtt{0}, \mathtt{1}\}$ and $\oplus$ to denote addition in this field. We use $\mathbb{F}_2^n$ to denote the vector space over $\mathbb{F}_2$ with dimension n.

Let $a, b \in \mathbb{F}_2^n$ be two n-variable binary vectors. We denote the i-th element of a by $a[i]$, that means $a = (a[0], \dots, a[n-1])$ and we use $\overline{a}$ to denote the complement value of a, i.e., $\overline{a} = (a[0] \oplus 1, \dots, a[n-1] \oplus 1)$. We use $\mathrm{hw}(a)$ and $\mathrm{hp}(a)$ to denote the *Hamming weight* and *Hamming parity* of a, respectively, defined as $\mathrm{hw}(a) = \sum_{i=0}^{n-1} a[i]$ and $\mathrm{hp}(a) = \bigoplus_{i=0}^{n-1} a[i]$.

We denote the inner product between a and b with $\langle a, b \rangle$ defined as $\langle a, b \rangle = \bigoplus_{i=0}^{n-1} a[i]b[i]$; and to denote concatenation of two vectors a and b, we use $a \,\|\, b$, which is equivalent to $(a[0], \dots, a[n-1], b[0], \dots, b[n-1])$.

Boolean Functions

The functions from the vector space $\mathbb{F}_2^n$ to the binary field $\mathbb{F}_2$ are called *boolean functions* with n-variables. We use $\mathcal{B}_n$ to denote the set of all n-variable boolean functions. *Truth table* is the most basic way to represent a boolean function. The truth table of $f \in \mathcal{B}_n$ is a binary vector $T_f \in \mathbb{F}_2^{2^n}$ such that for any $x \in \mathbb{F}_2^n$, $T_f[x]$ shows the value of $f(x)$.

Balanced boolean functions are the ones which for half of the inputs $x \in \mathbb{F}_2^n$, it maps to $\mathtt{1}$, and for the other half, it maps to $\mathtt{0}$.

Algebraic normal form (ANF) is another often used representation of boolean functions in cryptography. It is the n-variable polynomial representation over $\mathbb{F}_2$ of the form

$$f(x) = \bigoplus_{I \in \mathbb{F}_2^n} a_I x^I = \bigoplus_{I \in \mathbb{F}_2^n} a_I \Big(\prod_{i=0}^{n-1} x_i^{I[i]} \Big),$$

where x_i is the i-th variable of x, that is $x = (x_0, \dots, x_{n-1})$. By x^I, we denote the monomial $x_0^{I[0]} \cdots x_{n-1}^{I[n-1]}$ that corresponds to the monomial with x_i variables with $I[i] = \mathtt{1}$. Note that each a_I is a binary value and every coordinate x_i appears in this polynomial with exponents at most 1.

Algebraic normal form degree of a boolean function f is the maximum Hamming weight of all existing monomials in the ANF representation of the function, that is $\max_{I \in \mathbb{F}_2^n, a_I = 1} \mathrm{hw}(I)$, which we will simply call it algebraic degree of f.

Classifying boolean functions by their algebraic degree, the ones with degree zero, one, two, or three are called *constant*, *affine*, *quadratic*, and *cubic* functions, respectively. Affine functions are the extension of linear functions by adding a constant at the output, and can be displayed as $\langle \alpha, x \rangle \oplus c$ with $\alpha \in \mathbb{F}_2^n$ and $c \in \mathbb{F}_2$.

Walsh transform is a powerful tool for studying various properties of boolean functions, as it is closely related to the concept of linear correlation. Given a boolean function $f \in \mathcal{B}_n$

and an element $\alpha \in \mathbb{F}_2^n$, the Walsh transform of f at α is defined by

$$\widehat{f}(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \alpha, x \rangle \oplus f(x)} = \left|\{x \mid f(x) = \langle \alpha, x \rangle\}\right| - \left|\{x \mid f(x) \neq \langle \alpha, x \rangle\}\right| .$$

To make studying the properties of boolean functions easier, they can be partitioned according to an equivalence relation that preserves the properties of interest. Various equivalence relations have been used in the literature, but in this paper, we only use the extended variable-permutation equivalence as defined in [LP07].

Definition 1 (extended variable-permutation equivalence)**.** Two boolean functions f and g with n variables are said to be *extended variable-permutation equivalent* if there exist a mapping P corresponding to permutation of n variables, $a \in \mathbb{F}_2^n$, and $b \in \mathbb{F}_2$ such that for all $x \in \mathbb{F}_2^n$, we have $g(x) = f \circ P(x \oplus a) \oplus b$. In other words, g can be obtained from f by permuting and adding a constant in the input, with a possible inversion of the output.

In an equivalence relation, all functions that are equivalent to each other form an equivalence class, which can be represented by a single function in the class known as its *representative*. It is common practice to choose the lexicographically smallest function in the equivalence class as the representative. In this paper, we follow the same convention for representatives. We use the notation $\mathcal{B}_n^*$ to refer to the set of representatives in $\mathcal{B}_n$.

Correlation Immune and Resilient Functions

Definition 2 (correlation immune and resilient boolean function [Sie84, CGH⁺85])**.** A boolean function f is called t-th order correlation immune if its output distribution probability remains unchanged when at most t (or, equivalently, exactly t) of its input variables are fixed. It is called t-resilient if it is balanced and t-th order correlation immune. Equivalently, $f \in \mathcal{B}_n$ is t-th order correlation immune if $\widehat{f}(u) = 0$ for all $u \in \mathbb{F}_2^n$ with $1 \leq \mathrm{hw}(u) \leq t$, and it is t-resilient if $\widehat{f}(u) = 0$ for all $u \in \mathbb{F}_2^n$ with $\mathrm{hw}(u) \leq t$.

Note that when a function is t-th order correlation immune (or t-resilient), it does not necessarily mean that t is the maximum correlation immunity order of the function. To make this distinction clear, we use the term *maximum t-resilient* to refer to a function that is t-resilient, but not $(t+1)$-resilient.

By definition, a boolean function is 0-resilient if and only if it is balanced. Therefore, the set of all n-variable 0-resilient functions is the same as the set of all n-variable balanced functions.

