Constructions of Efficiently Implementable Boolean Functions with Provable Nonlinearity/Resiliency/Algebraic Immunity Trade-Offs

Palash Sarkar

Indian Statistical Institute, 203, B.T. Road, Kolkata, India 700108
Email: palash@isical.ac.in

November 11, 2026

Abstract

We describe new families of efficiently implementable Boolean functions achieving provable trade-offs between resiliency, nonlinearity, and algebraic immunity. Compared to one direction of research on balanced and 1-resilient functions with optimal algebraic immunity, the new functions provide lower algebraic immunity, but much lower – in fact, almost optimal – linear bias and importantly are much more efficient to implement. Compared to another direction of research based on constructions using direct sum, the new functions while achieving similar efficiency, has less number of variables for the same order of resiliency, same lower bound on algebraic immunity, and similar nonlinearity.
Keywords: Boolean function, resiliency, nonlinearity, algebraic immunity, efficient implementation.

1 Introduction

Boolean functions have widespread applications in various areas of computer science and engineering. For cryptographic applications, especially in the construction of stream ciphers, certain properties of Boolean functions have been identified as necessary for providing resistance to known attacks. Three such extensively studied properties are resiliency, nonlinearity and algebraic immunity. Over the last few decades, extensive research has been carried out on various aspects of Boolean functions possessing one or more of these three properties. We refer to the excellent book [5] for a very comprehensive and unified treatment of cryptographic properties of Boolean functions.

It is easy to construct functions which maximise any one of the three properties of resiliency, nonlinearity and algebraic immunity. For an n-variable function, the maximum possible order of resiliency is $n-1$ and the only functions which achieve this order of resiliency are the two affine functions which are non-degenerate on all the n variables. For even n, the maximum possible nonlinearity is achieved by bent functions [23] and there are many well known constructions of bent functions. The maximum possible algebraic immunity [12] of an n-variable function is $\lceil n/2 \rceil$, and the majority function achieves this value of algebraic immunity [13]. Functions that maximise

one property usually perform poorly with respect to the other two. Affine functions have minimum nonlinearity and algebraic immunity, bent functions are not balanced (i.e. not 0-resilient), while the majority function has poor nonlinearity and resiliency.

While resiliency, nonlinearity and algebraic immunity are security properties of Boolean functions, there is another aspect of Boolean functions which is also of great practical importance. For use in actual design of cryptographic systems, it is crucial that any Boolean function used in the design is efficient to implement. A measure of implementation efficiency is the number of 2-input gates that is required to implement a Boolean function. From a cryptographic point of view, along with good security properties, a Boolean function also needs to have a low gate count.

Previous works have proceeded in two directions. The first direction is focused on constructing functions which achieve optimal algebraic immunity, while at the same time achieving guarantees on the nonlinearity of the constructed functions. For balanced functions, this line of work has been reported in [6, 33, 4, 28, 26, 34, 17], and for 1-resilient functions on an even number of variables, this line of work has been reported in [32, 35, 29, 36, 30]. These works do not consider efficiency of implementation to be an issue and the constructed functions essentially require exponential (in the number of variables) size circuits to implement. Also, to the best of our knowledge, work in this direction has not considered order of resiliency greater than 1. More generally, the motivation for this line of research is essentially theoretical, and progress is difficult and slow.

The second independent direction of work [21, 20, 8, 15] has proposed functions which are constructed as a direct sum of different kinds of functions, In particular, the construction in [15] (which is also in the long version of [20]) is the direct sum of a linear function, a bent function, and a threshold function, which provides separate control over the three parameters of resiliency, nonlinearity, and algebraic immunity. The functions constructed by the methods in [21, 20, 8, 15] are efficient to implement. These functions are targeted towards the application of homomorphic encryption.

In this paper, we contribute to the second direction of research. A recent work [10] proposed the construction of efficiently implementable bent functions with provable lower bound on algebraic immunity. We leverage this result to provide constructions of functions which are efficient to implement and provide provable trade-offs between resiliency, nonlinearity, and algebraic immunity.

Two of our constructions are basic. The first construction increases resiliency by simply adding new variables, which provides a separate control over resiliency to the bent functions from [10]. For the resulting functions, the number of variables and the order of resiliency do not have the same parity. A second construction of a similar type shows how to increase resiliency by first adding a 5-variable, 1-resilient function to the bent function from [10], and then adding an appropriate number of additional variables. For the second construction, the number of variables and the order of resiliency have the same parity. Both the constructions provide functions which are essentially as efficient to implement as the bent functions from [10]. We provide a detailed comparison of the first construction to the approach in [20, 15] where the threshold function of [20, 15] is instantiated with the majority function. We also show that for balanced functions, compared to [6, 33, 4, 28, 26, 34, 17], the new functions provide same algebraic immunity, much lower linear bias, and are more efficient to implement even though the new functions require more variables. A similar comparison holds for the approach to construct 1-resilient functions taken in [32, 35, 29, 36, 30].

Both the above constructions achieve resiliency by essentially adding variables. This creates

simple linear structures, which may be considered undesirable. We provide two iterated constructions by incorporating the bent function constructed in [10] into a two-decade old construction idea sketched in [22]. The resulting constructions provide functions which are also efficient to implement, and which provide resiliency/nonlinearity/algebraic immunity trade-offs that are different from the first two constructions. Compared to the basic constructions, the iterated constructions yield functions, which for the same order of resiliency, and the same lower bound on algebraic immunity, provide lower linear bias at the cost of a few extra variables and 2-input gates.

From the viewpoint of theoretical computer science, the properties of algebraic immunity and resiliency were shown [1] to be key parameters of the "local function" required in Goldreich's construction [16] of pseudorandom generators. Motivated by [1], a recent study [14] dealt quite extensively with the provable trade-off between resiliency and algebraic immunity. This line of work does not consider nonlinearity of the functions. Since we consider nonlinearity as a parameter of the trade-off, the functions constructed in [14] are not directly comparable to our work.

Outline of the paper

The background and preliminaries are described in Section 2. In Section 3, we present the two basic constructions. The iterated constructions are presented in Section 4. The four constructions are compared in Section 5. Finally, Section 6 concludes the paper.

2 Background and Preliminary Results

In this section, we provide the basic definitions and present some preliminary results. For further and extensive details on cryptographic properties of Boolean functions we refer to [5].

The cardinality of a finite set S will be denoted by $\#S$. $\mathbb{F}_2$ denotes the finite field of two elements; $\mathbb{F}_2^n$, where n is a positive integer, denotes the vector space of dimension n over $\mathbb{F}_2$. The addition operator over both $\mathbb{F}_2$ and $\mathbb{F}_2^n$ will be denoted by $\oplus$. The product (which is also the logical AND) of $x, y \in \mathbb{F}_2$ will simply be written as xy. A bit vector of dimension n, i.e. an element of $\mathbb{F}_2^n$ will also be considered to be an n-bit binary string.

Let n be a positive integer. The support of $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$ is $\mathsf{supp}(\mathbf{x}) = \{1 \leq i \leq n : x_i = 1\}$, and the weight of $\mathbf{x}$ is $\mathsf{wt}(\mathbf{x}) = \#\mathsf{supp}(\mathbf{x})$. By $\mathbf{0}_n$ and $\mathbf{1}_n$ we will denote the all-zero and all-one strings of length n respectively. For $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, with $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ the distance between $\mathbf{x}$ and $\mathbf{y}$ is $d(\mathbf{x}, \mathbf{y}) = \#\{i : x_i \neq y_i\}$; the inner product of $\mathbf{x}$ and $\mathbf{y}$ is $\langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 \oplus \cdots \oplus x_n y_n$.

Boolean function. An n-variable Boolean function f is a map $f : \mathbb{F}_2^n \to \mathbb{F}_2$. The weight of f is $\mathsf{wt}(f) = \#\{\mathbf{x} \in \mathbb{F}_2^n : f(\mathbf{x}) = 1\}$; f is said to be *balanced* if $\mathsf{wt}(f) = 2^{n-1}$. The canonical ordering of the elements of $\mathbb{F}_2^n$ is the ordering where for $0 \leq i < 2^n$, the i-th element in the ordering is the n-bit binary representation of i. With respect to the canonical ordering an n-variable function f can be represented by a binary string of length 2^n, where the i-th bit of the string is the value of f on the i-th element of the canonical representation. We call such a string to be the *string (or truth table) representation* of f.

The *algebraic normal form (ANF) representation* of an n-variable Boolean function f is the representation of f as an element of the polynomial ring $\mathbb{F}_2[X_1,\ldots,X_n]/(X_1^2 \oplus X_1,\ldots,X_n^2 \oplus X_n)$ in the following manner: $f(X_1,\ldots,X_n) = \bigoplus_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} a_{\boldsymbol{\alpha}}\mathbf{X}^{\boldsymbol{\alpha}}$, where $\mathbf{X} = (X_1,\ldots,X_n)$; for $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_n) \in \mathbb{F}_2^n$, $\mathbf{X}^{\boldsymbol{\alpha}}$ denotes the monomial $X_1^{\alpha_1}\cdots X_n^{\alpha_n}$; and $a_{\boldsymbol{\alpha}} \in \mathbb{F}_2$. The (algebraic) degree of f is $\mathsf{deg}(f) = \max\{\mathsf{wt}(\boldsymbol{\alpha}) : a_{\boldsymbol{\alpha}} = 1\}$; we adopt the convention that the zero function has degree 0. The degree (or sometimes also called the length) of the monomial $\mathbf{X}^{\boldsymbol{\alpha}}$ is $\mathsf{wt}(\boldsymbol{\alpha})$.

In the following, we will define methods to combine two or more Boolean functions to construct a new Boolean function. Some of these methods will be the focus of our study and we will call such methods to be constructions, while the others are intermediate methods and we will simply refer to them as methods.

It is useful to introduce a notation for the concatenation of two functions. Let g and h be n-variable functions. Define an $(n+1)$-variable function in the following manner.

$$f(X_1,\ldots,X_{n+1}) \;=\; (1 \oplus X_{n+1})g(X_1,\ldots,X_n) \oplus X_{n+1}h(X_1,\ldots,X_n).$$

We denote f as $\mathsf{Concat}(g,h)$. Note that the string representation of $f = \mathsf{Concat}(g,h)$ is obtained by concatenating the string representations of g and h.

An n-variable function $f(X_1,\ldots,X_n)$ is said to be *non-degenerate* on the variable X_i, $1 \le i \le n$, if there are $\boldsymbol{\alpha},\boldsymbol{\beta} \in \mathbb{F}_2^n$ which differ only in the i-th position and $f(\boldsymbol{\alpha}) \neq f(\boldsymbol{\beta})$; if there are no such $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, then f is said to be degenerate on the variable X_i.

