On Boolean Functions with Low Polynomial Degree and Higher Order Sensitivity (Extended Abstract)

Subhamoy Maitra[1], Chandra Sekhar Mukherjee[2], Pantelimon Stănică[3], Deng Tang[4,5]

[1] Indian Statistical Institute, Kolkata, India, subho@isical.ac.in
[2] University of Southern California, USA chandrasekhar.mukherjee07@gmail.com
[3] Naval Postgraduate School, Monterey, USA, pstanica@nps.edu
[4] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China.
[5] Science and Technology on Communication Security Laboratory, Chengdu 610041, Sichuan, China, dtang@foxmail.com

Abstract. In this paper, we explore the tools from cryptology and complexity theory in the domain of Boolean functions with low polynomial degree and high sensitivity. It is well known that the polynomial degree of a Boolean function and its resiliency are directly connected. Using this connection we analyze the polynomial degree vs sensitivity values through the lens of resiliency, demonstrating certain existence and non-existence results on small number of variables (upto 10). In this process, borrowing an idea from complexity theory, we show that one can implement resilient Boolean functions on a large number of variables with linear size and logarithmic depth. Finally, we extend the notion of sensitivity to higher order and present relevant results in that direction.

Keywords: Boolean Function, Polynomial Degree, Resiliency, Sensitivity, Separation.

1 Introduction

Real polynomial degree ($\mathsf{pdeg}(f)$) and sensitivity ($s(f)$) are two central properties of Boolean functions in complexity theory, and have been studied extensively over the past three decades. These notions have important implications in the domain of query complexity [2] (and not only), where finding functions with lower polynomial degree than sensitivity generates more candidates for obtaining super-linear separation between two of the query complexity models, the classical deterministic and exact quantum models. A detailed study of these properties and the relations can be found in [2].

Determining the maximum possible separation between sensitivity and polynomial degree is an open problem that has been studied widely. Implicitly, the problem reduces to finding the separation between the number of variables and

the real polynomial degree in a fully sensitive function (a function f on n variables with $s(f) = n$). Informally, the sufficient and necessary condition for obtaining full sensitivity in a function f on n variables is to have an input point $\mathbf{x} \in \{0,1\}^n$ such that $f(\mathbf{x}) = \overline{f(\mathbf{x}^i)}, 1 \leq i \leq n$, where $\mathbf{x}^i$ is obtained by altering the value of the i-th bit of $\mathbf{x}$. Thus, we fix the output corresponding to some $n+1$ input points, of the total 2^n input points. Interestingly, this greatly restricts the real polynomial degree of the function. Without any restriction, a function that depends on all of its variables can have $\mathsf{pdeg}(f)$ as low as $\mathcal{O}(\log n)$. However, one of the seminal papers [7] in the study of Boolean functions dictates that $\mathsf{pdeg}(f) = \Omega\left(s(f)^{\frac{1}{2}}\right)$. Furthermore, apart from the famous recursive amplification method (which is also known as the function composition), there does not exist any known method to obtain functions with non-constant separation between $s(f)$ and $\mathsf{pdeg}(f)$. In fact, the maximum separation known between $s(f)$ and $\mathsf{pdeg}(f)$ is achieved by finding a 6 variable function with $s(f) = 6$ and $\mathsf{pdeg}(f) = 3$ (due to Kushilevitz [8]) and then recursively amplifying it. This results in a function f on $n = 6^d$ variables with full sensitivity (n) and polynomial degree of 3^d so that $s(f) = \mathsf{pdeg}(f)^{\log_3 6} \approx \mathsf{pdeg}(f)^{1.63}$ [8]. On the other hand, the real polynomial degree is intrinsically connected to the cryptographically important property of resiliency. Specifically if a function f on n variables has $\mathsf{pdeg}(f) = m$ then the function $g = f \oplus \mathcal{L}_n$ is $(n-m-1)$-resilient, where $\mathcal{L}_n$ is the all variable (symmetric) linear function. In this paper we refer to g and f as each other's dual. In this regard, we define the dual sensitivity ($ds(f)$) property, where a function g has full dual sensitivity if and only if there exists a point $\mathbf{x} \in \{0,1\}^n$ such that $g(\mathbf{x}) = g(\mathbf{x}^i), 1 \leq i \leq n$. This results in a one-to-one connection between the $\mathsf{pdeg}(f) - s(f)$ relationship and resiliency order-$ds(g)$ relationship, where g is the dual of f. We should remark here that the resilient Boolean functions have received a lot of interest in construction of symmetric ciphers as evident from [5]. Related to algebraic degree, one may note that the cubic (and quadratic) functions of highest resiliency are fully classified in [3].