Lemma 1. *t-resilience is invariant under the extended variable-permutation equivalence.*

Proof. If f and g are two equivalent boolean functions in $\mathcal{B}_n$, then there exist P, a mapping corresponding to a permutation of n variables, $a \in \mathbb{F}_2^n$, and $b \in \mathbb{F}_2$ such that for all $x \in \mathbb{F}_2^n$, we have $g(x) = f \circ P(x \oplus a) \oplus b$. Then,

$$\begin{aligned}
\widehat{g}(\alpha) &= \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \alpha, x \rangle \oplus g(x)} && = \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \alpha, x \rangle \oplus f \circ P(x \oplus a) \oplus b} \\
&= (-1)^b \cdot \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \alpha, x \rangle \oplus f \circ P(x \oplus a)} && = (-1)^b \cdot \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \alpha, x \oplus a \rangle \oplus f \circ P(x)} \\
&= (-1)^{\langle a, \alpha \rangle \oplus b} \cdot \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \alpha, x \rangle \oplus f \circ P(x)} && = (-1)^{\langle a, \alpha \rangle \oplus b} \cdot \widehat{f}(P(\alpha)) .
\end{aligned}$$

f is a t-resilient function if and only if $\widehat{f}(\alpha)$ is zero for any $\alpha \in \mathbb{F}_2^n$ with $\mathrm{hw}(\alpha) \leq t$. Since P is a mapping corresponding to a permutation of variables, it does not change the Hamming weight value. Hence, for any $\alpha \in \mathbb{F}_2^n$ with $\mathrm{hw}(\alpha) \leq t$, $\widehat{g}(\alpha)$ is also zero, meaning that g is a t-resilient function. □

Based on Lemma 1, we only need to consider correlation immune or resilient functions up to extended variable-permutation equivalence. In the rest of paper, when we refer to two functions being equivalent, we mean up to the extended variable-permutation equivalence.

We will use $\mathcal{R}_{n,t}$ to denote the set of all n-variable t-resilient boolean functions, and $\mathcal{R}^*_{n,t}$ to denote the set of all representatives in $\mathcal{R}_{n,t}$. Note that since each $(t+1)$-resilient function is also a t-resilient function, we have $\mathcal{R}_{n,t+1} \subset \mathcal{R}_{n,t}$ and $\mathcal{R}^*_{n,t+1} \subset \mathcal{R}^*_{n,t}$. Besides, $\mathcal{R}_{n,t} - \mathcal{R}_{n,t+1}$ (and $\mathcal{R}^*_{n,t} - \mathcal{R}^*_{n,t+1}$) is the set of all n-variable maximum t-resilient (representative) boolean functions.

A boolean function that can be represented as the direct sum of two smaller-dimension boolean functions is called a *decomposable function.* In other words, $h \in \mathcal{B}_{n+m}$ is called a decomposable function if it can be written as the direct sum of $f \in \mathcal{B}_n$ and $g \in \mathcal{B}_m$; i.e., for all $x \in \mathbb{F}_2^n$ and $y \in \mathbb{F}_2^m$, we have $h(x,y) = f(x) \oplus g(y)$.

Lemma 2. *[Sie84] The function h which is the direct sum of two functions $f \in \mathcal{R}_{n,t}$ and $g \in \mathcal{R}_{m,u}$, is a $(t+u+1)$-resilient function.*

Let $f \in \mathcal{B}_n$ and $g \in \mathcal{B}_{n+1}$. We call g is the type-0 extension of f if for all $x \in \mathbb{F}_2^n$ and $x_n \in \mathbb{F}_2$, g is defined as $g(x,x_n) = f(x)$, and we call it is the type-1 extension of f if $g(x,x_n) = f(x) \oplus x_n$. Note that if f is a t-resilient function, then its type-0 and type-1 extensions are t- and $(t+1)$-resilient functions, respectively.

Proposition 1. *[Sie84] Any t-th order correlation immune n-variable boolean function has algebraic degree of at most $n-t$. Additionally, any t-resilient function has algebraic degree at most $n-t-1$ if $t < n-1$, and has degree 1 (i.e., is affine) if $t = n-1$.*

Based on Proposition 1, Siegenthaler classified all n-, $(n-1)$-, and $(n-2)$-th order correlation immune n-variable boolean functions.

Lemma 3. *[Sie84] An n-variable boolean function is n-th order correlation immune if and only if it is a constant function. A non-constant n-variable boolean function is $(n-1)$-th correlation immune if and only if it is equal to $x_0 \oplus \ldots \oplus x_{n-1} \oplus c$ with $c \in \mathbb{F}_2$. Moreover, a non-constant boolean function is maximum $(n-2)$-th correlation immune if and only if it is equal to $x_0 \oplus \ldots \oplus x_{j-1} \oplus x_{j+1} \oplus \ldots \oplus x_{n-1} \oplus c$ with $0 \le j < n$ and $c \in \mathbb{F}_2$. Therefore, the only function in $\mathcal{R}^*_{n,n-1}$ is $f(x) = x_0 \oplus \ldots \oplus x_{n-1}$, and the only function in $\mathcal{R}^*_{n,n-2} - \mathcal{R}^*_{n,n-1}$ is $f(x) = x_1 \oplus \ldots \oplus x_{n-1}$.*

Siegenthaler also introduced a construction for building $(n+1)$-variable $(t+1)$-resilient functions using n-variable t-resilient functions, which is known as the *Siegenthaler's construction.* This construction is explained in detail in Theorem 1 which is the main principle used in the next section to construct $\mathcal{R}^*_{n+1,t+1}$ using all the functions in $\mathcal{R}^*_{n,t}$.

Theorem 1. *[Sie84] Let $f \in \mathcal{B}_{n+1}$, and $f_0 \in \mathcal{B}_n$ and $f_1 \in \mathcal{B}_n$ be the two functions derived from f using the following equation:*

$$f(x,x_n) = \overline{x_n} \cdot f_0(x) \,\oplus\, x_n \cdot f_1(x) \quad \forall\, x \in \mathbb{F}_2^n \text{ and } x_n \in \mathbb{F}_2\,.$$

If both f_0 and f_1 are t-resilient functions, then f is also a t-resilient function. Furthermore, f is $(t+1)$-resilient if and only if:

- *both f_0 and f_1 are t-resilient functions, and*
- *for any $\alpha \in \mathbb{F}_2^n$ with $\mathrm{hw}(\alpha) = t+1$, $\widehat{f_1}(\alpha) = -\widehat{f_0}(\alpha)$.*

Note that by following Siegenthaler's construction, *any* boolean function with $n+1$ variables $(n > 1)$ can be decomposed into two functions with n variables. Additionally, the truth table of f can be obtained by concatenating the truth tables of f_0 and f_1, i.e., $T_f = T_{f_0} \,\|\, T_{f_1}$.

Consider the Siegenthaler's construction for a fixed $f_0 \in \mathcal{R}_{n,t}$. One trivial solution for $f_1 \in \mathcal{R}_{n,t}$ to make the resulting $(n+1)$-variable function f a $(t+1)$-resilient function is to define $f_1(x) = f_0(x) \oplus 1$ for all $x \in \mathbb{F}_2^n$.