Functions of degree at most 1 are said to be affine functions. Affine functions with $a_{\mathbf{0}_n} = 0$ are said to be linear functions. Each $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_n) \in \mathbb{F}_2^n$, defines the linear function $\langle\boldsymbol{\alpha},\mathbf{X}\rangle = \langle\boldsymbol{\alpha},(X_1,\ldots,X_n)\rangle = \alpha_1X_1 \oplus \cdots \oplus \alpha_nX_n$. If $\mathsf{wt}(\boldsymbol{\alpha}) = w$, then the function $\langle\boldsymbol{\alpha},\mathbf{X}\rangle$ is non-degenerate on exactly w of the n variables.

For $\mathbf{X} = (X_1,\ldots,X_n)$, by $L_n(\mathbf{X})$ we will denote the following function.

$$L_n(\mathbf{X}) \;=\; X_1 \oplus \cdots X_n. \tag{1}$$

Nonlinearity and Walsh transform. The distance between two n-variable functions f and g is $d(f,g) = \#\{\mathbf{x} \in \mathbb{F}_2^n : f(\mathbf{x}) \neq g(\mathbf{x})\} = \mathsf{wt}(f \oplus g)$. The *nonlinearity* of an n-variable function f is defined to be $\mathsf{nl}(f) = \min_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\{d(f,\langle\boldsymbol{\alpha},\mathbf{X}\rangle), d(f,1\oplus\langle\boldsymbol{\alpha},\mathbf{X}\rangle)\}$, i.e. the nonlinearity of f is the minimum of the distances of f to all the affine functions.

The Walsh transform of an n-variable function f is the map $W_f : \mathbb{F}_2^n \to \mathbb{Z}$, where for $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, $W_f(\boldsymbol{\alpha}) = \sum_{\mathbf{x}\in\mathbb{F}_2^n}(-1)^{f(\mathbf{x})\oplus\langle\boldsymbol{\alpha},\mathbf{x}\rangle}$. From the definition it follows that $W_f(\boldsymbol{\alpha}) = 2^n - 2d(f,\langle\boldsymbol{\alpha},\mathbf{X}\rangle)$. Consequently, $\mathsf{nl}(f) = 2^{n-1} - \frac{1}{2}\max_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}|W_f(\boldsymbol{\alpha})|$. The nonlinearity of f is invariant under an invertible linear transformation on the variables of f.

Linear bias. We define the *linear bias* of an n-variable function f to be $\mathsf{LB}(f) = \max_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}|W_f(\boldsymbol{\alpha})|/2^n = 1 - \mathsf{nl}(f)/2^{n-1}$. From a cryptographic point of view, the linear bias, rather than the nonlinearity, is of importance, since it is the linear bias which is used to quantify the resistance to (fast) correlation attacks.

Bent functions. An n-variable function f is said to be bent [23] if $W_f(\boldsymbol{\alpha}) = \pm 2^{n/2}$ for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n$. From the definition it follows that bent functions can exist only if n is even. An n-variable bent

function has nonlinearity $2^{n-1} - 2^{n/2-1}$ (resp. linear bias $2^{-n/2}$), and this is the maximum possible nonlinearity (resp. least possible linear bias) that can be achieved by any n-variable function.

Maiorana-McFarland bent functions. The well known Maiorana-McFarland class of bent functions is defined as follows. For $k \geq 1$, let $\psi : \{0,1\}^k \to \{0,1\}^k$ be a bijection and $h : \{0,1\}^k \to \{0,1\}$ be a Boolean function. Let $\mathbf{X} = (X_1, \ldots, X_k)$ and $\mathbf{Y} = (Y_1, \ldots, Y_k)$. For $k \geq 1$, (ψ, h)-MM_{2k} is defined to be the following function.

$$(\psi, h)\text{-}\mathsf{MM}_{2k}(\mathbf{X}, \mathbf{Y}) \quad = \quad \langle \psi(\mathbf{X}), \mathbf{Y} \rangle \oplus h(\mathbf{X}). \tag{2}$$

Note that the degree of (ψ, h)-MM_{2k} is $\max(1 + \max_{1 \leq i \leq k} \mathsf{deg}(\psi_i), \mathsf{deg}(h))$, where $\psi_1, \ldots, \psi_k$ are the coordinate functions of ψ.

By $Q_{2k}(\mathbf{X}, \mathbf{Y})$ we will denote (ψ, h)-MM_{2k} with ψ to be the identity permutation, and h to be the constant all-zero function, i.e.

$$Q_{2k}(\mathbf{X}, \mathbf{Y}) \quad = \quad X_1 Y_1 \oplus \cdots X_k Y_k. \tag{3}$$

Almost optimal linear bias. For a positive integer n, the covering radius bound (see Theorem 3 in Chapter 3 of [5]) on the nonlinearity of an n-variable function f is the following: $\mathsf{nl}(f) \leq 2^{n-1} - 2^{n/2-1}$, and equivalently, $\mathsf{LB}(f) \geq 2^{-n/2}$. The bound is achieved if and only if f is bent. We say that f has *almost optimal linear bias* if $2^{-n/2} \leq \mathsf{LB}(f) \leq 2^{-n/2+1}$, i.e. if the linear bias of f is at most two times the lower bound arising from the covering radius bound. If n is even, and $\mathsf{nl}(f) = 2^{n-1} - 2^{n/2}$, then $\mathsf{LB}(f) = 2^{-n/2+1}$, while if n is odd, and $\mathsf{nl}(f) = 2^{n-1} - 2^{(n-1)/2}$, then $\mathsf{LB}(f) = 2^{-n/2+1/2} < 2^{-n/2+1}$; in both cases the linear bias is almost optimal.

Resilient functions. Let n be a positive integer and m be an integer such that $0 \leq m < n$. An n-variable function f is said to be m-resilient if $W_f(\boldsymbol{\alpha}) = 0$ for all $\boldsymbol{\alpha}$ satisfying $\mathsf{wt}(\boldsymbol{\alpha}) \leq m$. Equivalently, f is m-resilient if and only if $d(f, \langle \boldsymbol{\alpha}, \mathbf{X} \rangle) = 2^{n-1}$ for $\boldsymbol{\alpha}$ satisfying $\mathsf{wt}(\boldsymbol{\alpha}) \leq m$, i.e. if the distance between f and any linear function which is non-degenerate on at most m variables is equal to 2^{n-1}. Siegenthaler's bound [27] relates n, m and the degree d of f in the following manner:

$$\text{if } m = n-1, \text{ then } d = 1, \text{ and if } m \leq n-2, \text{ then } d \leq n - m - 1. \tag{4}$$

Divisibility results obtained in [25, 3, 9] show that the Walsh transform values of an n-variable, m-resilient function f having degree d is divisible by $2^{m+2+\lfloor (n-m-2)/d \rfloor}$.

Algebraic immunity. The algebraic immunity of an n-variable function f is defined [12] as follows: $\mathsf{AI}(f) = \min_{g \neq 0} \{\mathsf{deg}(g) : \text{ either } gf = 0, \text{ or } g(f \oplus 1) = 0\}$. It is known [12] that $\mathsf{AI}(f) \leq \lceil n/2 \rceil$. (The result follows from a restatement of Theorem 6.0.1 of [12] in the extended version of [12] available at `http://ntcourtois.free.fr/toyolili.pdf`.) If the bound is achieved, then we say that f has optimal algebraic immunity.

Let $\mathsf{AN}(f) = \min_{g \neq 0} \{\mathsf{deg}(g) : fg = 0\}$, and $\Delta_{\mathsf{AN}}(f) = \mid \mathsf{AN}(f) - \mathsf{AN}(f \oplus 1) \mid$. Note that $\mathsf{AI}(f) = \min\{\mathsf{AN}(f), \mathsf{AN}(f+1)\}$.

Fast algebraic immunity. Given an n-variable function f, suppose that there are n-variable functions $g \neq 0$ and h of degrees e and d respectively such that $gf = h$. If $e + d \geq n$, then the existence of g and h satisfying $gf = h$ is guaranteed [11]. The fast algebraic immunity (FAI) of f is defined in the following manner: $\mathsf{FAI}(f) = \min(2\mathsf{AI}(f), \min_{g \neq 0}\{\mathsf{deg}(g) + \mathsf{deg}(fg) : 1 \leq \mathsf{deg}(g) < \mathsf{AI}(f)\})$. The following bounds hold for $\mathsf{FAI}(f)$: $1 + \mathsf{AI}(f) \leq \mathsf{FAI}(f) \leq 2\,\mathsf{AI}(f)$. In particular, the lower bound of $1 + \mathsf{AI}(f)$ on $\mathsf{FAI}(f)$ suggests that a target lower bound (which is not necessarily optimal) on $\mathsf{FAI}(f)$ may be achieved by designing functions with a desired value of algebraic immunity.

Majority function. For $n \geq 1$, let $\mathsf{Maj}_n : \{0,1\}^n \to \{0,1\}$ be the majority function defined in the following manner. For $\mathbf{x} \in \{0,1\}^n$, $\mathsf{Maj}(\mathbf{x}) = 1$ if and only if $\mathsf{wt}(\mathbf{x}) > \lfloor n/2 \rfloor$.

Theorem 1 (From Theorems 1, 2, and 3 of [13]) *Let n be a positive integer.*

1. Maj_n *has the maximum possible AI of* $\lceil n/2 \rceil$.
2. *The degree of* Maj_n *is equal to* $2^{\lfloor \log_2 n \rfloor}$.
3. $\mathsf{nl}(\mathsf{Maj}_n) = 2^{n-1} - \binom{n-1}{\lfloor n/2 \rfloor}$, *equivalently* $\mathsf{LB}(\mathsf{Maj}_n) = \binom{n-1}{\lfloor n/2 \rfloor}/2^{n-1}$.

Proposition 1 (Theorem 4.1 of [37]) Maj_n *can be implemented using $O(n)$ NAND gates.*

By $G(n)$ we will denote the number of NAND gates required to implement Maj_n. For some concrete values of n, estimates of $G(n)$ were obtained in [10, Section V]. The strategy used was the following. A circuit to implement Maj_n has two parts, one part to compute the weight of an n-bit string, and the other part to determine whether the weight is more than $\lfloor n/2 \rfloor$. The second part requires $O(\log_2 n)$ gates, and $G(n)$ is determined mainly by the size of the circuit to compute the weight of an n-bit string. Using the algorithm in [2] for finding the weight of a binary string, it was shown in [10, Section V] that the circuit to compute the weight of an n-bit string can be implemented using full and half adders. It turns out that the total of number of full and half adders is less than n; for example, it was shown that the numbers of full and half adders for $n = 37$ are 31 and 3 respectively – see [10, Section V] for other examples. A full adder requires 9 NAND gates, while a half adder requires 5 NAND gates. So overall $G(n)$ is at most about $10n$ NAND gates.