In this paper, we show that the techniques from complexity theory and cryptology can supplement each other with respect to combinatorial aspects of Boolean functions which are important in their own interests, and extend the notion of sensitivity to higher order towards a better understanding how fixing of outputs with respect to flipping of more than one input bits can effect the lower bound on the polynomial degree of a function. That is, we use different properties of resiliency and polynomial degree to obtain results and constructions that apply to both paradigms of cryptology and complexity theory.

1.1 Contributions and Organization

We revisit the one to one connection between "low polynomial degree-high sensitivity" and "high resiliency-dual sensitivity" of Boolean functions, via their Fourier spectrum based definitions. This connection is exploited in Section 2 to search for separation between $s(f)$ and $\mathsf{pdeg}(f)$ for functions on all small number of variables through the resiliency approach. We find new classes of functions

with maximum $s(f) - pdeg(f)$ separation and obtain super-linear separation between n and $\mathsf{pdeg}(f)$ for fully second order sensitive functions. Specifically we find the following which were not known earlier.

- There exists second order six variable functions with $\mathsf{pdeg}(f) = 3$. That is the maximum known separation for n and pdeg is same for first and second order sensitivity.
- There does not exist any 7 variable fully sensitive function with $\mathsf{pdeg}(f) = 3$.

Further, we analyze the recursive amplification method explained in [7] (see also further explanation in the proof of [8, Lemma 1]). We show that its generalization allows us to obtain additional classes of functions with super-linear separation between $s(f)$ and $\mathsf{pdeg}(f)$. Next, using this method we design efficient circuits (linear size and logarithmic depth (in n) for highly resilient functions. In the working full version [6] (this includes elaborate discussion about the motivation and background), we discuss the cryptographic properties of these functions, such as nonlinearity and will describe different trade-offs.

We use our resiliency based search method to obtain second order sensitive 6 variable functions with $\mathsf{pdeg}(f) = 3$. Coupled with the modified recursive amplification that we propose in Section 3, this gives us second order functions f^u with $n = 6^u$ variables and $\mathsf{pdeg}(f^u) = n^{\frac{\log 3}{\log 6}}$, matching the best known bound between n and $\mathsf{pdeg}(f)$ for first order sensitivity. This raises the question of whether asymptotic separation between n and $\mathsf{pdeg}(f)$ is a strictly decreasing function when plotted against sensitivity order.

1.2 Preliminaries

The definitions of resiliency and polynomial degree are based on the Fourier spectrum of a Boolean function [9]. A function f has polynomial degree k if and only if its Fourier spectrum values $W_f(\mathbf{x})$ are 0 for all $\mathbf{x} \in \{0,1\}^n$ with Hamming weight wt of $\mathbf{x}$ being greater than k. Recall that the Fourier transform of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ at the point $\mathbf{u}$ is $W_f(\mathbf{u}) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) + \mathbf{u} \cdot \mathbf{x}}$. A function f is k-resilient if and only if its Fourier spectrum values are 0 for all $\mathbf{x} : wt(\mathbf{x}) \leq k$. Generally, we will interpret a function to be k-resilient here when it is k-resilient but not $(k+1)$-resilient. Given a function f, $W_f(\mathbf{x}) = W_{f \oplus \mathcal{L}_n}(\bar{\mathbf{x}})$ where $\mathcal{L}_n$ is the linear function on n variables with $\bar{\mathbf{x}}$ obtained by flipping each bit of $\hat{\boldsymbol{x}} \in \mathbb{F}_2^n$. These structural arguments gave rise to the famous result connecting the resiliency order and polynomial degree of Boolean functions.

Theorem 1 ([9], page 150). *If a function g is k-th order resilient then the function $f = g \oplus \mathcal{L}_n$ will have a polynomial degree equal to $n - k - 1$, where $\mathcal{L}_n = \oplus_{i=1}^n x_i$.*

1.3 Sensitivity

Sensitivity $s(f)$ is one of the most studied properties of Boolean function. For any $\mathbf{x} \in \mathbb{F}_2^n$, we let $\mathbf{x}^i$ to be $\mathbf{x}$ with the i-th bit of $\mathbf{x}$ flipped (complemented).