In this case, the resulting $(n+1)$-variable $(t+1)$-resilient function is given by $f(x, x_n) = f_0(x) \oplus x_n$ for all $x \in \mathbb{F}_2^n$ and $x_n \in \mathbb{F}_2$, which is the type-1 extension of f_0. Therefore,

$$\left\{f \in \mathcal{B}_{n+1} \,|\, f(x, x_n) = g(x) \oplus x_n \;\; \forall\, x \in \mathbb{F}_2^n \text{ and } x_n \in \mathbb{F}_2\,,\ \text{with } g \in \mathcal{R}^*_{n,t}\right\} \subset \mathcal{R}^*_{n+1,t+1}\,.$$

Another solution, introduced in [CCCS91], is defined by $f_1(x) = f_0(\overline{x}) \oplus \epsilon$ for all $x \in \mathbb{F}_2^n$, with $\epsilon = t \mod 2$. Then the Walsh transform of f_1 for any $\alpha \in \mathbb{F}_2^n$ will be

$$\begin{aligned}
\widehat{f_1}(\alpha) &= \sum_{x\in\mathbb{F}_2^n} (-1)^{\langle \alpha, x\rangle \oplus f_1(x)} &&= \sum_{x\in\mathbb{F}_2^n} (-1)^{\langle \alpha, x\rangle \oplus f_0(\overline{x}) \oplus \epsilon} \\
&= (-1)^{\epsilon} \cdot \sum_{x\in\mathbb{F}_2^n} (-1)^{\langle \alpha, \overline{x}\rangle \oplus f_0(x)} &&= (-1)^{\mathrm{hp}(\alpha)\oplus\epsilon} \cdot \sum_{x\in\mathbb{F}_2^n} (-1)^{\langle \alpha, x\rangle \oplus f_0(x)} \\
&= (-1)^{\mathrm{hp}(\alpha)\oplus\epsilon} \cdot \widehat{f_0}(\alpha) &&= (-1)^{\mathrm{hw}(\alpha)+t} \cdot \widehat{f_0}(\alpha)\,,
\end{aligned}$$

and the resulting $(n+1)$-variable $(t+1)$-resilient function is given by:

$$\begin{aligned}
f(x, x_n) &= \overline{x_n} \cdot f_0(x) \,\oplus\, x_n \cdot f_1(x) &&= \overline{x_n} \cdot f_0(x) \,\oplus\, x_n \cdot \big(f_0(\overline{x}) \oplus \epsilon\big) \\
&= (x_n \oplus 1) \cdot f_0(x) \,\oplus\, x_n \cdot \big(f_0(\overline{x}) \oplus \epsilon\big) &&= x_n \cdot \big(f_0(x) \oplus f_0(\overline{x}) \oplus \epsilon\big) \oplus f_0(x)\,.
\end{aligned}$$

Note that if $f_0(x) \oplus f_0(\overline{x}) = \overline{\epsilon}$ for all $x \in \mathbb{F}_2^n$, then the aforementioned f function is equal to the type-1 extension of the f_0 function.

Based on Siegenthaler's construction, Camion, Carlet, Charpin, and Sendrier classified all the functions in $\mathcal{R}^*_{n,n-3}$ [CCCS91].

Lemma 4. *[CCCS91] $\mathcal{R}^*_{n,n-3} - \mathcal{R}^*_{n,n-2}$ includes only five functions:*

- $f(x) = x_2 \oplus \ldots \oplus x_{n-1}$ $\quad (n \geq 3)\,,$
- $f(x) = x_0x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1}$ $\quad (n \geq 3)\,,$
- $f(x) = x_0x_1 \oplus x_0x_2 \oplus x_2 \oplus \ldots \oplus x_{n-1}$ $\quad (n \geq 3)\,,$
- $f(x) = x_0x_1 \oplus x_0x_2 \oplus x_1x_2 \oplus x_3 \oplus \ldots \oplus x_{n-1}$ $\quad (n \geq 3)\,,$
- $f(x) = x_0x_1 \oplus x_0x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_2 \oplus \ldots \oplus x_{n-1}$ $\quad (n \geq 4)\,.$

3 An Algorithm for Classifying $\mathcal{R}^*_{n,n-m}$

In this section, we explain our approach for classifying all n-variable $(n-m)$-resilient boolean functions up to the extended variable-permutation equivalence. This approach follows the principle of Siegenthaler's constructions introduced in Theorem 1. We introduce several speed-up techniques that help us to develop the basic search algorithm into an efficient one that enables us to compute $\mathcal{R}^*_{n,n-4}$ for any value of n.

Basic Approach based on Siegenthaler's Construction

Based on Siegenthaler's construction, to compute $\mathcal{R}^*_{n,n-m}$ for a fixed value of m and all values of $n \geq m$, we start by using $\mathcal{B}^*_{m-1}$ to compute $\mathcal{R}^*_{m,0}$. Then, we use $\mathcal{R}^*_{m,0}$ to compute $\mathcal{R}^*_{m+1,1}$, and so on. We continue these steps until the number of representatives in $\mathcal{R}^*_{n+1,n-m+1}$ is the same as the number of representatives in $\mathcal{R}^*_{n,n-m}$, i.e., each representative in $\mathcal{R}^*_{n+1,n-m+1}$ is a type-1 extension of a representative in $\mathcal{R}^*_{n,n-m}$. Note that Tarannikov and Kirienko showed in [TK00] that such an n value exists, which they denoted by $p(m)$, and proved that $p(4) = 10$.

To explain our approach, consider that we have already computed $\mathcal{R}^*_{n,t}$. To compute all functions in $\mathcal{R}^*_{n+1,t+1}$, the simplest approach is to take two functions f_0 and f_1 from $\mathcal{R}_{n,t}$ and check the condition for the values of their Walsh transform at α points where $\mathrm{hw}(\alpha) = t+1$. Then, we check if the resulting function of Siegenthaler's construction, f, is a representative function.

Lemma 5. *Let $f \in \mathcal{B}_{n+1}$ and $f_0 \in \mathcal{B}_n$ and $f_1 \in \mathcal{B}_n$ be the functions derived from f using the following equation:*

$$f(x, x_n) = \overline{x_n} \cdot f_0(x) \,\oplus\, x_n \cdot f_1(x) \quad \forall\, x \in \mathbb{F}_2^n \text{ and } x_n \in \mathbb{F}_2\,.$$

If f is a representative function, then f_0 is also a representative function and is lexicographically smaller than or equal to the representative function for the class of f_1.

Proof. If f is an $(n+1)$-variable representative, then the truth table for $f(x, x_n)$ is lexicographically smaller than or equal to the truth table for any function equivalent to f. Note that the truth table of $f(x, x_n)$ can be represented as $(T_{f_0(x)} \,\|\, T_{f_1(x)})$.

Let $f_i(x) = f_i^* \circ P_i(x \oplus a_i) \oplus b_i$ for all values of $x \in \mathbb{F}_2^n$ and $i \in \{0, 1\}$, with $f_i^* \in \mathcal{R}^*_{n,t}$, P_i, a mapping corresponding to a permutation of n variables, $a_i \in \mathbb{F}_2^n$, and $b_i \in \mathbb{F}_2$. Then, we have $f_i^*(x) = f_i\big(P_i^{-1}(x) \oplus a_i\big) \oplus b_i$ for all values of $x \in \mathbb{F}_2^n$ and $i \in \{0, 1\}$.