Threshold function. For $n \geq 1$, and $1 \leq d \leq n$, let $\mathsf{Th}_{n,d} : \{0,1\}^n \to \{0,1\}$ be the threshold function defined in the following manner. For $\mathbf{x} \in \{0,1\}^n$, $\mathsf{Th}_{n,d}(\mathbf{x}) = 0$ if and only if $\mathsf{wt}(\mathbf{x}) < d$. Note that $\mathsf{Maj}_n = \mathsf{Th}_{n,\lceil n/2 \rceil}$. The following result provides the algebraic immunity of $\mathsf{Th}_{n,d}$ (see also [18, Lemma 4]).

Proposition 2 (Lemma 6 of [7]) *Let $n > 1$ and $1 \leq d \leq n$ be positive integers. Then* $\mathsf{AN}(\mathsf{Th}_{n,d}) = n - d + 1$, $\mathsf{AN}(1 \oplus \mathsf{Th}_{n,d}) = d$. *Hence,* $\mathsf{AI}(\mathsf{Th}_{n,d})$, *and* $\Delta_{\mathsf{AN}}(\mathsf{Th}_{n,d}) = \mid n + 1 - 2d \mid$.

Direct sum. A simple way to construct a function is to add together two functions on disjoint sets of variables. The constructed function is called the direct sum of the two smaller functions. Let n_1 and n_2 be positive integers and g and h be functions on n_1 and n_2 variables respectively. Define

$$f(X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2}) \;=\; g(X_1, \ldots, X_{n_1}) \oplus h(Y_1, \ldots, Y_{n_2}). \tag{5}$$

Basic properties of a function constructed as a direct sum is given in the following result.

Proposition 3 (Proposition 1 of [24]) *Let f be constructed as in* (5)*. Then*

1. $\mathsf{deg}(f) = \max\{\mathsf{deg}(g), \mathsf{deg}(h)\}$.
2. $\mathsf{nl}(f) = 2^{n_1}\mathsf{nl}(h) + 2^{n_2}\mathsf{nl}(g) - 2\mathsf{nl}(g)\mathsf{nl}(h)$. *Equivalently,* $\mathsf{LB}(f) = \mathsf{LB}(g) \cdot \mathsf{LB}(h)$.
3. *f is balanced if and only if at least one of g or h is balanced.*
4. *If g is m_1-resilient and h is m_2-resilient, then f is m-resilient, where $m = \max\{m_1, m_2\}$.*

A special case of direct sum is the following simple construction: $f(W, \mathbf{X})$ is defined to be $f(W, \mathbf{X}) = W \oplus g(\mathbf{X})$. The following properties of such an f can be proved directly, and also follows as a consequence of Proposition 3: $\mathsf{nl}(f) = 2 \cdot \mathsf{nl}(g)$ and $\mathsf{LB}(f) = \mathsf{LB}(g)$; f is balanced, and if g is m-resilient, then f is $(m+1)$-resilient.

Upper and lower bounds on the algebraic immunity of a function constructed as a direct sum are given by the following result.

Proposition 4 (Lemma 3 of [21]) *For f constructed as in* (5)*,* $\max\{\mathsf{AI}(g), \mathsf{AI}(h)\} \leq \mathsf{AI}(f) \leq \mathsf{AI}(g) + \mathsf{AI}(h)$.

A sharper lower bound on the algebraic immunity of direct sum is known.

Proposition 5 (Corollary 3 of [18]) *For f constructed as in* (5)*, suppose* $\mathsf{AI}(g) \geq \mathsf{AI}(h)$*, then* $\mathsf{AI}(g) + \min\{\Delta_{\mathsf{AN}}(g), \mathsf{AI}(h)\} \leq \mathsf{AI}(f)$.

If h is a non-constant function, and $\Delta_{\mathsf{AN}}(g)$ is positive, then the lower bound in Proposition 5 is greater than the lower bound in Proposition 4.

Tunable direct sum. The direct sum of different functions has been considered in a number of previous works [21, 20, 8, 15]. Direct sum of monomials was considered in [21, 20], the direct sum of linear, quadratic, and triangular functions was considered in [21], the direct sum of linear and threshold functions was considered in [8], and the direct sum of linear, quadratic, and threshold functions was considered in [15] (and also in the longer version of [20]). Based on [15] we define the following construction.

Construction 1 *Let $m \geq 0$, $k, s \geq 1$, and $1 \leq d \leq s$ be integers. For $\mathbf{W} = (W_1, \ldots, W_m)$, $\mathbf{X} = (X_1, \ldots, X_k)$, $\mathbf{Y} = (Y_1, \ldots, Y_k)$, and $\mathbf{Z} = (Z_1, \ldots, Z_s)$, define a Boolean function f on $m + 2k + s$ variables as follows.*

$$f(\mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}) \quad = \quad L_m(\mathbf{W}) \oplus Q_{2k}(\mathbf{X}, \mathbf{Y}) \oplus \mathsf{Th}_{s,d}(\mathbf{Z}). \tag{6}$$

We state the following result in the case where s is odd and $\mathsf{Th}_{s,d} = \mathsf{Maj}_s$. Note that for odd s, $\mathsf{Maj}_s(\mathbf{Z})$ is balanced. From Theorem 1, and Propositions 3 and 4, we have the following result.

Proposition 6 *In Construction 1, let $k > 1$ and s be odd and suppose Maj_s is used to instantiate $\mathsf{Th}_{s,d}$ in Construction 1, and let f be the function that is obtained. Then we have the following.*

1. *f is m-resilient.*
2. $\mathsf{nl}(f) = 2^m\left(2^{2k}\left(2^{s-1} - \binom{s-1}{\lfloor s/2 \rfloor}\right) + 2^s(2^{2k-1} - 2^{k-1}) - 2(2^{2k-1} - 2^{k-1})\left(2^{s-1} - \binom{s-1}{\lfloor s/2 \rfloor}\right)\right)$*, and hence* $\mathsf{LB}(f) = 2^{-k}\binom{s-1}{(s-1)/2}/2^{s-1}$.

3. *f has algebraic immunity to be at least $\lceil s/2 \rceil$.*

4. *f can be implemented using k AND, $k+m$ XOR, and $G(s)$ NAND gates.*

Applying Stirling's approximation $\binom{2j}{j} \approx 4^j/\sqrt{\pi j}$, we obtain

$$| \log_2 \mathsf{LB}(f) | \;\approx\; k + \tfrac{1}{2}\log_2\big(\pi(s-1)/2\big). \tag{7}$$

From (7) one may note that the linear bias of any function obtained from Construction 7 is essentially determined by k and the effect of s is only logarithmic.

Remark 1 *If instead of Maj_s, Construction 1 is instantiated with a more general $\mathsf{Th}_{s,d}$, then the parameters guaranteed in Proposition 6 will change. For $k > 0$, using Propositions 2 and 5, the algebraic immunity of $Q_{2k} \oplus \mathsf{Th}_{s,d}$ (and hence the algebraic immunity of f) is at least $1 + \lceil s/2 \rceil$ if $d = \lfloor s/2 \rfloor$ (where we need to use the fact that the algebraic immunity of Q_{2k} is 2). The resulting $Q_{2k} \oplus \mathsf{Th}_{s,d}$, on the other hand, is not balanced, and so the resiliency of f is $m-1$. Also, the expression for the linear bias will change. Such changes in parameters arising from using a more general $\mathsf{Th}_{s,d}$ instead of Maj_s in Construction 1 will not affect the general nature of the comparison between Construction 1 and the constructions that we propose later.*

Maiorana-McFarland with Majority. The following lower bound on the algebraic immunity of a special class of Maiorana-McFarland bent functions was obtained in [10].

Theorem 2 (Theorem 2 of [10]) *Let $k \geq 2$, $n = 2k$, and $\psi : \{0,1\}^k \to \{0,1\}^k$ be an affine map, i.e. each of the coordinate functions of ψ is an affine function of the input variables. Then*

$$\mathsf{AI}((\psi, h)\text{-}\mathsf{MM}_{2k}) \;\geq\; \mathsf{AI}(h). \tag{8}$$

Consequently, $\mathsf{AI}((\psi, \mathsf{Maj})\text{-}\mathsf{MM}_{2k}) \geq \mathsf{AI}(\mathsf{Maj}_k) = \lceil k/2 \rceil$.

Theorem 2 holds for any affine map ψ. From efficiency considerations, we will consider ψ to be a bit permutation, i.e. there is a permutation ρ of $\{1, \ldots, k\}$ such that for any $(x_1, \ldots, x_k) \in \mathbb{F}_2^n$, $\psi(x_1, \ldots, x_k) = (x_{\rho(1)}, \ldots, x_{\rho(k)})$. Implementation of a bit permutation requires only the proper connection pattern, and does not require any gates. For the case where ψ is a bit permutation, it was shown in [19] that the algebraic immunity of $(\psi, \mathsf{Maj})\text{-}\mathsf{MM}_{2k}$ is at most $1 + 2^{\lceil \log(n/4) \rceil}$.

Gate count. From an implementation point of view, it is of interest to obtain functions which are efficient to implement. A measure of implementation efficiency is the circuit size which is given by the number of 2-input gates required to implement an n-variable function f. The truth table representation of f requires a circuit size of $\Omega(2^n)$. For even moderate values of n, such a representation results in a very large circuit. In this paper, we consider the following 2-input gates: XOR, AND, and NAND. The size of the circuit to implement the majority function is considered in terms of NAND gates. For the other components we use only XOR and AND gates.

We note that the circuit size is only one measure of efficiency. This measure is relevant for usual stream cipher applications. On the other hand, for applications to homomorphic encryption, a relevant measure is the multiplicative depth. The works [21, 20, 8, 15] are focused towards homomorphic encryption.

Remark 2 *In the later sections, we describe constructions which are based on* (ψ, Maj)-MM_{2k}. *All of these constructions can also be based on* $(\psi, \mathsf{Th}_{k,d})$-$\mathsf{MM}_{2k}$ *for some suitable more homomorphic friendly choice of* $\mathsf{Th}_{k,d}$.

3 Basic Constructions of m-Resilient Functions

In the present section, we provide two basic constructions of m-resilient functions with guarantees on linear bias and algebraic immunities.

3.1 Case $n \not\equiv m \bmod 2$

The following construction builds on the basic fact that adding new variables increases the order of resiliency.

Construction 2 *Let* m *be a non-negative integer, and* $n > m$ *be another integer such that* $n \not\equiv m \bmod 2$. *Let* $k = (n-m-1)/2$. *Let* $\psi : \{0,1\}^k \to \{0,1\}^k$ *be a bit permutation. The* n*-variable function* f *is defined as follows.*

$$\begin{aligned} & f(X_1, \ldots, X_{m+1}, U_1, \ldots, U_k, V_1, \ldots, V_k) \\ = \;& X_1 \oplus \cdots \oplus X_{m+1} \oplus (\psi, h)\text{-}\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k). \end{aligned} \tag{9}$$

The properties of the function obtained from Construction 2 are stated in the following result.