The sensitivity of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ at a point $\mathbf{x}$ can be defined as $s(f, \mathbf{x}) = |i \in [n] : f(\mathbf{x}) \neq f(\mathbf{x}^i)|$, and the sensitivity of a function is $s(f) = \max_{\mathbf{x} \in \mathbb{F}_2^n} s(f, \mathbf{x})$. It is natural to consider the situation where we want the function to have the same value even if multiple input bits of $\mathbf{x}$ are flipped regardless of their position. In this direction, we define the k-th order sensitivity of a Boolean function.

For any set $S \subseteq [n]$ and the input point $\mathbf{x} \in \mathbb{F}_2^n$ we define $\mathbf{x}^{(S)}$ as the input point obtained by flipping the j-th bit of $\mathbf{x}$ for all $j \in S$. Sensitivity is defined around the notion of flipping any single component corresponding to a given input where the output of the function remains unchanged. In this regard we define k-th order sensitivity of a function in the following manner.

Definition 1. *We call a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, fully k-th order sensitive if there exists $\mathbf{x} \in \mathbb{F}_2^n$ such that $f(\mathbf{x}) \neq f\left(\mathbf{x}^{(S)}\right),\ \forall S \subseteq [n],\ 1 \leq |S| \leq k$.*

That is, f is fully (we may omit 'fully' in the following text) k-th order sensitive if there exists an input so that flipping any $i \leq k$ of the component bits of the input, changes the function's output. Thus a first order sensitive function is simply a function with $s(f) = n$. The main implication of k-th order sensitivity is that, it indeed further restricts how low the degree of the real polynomial corresponding to the function can be. Without any restrictions we know $\mathsf{pdeg}(f)$ can be as low as $\log n$ for functions that depend on n variables [4]. If one fixes $s(f) = n$ then the polynomial degree is $\Omega\left(\sqrt{n}\right)$ [7]. In this paper we introduce the concept of higher order sensitivity and find certain results of separation in that direction.

1.4 Dual sensitivity

Given a function f on n variables with polynomial degree m, we call the function $g = f \oplus \mathcal{L}_n$, the dual of f, which has $n - m - 1$ resiliency. The dual sensitivity of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ at a point $\mathbf{x}$ is defined as $ds(f, \mathbf{x}) = |\{i \in [n] : f(\mathbf{x}) = f(\mathbf{x}^i)\}|$. The dual sensitivity of f is $ds(f) = \max_{\mathbf{x} \in \mathbb{F}_2^n} ds(f, \mathbf{x})$. This notion can be extended to k-th order dual sensitivity in the following manner.

Definition 2. *We say a function f is k-th order dual sensitive if there exists $\mathbf{x} \in \mathbb{F}_2^n$, such that, for all $j : 1 \leq j \leq k$ we have:*

- *if $j \equiv 0 \bmod 2$, then $f(\mathbf{x}) \neq f\left(\mathbf{x}^{(S)}\right),\ \forall S \subseteq [n]$ with $|S| = j$.*
- *if $j \equiv 1 \bmod 2$, then $f(\mathbf{x}) = f\left(\mathbf{x}^{(S)}\right),\ \forall S \subseteq [n]$ with $|S| = j$.*

That is, a function is k-th order dual sensitive if there is an input point such that if we flip the values of any odd number $\leq k$ of input bits then the function's output remains unchanged and if we flip any even number $\leq k$ of input bits then the function's output gets complemented.

Proposition 1. *A function f on n variables is k-th order sensitive if and only if its dual $g = f \oplus \mathcal{L}_n$ is k-th order dual sensitive.*

Let us now present the following notations and a corresponding technical result.

- An (n, k, p)-function is a Boolean function of n variables that is k-th order sensitive and has real polynomial degree at most p.
- An $[n, k, m]$-function is a Boolean function of n variables that is k-th order dual sensitive and m-resilient.

Proposition 2. *The n-variable function f is an (n, k, p)-function if and only if $g = f \oplus \mathcal{L}_n$ is an $[n, k, n - p - 1]$-function.*

Now we move into the search based results.

2 Search on small number of variables

As we shall observe in Section 3, upon some modifications, the recursive amplification method can be used to obtain k-th order sensitive functions f^u on d^u variables with polynomial degree of p^u, starting from a function f on d variables and $\mathsf{pdeg}(f) = p$. Here f is called the base function. Thus results of low pdeg of k-th order sensitive functions on small variables directly generate super-linear separations between n and pdeg for k-th order sensitive functions.