The truth table of $f\big(P_0^{-1}(x) \oplus a_0, x_n\big) \oplus b_0$ is equal to $(T_{f_0^*(x)} \,\|\, T_{f_1(P_0^{-1}(x)\oplus a_0)\oplus b_0})$, which lexicographically must be greater than or equal to $(T_{f_0(x)} \,\|\, T_{f_1(x)})$; that is,

$$(T_{f_0^*(x)} \,\|\, T_{f_1(P_0^{-1}(x)\oplus a_0)\oplus b_0}) \geq (T_{f_0(x)} \,\|\, T_{f_1(x)}) \quad \Rightarrow \quad T_{f_0^*(x)} \geq T_{f_0(x)}\,.$$

However, since f_0^* is an n-variable representative, the the truth table for $f_0^*(x)$ is lexicographically smaller than or equal to the truth table for $f_0(x)$; i.e., that is $T_{f_0^*(x)} \leq T_{f_0(x)}$. Combining these two inequalities, we have $T_{f_0^*(x)} = T_{f_0(x)}$ or, equivalently, $f_0^* = f_0$ which means that f_0 is a representative function.

Besides, the truth table of $f\big(P_1^{-1}(x) \oplus a_1, \overline{x_n}\big) \oplus b_1$ is equal to $(T_{f_1^*(x)} \,\|\, T_{f_0(P_1^{-1}(x)\oplus a_1)\oplus b_1})$, which lexicographically must be greater than or equal to $(T_{f_0(x)} \,\|\, T_{f_1(x)})$. Hence, $T_{f_1^*(x)} \geq T_{f_0(x)} = T_{f_0^*(x)}$ which means that the representative function for the class of f_1 is lexicographically greater than or equal to the representative function f_0. □

Based on Lemma 5, to compute $\mathcal{R}^*_{n+1,t+1}$, we do not need to go through all $|\mathcal{R}_{n,t}|^2$ possible choices for (f_0, f_1). It is enough to take two representative functions f_0^* and f_1^*, with f_0^* being lexicographically smaller or equal to f_1^*. For each function f_1 equivalent to f_1^*, we check the condition for values of the Walsh transform at α points with $\mathrm{hw}(\alpha) = t+1$ for f_0^* and f_1 functions. Then, we check if the resulting function of Siegenthaler's construction, f, is a representative function.

Note that there are at most $2^{n+1} \cdot n!$ functions equivalent to each representative function. This means that for each (f_0^*, f_1^*) representative pair, we need to repeat the condition check $2^{n+1} \cdot n!$ times. For all the operations and computations required for a fixed (f_0^*, f_1^*) representative pair, we refer to it as the iteration for (f_0^*, f_1^*) representative pair.

In the following, we first focus on reducing the number of representative pairs from $\mathcal{R}^*_{n,t}$ that need to be evaluated to build $\mathcal{R}^*_{n+1,t+1}$. Then, we focus on the amount of computation needed within each iteration for a representative pair.

As a result of Lemma 5, we need to go through $|\mathcal{R}^*_{n,t}| \cdot \big(|\mathcal{R}^*_{n,t}| + 1\big)/2$ iterations for representative pairs in $\mathcal{R}^*_{n,t}$. Based on the numbers reported in Table 2 for $|\mathcal{R}^*_{n,n-4}|$, to compute $\mathcal{R}^*_{n+1,n-3}$ with $n = 3$ to $n = 10$, we need to go through 105, 1 711, 32 896, 167 331, 259 560, 284 635, 289 180, and 289 941 iterations, respectively.

Technique 1 for Reducing Number of Representative Pairs

Lemma 6. *Let $f \in \mathcal{B}_{n+1}$ and $f_0 \in \mathcal{B}_n$ and $f_1 \in \mathcal{B}_n$ be the two functions derived from f using the following equation:*

$$f(x, x_n) = \overline{x_n} \cdot f_0(x) \oplus x_n \cdot f_1(x) \quad \forall\, x \in \mathbb{F}_2^n \textit{ and } x_n \in \mathbb{F}_2\,.$$

If f is representative and f_0 is a type-1 extension of a function in $\mathcal{B}_{n-1}$, then f is the type-1 extension of f_0.

Proof. Let f_{00}, f_{10} f_{01} and f_{11} be the four $(n-1)$-variable boolean functions derived from f using the following equation for all $x' \in \mathbb{F}_2^{n-1}$ and $x_n, x_{n-1} \in \mathbb{F}_2$:

$$\begin{aligned} &f(x', x_{n-1}, x_n) = \\ &\qquad \overline{x_{n-1}} \cdot \overline{x_n} \cdot f_{00}(x') \oplus x_{n-1} \cdot \overline{x_n} \cdot f_{10}(x') \oplus \overline{x_{n-1}} \cdot x_n \cdot f_{01}(x') \oplus x_{n-1} \cdot x_n \cdot f_{11}(x')\,. \end{aligned}$$

Then the truth table of $f(x', x_{n-1}, x_n)$ can be represented by $(T_{f_{00}} \,\|\, T_{f_{10}} \,\|\, T_{f_{01}} \,\|\, T_{f_{11}})$, and since f is representative, its truth table is lexicographically smaller than or equal to the truth table for any function equivalent to f. Furthermore, since f_0 is a type-1 extension of an $(n-1)$-variable function, we have $f_{10}(x') = f_{00}(x') \oplus 1$ for all $x' \in \mathbb{F}_2^{n-1}$, or equivalently, $T_{f_{10}} = \overline{T_{f_{00}}}$. Then, we have

$$\begin{aligned} T_{f(x', x_{n-1}, x_n)} &= (T_{f_{00}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{01}} \,\|\, T_{f_{11}}) \\ T_{f(x', x_n, x_{n-1})} &= (T_{f_{00}} \,\|\, T_{f_{01}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{11}}) \geq (T_{f_{00}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{01}} \,\|\, T_{f_{11}}) \quad \Rightarrow\; T_{f_{01}} \geq \overline{T_{f_{00}}} \\ T_{f(x', \overline{x_{n-1}}, x_n) \oplus 1} &= (T_{f_{00}} \,\|\, \overline{T_{f_{00}}} \,\|\, \overline{T_{f_{11}}} \,\|\, \overline{T_{f_{01}}}) \geq (T_{f_{00}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{01}} \,\|\, T_{f_{11}}) \quad \Rightarrow\; \overline{T_{f_{11}}} \geq T_{f_{01}} \\ T_{f(x', \overline{x_n}, x_{n-1}) \oplus 1} &= (T_{f_{00}} \,\|\, \overline{T_{f_{11}}} \,\|\, \overline{T_{f_{00}}} \,\|\, \overline{T_{f_{01}}}) \geq (T_{f_{00}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{01}} \,\|\, T_{f_{11}}) \quad \Rightarrow\; \overline{T_{f_{11}}} \geq \overline{T_{f_{00}}} \\ T_{f(x', x_n, \overline{x_{n-1}})} &= (T_{f_{01}} \,\|\, T_{f_{00}} \,\|\, T_{f_{11}} \,\|\, \overline{T_{f_{00}}}) \geq (T_{f_{00}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{01}} \,\|\, T_{f_{11}}) \quad \Rightarrow\; T_{f_{01}} \geq T_{f_{00}} \\ T_{f(x', \overline{x_{n-1}}, \overline{x_n})} &= (T_{f_{11}} \,\|\, T_{f_{01}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{00}}) \geq (T_{f_{00}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{01}} \,\|\, T_{f_{11}}) \quad \Rightarrow\; T_{f_{11}} \geq T_{f_{00}} \end{aligned}$$