Proposition 7 *Let* f *be the* n*-variable function obtained as in Construction 2. Then* f *is an* m*-resilient function with linear bias equal to* $2^{-(n-m-1)/2}$. *Further, if* $h = \mathsf{Maj}_k$, *then the algebraic immunity of* f *is at least* $\lceil (n-m-1)/4 \rceil$, *and* f *can be implemented using* k *AND,* $k+m+1$ *XOR, and* $G(k)$ *NAND gates.*

Proof: Since $m+1$ new variables are added to a bent function, the order of resiliency is m. The linear bias of the bent function on $2k$ variables is 2^{-k}, where $k = (n-m-1)/2$, and since the new variables are simply added, the linear bias remains unchanged. From Proposition 4 the algebraic immunity of f is at least the algebraic immunity of the bent function; from Theorem 2 the algebraic immunity of the bent function is at least the algebraic immunity of $h = \mathsf{Maj}_k$; and from Theorem 1 the algebraic immunity of Maj_k is equal to $\lceil k/2 \rceil$. Maj_k requires $G(k)$ NAND gates. The counts of the other gates follow easily from Construction 2. □

Comparison to Proposition 6. Both Propositions 6 and 7 increase resiliency by adding variables. Further, both Propositions 6 and 7 rely on the majority function to achieve a guaranteed lower bound on algebraic immunity. So for the purpose of comparison, we assume that the order of resiliency, and the lower bound on the algebraic immunity is the same for both the cases.

Let a_0 be the target lower bound on the algebraic immunity. Then in Proposition 6 we have $s = 2a_0 - 1$, and in Proposition 7 we have $k = 2a_0 - 1$. In Table 1 we compare the parameters of Propositions 6 and 7. The column labelled 'difference' provides the difference between the parameters in Proposition 6 and 7.

Fixing a_0 and m fixes all the parameters in Proposition 7, while in Proposition 6, the parameter k is still free. As k increases, for Proposition 6 both the number of variables and the circuit size increase, while the linear bias decreases (equivalently, $|\log_2 \mathsf{LB}|$ increases). The comparison between Propositions 6 and 7 is based on k and is summarised as follows.

1. For $k < a_0$, Proposition 6 has both smaller number of variables, and smaller circuit size, but the linear bias is much larger; in Proposition 6 it is 2^{-k-c}, while in Proposition 7 it is 2^{-2a_0+1}.
2. For $k = a_0$, both Propositions 6 and 7 have the same number of variables; Proposition 6 has a circuit which is smaller by $2a_0 - 1$ AND and XOR gates; and the linear bias of Proposition 6 is $2^{-(a_0+c)}$, and that of Proposition 7 is 2^{-2a_0+1}. To understand this concretely, suppose $a_0 = 30$, then Proposition 7 requires 59 more AND and XOR gates than Proposition 6, but Proposition 7 has linear bias 2^{-59}, whereas Proposition 6 has linear bias $2^{-30}/\sqrt{29\pi}$. So compared to Proposition 6, a few extra gates required in Proposition 7 lead to a substantial reduction in the linear bias.
3. For $k = \lceil 2a_0 - 1 - c \rceil$, both Propositions 6 and 7 have almost the same linear bias and also almost the same circuit size. The number of variables required by Proposition 6 is about $2a_0 - 2 - 2c$ more than that of Proposition 7. Again taking $a_0 = 30$, we see that Proposition 6 requires about 50 more variables compared to Proposition 7.
4. As k increases beyond $\lceil 2a_0 - 1 - c \rceil$, in comparison to Proposition 7, the linear bias of Proposition 6 goes down further, at the cost of increase in the number of variables and the circuit size.

	Proposition 6 $s = 2a_0 - 1$, k free	Proposition 7 $k = 2a_0 - 1$	difference
n	$m + 2k + 2a_0 - 1$	$m + 4a_0 - 1$	$2k - 2a_0$
resiliency	m	m	0
$\lvert\log_2 \mathsf{LB}\rvert$	$\approx k + c$	$2a_0 - 1$	$\approx k + c - 2a_0 + 1$
AI (lower bound)	a_0	a_0	0
#AND	k	$2a_0 - 1$	$k - 2a_0 + 1$
#XOR	$k + m$	$2a_0 + m$	$k - 2a_0$
size $- G(2a_0 - 1)$	$2k + m$	$4a_0 + m - 1$	$2k - 4a_0 + 1$

Table 1: Parameters of Proposition 6 and Proposition 7. Here $c = \frac{1}{2}\log_2(\pi(a_0 - 1))$ (see (7)), and $G(2a_0 - 1)$ is the number of NAND gates required for Maj_{2a_0-1}.

To summarise, in Proposition 6 the parameter k provides a separate control. From Point 4 above, for the same order of resiliency, the same lower bound on algebraic immunity, similar linear bias and efficiency, Proposition 6 requires substantially more variables compared to Proposition 7.

Balanced functions. The special case of balanced functions was considered in [10] and is given in the following result.

Corollary 1 ([10]) *Let $n \equiv 1 \bmod 2$. It is possible to construct an n-variable, balanced function with linear bias equal to $2^{-(n-1)/2}$ (which is almost optimal), algebraic immunity at least $\lceil (n-1)/4 \rceil$, and can be implemented using $(n-1)/$ AND, $(n+1)/2$ XOR, and $G((n-1)/2)$ NAND gates.*

Proof: Putting $m = 0$ in Proposition 7 provides the result. □

Several papers [6, 33, 4, 28, 26, 34, 17] proposed constructions of n-variable balanced functions which achieve optimal algebraic immunity $\lceil n/2 \rceil$ with provable upper bounds on the linear bias. The linear bias of the n-variable function constructed in the last of this line of papers, i.e. in [17], is at most $(0.402963 + \ln 2/\pi)2^{-(n-2)/2} + \delta$, where δ is a small (though quite complicated) quantity.

For odd n, Corollary 1 provides constructions of n-variable balanced functions with almost optimal linear bias of $2^{-(n-1)/2}$ and a lower bound of $\lceil (n-1)/4 \rceil$ on algebraic immunity. The linear biases of the functions obtained from the previous constructions (both the upper bound as well as actual values for concrete functions) are higher than the linear biases of the functions obtained from Corollary 1. So Corollary 1 and the previous constructions provide different points on the nonlinearity/linear bias trade-off curve for balanced functions.

The main advantage of the functions constructed using Corollary 1 is that these functions can be constructed using $O(n)$ gates, while the functions constructed using the methods in [6, 33, 4, 28, 26, 34, 17] are essentially based on discrete logarithm computation and require super-polynomial size circuits. To see this advantage in concrete terms, we consider the case of the 28-variable function (say f_{28}) reported in Table 2 of [17] which has algebraic immunity 14 and nonlinearity 134201460 (equivalently linear bias equal to about $2^{-13.01}$). According to the description of hardware implementation in Section 4.2 of [17], implementation of f_{28} will require a look-up table of size 2^{28} along with other gates. Taking $n = 55$ in Corollary 1, we obtain a 55-variable balanced function (say f_{55}) having linear bias 2^{-27} and algebraic immunity at least 14. Further, f_{55} can be implemented using 27 AND, 28 XOR, and $G(27)$ NAND gates. In comparison to f_{28}, the function f_{55} has a much lower linear bias and is much more efficient to implement.

1-Resilient functions. The special case of 1-resilient function obtained from Proposition 7 is given in the following result.

Corollary 2 *Let $n \equiv 0 \bmod 2$. It is possible to construct an n-variable, 1-resilient function with linear bias equal to $2^{-(n-2)/2}$ (which is almost optimal), algebraic immunity at least $\lceil (n-2)/4 \rceil$, and can be implemented using $(n-2)/2$ AND, $n/2$ XOR, and $G((n-2)/2)$ NAND gates.*

Proof: Putting $m = 1$ in Proposition 7 provides the result. □

For even n, several papers [32, 35, 29, 36, 30] have proposed constructions of 1-resilient functions with optimal algebraic immunity $n/2$ and provable upper bounds on linear bias. To the best of our knowledge the last of such results is [30] which improves upon works prior to it by providing lower linear bias, and also guarantees the fast algebraic immunity to be at least $n-6$. The upper bound on the linear bias of n-variable functions constructed in [30] is $((1 + (n/2)\ln 2)/\pi + (\pi + 16)/32)2^{-n/2+1} + 2^{-(n-1)}$. The linear bias of the functions constructed in Corollary 2 is $2^{-(n-2)/2}$ which is almost optimal, the algebraic immunity is at least $\lceil (n-2)/4 \rceil$, and hence the fast algebraic immunity is at least $1 + \lceil (n-2)/4 \rceil$. So for 1-resilient functions, the functions constructed in [30]

and those in Corollary 2 represent two distinct trade-off points with respect to algebraic immunity, fast algebraic immunity and linear bias.

The main advantage of the functions constructed using Corollary 2 is that these functions can be constructed using $O(n)$ gates, while the method in [30] is based on defining the support of the desired Boolean function using powers of a primitive element over the field $\mathbb{F}_{2^{n/2}}$; in particular, from the description of Construction 2 in [30] it appears that the only reasonable method of implementing the function is to use a truth table requiring $\Omega(2^n)$ gates.

We provide a concrete example to highlight the difference between the functions constructed using Corollary 2 and Theorem 6 of [30] (which is based on Construction 2 of [30]). Suppose $n = 32$. Theorem 6 of [30] provides a 32-variable, 1-resilient function (say g_{32}) with algebraic immunity 16, fast algebraic immunity at least 26 and nonlinearity at least 2147192232 (equivalently, linear bias at most about $2^{-12.85}$) which can be implemented using around 2^{32} gates. Taking $n = 100$ in Corollary 2 we obtain a 100-variable, 1-resilient function (say g_{100}) with algebraic immunity at least 25 (and hence fast algebraic immunity at least 26), and almost optimal linear bias equal to 2^{-49}. Further, g_{100} can be implemented using 49 AND, 50 XOR, and $G(49)$ NAND gates. So comparing g_{32} to g_{100}, we see that g_{100} has a much lower linear bias and is much more efficient to implement. The trade-off is that the number of variables required is more.

Remark 3 *Proposition 7 guarantees algebraic immunity of the constructed n-variable, m-resilient function to be at least $\lceil (n-m-1)/4 \rceil$. For concrete values of n, the actual value of algebraic immunity can actually be greater than the lower bound. For example, if we take $n = 10$ and $m = 1$, then the lower bound on algebraic immunity of the 10-variable function constructed using Proposition 7 is 2; we constructed the 10-variable function and computed its actual algebraic immunity which turns out to be 3.*

3.2 Case $n \equiv m \bmod 2$

Proposition 7 covers the case where n and m do not have the same parity. The case where n and m have the same parity can also be covered in a similar manner. For such a construction we use a 5-variable function given by the following result.