For example, if we obtain a fully sensitive (first oder sensitive) function on 7 variables with $\mathsf{pdeg}(f) = 3$, or a 10 variable first order sensitive function with polynomial degree 4, then it would improve upon the best known separation between $s(f)$ and $\mathsf{pdeg}(f)$. In this direction, the functions on up to 5 variables can be exhaustively searched to obtain all existing combinations. However, for functions on 6 and more variables, an exhaustive search is not possible given the size of search space (2^{64} for $n = 6$, 2^{128} for $n = 7$ and so on), and we instead use the properties of resiliency and dual sensitivity to completely exhaust the case of fully sensitive functions for $n = 6$ and $n = 7$ in terms of obtaining all functions and proving non existence, respectively.

Detailed results for 4 and 5 variables are available in [6]. Let us now look into the case of 6 variable functions, for which we have the best base function for first order sensitivity, the Kushilevitz functions [8, Footnote on p. 560]. The importance of such 6-variable functions from complexity theoretic views are explained in [4], too.

There are total 2^{64} Boolean functions on 6 variables, and checking the resiliency and sensitivity of all possible functions requires computational resources that is unattainable. We instead use properties of dual sensitivity and resiliency to obtain all possible $[6, 1, 2]$-functions by concatenating the truth tables of two 5 variable functions. Any 6 variable function f can be written as $f(x_1, \ldots, x_6) = (1 \oplus x_6)f_2(x_1, \ldots, x_5) \oplus x_6 f_2(x_1, \ldots, x_5)$ Where f_1 and f_2 are functions on 5 variables. Then we have the following constraints on the properties of f.

1. If f is 2-resilient then either both f_1 and f_2 are 1-resilient or both are 2-resilient [5].

2. If f is fully dual sensitive, then at least one of f_1 and f_2 are fully dual sensitive. This is easy to see as if neither f_1 nor f_2 are dual sensitive then there is no input point for which the whole function can have full dual sensitivity.

Now we have approximately 2^{13} many $[5,1,1]$-functions and approximately 2^{18} many $[5,0,1]$-functions, which reduces the effective search space to approximately 2^{33} from the naive 2^{64}. Using these constraints we get the full characterization of $[6,-,2]$-functions, which was not previously reported.

- We find that there are $33632(\approx 2^{15.03})$ many $[6,1,2]$-functions. Here it should be noted that the dual of any such function is a $(6,1,3)$-function. We can use the modified recursive amplification technique of Theorem 2 on all such functions to obtain $(6^u,1,3^u)$-functions, which gives us the best known separation between sensitivity and polynomial degree, same as the function by Kushilevitz [8].
- We also get $192(\approx 2^{7.6})$ many $[6,2,2]$-functions, and this, via Theorem 2 of Section 3, gives us the maximum super-linear separation between number of variables and real polynomial degree in second order sensitive functions, which is $\mathsf{pdeg}(f) = n^{\frac{\log 3}{\log 6}}$, which is also the currently best known separation for first order sensitivity.
  Furthermore, there is no $[6,>2,2]$-function.

2.1 Nonexistence of $(7,1,3)$-functions

The existence or non-existence of a $(7,1,3)$-function is central to understanding the maximum separation between $s(f)$ and $\mathsf{pdeg}(f)$. If there does exist a $(7,1,3)$-function then we can obtain a $(7^u,1,3^u)$-function using the recursive amplification method, which gives $s(f) = \mathsf{pdeg}(f)^{\frac{\log 7}{\log 3}}$, improving on the best known result. However, the total number of functions on 7 variables is 2^{128} and therefore checking all functions for this profile is not possible. In this direction we use a mixed integer linear program to investigate the existence of such a function. Let f be a Boolean function on 7 variables. Then, f has full sensitivity and polynomial degree 3 if and only if there is a vector $\mathbf{x} \in \mathbb{F}_2^7$ such that $f(\mathbf{x}) \oplus (\mathbf{x}^i) = 1$ for every $1 \leq i \leq 7$ and $W_f(\mathbf{u}) = 0$ for every $\mathbf{u} \in \mathbb{F}_2^7$ with Hamming weight no less than 4. For every $\mathbf{x} \in \mathbb{F}_2^7$, by mixed integer linear program method with constraints on the Walsh spectrum, the GUROBI software shows that there is no 7 variable function with full sensitivity and polynomial degree 3. Ranging over all vectors $\mathbf{x}$ in $\mathbb{F}_2^7$, we confirm that there is no such function.