$\overline{T_{f_{11}}} \geq \overline{T_{f_{00}}}$ is equivalent to $T_{f_{11}} \leq T_{f_{00}}$, and since we also have $T_{f_{11}} \geq T_{f_{00}}$, it follows that $T_{f_{11}} = T_{f_{00}}$, or equivalently, $f_{11} = f_{00}$. Similarly, combining $T_{f_{01}} \geq \overline{T_{f_{00}}}$ and $\overline{T_{f_{11}}} \geq T_{f_{01}}$ with $f_{11} = f_{00}$ yields $T_{f_{01}} = \overline{T_{f_{00}}}$, or equivalently, $f_{01} = f_{00} \oplus 1$. Altogether, the truth table of $f(x', x_{n-1}, x_n)$ will be $(T_{f_{00}} \,\|\, \overline{T_{f_{00}}} \,\|\, \overline{T_{f_{00}}} \,\|\, T_{f_{00}})$ which is equivalent to $f(x', x_{n-1}, x_n) = f_{00}(x') \oplus x_{n-1} \oplus x_n = f_0(x) \oplus x_n$ for all $x' \in \mathbb{F}_2^{n-1}$ and $x_{n-1}, x_n \in \mathbb{F}_2$. □

Lemma 6 enables us to reduce the number of representative pairs from $\mathcal{R}^*_{n,t}$ that need to be evaluated to construct $\mathcal{R}^*_{n+1,t+1}$ in the following manner. We know that the type-1 extension of each function in $\mathcal{R}^*_{n,t}$ is already included in $\mathcal{R}^*_{n+1,t+1}$. Hence, we only need to find all the representatives in $\mathcal{R}^*_{n+1,t+1}$ that are not type-1 extensions of representative functions in $\mathcal{R}^*_{n,t}$. We denote the set of such functions by $\mathcal{R}^\dagger_{n+1,t+1}$, i.e.,

$$\begin{aligned} &\mathcal{R}^\dagger_{n+1,t+1} = \mathcal{R}^*_{n+1,t+1} \\ &\qquad - \left\{ f \in \mathcal{B}_{n+1} \,\middle|\, f(x, x_n) = g(x) \oplus x_n \;\; \forall\, x \in \mathbb{F}_2^n \text{ and } x_n \in \mathbb{F}_2\,,\ \text{with } g \in \mathcal{R}^*_{n,t} \right\}. \end{aligned}$$

Based on Lemma 6, any representative pair (f_0^*, f_1^*) from $\mathcal{R}^*_{n,t}$ with f_0^* being a type-1 extension (of a function in $\mathcal{R}^*_{n-1,t-1}$) only produces a single representative in $\mathcal{R}^*_{n+1,t+1}$, and this representative is the type-1 extension of f_0^*. Therefore, to compute $\mathcal{R}^\dagger_{n+1,t+1}$, we only need to consider representative pairs (f_0^*, f_1^*) from $\mathcal{R}^*_{n,t}$ where f_0^* is not a type-1 extension (i.e., $f_0^* \in \mathcal{R}^\dagger_{n,t}$) and it is lexicographically smaller or equal to f_1^*.

Applying this technique guarantees that the number of representative pairs which need to be evaluated for computing $\mathcal{R}^\dagger_{n+1,t+1}$ (and accordingly for computing $\mathcal{R}^*_{n+1,t+1}$) is less than $|\mathcal{R}^\dagger_{n,t}| \cdot |\mathcal{R}^*_{n,t}|$. We emphasize that this number is only an upper bound since we only need to use (f_0^*, f_1^*) representative pairs which satisfy the condition f_0^* is smaller or equal to f_1^*, and determining the exact value requires knowledge of functions in $\mathcal{R}_{n,t}^*$.

The exact number of such representative pairs needed for computing $\mathcal{R}_{n+1,n-3}^{\dagger}$ with $n = 3$ until $n = 10$ is reported in Table 2. Specifically, we need to go through 90, 1 429, 26 385, 89 855, 43 874, 8 009, 773, and 62 iterations, respectively. As you see, applying only technique 1 results in a significant reduction in the number of representative pairs from $\mathcal{R}_{n,t}^*$ that need to be considered for computing $\mathcal{R}_{n+1,t+1}^{\dagger}$, particularly as the value of n increases.

Technique 2 for Reducing Number of Representative Pairs

Let $f_0^* \in \mathcal{R}_{n,t}^*$ such that, using the Siegenthaler's construction, there exists exactly one f_1 among all functions in $\mathcal{R}_{n,t}$ that can be used to construct an $(n+1)$-variable $(t+1)$-resilient function f. Note that this implies $f_1(x) = f_0^*(x) \oplus 1 = f_0^*(\overline{x}) \oplus \epsilon$ with $\epsilon = t \bmod 2$. Accordingly, $f(x, x_n) = f_0^*(x) \oplus x_n$ for all $x \in \mathbb{F}_2^n$ and $x_n \in \mathbb{F}_2$, which is the type-1 extension of f_0^*. We refer to such a function f_0^* as a *single-solution* representative.

Lemma 7. *Let $f_0^* \in \mathcal{R}_{n,t}^*$ be a single-solution representative, and $f^* \in \mathcal{R}_{n+1,t+1}^*$ be the type-1 extension of f_0^*. Then, for any function $g^* \in \mathcal{R}_{n+1,t+1}^* \setminus \{f^*\}$, the iterations related to $(n+1)$-variable representative pairs (f^*, g^*) and (g^*, f^*) cannot produce an $(n+2)$-variable $(t+2)$-resilient function.*

Proof. Assume that the representative pair (f^*, g^*) can build an $(n+2)$-variable $(t+2)$-resilient function. Then, there is a function g equivalent to g^* such that $g(x, x_n) = \overline{x_n} \cdot g_0(x) \oplus x_n \cdot g_1(x)$ for all $x \in \mathbb{F}_2^n$ and $x_n \in \mathbb{F}_2$, where g_0 and g_1 are both t-resilient. Furthermore, there is an $(n+2)$-variable $(t+2)$-resilient function h such that, for all $x \in \mathbb{F}_2^n$ and $x_n, x_{n+1} \in \mathbb{F}_2$, we have

$$\begin{aligned} h(x, x_n, x_{n+1}) &= \overline{x_{n+1}} \cdot f^*(x, x_n) \oplus x_{n+1} \cdot g(x, x_n) = \\ &\overline{x_n} \cdot \overline{x_{n+1}} \cdot f_0^*(x) \oplus x_n \cdot \overline{x_{n+1}} \cdot (f_0^*(x) \oplus 1) \oplus \overline{x_n} \cdot x_{n+1} \cdot g_0(x) \oplus x_n \cdot x_{n+1} \cdot g_1(x)\,. \end{aligned}$$

Since h is a $(t+2)$-resilient function, the following two $(n+1)$-variable functions must also be $(t+1)$-resilient:

- h' defined by $h'(x, x_n) = \overline{x_n} \cdot f_0^*(x) \oplus x_n \cdot g_0(x)$ for all $x \in \mathbb{F}_2^n$ and $x_n \in \mathbb{F}_2$,
- h'' defined by $h''(x, x_n) = \overline{x_n} \cdot \big(f_0^*(x) \oplus 1\big) \oplus x_n \cdot g_1(x)$ for all $x \in \mathbb{F}_2^n$ and $x_n \in \mathbb{F}_2$.