Proposition 8 *Define*

$$f_5(X_1, X_2, Z_1, Z_2, Z_3) = Z_1 \oplus Z_2 \oplus X_1(Z_1 \oplus Z_3) \oplus X_2(Z_2 \oplus Z_3) \oplus X_1X_2(Z_1 \oplus Z_2 \oplus Z_3). \quad (10)$$

The function f_5 is a 5-variable, 1-resilient function having degree 3, algebraic immunity 2, nonlinearity 12 (and hence almost optimal linear bias equal to 2^{-2}), and can be implemented using 7 XOR and 4 AND gates.

Proof: We note that f_5 can be written in the following manner.

$$\begin{aligned} f_5(X_1, X_2, Z_1, Z_2, Z_3) &= (1 \oplus X_1)(1 \oplus X_2)\big(Z_1 \oplus Z_2\big) \oplus (1 \oplus X_1)X_2\big(Z_1 \oplus Z_3\big) \\ &\quad \oplus X_1(1 \oplus X_2)\big(Z_2 \oplus Z_3\big) \oplus X_1X_2\big(Z_1 \oplus Z_2 \oplus Z_3\big). \end{aligned}$$

This shows that f_5 is the concatenation of 4 distinct linear functions each of which is non-degenerate on at least 2 variables. It follows that f_5 is 1-resilient. Let the 4 distinct linear functions making

up f be h_1, h_2, h_3 and h_4. Any 5-variable affine function ℓ can be written as concatenation of g_1, g_2, g_3, g_4, where each g_i is either g or $1 \oplus g$, where g is some 3-variable linear function. So $d(f, \ell) = \sum_{i=1}^{4} d(h_i, g_i)$. If neither g nor $1 \oplus g$ is equal to any of the h_i's, then $d(f, \ell) = 16$; if g or $1 \oplus g$ is equal to one of the h_i's, then $d(f, \ell) = 12$; and if g or $1 \oplus g$ is equal to one of the $(1 \oplus h_i)$'s, then $d(f, \ell) = 20$. Since these are the only three possibilities, it follows that the nonlinearity of f is 12 and hence f_5 has almost optimal linear bias equal to 2^{-2}.

The degree of f_5 is 3. We computed the algebraic immunity of f_5 using a computer and found it to be 2.

The expression for f_5 given by (10) can be implemented using 7 XOR and 4 AND gates. □

Using Proposition 8, we provide the construction for the case where n and m have the same parity.

Construction 3 *Let m be a positive integer, and let $n \geq m + 4$ be such that $n \equiv m \bmod 2$. Let $k = (n - m - 4)/2$. Let $\psi : \{0,1\}^k \to \{0,1\}^k$ be a bit permutation. Let f_5 be the function defined in* (10)*. The n-variable function f is defined as follows.*

$$\begin{aligned} & f(Y_1, \ldots, Y_{m-1}, X_1, X_2, Z_1, Z_2, Z_3, U_1, \ldots, U_k, V_1, \ldots, V_k) \\ &= \; Y_1 \oplus \cdots \oplus Y_{m-1} \oplus f_5(X_1, X_2, Z_1, Z_2, Z_3) \oplus (\psi, h)\text{-}\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k). \end{aligned} \quad (11)$$

The properties of the function obtained from Construction 3 are stated in the following result.

Proposition 9 *Let f be the n-variable function obtained as in Construction 2. Then f is an m-resilient function with linear bias equal to $2^{-(n-m)/2}$. Further, if $h = \mathsf{Maj}_k$, then the algebraic immunity of f is at least $\lceil (n-m-4)/4 \rceil$, and f can be implemented using $(k+4)$ AND, $(k+m+7)$ XOR, and $G(k)$ NAND gates.*

Proof: The function f_5 and $m - 1$ new variables are added to the bent function. The function f_5 is itself 1-resilient, and adding the $m - 1$ new variables increases resiliency to m. The linear bias of the bent function on $2k$ variables is 2^{-k}, where $k = (n - m - 4)/2$. The linear bias of f_5 is 2^{-2}, and so the linear bias of the sum of f_5 and the bent function is 2^{-k-2}. Addition of the new variables does not change the linear bias.

From Proposition 4 the algebraic immunity of f is at least the algebraic immunity of the bent function; from Theorem 2 the algebraic immunity of the bent function is at least the algebraic immunity of $h = \mathsf{Maj}_k$; and from Theorem 1 the algebraic immunity of Maj_k is equal to $\lceil k/2 \rceil$.

Maj_k requires $G(k)$ NAND gates. The counts of the other gates follow from directly from Construction 3 and Proposition 8. □

4 Iterated Construction

Both Propositions 7 and 9 essentially simply add new variables to increase resiliency. This creates linear structures (see Proposition 28 of [5]) and may be considered undesirable. In this section, we describe a different method for increasing resiliency. This results in functions which provide a different type of trade-off between resiliency, nonlinearity, and algebraic immunity. For the construction, we resurrect the idea of an iterated construction of resilient functions which was

only briefly sketched in [22]. The idea in [22] was itself based on a more general theoretical result from [31]. The description in [22] briefly considered resiliency and nonlinearity, but not algebraic immunity (in fact, the work [22] predates the introduction of the notion of algebraic immunity).

While the construction idea is from [22], the detailed description that we present below was not provided in [22].

Method 1 *Let g and h be two n-variable functions, and let f be an $(n+1)$-variable function obtained as* $\mathsf{Concat}(g,h)$*, i.e.*

$$f(X_1,\ldots,X_{n+1}) = (1 \oplus X_{n+1})g(X_1,\ldots,X_n) \oplus X_{n+1}h(X_1,\ldots,X_n).$$

Define $(n+3)$-variable functions G and H as follows.

$$\begin{aligned} & G(X_1,\ldots,X_{n+3}) \\ &= X_{n+3} \oplus X_{n+2} \oplus f(X_1,\ldots,X_{n+1}) \\ &= X_{n+3} \oplus X_{n+2} \oplus (1 \oplus X_{n+1})g(X_1,\ldots,X_n) \oplus X_{n+1}h(X_1,\ldots,X_n), \qquad (12) \\ & H(X_1,\ldots,X_{n+3}) \\ &= X_{n+3} \oplus X_{n+1} \oplus (1 \oplus X_{n+3} \oplus X_{n+2})g(X_1,\ldots,X_n) \oplus (X_{n+3} \oplus X_{n+2})h(X_1,\ldots,X_n). \qquad (13) \end{aligned}$$

By $\mathsf{Step}(g,h)$ *we denote the pair of functions (G,H) obtained in* (12) *and* (13)*. Define an $(n+4)$-variable function F as* $\mathsf{Concat}(G,H)$*, i.e.*

$$F(X_1,\ldots,X_{n+4}) = (1 \oplus X_{n+4})G(X_1,\ldots,X_{n+3}) \oplus X_{n+4}H(X_1,\ldots,X_{n+3}). \qquad (14)$$

The following result relates the properties of g and h to that of G, H and F.

Theorem 3 *Let n be a positive integer, and g and h be two n-variable functions. Let $(G,H) = \mathsf{Step}(g,h)$ and $F = \mathsf{Concat}(G,H)$ be constructed as in Method 1. Then the following holds.*

1. *If g and h are m-resilient, then G and H are $(m+2)$-resilient.*
2. $\mathsf{nl}(G) = \mathsf{nl}(H) = 4\mathsf{nl}(f)$*, and equivalently* $\mathsf{LB}(G) = \mathsf{LB}(H) = \mathsf{LB}(f)$.
3. *Let $\ell(X_1,\ldots,X_{n+3})$ be a linear function. Then either $G \oplus \ell$ or $H \oplus \ell$ is balanced.*
4. *Obtaining G and H from g and h requires 8 XOR and 4 AND gates.*
5. *If g and h are m-resilient, then F is $(m+2)$-resilient.*
6. $\mathsf{nl}(F) = 2^{n+2} + \mathsf{nl}(G) = 2^{n+2} + 4\mathsf{nl}(H)$*, and equivalently* $\mathsf{LB}(F) = \mathsf{LB}(G)/2 = \mathsf{LB}(H)/2$.
7. *Obtaining F from G and H requires 2 XOR and 2 AND gates.*

Proof:

Proof of the first two points. If g and h are m-resilient, then so is f. Note that G is obtained by adding two new variables to f. It then follows that G is $(m+2)$-resilient and further $\mathsf{nl}(G) = 4\mathsf{nl}(f)$. The function H is obtained from the function G by the following invertible linear transformation on the variables: $X_{n+1} \to X_{n+3} \oplus X_{n+2}$, $X_{n+2} \to X_{n+1}$, $X_{n+3} \to X_{n+3}$. Since nonlinearity is invariant under an invertible linear transformation on the variables, it follows that $\mathsf{nl}(H) = \mathsf{nl}(G)$.

The function $H(X_1,\ldots,X_{n+3})$ can be written as $H(X_1,\ldots,X_{n+3}) = X_{n+1} \oplus H'(X_1,\ldots,X_n, X_{n+2}, X_{n+3})$, where

$$H'(X_1,\ldots,X_n,X_{n+2},X_{n+3})$$

$$
\begin{aligned}
&= (1 \oplus X_{n+3})(1 \oplus X_{n+2})g(X_1, \ldots, X_n) \oplus (1 \oplus X_{n+3})X_{n+2}h(X_1, \ldots, X_n) \\
&\quad \oplus X_{n+3}(1 \oplus X_{n+2})(1 \oplus h(X_1, \ldots, X_n)) \oplus X_{n+3}X_{n+2}(1 \oplus g(X_1, \ldots, X_n)).
\end{aligned}
$$

Since H is obtained from H' by adding a new variable, it follows that to show H is $(m+2)$-resilient, it is sufficient to show H' is $(m+1)$-resilient. To show that H' is $(m+1)$-resilient, it is sufficient to show that $H' \oplus \ell'$ is balanced for all $(n+2)$-variable linear functions $\ell'(X_1, \ldots, X_n, X_{n+2}, X_{n+3})$ which are non-degenerate on at most $m+1$ variables. Write

$$
\begin{aligned}
&\ell'(X_1, \ldots, X_n, X_{n+2}, X_{n+3}) \\
&= (1 \oplus X_{n+3})(1 \oplus X_{n+2})\ell_1(X_1, \ldots, X_n) \oplus (1 \oplus X_{n+3})X_{n+2}\ell_2(X_1, \ldots, X_n) \\
&\quad \oplus X_{n+3}(1 \oplus X_{n+2})\ell_3(X_1, \ldots, X_n) \oplus X_{n+3}X_{n+2}\ell_4(X_1, \ldots, X_n),
\end{aligned}
$$

where $\ell_1, \ell_2, \ell_3, \ell_4$ are linear functions. Note that for $1 \leq i < j \leq 4$, either $\ell_i = \ell_j$ or $\ell_i = 1 \oplus \ell_j$. We have

$$
\mathsf{wt}(H' \oplus \ell') = \mathsf{wt}(g \oplus \ell_1) + \mathsf{wt}(h \oplus \ell_2) + (2^n - \mathsf{wt}(h \oplus \ell_3)) + (2^n - \mathsf{wt}(g \oplus \ell_4)). \tag{15}
$$

Suppose that ℓ' is degenerate on both X_{n+2} and X_{n+3}. In this case, all the ℓ_i's are equal, and so from (15) it follows that $\mathsf{wt}(H' \oplus \ell) = 2^{n+1}$. Next suppose that ℓ is non-degenerate on at least one of X_{n+2} or X_{n+3}. In this case, each of the ℓ_i's is non-degenerate on at most m variables. Since both g and h are m-resilient, it follows that $\mathsf{wt}(g \oplus \ell_1)$, $\mathsf{wt}(h \oplus \ell_2)$, $\mathsf{wt}(h \oplus \ell_3)$ and $\mathsf{wt}(g \oplus \ell_4)$ are all equal to 2^{n-1}, and so $\mathsf{wt}(H' \oplus \ell) = 2^{n+1}$. This shows that H' is $(m+1)$-resilient and hence H is $(m+2)$-resilient.