However, it not possible to search for all fully sensitive and higher order sensitive functions on more than 7 variables because of the size of the search space. In this regard we search for $8, 9$ and 10 variables rotation symmetric Boolean functions (RSBFs), which is another cryptographically important class of functions, to obtain fully sensitive (first order sensitive functions) using the least possible polynomial degree (maximum resiliency in the dual function). The detailed results are available in [6].

Particularly, we check that there does not exist any $[10, 1, 5]$-rotation symmetric function. It should be noted that if we can obtain a $[10, 1, 5]$-function (provided such a function exists) then that would improve on the best known separation between sensitivity and polynomial degree. This is because we can then get a $(10, 1, 4)$-function and then recursively amplify the function using the amplification process described in Theorem 2 to get a $(10^u, 1, 4^u)$-function, thus giving $s(f) = (\mathsf{pdeg}(f))^{\frac{\log 10}{\log 4}} \approx (\mathsf{pdeg}(f))^{1.66}$ and this would be an improvement on the best known result $s(f) = (\mathsf{pdeg}(f))^{\frac{\log 6}{\log 3}} \approx (\mathsf{pdeg}(f))^{1.63}$ starting from 6-variable Kushilevitz functions [8, Footnote on p. 560]. A search on 10-variable functions outside the RSBF class is quite challenging in terms of computational efforts.

3 The recursive amplification method

We have noted that fixing the value of a function corresponding to $n + 1$ input points to make a function fully sensitive, will restrict the polynomial degree to $\Omega(\sqrt{n})$. The best known results in this paradigm is derived through the recursive amplification method, which is also the function composition method. This is a well known technique [7] that is used to obtain super-linear separation between n and $\mathsf{pdeg}(f)$ and is also used to obtain super-linear separation between $s(f)$ and $\mathsf{pdeg}(f)$. In this section we use this technique and obtain the following results:

1. A slight modification of the recursive amplification method to obtain super-linear separation between $s(f)$ and $\mathsf{pdeg}(f)$ by starting from any candidate base function.
2. We build highly resilient functions with good nonlinearity, $\mathcal{O}(n)$ circuit size and $\mathcal{O}(\log n)$ circuit depth.
3. We obtain super-linear separation between number of variables(n) and polynomial degree ($\mathsf{pdeg}(f)$) for functions with constant order sensitivity.

Recursive amplification was used to obtain the largest known separation between sensitivity and polynomial degree of Boolean functions [2,7,8], as well as the first example of separation between exact quantum query complexity and deterministic query complexity [1], among other separation results.

The basic construction of [7] is as follows. Let f be a function on d variables $x_1, x_2, \ldots, x_d$ with polynomial degree p. Then the recursive amplification method generates the function f^u on d^u variables as: $f^1 = f$, and

$$f^{i+1}\left(x_1, \ldots, x_{d^{i+1}}\right) = f\left(f^i\left(x_1, \ldots, x_{d^i}\right), \ldots, f^i\left(x_{(d-1)d^i+1}, \ldots, x_{d^{i+1}}\right)\right). \quad (1)$$

One may note that for any starting f, we have $\mathsf{pdeg}(f^u) = p^u$. Thus if the sensitivity also gets amplified, we could start with any d variable function with and obtain f^u with super-linear $s(f^u) - \mathsf{pdeg}(f^u)$ whenever $s(f) > \mathsf{pdeg}(f)$. However, sensitivity is not always amplified in the similar manner, and $s(f^u)$ can be arbitrarily low. To this end, we propose a construction so that we can get super-linear separation between $s()$ and $\mathsf{pdeg}()$ starting from any function.

Furthermore, the results also follow for higher-order sensitivity. Our construction and its proof are presented below. One may note the generalization of our technique over the existing idea in [7], as explained above in (1).