Since f_0^* is a single-solution function, h' and h'' can be $(t+1)$-resilient if $g_0(x) = f_0^*(x) \oplus 1$ and $g_1(x) = f_0^*(x)$ for all $x \in \mathbb{F}_2^n$. This implies that $g(x, x_n) = f^*(x, x_n) \oplus 1$ for all $x \in \mathbb{F}_2^n$ and $x_n \in \mathbb{F}_2$, and thus $g^* = f^*$, which contradicts the assumption of the lemma that $g^* \in \mathcal{R}_{n+1,t+1}^* \setminus \{f^*\}$. A similar approach holds for the case of the representative pair (g^*, f^*), that this representative pair can build an $(n+2)$-variable $(t+2)$-resilient function only if $f^* = g^*$. □

We use Lemma 7 for further reduction on the number of representative pairs needed to build $\mathcal{R}_{n+2,t+2}^{\dagger}$. At the step of computing $\mathcal{R}_{n+1,t+1}^{\dagger}$, for a fixed f_0^* representative in $\mathcal{R}_{n,t}^{\dagger}$, while iterating through all (f_0^*, f_1^*) representative pairs, we check if f_0^* is a single-solution representative or not. Based on this saved information about each $f_0^* \in \mathcal{R}_{n,t}^{\dagger}$, we form the following set at the end of the current step:

$$\mathcal{R}_{n+1,t+1}^{\ddagger} = \mathcal{R}_{n+1,t+1}^* - \{\text{type-1 extension of all } f \in \mathcal{R}_{n,t}^* \text{ such that } f \text{ is single-solution}\}\,.$$

Then, in the next step, for computing $\mathcal{R}_{n+2,t+2}^{\dagger}$, we only need to iterate through all (g_0^*, g_1^*) representative pairs with $g_0^* \in \mathcal{R}_{n+1,t+1}^{\dagger}$, $g_1^* \in \mathcal{R}_{n+1,t+1}^{\ddagger}$, and g_0^* lexicographically smaller than or equal to g_1^*.

Note that for all possible values of n and t, we always have $\mathcal{R}^{\dagger}_{n,t} \subset \mathcal{R}^{\ddagger}_{n,t} \subset \mathcal{R}^{*}_{n,t}$.

By applying this technique, we can reduce the number of representative pairs that need to be evaluated for computing $\mathcal{R}^{\dagger}_{n+2,t+2}$ with negligible overhead computation. The exact number of such representative pairs for computing $\mathcal{R}^{\dagger}_{n+2,n-2}$ with $n = 3$ up to $n = 9$ is reported in Table 2. That is, we need to go through 1 266, 24 356, 79 631, 28 450, 1 919, 61, and 3 iterations, respectively.

Technique 3 for Reducing Number of Representative Pairs

Lemma 8. *In Siegenthaler's construction, the two functions f_0 and f_1 from $\mathcal{R}_{n,t}$ can form $(n+1)$-variable $(t+1)$-resilient functions if the distribution of magnitudes for their representatives' Walsh transform at the α points with $\mathrm{hw}(\alpha) = t+1$ are equal.*

Proof. For each $i \in \{0,1\}$, we define the multiset A_i as follows:

$$A_i = \left\{ |\widehat{f}_i(\alpha)| \;\middle|\; \alpha \in \mathbb{F}_2^n \text{ with } \mathrm{hw}(\alpha) = t+1 \right\}.$$

According to Theorem 1, f_0 and f_1 from $\mathcal{R}_{n,t}$ can construct $(n+1)$-variable $(t+1)$-resilient functions in Siegenthaler's construction if $A_0 = A_1$, and this is due to that for all $\alpha \in \mathbb{F}_2^n$ with $\mathrm{hw}(\alpha) = t+1$, we have $\widehat{f}_1(\alpha) = -\widehat{f}_0(\alpha)$.

Let $f_i(x) = f_i^* \circ P_i(x \oplus a) \oplus b_i$ for all $x \in \mathbb{F}_2^n$ and $i \in \{0,1\}$, where $f_i^* \in \mathcal{R}^*_{n,t}$, P_i is a mapping corresponding to a permutation of n variables, $a_i \in \mathbb{F}_2^n$, and $b_i \in \mathbb{F}_2$. Hence, for $\alpha \in \mathbb{F}_2^n$ and $i \in \{0,1\}$, we have $\widehat{f}_i(\alpha) = (-1)^{\langle a_i, \alpha \rangle \oplus b_i} \cdot \widehat{f_i^*}(P_i(\alpha))$ and consequently, $|\widehat{f}_i(\alpha)| = |\widehat{f_i^*}(P_i(\alpha))|$. Therefore,

$$\begin{aligned} A_i &= \left\{ |\widehat{f_i^*}(P_i(\alpha))| \;\middle|\; \alpha \in \mathbb{F}_2^n \text{ with } \mathrm{hw}(\alpha) = t+1 \right\} \\ &= \left\{ |\widehat{f_i^*}(\alpha)| \;\middle|\; \alpha \in \mathbb{F}_2^n \text{ with } \mathrm{hw}\left(P_i^{-1}(\alpha)\right) = t+1 \right\} \\ &= \left\{ |\widehat{f_i^*}(\alpha)| \;\middle|\; \alpha \in \mathbb{F}_2^n \text{ with } \mathrm{hw}(\alpha) = t+1 \right\}. \end{aligned}$$

Note that the last equality is a result of the fact that P_i is a mapping corresponding to a permutation of variables, which preserves the Hamming weight value. As $A_0 = A_1$, we have

$$\left\{ |\widehat{f_0^*}(\alpha)| \;\middle|\; \alpha \in \mathbb{F}_2^n \text{ with } \mathrm{hw}(\alpha) = t+1 \right\} = \left\{ |\widehat{f_1^*}(\alpha)| \;\middle|\; \alpha \in \mathbb{F}_2^n \text{ with } \mathrm{hw}(\alpha) = t+1 \right\}.$$

This implies that the distribution of magnitudes for representatives' Walsh transform at the points with Hamming weight $t+1$ is the same for both functions. □

We can apply Lemma 8 to reduce the number of representative pairs that need to be evaluated from $\mathcal{R}^*_{n,t}$ in order to construct $\mathcal{R}^*_{n+1,t+1}$. For each representative pair remaining after technique 2, we check the distribution of magnitudes for the Walsh transform at points with Hamming weight $t+1$.