Proof of the third point. Suppose the linear function $\ell(X_1, \ldots, X_{n+3})$ is degenerate on X_{n+3}. Then $G \oplus \ell$ can be written as X_{n+3} plus a function which does not involve X_{n+3}, and hence $G \oplus \ell$ is balanced. Similarly, if ℓ is degenerate on X_{n+2}, then $G \oplus \ell$ can be written as X_{n+2} plus a function which does not involve X_{n+2}, and hence $G \oplus \ell$ is balanced. Further, if ℓ is degenerate on X_{n+1}, then $H \oplus \ell$ can be written as X_{n+1} plus a function which does not involve X_{n+1}, and hence $H \oplus \ell$ is balanced. So suppose ℓ is non-degenerate on all three of the variables X_{n+1}, X_{n+2} and X_{n+3}, i.e. $\ell(X_1, \ldots, X_{n+3}) = X_{n+3} \oplus X_{n+2} \oplus X_{n+1} \oplus \sigma(X_1, \ldots, X_n)$, where σ is a linear function. We argue that $H \oplus \ell$ is balanced. From the definition of H, we have $H \oplus \ell = X_{n+2} \oplus (1 \oplus X_{n+3} \oplus X_{n+2})g \oplus (X_{n+3} \oplus X_{n+2})h \oplus \sigma$ (which is degenerate on X_{n+1}). Considering the four possible values of X_{n+3} and X_{n+2}, the sub-functions of $H \oplus \ell$ that are obtained are $g \oplus \sigma$ (corresponding to $X_{n+3} = 0$, $X_{n+2} = 0$), $1 \oplus h \oplus \sigma$ (corresponding to $X_{n+3} = 0$, $X_{n+2} = 1$), $h \oplus \sigma$ (corresponding to $X_{n+3} = 1$, $X_{n+2} = 0$), and $1 \oplus g \oplus \sigma$ (corresponding to $X_{n+3} = 1$, $X_{n+2} = 1$). So $\mathsf{wt}(H \oplus \ell) = 2(\mathsf{wt}(g \oplus \sigma) + \mathsf{wt}(1 \oplus h \oplus \sigma) + \mathsf{wt}(h \oplus \sigma) + \mathsf{wt}(1 \oplus g \oplus \sigma)) = 2^{n+2}$, where the factor of 2 arises from the variable X_{n+1} on which $H \oplus \ell$ is degenerate. Hence, $H \oplus \ell$ is balanced.

Proof of the fourth point. The counts of the XOR and AND gates are clear from (12) and (13).

Proof of the fifth and sixth points. Next we provide the arguments for the properties of F. Since H and G are $(m+2)$-resilient, it follows that F is also $(m+2)$-resilient. Let $\mu(X_1, \ldots, X_{n+4})$ be an affine function on $(n+4)$ variables. Then μ can be written as $\mu = c \oplus dX_{n+4} \oplus \ell(X_1, \ldots, X_{n+3})$, where $c, d \in \mathbb{F}_2$, and ℓ is a linear function on $n+3$ variables. Then $\mathsf{wt}(F \oplus \mu) = \mathsf{wt}(c \oplus G \oplus \ell) +$

$\mathsf{wt}(c \oplus d \oplus H \oplus \ell)$. From the third point of the theorem, we have that either $G \oplus \ell$, or $H \oplus \ell$ is balanced. Suppose $G \oplus \ell$ is balanced. Then $\mathsf{wt}(F \oplus \mu) = 2^{n+2} + \mathsf{wt}(c \oplus d \oplus H \oplus \ell) \geq 2^{n+2} + \mathsf{nl}(H)$, where the equality is achieved for c, d and ℓ such that $\mathsf{nl}(H) = \mathsf{wt}(c \oplus d \oplus H \oplus \ell)$. Similarly, if $H \oplus \ell$ is balanced, then $\mathsf{wt}(F \oplus \mu) = 2^{n+2} + \mathsf{wt}(c \oplus G \oplus \ell)$, where the equality is achieved for c and ℓ such that $\mathsf{nl}(G) = \mathsf{wt}(c \oplus G \oplus \ell)$. From these two cases, the statement on the nonlinearity of F follows.

Proof of the seventh point. This is clear from the construction. □

To obtain a lower bound on the algebraic immunities of G and H obtained from Method 1, we first prove the following general result.

Proposition 10 *Let n, n_1 and n_2 be positive integers with $n = n_1 + n_2$, and let $f(X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2})$ be an n-variable function. We write*

$$f(X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2}) = \bigoplus_{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_{n_2})} (1 \oplus \alpha_1 \oplus Y_1) \cdots (1 \oplus \alpha_{n_2} \oplus Y_{n_2}) f_{\boldsymbol{\alpha}}(X_1, \ldots, X_{n_1}), \quad (16)$$

where for $\boldsymbol{\alpha} \in \mathbb{F}_2^{n_2}$, $f_{\boldsymbol{\alpha}}(X_1, \ldots, X_{n_1}) = f(X_1, \ldots, X_{n_1}, \alpha_1, \ldots, \alpha_{n_2})$. Then $\mathsf{AI}(f) \geq \min_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n_2}} \mathsf{AI}(f_{\boldsymbol{\alpha}})$.

Proof: Suppose g is a non-zero annihilator of f. We write

$$g(X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2}) = \bigoplus_{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_{n_2})} (1 \oplus \alpha_1 \oplus Y_1) \cdots (1 \oplus \alpha_{n_2} \oplus Y_{n_2}) g_{\boldsymbol{\alpha}}(X_1, \ldots, X_{n_1}),$$

where $g_{\boldsymbol{\alpha}}(X_1, \ldots, X_{n_1}) = g(X_1, \ldots, X_{n_1}, \alpha_1, \ldots, \alpha_{n_2})$.

Since $gf = 0$, it follows that $g_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}} = 0$ for all $\boldsymbol{\alpha} \in \mathbb{F}_2^{n_2}$. Further, since $g \neq 0$, there must be some $\boldsymbol{\alpha}$ such that $g_{\boldsymbol{\alpha}} \neq 0$. Then $g_{\boldsymbol{\alpha}}$ is a non-zero annihilator of $f_{\boldsymbol{\alpha}}$ and so $\mathsf{deg}(g_{\boldsymbol{\alpha}}) \geq \mathsf{AI}(f_{\boldsymbol{\alpha}})$. Clearly, $\mathsf{deg}(g) \geq \mathsf{deg}(g_{\boldsymbol{\alpha}})$, and so $\mathsf{deg}(g) \geq \mathsf{AI}(f_{\boldsymbol{\alpha}})$.

A similar reasoning shows that if g is a non-zero annihilator of $1 \oplus f$, then $\mathsf{deg}(g) \geq \mathsf{AI}(f_{\boldsymbol{\beta}})$ for some $\boldsymbol{\beta} \in \mathbb{F}_2^{n_2}$.

Since $\mathsf{AI}(f)$ is the minimum of the degrees of all the non-zero annihilators of f, and the degrees of all the non-zero annihilators of $1 \oplus f$, the result follows. □

The lower bound on the algebraic immunity of a direct sum which was obtained in [21] and is stated in Proposition 4 can be seen as a corollary of Proposition 10. The idea is that the sub-functions of f obtained by fixing $X_1, \ldots, X_{n_1}$ to arbitrary values are either $h(Y_1, \ldots, Y_{n_2})$ or $1 \oplus h(Y_1, \ldots, Y_{n_2})$. So from Proposition 10, we have $\mathsf{AI}(f) \geq \mathsf{AI}(h)$. Similarly, by fixing $Y_1, \ldots, Y_{n_2}$ to arbitrary values, we obtain $\mathsf{AI}(f) \geq \mathsf{AI}(g)$.

Theorem 4 *Let G, H and F be the functions constructed as in Method 1. Then*

1. $\mathsf{AI}(G), \mathsf{AI}(H) \geq \min\{\mathsf{AI}(g), \mathsf{AI}(h)\}$.
2. $\mathsf{AI}(F) \geq \min\{\mathsf{AI}(G), \mathsf{AI}(H)\} \geq \min\{\mathsf{AI}(g), \mathsf{AI}(h)\}$.

Proof: By setting the variables $X_{n+3}, X_{n+2}, X_{n+1}$ to arbitrary values, the sub-functions of G that are obtained are g, $1 \oplus g$, h, and $1 \oplus h$. Since g and $1 \oplus g$ have the same algebraic

immunity, and h and $1 \oplus h$ also have the same algebraic immunity, the lower bound on $\mathsf{AI}(G)$ follows from Proposition 10. The lower bounds on $\mathsf{AI}(H)$ and $\mathsf{AI}(F)$ follow in a similar manner using Proposition 10. $\square$

Method 1 can be iterated to obtain functions on progressively larger number of variables. The following construction describes the idea.

Method 2 *Let n and t be positive integers, and let g and h be two n-variable functions. Consider the following iterated construction.*

$g^{(0)} \leftarrow g$; $h^{(0)} \leftarrow h$; $f^{(0)} \leftarrow \mathsf{Concat}(g^{(0)}, h^{(0)})$;
for $i \leftarrow 1$ *to* t *do*
  $(g^{(i)}, h^{(i)}) \leftarrow \mathsf{Step}(g^{(i-1)}, h^{(i-1)})$; $f^{(i)} \leftarrow \mathsf{Concat}(g^{(i)}, h^{(i)})$;
end for;
return $f^{(t)}$.

We denote the function $f^{(t)}$ by $\mathsf{Iter}_t(g, h)$.

The properties of the function constructed using Method 2 are given in the following result.