Theorem 2. *Let f be a d-variable k-th order sensitive function with $\mathsf{pdeg}(f_1) = p$ and $\mathbf{y} = (y_1, y_2, \ldots, y_d)$ being one input with respect to which the function exhibits k-th order sensitivity. We define the function f^u on d^u variables such that:*

1. $\mathbf{y}^1 = \mathbf{y}$, *and* $\mathbf{y}^i \in \mathbb{F}_2^{d^i}$ *is obtained by concatenating d copies of* $\mathbf{y}^{i-1}$;
2. $f^1 = f$, *and* $\mu_{i-1} = f^{i-1}(\mathbf{y}^{i-1})$;
3. $f^i = f\Big(f^{i-1}(x_1, \ldots, x_{d^{i-1}}) \oplus \mu_{i-1} \oplus y_1, \ldots, f^{i-1}(x_{jd^{i-1}+1}, \ldots, x_{(j+1)d^{i-1}}) \oplus \mu_{i-1} \oplus y_j, \ldots, f^{i-1}(x_{(d-1)d^{i-1}+1}, \ldots, x_{d^i}) \oplus \mu_{i-1} \oplus y_d\Big)$.

Then f^u is a k-th order sensitive function on $n = d^u$ variables and $\mathsf{pdeg}(f) = p^u$ with k-th order sensitivity achieved at the input point $\mathbf{y}^u$.

Proof. Here, we have the base function f on d variables. Let us denote by $[\mathbf{x}]_k$ any input point that can be obtained by flipping at least one and at most k bits of $\mathbf{x} \in \mathbb{F}_2^n$. Thus if a function f is k-th order sensitive at the point $\mathbf{y}$ then $f([\mathbf{y}]_k) = \overline{f(\mathbf{y})}$ by definition. We now prove the result using induction on u. The result holds for $u = 1$ by definition. Assume the result holds for $u-1$ and we need to show that the function f^u has k-th order sensitivity at $\mathbf{y}^u$. The value of the function at $\mathbf{y}^u$ is $f^u(\mathbf{y}^u) = f\Big(f^{u-1}(\mathbf{y}^{u-1}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus y_1, \ldots, f^{u-1}(\mathbf{y}^{u-1}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus y_d\Big) = f(\mathbf{y})$.

Let us now select any $i \leq k$ variables whose value we wish to flip resulting in an input point of the form $[\mathbf{y}^u]_k$. We define the d tuple $S = (s_1, s_2, \ldots, s_d)$ where s_i denotes the number of bits to be flipped between $x_{(i-1)d^{u-1}+1}$ and $x_{id^{u-1}}$. Thus $0 \leq s_i \leq k, \forall i$. If $s_i = 0$ then $f^{u-1}(x_{id^{u-1}+1}, \ldots, x_{(i+1)d^{u-1}}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus a_i = f^{u-1}(\mathbf{y}^{u-1}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus a_i = a_i$.

If $1 \leq s_i \leq k$ then $f^{u-1}(x_{id^{u-1}+1}, \ldots, x_{(i+1)d^{u-1}}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus a_i = f^{u-1}([\mathbf{y}^{u-1}]_k) \oplus f^{u-1}(y^{u-1}) \oplus a_i = \overline{a_i}$.

The number of nonzero values in S are at most k, which would change at most k of the d points y_i in the base function's input to $\overline{y_i}$ and result in an input to f of the form of $f([y]_k)$. Thus for the function f^u we have $f^u([\mathbf{y}^u]_k) = f([\mathbf{y}]_k) = \overline{f(\mathbf{y})} = \overline{f^u(\mathbf{y}^u)}$.

The polynomial degree result holds from the basic definition of recursive amplification as $\mathsf{pdeg}(f) = \mathsf{pdeg}(\overline{f})$ and this completes the proof. □

As an example of Theorem 2, Figure 1 provides an outline of building a 9-variable function using 4 instances of the circuit C_f corresponding to a 3-variable function f.

Theorem 3. *Given a function f on d variables with $\mathsf{pdeg}(f) = p < d$ we can obtain a function on g^u on $n = d^u$ variables with resiliency $n - n^{\frac{\log p}{\log d}} - 1$ such that there is a circuit of linear size and logarithmic depth in n considering f can be implemented with constant resources.*

Proof. Following Theorem 2, we denote by f^u the function obtained through recursively amplifying the function f, u times, which gives us a function on d^u variables with $\mathsf{pdeg}(f^u) = n^{\frac{\log p}{\log d}}$. Let us assume the circuit corresponding to the base function on d variables consists of some c_d gates and has a depth of t_d. This circuit takes in d input variable bits and outputs a single bit. Then the circuit corresponding to f^u can be built using the circuits for f in a layered manner in the following way:

- In the first layer there are total d^{u-1} circuits each taking in d variables each as input bits.
- In the i-th layer there d^{u-i-1} circuits each taking as input d of the d^{u-i} output bits from the previous layer.
- The final layer contains a single circuit, whose output is the output of the final function.