Note that for a fixed m and $t = n - m$, the number of points with Hamming weight of $t+1$ is equal to

$$\binom{n}{n-m+1} = \frac{n \cdot (n-1) \cdot \cdots \cdot (n-m+2)}{(m-1)!}.$$

As n increases, the number of such points also increases. Therefore, the probability that two representatives from $\mathcal{R}^*_{n,t}$ have the same distribution at points with Hamming weight $n-m+1$ decreases significantly.

We have reported the number of representative pairs for the case of $m = 4$ in Table 2.

The number of pairs that need to be evaluated for computing $\mathcal{R}^{\dagger}_{n+1,n-3}$ with n ranging from 3 to 10 are 23, 133, 1 911, 6 423, 1 779, 149, 8, and 1, respectively.

Computations for Each Iteration with a Representative Pair

Let (f_0^*, f_1^*) be a representative pair from $\mathcal{R}_{n,t}^*$ that satisfies all the conditions from the previous three techniques. To determine whether this representative pair can be used to construct $(n+1)$-variable $(t+1)$-resilient functions using the Siegenthaler's construction, we need to examine all (at most) $2^{n+1} \cdot n!$ equivalent functions within the class of f_1^*.

Let f_1 be an equivalent function to f_1^*, where $f_1(x) = f_1^* \circ P(x \oplus a) \oplus b$ for all $x \in \mathbb{F}_2^n$, with P representing a mapping corresponding to a permutation of n variables, $a \in \mathbb{F}_2^n$, and $b \in \mathbb{F}_2$. Using Siegenthaler's construction, f_0^* and f_1 can construct a $(t+1)$-resilient function if for all $\alpha \in \mathbb{F}_2^n$ with $\mathrm{hw}(\alpha) = t+1$, we have $\widehat{f_1}(\alpha) = -\widehat{f_0^*}(\alpha)$, which means $|\widehat{f_1}(\alpha)| = |\widehat{f_0^*}(\alpha)|$. Moreover, since $\widehat{f_1}(\alpha) = (-1)^{\langle a,\alpha\rangle \oplus b} \cdot \widehat{f_1^*}\big(P(\alpha)\big)$, we must have $|\widehat{f_0^*}(\alpha)| = |\widehat{f_1^*}\big(P(\alpha)\big)|$ for each $\alpha \in \mathbb{F}_2^n$ with $\mathrm{hw}(\alpha) = t+1$. This condition only depends on the mapping P and is independent of the values for a or b. Therefore, instead of checking all $2^{n+1} \cdot n!$ equivalent functions, we only need to consider all $n!$ possible choices for the mapping P and verify if $|\widehat{f_0^*}(\alpha)| = |\widehat{f_1^*}\big(P(\alpha)\big)|$ for all $\alpha \in \mathbb{F}_2^n$ with $\mathrm{hw}(\alpha) = t+1$.

It should be noted that when checking this condition for different P mappings, there is no need to repeat the computation of the Walsh transform for the function $f_1^* \circ P$ for each choice of P mapping. It is sufficient to have the Walsh transform values of the functions f_0^* and f_1^* (at α points with $\mathrm{hw}(\alpha) = t+1$) computed previously, at the starting point for the step of computing $\mathcal{R}_{n+1,t+1}^*$.

For a representative pair (f_0^*, f_1^*) and a mapping P, in the case where the condition for magnitudes of the Walsh transforms at the α points is satisfied, we then check if there exist any $a \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2$ that satisfy the condition for signs of the Walsh transform at these points. If such values for a and b exist, then we have successfully built an $(n+1)$-variable $(t+1)$-resilient function. Note that this step could be made more efficient, but since it is not the bottleneck for the computational complexity of our algorithm, we leave it in its current simple form.

The bottleneck for the computational complexity of our algorithm (up to this point) is when we go through all $n!$ possible mappings of P for each representative pair (f_0^*, f_1^*). However, since there are very few mappings P that can pass the condition for magnitudes of the Walsh transform at the α points with $\mathrm{hw}(\alpha) = t+1$, going through all choices for a and b will not increase the computational complexity of the algorithm. In other words, if N_3 denotes the number of remaining representative pairs after technique 3 is applied, the computational complexity of the step for building $\mathcal{R}_{n+1,t+1}^*$ will be $N_3 \cdot n!$ times the cost of a few look-up tables.

Checking Representativeness of a Function

All $(n+1)$-variable $(t+1)$-resilient functions produced within each iteration are not necessarily representative functions. Therefore, for each $(n+1)$-variable $(t+1)$-resilient function f built by Siegenthaler's construction, we need to check if it is representative. To do this, we go through all possible choices for P, a mapping corresponding to a permutation of $n+1$ variables, and $a \in \mathbb{F}_2^{n+1}$. For each mapping P and constant a, we fix $b \in \mathbb{F}_2$ to the value of $f \circ P(a)$ and compute the function $f'(x) = f \circ P(x \oplus a) \oplus b$ for all values of $x \in \mathbb{F}_2^n$. We then check if the function f' is lexicographically smaller than f.

Note that if there is a single choice for mapping P and constant a such that the corresponding equivalent function f' is smaller than f, it is enough to decide that f is not a representative function. Also, it is not necessary to compute all the truth table for function f' to compare it with f. We only need to compute its truth table until the point $y \in \mathbb{F}_2^n$ such that $f(x) = f'(x)$ for all $x \in \mathbb{F}_2^{n+1}$ with $x < y$ and $f(y) \neq f'(y)$. Note that the point y always exists if $f \neq f'$.

If f is a type-1 extension of an n-variable representative function f_0^*, then it is definitely a representative function. Hence, before checking representativeness of functions, we check for being a type-1 extension. Therefore, only the representative functions in $\mathcal{R}^{\dagger}_{n+1,t+1}$ will go through all possible choices for mapping P and constant a.

By applying these techniques, with comparatively smaller complexity, we can check if a function is not a representative. However, if the function is in $\mathcal{R}^{\dagger}_{n+1,t+1}$, we need to go through all possible choices for mapping P and constant a to make sure that it is a representative function. This means that the computational complexity of this part is about $|\mathcal{R}^{\dagger}_{n+1,t+1}| \cdot (n+1)! \cdot 2^{n+1}$ times of partially computing an equivalent function.

Ending Step of the Algorithm

Lemma 7 not only helps us develop our algorithm for building $\mathcal{R}^{*}_{n,n-m}$ for each n value, but it also provides information about the step in our algorithm where we should stop.