Theorem 5 *Let n and t be positive integers, g and h be n-variable functions. Let $f = \mathsf{Concat}(g, h)$, and $f^{(t)} = \mathsf{Iter}_t(g, h)$. Then the following holds.*

1. *The function $f^{(t)}$ is an $(n + 3t + 1)$-variable function.*
2. *If g and h are m-resilient, then $f^{(t)}$ is $(m + 2t)$-resilient.*
3. $\mathsf{LB}(f^{(t)}) = 2^{-t} \cdot \mathsf{LB}(f)$.
4. $\mathsf{AI}(f^{(t)}) \geq \min\{\mathsf{AI}(g), \mathsf{AI}(h)\}$.
5. *Obtaining $f^{(t)}$ from g and h requires $8t + 2$ XOR and $4t + 2$ AND gates.*

Proof: The result follows from Theorem 3 by induction on t. $\square$

By appropriately choosing the initial functions g and h, Theorem 5 can be used to obtain the resiliency, linear bias and lower bound on algebraic immunity of the function $\mathsf{Iter}_t(g, h)$. Based on this idea, we define the following two constructions.

Construction 4 *Let n and t be positive integers such that $n - 3t - 1 \geq 3$ is an odd integer. Let $k = (n - 3t - 2)/2$, and $\psi : \{0,1\}^k \to \{0,1\}^k$ be a bit permutation. Define*

$$\begin{aligned} g^{(0)}(X_1, U_1, \ldots, U_k, V_1, \ldots, V_k) &= X_1 \oplus (\psi, \mathsf{Maj}_k)\text{-}\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k) \\ h^{(0)}(X_1, U_1, \ldots, U_k, V_1, \ldots, V_k) &= 1 \oplus X_1 \oplus (\psi, \mathsf{Maj}_k)\text{-}\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k) \\ f^{(t)} &= \mathsf{Iter}_t(g^{(0)}, h^{(0)}). \end{aligned}$$

Construction 5 *Let n and t be positive integers such that $n - 3t - 1 \geq 4$ is an even integer. Let $k = (n - 3t - 5)/2$, and $\psi : \{0,1\}^k \to \{0,1\}^k$ be a bit permutation. Define*

$$\begin{aligned} & g^{(0)}(X_1, Z_1, Z_2, Z_3, U_1, \ldots, U_k, V_1, \ldots, V_k) \\ &\quad = Z_1 \oplus Z_2 \oplus X_1(Z_1 \oplus Z_3) \oplus (\psi, \mathsf{Maj}_k)\text{-}\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k) \\ & h^{(0)}(X_1, Z_1, Z_2, Z_3, U_1, \ldots, U_k, V_1, \ldots, V_k) \\ &\quad = Z_1 \oplus Z_3 \oplus X_1 Z_2 \oplus (\psi, \mathsf{Maj}_k)\text{-}\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k) \\ f^{(t)} &= \mathsf{Iter}_t(g^{(0)}, h^{(0)}). \end{aligned}$$

The properties of the functions obtained using Constructions 4 and 5 are stated in the following theorem.

Theorem 6 *Let n and t be positive integers.*

1. *Suppose $n-3t-1 \geq 3$ is an odd integer and $f^{(t)}$ is the function obtained from Construction 4. Then $f^{(t)}$ is an n-variable, $2t$-resilient function with linear bias equal to $2^{-(n-t-2)/2}$, algebraic immunity at least $\lceil (n-3t-2)/4 \rceil$, and can be implemented using $4t+k+2$ AND, $8t+k+3$ XOR, and $G(k)$ NAND gates.*

2. *Suppose $n-3t-1 \geq 4$ is an even integer and $f^{(t)}$ is the function obtained from Construction 5. Then $f^{(t)}$ is an n-variable, $(2t+1)$-resilient function with linear bias equal to $2^{-(n-t-1)/2}$, algebraic immunity at least $\lceil (n-3t-5)/4 \rceil$, and can be implemented using $4t+k+4$ AND, $8t+k+8$ XOR, and $G(k)$ NAND gates.*

Proof: First suppose $n-3t-1$ is odd.

Note that due to the addition of the variable X_1, both $g^{(0)}$ and $h^{(0)}$ are balanced, i.e. 0-resilient. From Proposition 4, the algebraic immunities of both $g^{(0)}$ and $h^{(0)}$ are at least the algebraic immunity of (ψ, Maj_k)-$\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k)$; from Theorem 2 the algebraic immunity of (ψ, Maj_k)-$\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k)$ is at least the algebraic immunity of Maj_k; and from Theorem 1, the algebraic immunity of Maj_k is at least $\lceil k/2 \rceil = \lceil (n-3t-2)/4 \rceil$. So the algebraic immunities of both $g^{(0)}$ and $h^{(0)}$ are at least $\lceil (n-3t-2)/4 \rceil$. Define

$$\begin{aligned} & f^{(0)}(X_1, X_2, U_1, \ldots, U_k, V_1, \ldots, V_k) \\ &= (1 \oplus X_2) g^{(0)}(X_1, U_1, \ldots, U_k, V_1, \ldots, V_k) \oplus X_2 h^{(0)}(X_1, U_1, \ldots, U_k, V_1, \ldots, V_k). \end{aligned}$$

Simplifying we have $f^{(0)}(X_1, X_2, U_1, \ldots, U_k, V_1, \ldots, V_k) = X_1 \oplus X_2 \oplus (\psi, \mathsf{Maj}_k)$-$\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k)$. From Theorem 7, the linear bias of $f^{(0)}$ is equal to $2^{-(n-3t-2)/2}$. Note that $g^{(0)}$ and $h^{(0)}$ are functions of $n-3t-1$ variables and $f^{(0)}$ is a function of $n-3t$ variables. Since $f^{(t)} = \mathsf{Iter}_t(g^{(0)}, h^{(0)})$, from Theorem 5 it follows that $f^{(t)}$ is a function of n variables. Further, also from Theorem 5, the resiliency of $f^{(t)}$ is $2t$, linear bias is equal to $2^{-t}\mathsf{LB}(f^{(0)}) = 2^{-(n-t-2)/2}$, and algebraic immunity is at least $\lceil (n-3t-2)/4 \rceil$. The implementation of $f^{(t)}$ requires the implementation of Maj_k, and the implementation of the bit permutation ψ. The implementation of the bit permutation ψ does not require any gates (only the appropriate connection pattern needs to be implemented). Maj_k requires $G(k)$ NAND gates. The counts of the other gates follow directly from Construction 4 and Theorem 5.

Next suppose $n-3t-1$ is even. By an argument similar to the case for $n-3t-1$ is odd, the algebraic immunities of both $g^{(0)}$ and $h^{(0)}$ are at least $\lceil k/2 \rceil = \lceil (n-3t-5)/4 \rceil$. Define

$$\begin{aligned} & f^{(0)}(X_1, X_2, Z_1, Z_2, Z_3, U_1, \ldots, U_k, V_1, \ldots, V_k) \\ &= (1 \oplus X_2) g^{(0)}(X_1, Z_1, Z_2, Z_3, U_1, \ldots, U_k, V_1, \ldots, V_k) \\ &\quad \oplus X_2 h^{(0)}(X_1, Z_1, Z_2, Z_3, U_1, \ldots, U_k, V_1, \ldots, V_k). \end{aligned}$$

Simplifying we have

$$f^{(0)}(X_1, X_2, Z_1, Z_2, Z_3, U_1, \ldots, U_k, V_1, \ldots, V_k)$$

$$= \quad f_5(X_1, X_2, Z_1, Z_2, Z_3) \oplus (\psi, h)\text{-}\mathsf{MM}_{2k}(U_1, \ldots, U_k, V_1, \ldots, V_k).$$

From Theorem 9, $f^{(0)}$ is 1-resilient. Note that $Z_1 \oplus Z_2 \oplus X_1(Z_1 \oplus Z_3) = (1 \oplus X_1)(Z_1 \oplus Z_2) \oplus X_1(Z_2 \oplus Z_3)$, which is the concatenation of two linear functions both of which are non-degenerate on 2 variables, and hence is 1-resilient. Since $g^{(0)}$ is the direct sum of the 1-resilient function $Z_1 \oplus Z_2 \oplus X_1(Z_1 \oplus Z_3)$ and a bent function, it follows that $g^{(0)}$ is 1-resilient. Similarly, we may write $Z_1 \oplus Z_3 \oplus X_1 Z_2 = (1 \oplus X_1)(Z_1 \oplus Z_3) \oplus X_1(Z_1 \oplus Z_2 \oplus Z_3)$ to see that $h^{(0)}$ is the direct sum of a 1-resilient function and a bent function, and so it follows that $h^{(0)}$ is 1-resilient. From Theorem 9, the linear bias of $f^{(0)}$ is equal to $2^{-(2k+6-2)/2} = 2^{-(n-3t-1)/2}$. As in the case for $n - 3t - 1$ being odd, from Theorem 5, $f^{(t)}$ is a function of n variables, which is $(2t+1)$-resilient, having linear bias equal to $2^{-t}\mathsf{LB}(f^{(0)}) = 2^{-(n-t-1)/2}$, algebraic immunity at least $\lceil (n - 3t - 5)/4 \rceil$.

As in the previous case, the bit permutation ψ does not require any gate, Maj_k requires $G(k)$ NAND gates, and the counts of the other gates follow directly from Construction 5 and Theorem 5. □

5 Comparison

In this section, we compare the four constructions, namely Construction 2 to 5. All the four constructions make use of the majority function to achieve a guaranteed lower bound on algebraic immunity. Further, for all the constructions, it is the majority function which requires the maximum number of gates. For comparing among the four constructions, we assume that all of them achieve the same guaranteed lower bound (say a_0) on the algebraic immunity. This requires setting $k = 2a_0 - 1$ in all the constructions. Under this setting, the parameters of the different constructions are shown in Table 2.

	Proposition 7 $k = 2a_0 - 1$	Proposition 9 $k = 2a_0 - 1$	Theorem 6(1) $k = 2a_0 - 1$	Theorem 6(2) $k = 2a_0 - 1$
n	$m + 4a_0 - 1$	$m + 4a_0 + 2$	$3t + 4a_0$	$3t + 4a_0 + 3$
resiliency	m	m	$2t$	$2t + 1$
$\lvert\log_2 \mathsf{LB}\rvert$	$2a_0 - 1$	$2a_0 + 1$	$t + 2a_0 - 1$	$t + 2a_0 + 1$
AI (lower bound)	a_0	a_0	a_0	a_0
#AND	$2a_0 - 1$	$2a_0 + 3$	$4t + 2a_0 + 1$	$4t + 2a_0 + 3$
#XOR	$2a_0 + m$	$2a_0 + m + 6$	$8t + 2a_0 + 2$	$8t + 2a_0 + 7$
size $- G(2a_0 - 1)$	$4a_0 + m - 1$	$4a_0 + m + 9$	$12t + 4a_0 + 3$	$12t + 4a_0 + 10$

Table 2: Parameters of Proposition 7, Proposition 9 and Theorem 6(1), 6(2), all with $k = 2a_0 - 1$. The number of NAND gates is $G(2a_0 - 1)$ in all four cases.

One may observe that the trade-offs achieved by Proposition 7 and 9 are different from those achieved by the two cases of Theorem 6. For concreteness let us consider Proposition 7 and Theorem 6(1), the comparison in the other cases being similar.