Then the total number of circuit instances of f to be used is $\sum_{i=0}^{u-1} d^i = \frac{d^u-1}{u-1}$ and the gate count is $c_d \times \frac{d^u-1}{u-1} = \mathcal{O}(d^u) = \mathcal{O}(n)$. Moreover, the depth of this circuit is $u \times t_d$ as the circuit for f is set up in u layers, which gives as a circuit for f^u with $\mathcal{O}(\log_d n)$ depth.

Now if we XOR the parity of all the input bits to this output we obtain a $n - n^{\frac{\log p}{\log d}} - 1$ function g^u via the resiliency-polynomial degree connection. That is, g^u is the dual of f^u where f^u is the function on n^u variables obtained by recursively amplifying f. The parity of the input bits can be simply obtained in parallel using n gates and $\log n$ depth, which gives us the result. □

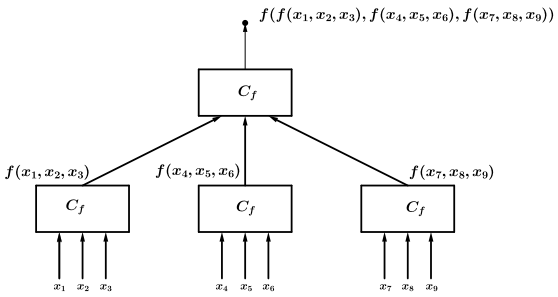

Fig. 1: Example of a circuit corresponding to recursive amplification.

Building efficient low depth circuits for cryptographically important functions with large number of input variables is a challenging problem. In this regard one may refer to an earlier construction in [10]. That work [10] shows how to start with an m-resilient function on some d variables and generate an $(m+u)$-resilient function on $n = (d+u)$ variables that requires $\mathcal{O}(u)$ depth, which is

effectively $\mathcal{O}(n)$ as d is constant for any given construction. Improving on this, we have the result as in Theorem 3. We refer to [6] for elaborate discussion on the nonlinearity lower bounds one can derive for these highly resilient functions, along with algebraic degree-resiliency trade-offs.

Finally, we note that one can obtain super-linear separation between n and $\mathsf{pdeg}(f)$ for functions with any constant order sensitivity. The details are available in [6]. This raises the interesting problem of understanding the nature of the maximum super-linear separation possible between n and $\mathsf{pdeg}(f)$ with increasing, constant order sensitivity k.

References

1. A. Ambainis, *Superlinear advantage for exact quantum algorithms*, Proceedings of the forty-fifth annual ACM symposium on Theory of Computing (STOC'13), pp. 891–900, 2013.
2. H. Buhrman and R. De Wolf, *Complexity measures and decision tree complexity: a survey*, Theoretical Computer Science, 288:1 (2002), 21–43.
3. C. Carlet and P. Charpin. *Cubic Boolean functions with highest resiliency*, IEEE Transactions on Information Theory, Vol. 51, No 2, pp. 562-571, February 2005.
4. P. Hatami, R. Kulkarni and D. Pankratov, *Variations on the Sensitivity Conjecture*, `https://arxiv.org/abs/1011.0354`, 2010.
5. S. Maitra and P. Sarkar, *Highly Nonlinear Resilient Functions Optimizing Siegenthaler's Inequality*, Advances in Cryptology, CRYPTO '99. Lecture Notes in Computer Science, vol 1666. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48405-1_13
6. S. Maitra, C. S. Mukherjee, P. Stănică and D. Tang, *On Boolean Functions with Low Polynomial Degree and Higher Order Sensitivity*, `https://arxiv.org/abs/2107.11205`, 2021.
7. N. Nisan and M. Szegedy, *On the degree of Boolean functions as real polynomials*, Comput. Complexity 4 (1994), 301–313. First published in STOC 1992. Available at `https://link.springer.com/article/10.1007%2FBF01263419`
8. N. Nisan and A. Wigderson, *On rank vs. communication complexity*, Combinatorica 15 (1995), 557–565.
9. R. O'Donnell, Analysis of Boolean functions, Cambridge University Press, 2014.
10. P. Sarkar and S. Maitra, *Efficient implementation of cryptographically useful "large" Boolean functions*, IEEE Transactions on Computers 52:4 (2003), 410–417.