Lemma 9. *If $|\mathcal{R}^{*}_{n+1,n-m+1}| = |\mathcal{R}^{*}_{n,n-m}|$, then for any $n' > n$, we have $|\mathcal{R}^{*}_{n',n'-m}| = |\mathcal{R}^{*}_{n,n-m}|$ and, more precisely,*

$$\mathcal{R}^{*}_{n',n'-m} = \{\textit{type-1 extension of all functions in } \mathcal{R}^{*}_{n'-1,n'-m-1}\}\,.$$

Proof. If there is an n such that $|\mathcal{R}^{*}_{n+1,n-m+1}| = |\mathcal{R}^{*}_{n,n-m}|$, it means that all the n-valriable representatives are single-solution representatives. Therefore, based on Lemma 7, all the representatives in $\mathcal{R}^{*}_{n+2,t+2}$ are the type-1 extension of each representative in $\mathcal{R}^{*}_{n+1,t+1}$. □

This means that to classify $\mathcal{R}^{*}_{n,n-m}$ for all values of $n \geq m$, we only need to continue our algorithm until the step of computing $\mathcal{R}^{*}_{n+1,n-m+1}$ such that $\mathcal{R}^{\dagger}_{n+1,n-m+1} = \emptyset$, or equivalently $|\mathcal{R}^{*}_{n+1,n-m+1}| = |\mathcal{R}^{*}_{n,n-m}|$.

Results for the Case of $R^{*}_{n,n-4}$

We apply our algorithm to classify all n-variable $(n-4)$-resilient functions (up to extended variable-permutation equivalence). We begin by using all 3-variable representative functions, denoted by $\mathcal{R}^{*}_{3,-1}$, to build $\mathcal{R}^{*}_{4,0}$ and repeat for another 7 steps until we reach the step of building $\mathcal{R}^{*}_{11,7}$ using $\mathcal{R}^{*}_{10,6}$. The algorithm stops at this step by reaching to $\mathcal{R}^{\dagger}_{11,7} = \emptyset$.

The number of representatives in each $\mathcal{R}^{*}_{n,n-m}$ with $4 \leq n \leq 11$ is summarized in page 12. We recall that $\mathcal{R}^{*}_{n,t}$, $\mathcal{R}^{\ddagger}_{n,t}$ and $\mathcal{R}^{\dagger}_{n,t}$ denote the set of all, not-single-solution, and not-type-1 extension n-variable t-resilient representatives, respectively.

The computational complexity of this search is summarized in Table 2, by separately reporting the cost for building $(n+1)$-variable $(t+1)$-resilient functions using Siegenthaler's construction and the cost of checking if the constructed functions are representatives. The first part's cost is about 2^{25} times a few look-up tables, and for the second part, it is about 2^{31} times of partially computing the truth-table of an equivalent function. One can apply complicated methods to check for representativeness of a function to further reduce the computational cost of the algorithm. However, since the total complexity of the algorithm falls within the range of computing it in less than an hour using single-thread computation in a typical PC or laptop, we leave it as is for now.

Results for $R^{*}_{5,0}$ and $R^{*}_{6,1}$

We have also applied our algorithm to classify all n-variable $(n-5)$-resilient functions (up to extended variable-permutation equivalence). To do so, we begin by using all 4-variable representative functions to construct $\mathcal{R}^{*}_{5,0}$, and then repeat this process to compute $\mathcal{R}^{*}_{6,1}$.

However, due to the sheer size of $\mathcal{R}^{*}_{6,1}$ which contains approximately 2^{30} representatives, we estimate that it will be impossible to save all representatives in $\mathcal{R}^{*}_{7,2}$. The exact number of representatives in $\mathcal{R}^{*}5,0$ and $\mathcal{R}^{*}_{6,1}$ are 86 603 and 1 035 596 784 respectively.

Table 1: Number of $(n-4)$-resilient n-variable representatives. The second part of the table shows number of representatives in $\mathcal{R}^{\dagger}_{n,t}$ for each algebraic degree and Walsh transform spectrum. By (x, y, z) for the Walsh transform spectrum, we mean x times appearance of 2^{n-2}, y times 2^{n-1} and z times $3 \cdot 2^{n-2}$ in absolute values of the Walsh transform.

n		3	4	5	6	7	8	9	10
$\lvert\mathcal{R}^{*}_{n,n-4}\rvert$		14	58	256	578	720	754	760	761
$\lvert\mathcal{R}^{\ddagger}_{n,n-4}\rvert$		14	53	240	509	416	114	19	3
$\lvert\mathcal{R}^{\dagger}_{n,n-4}\rvert$		10	44	198	322	142	34	6	1
cubic	$(16, 0, 0)$	–	–	25	199	112	31	6	1
cubic	$(12, 1, 0)$	–	–	106	85	21	2	–	–
cubic	$(\ 8, 2, 0)$	–	28	58	35	9	1	–	–
cubic	$(\ 7, 0, 1)$	4	9	4	1	–	–	–	–
quadratic	$(\ 0, 4, 0)$	5	7	5	2	–	–	–	–

Table 2: Number of representative pairs remaining after applying each technique for step of computing $\mathcal{R}^{*}_{n+1,t+1}$ together with the computational complexity of each step. N_0, N_1, N_2 and N_3 denote the number representative pairs before technique 1, after technique 1, after technique 2, and after technique 3, respectively. Besides, C_1 denotes the cost for building $(n+1)$-variable $(t+1)$-resilient functions using the Siegenthaler's construction and C_2 denotes the cost for checking if the constructed functions are representatives.

n	4	5	6	7	8	9	10	11
N_0	105	1 711	32 896	167 331	259 560	284 635	289 180	289 941
N_1	90	1 429	26 385	89 855	43 874	8 009	773	62
N_2	90	1 266	24 356	79 631	28 450	1 919	61	3
N_3	23	133	1 911	6 423	1 779	149	8	1
C_1	138	3 192	229 320	4 624 560	8 966 160	6 007 680	2 903 040	3 628 800
C_2	16 896	760 320	14 837 760	91 607 040	350 945 280	1 114 767 360	negligible	-

All the results of this paper are publicly available at the following link:

https://gitlab.science.ru.nl/shahramr/ResilientFunctions.git

Acknowledgments

This work described in this paper is supported by the Netherlands Organization for Scientific Research (NWO) under TOP grant TOP1.18.002 SCALAR.

References

[CC05] Claude Carlet and Pascale Charpin. Cubic boolean functions with highest resiliency. *IEEE Trans. Inf. Theory*, 51(2):562–571, 2005.

[CCCS91] Paul Camion, Claude Carlet, Pascale Charpin, and Nicolas Sendrier. On correlation-immune functions. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 86–100. Springer, 1991.

[CGH$^+$85] Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem of t-resilient functions (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 396–407. IEEE Computer Society, 1985.

[LP07] Gregor Leander and Axel Poschmann. On the classification of 4 bit s-boxes. In Claude Carlet and Berk Sunar, editors, *Arithmetic of Finite Fields, First International Workshop, WAIFI 2007, Madrid, Spain, June 21-22, 2007, Proceedings*, volume 4547 of *Lecture Notes in Computer Science*, pages 159–176. Springer, 2007.

[Sie84] Thomas Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Trans. Inf. Theory*, 30(5):776–780, 1984.

[TK00] Yuriy Tarannikov and Denis Kirienko. Spectral analysis of high order correlation immune functions. *IACR Cryptol. ePrint Arch.*, page 50, 2000.