Setting $2t = m$, ensures that both Proposition 7 and Theorem 6(1) achieve the same resiliency

(along with the same lower bound on algebraic immunity). Table 3 shows the comparison between the parameters, while the column labelled 'difference' shows the difference in the parameters.

	Proposition 7 $k = 2a_0 - 1$	Theorem 6(1) with $t = m/2$ $k = 2a_0 - 1$	difference (Thm. 6(1) − Prop. 6)
n	$m + 4a_0 - 1$	$\frac{3m}{2} + 4a_0$	$\frac{m}{2} + 1$
resiliency	m	m	0
$\lvert\log_2 \mathsf{LB}\rvert$	$2a_0 - 1$	$\frac{m}{2} + 2a_0 - 1$	$\frac{m}{2}$
AI (lower bound)	a_0	a_0	0
#AND	$2a_0 - 1$	$2m + 2a_0 + 1$	$2m + 2$
#XOR	$2a_0 + m$	$4m + 2a_0 + 2$	$3m + 2$
size $- G(2a_0 - 1)$	$4a_0 + m - 1$	$6m + 4a_0 + 3$	$5m + 4$

Table 3: Parameters of Proposition 6 and Theorem 6(1) with $2t = m$ and $k = 2a_0 - 1$. The number of NAND gates is $G(2a_0 - 1)$ in both cases.

For the same order of resiliency m and the same lower bound a_0 on algebraic immunity, Theorem 6(1) has linear bias smaller by a factor of $2^{-m/2}$, at the cost of $m/2 + 1$ more variables and $5m + 4$ more gates. For example, for $a_0 = 30$ and $m = 8$ (so $t = 4$), Proposition 7 gives $n = 127$, linear bias 2^{-59}, and $127 + G(59)$ gates, while Theorem 6(1) gives $n = 132$, linear bias 2^{-63}, and $171 + G(59)$ gates.

6 Conclusion

We have described several constructions which provide functions with provable trade-offs between resiliency, linear bias, and algebraic immunity *and* at the same time are efficient to implement. From a cryptographic perspective, the efficiency of implementation is important for functions to be actually used in practical designs. We have compared our constructions to the known constructions in the literature and have identified the trade-offs. We hope our work will stimulate further research in the area of efficiently implementable Boolean functions with improved trade-offs between the security parameters.

Acknowledgement

We gratefully acknowledge comments on an earlier version which helped to improve the paper.

References

[1] Benny Applebaum and Shachar Lovett. Algebraic attacks against random local functions and their countermeasures. *SIAM J. Comput.*, 47(1):52–79, 2018. 3

[2] Joan Boyar and René Peralta. The exact multiplicative complexity of the Hamming weight function. *Electron. Colloquium Comput. Complex.*, TR05-049, 2005. `https://dblp.org/rec/journals/eccc/ECCC-TR05-049.bib`. 6

[3] Claude Carlet. On the coset weight divisibility and nonlinearity of resilient and correlation-immune functions. In Tor Helleseth, P. Vijay Kumar, and Kyeongcheol Yang, editors, *Sequences and their Applications - Proceedings of SETA 2001, Bergen, Norway, May 13-17, 2001*, Discrete Mathematics and Theoretical Computer Science, pages 131–144. Springer, 2001. 5

[4] Claude Carlet. Comments on "constructions of cryptographically significant Boolean functions using primitive polynomials". *IEEE Trans. Inf. Theory*, 57(7):4852–4853, 2011. 2, 11

[5] Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, 2021. 1, 3, 5, 13

[6] Claude Carlet and Keqin Feng. An infinite class of balanced functions with optimal algebraic immunity, good immunity to fast algebraic attacks and good nonlinearity. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*, pages 425–440. Springer, 2008. 2, 11

[7] Claude Carlet and Pierrick Méaux. Boolean functions for homomorphic-friendly stream ciphers. In Cheikh Thiecoumba Gueye, Edoardo Persichetti, Pierre-Louis Cayrel, and Johannes Buchmann, editors, *Algebra, Codes and Cryptology*, pages 166–182, Cham, 2019. Springer International Publishing. 6

[8] Claude Carlet and Pierrick Méaux. A complete study of two classes of boolean functions: Direct sums of monomials and threshold functions. *IEEE Trans. Inf. Theory*, 68(5):3404–3425, 2022. 2, 7, 8

[9] Claude Carlet and Palash Sarkar. Spectral domain analysis of correlation immune and resilient Boolean functions. *Finite Fields and Their Applications*, 8:120–130, 2002. 5

[10] Claude Carlet and Palash Sarkar. The nonlinear filter model of stream cipher redivivus. *IEEE Trans. Inf. Theory*, 72(9):7102–7120, 2026. 2, 3, 6, 8, 10, 11

[11] Nicolas T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194. Springer, 2003. 6

[12] Nicolas T. Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359. Springer, 2003. 1, 5

[13] Deepak Kumar Dalai, Subhamoy Maitra, and Sumanta Sarkar. Basic theory in construction of Boolean functions with maximum possible annihilator immunity. *Des. Codes Cryptogr.*, 40(1):41–58, 2006. 1, 6

[14] Aurélien Dupin, Pierrick Méaux, and Mélissa Rossi. On the algebraic immunity - resiliency trade-off, implications for Goldreich's pseudorandom generator. *Des. Codes Cryptogr.*, 91(9):3035–3079, 2023. 3

[15] François Gérard, Agnese Gini, and Pierrick Méaux. Toolip: How to find new instances of filip cipher with smaller key size and new filters. In Serge Vaudenay and Christophe Petit, editors, *Progress in Cryptology - AFRICACRYPT 2024 - 15th International Conference on Cryptology in Africa, Douala, Cameroon, July 10-12, 2024, Proceedings*, volume 14861 of *Lecture Notes in Computer Science*, pages 21–45. Springer, 2024. 2, 7, 8

[16] Oded Goldreich. Candidate one-way functions based on expander graphs. *Electron. Colloquium Comput. Complex.*, TR00-090, 2000. 3

[17] Xuewei Hu, Bo Yang, and Meijuan Huang. A construction of highly nonlinear Boolean functions with optimal algebraic immunity and low hardware implementation cost. *Discret. Appl. Math.*, 285:407–422, 2020. 2, 11

[18] Pierrick Méaux. On the algebraic immunity of direct sum constructions. *Discret. Appl. Math.*, 320:223–234, 2022. 6, 7

[19] Pierrick Méaux. From at least $n/3$ to at most $3\sqrt{n}$: Correcting the algebraic immunity of the hidden weight bit function. In Daniel Escudero and Ivan Damgård, editors, *Progress in Cryptology - LATINCRYPT 2025 - 9th International Conference on Cryptology and Information Security in Latin America, Medellín, Colombia, October 1-3, 2025, Proceedings*, volume 16129 of *Lecture Notes in Computer Science*, pages 345–369. Springer, 2025. 8

[20] Pierrick Méaux, Claude Carlet, Anthony Journault, and François-Xavier Standaert. Improved filter permutators for efficient FHE: better instances and implementations. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *Progress in Cryptology - INDOCRYPT 2019 - 20th International Conference on Cryptology in India, Hyderabad, India, December 15-18, 2019, Proceedings*, volume 11898 of *Lecture Notes in Computer Science*, pages 68–91. Springer, 2019. 2, 7, 8

[21] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 311–343. Springer, 2016. 2, 7, 8, 16

[22] E. Pasalic, S. Maitra, T. Johansson, and P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bound on nonlinearity. *Electronic Notes in Discrete Mathematics*, 6:158–167, 2001. WCC2001, International Workshop on Coding and Cryptography. 3, 14

[23] Oscar S. Rothaus. On "bent" functions. *J. Comb. Theory, Ser. A*, 20(3):300–305, 1976. 1, 4

[24] Palash Sarkar and Subhamoy Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 485–506. Springer, 2000. 7

[25] Palash Sarkar and Subhamoy Maitra. Nonlinearity bounds and constructions of resilient Boolean functions. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 515–532. Springer, 2000. 5

[26] Jinyong Shan, Lei Hu, Xiangyong Zeng, and Chunlei Li. A construction of 1-resilient Boolean functions with good cryptographic properties. *J. Syst. Sci. Complex.*, 31(4):1042–1064, 2018. 2, 11

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

[28] Deng Tang, Claude Carlet, and Xiaohu Tang. Highly nonlinear Boolean functions with optimal algebraic immunity and good behavior against fast algebraic attacks. *IEEE Trans. Inf. Theory*, 59(1):653–664, 2013. 2, 11

[29] Deng Tang, Claude Carlet, and Xiaohu Tang. A class of 1-resilient Boolean functions with optimal algebraic immunity and good behavior against fast algebraic attacks. *Int. J. Found. Comput. Sci.*, 25(6):763–780, 2014. 2, 11

[30] Deng Tang, Claude Carlet, Xiaohu Tang, and Zhengchun Zhou. Construction of highly nonlinear 1-resilient Boolean functions with optimal algebraic immunity and provably high fast algebraic immunity. *IEEE Trans. Inf. Theory*, 63(9):6113–6125, 2017. 2, 11, 12

[31] Yuriy V. Tarannikov. On resilient Boolean functions with maximal possible nonlinearity. In Bimal K. Roy and Eiji Okamoto, editors, *Progress in Cryptology - INDOCRYPT 2000, First International Conference in Cryptology in India, Calcutta, India, December 10-13, 2000, Proceedings*, volume 1977 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 2000. 14

[32] Ziran Tu and Yingpu Deng. Boolean functions optimizing most of the cryptographic criteria. *Discret. Appl. Math.*, 160(4-5):427–435, 2012. 2, 11

[33] Qichun Wang, Jie Peng, Haibin Kan, and Xiangyang Xue. Constructions of cryptographically significant boolean functions using primitive polynomials. *IEEE Trans. Inf. Theory*, 56(6):3048–3053, 2010. 2, 11

[34] Qichun Wang and Pantelimon Stanica. A trigonometric sum sharp estimate and new bounds on the nonlinearity of some cryptographic Boolean functions. *Des. Codes Cryptogr.*, 87(8):1749–1763, 2019. 2, 11

[35] Tianze Wang, Meicheng Liu, and Dongdai Lin. Construction of resilient and nonlinear Boolean functions with almost perfect immunity to algebraic and fast algebraic attacks. In Miroslaw Kutylowski and Moti Yung, editors, *Information Security and Cryptology - 8th International Conference, Inscrypt 2012, Beijing, China, November 28-30, 2012, Revised Selected Papers*, volume 7763 of *Lecture Notes in Computer Science*, pages 276–293. Springer, 2012. 2, 11

[36] Zhao Wang, Xiao Zhang, Sitao Wang, Zhiming Zheng, and Wenhua Wang. Construction of Boolean functions with excellent cryptographic criteria using bivariate polynomial representation. *Int. J. Comput. Math.*, 93(3):425–444, 2016. 2, 11

[37] Ingo Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons, 1987. 6