Walsh-Hadamard Transform and Cryptographic Applications in Bias Computing *

Yi Lu[1†] and Yvo Desmedt[2,3]

[1] National Research Center of Fundamental Software, Beijing, China
[2] The University of Texas at Dallas, Richardson, TX, USA
[3] University College London, London, UK

Abstract. Walsh-Hadamard transform is used in a wide variety of scientific and engineering applications, including bent functions and cryptanalytic optimization techniques in cryptography. In linear cryptanalysis, it is a key question to find a good linear approximation, which holds with probability $(1 + d)/2$ and the bias d is large in absolute value. Lu and Desmedt (2011) take a step toward answering this key question in a more generalized setting and initiate the work on the generalized bias problem with linearly-dependent inputs. In this paper, we give fully extended results. Deep insights on assumptions behind the problem are given. We take an information-theoretic approach to show that our bias problem assumes the setting of the maximum input entropy subject to the input constraint. By means of Walsh transform, the bias can be expressed in a simple form. It incorporates Piling-up lemma as a special case. Secondly, as application, we answer a long-standing open problem in correlation attacks on combiners with memory. We give a closed-form exact solution for the correlation involving the multiple polynomial of any weight *for the first time*. We also give Walsh analysis for numerical approximation. An interesting bias phenomenon is uncovered, i.e., for even and odd weight of the polynomial, the correlation behaves differently. Thirdly, we introduce the notion of weakly biased distribution, and study bias approximation for a more general case by Walsh analysis. We show that for weakly biased distribution, Piling-up lemma is still valid. Our work shows that Walsh analysis is useful and effective to a broad class of cryptanalysis problems.
Keywords. (Sparse) Walsh-Hadamard Transform, Linear cryptanalysis, Bias analysis, Maximum entropy principle, Piling-up lemma.

1 Introduction

Walsh-Hadamard transform is powerful in a variety of applications in image and video coding, speech processing, data compression, communications [10, 41] (including the classic application bent functions [38] used

* Part of this material appeared in the proceedings [21]. In this paper, we give more theoretical results, new applications and new analytical results.

† E-mail: dr.yi.lu@ieee.org

in cryptography). Most recently, it is demonstrated that Walsh transform[4] plays an essential role in a generic sampling problem for which we know nothing about the signal source *a priori* and the sample size is bounded [20]. As this transform only performs addition and subtraction, it is extremely easy for digital implementation. Similar to the Fast Fourier Transform, it has a fast and efficient algorithm - Fast Walsh Transform (FWT). For an array of size N, where N is an integer power of two, the total number of arithmetic operations to compute FWT is $N \log_2 N$. The simplicity of Walsh transform intrigued research efforts in cryptanalytic techniques for symmetric crypto-systems two decades ago [3]. Interestingly, due to the hardware bottleneck, the computing technology before millennium thwarted applicability of Walsh transform to practical cryptanalysis. It is not until the beginning of new millennium that it has become feasible to do giga-scale Walsh-Hadamard transforms on a single PC. The topic of Walsh transform in practical cryptanalysis emerges since then (cf. [6, 7, 22, 23]).

To analyze the security of symmetric crypto-systems, there exist two mainstream generic techniques, i.e., differential cryptanalysis and linear cryptanalysis (cf. [31]). Linear cryptanalysis was invented by Matsui [25] for the 64-bit block cipher Data Encryption Standard (DES). It proves widely applicable to both block ciphers and stream ciphers. The basic idea is to find a linear approximation for (part of) the symmetric crypto-system, which holds with probability $(1 + d)/2$ for the correct guess on the key bit and probability $1/2$ for all the wrong guesses. The critical parameter d (i.e., the bias) affects both the data and time complexities. It is known that the data complexity needs to be on the order of $1/d^2$ for a high probability of success. Therefore, the key question in linear cryptanalysis is to find a good linear approximation, which holds with largebias d. Herein, we loosely say that the bias d is large if $|d|$ is large. It is not a trivial question, because of the fact[5] that there exists a large gap between the sizes of crypto-systems and those of the core functions which can be constructed with strong cryptographic strength. Further, as pointed out in [20], this key question is connected with the famous

[4] Throughout the paper, we occasionally abuse the use of Walsh-Hadamard transform by Walsh transform for short.

[5] On one hand, the size of internal states of crypto-systems evolves from the traditional 64 bits, to the less common 128 bits, the common 256 bits and the emerging 512 bits or more nowadays. On the other hand, the design of cryptographically strong functions targets at the main building blocks of the crypto-systems, which have small or medium sizes.

problem of sparse Walsh-Hadamard Transform [5, 16, 39] in the signal processing domain.

It is known that Walsh-Hadamard transform is useful in regular bias computing in the setting of uniformly-distributed inputs. In [21], Lu and Desmedt take a step toward answering this key question in a more generalized setting. Suppose[6] that the compound function consists of the core functions $F_1, \ldots, F_k$ (for fixed k), which are defined over the same space of modest size, e.g., the binary vector space of 32 bits. Assuming that the inputs are all independent, Maximov and Johansson showed that certain class of large distributions (of the compound function) can be efficiently computed by transform domain analysis [26]. In practical crypto-systems, by the design principle of confusion, the inputs, though random and uniformly distributed individually, are jointly dependent in a rather complicated manner. It remains an open question whether or not one can perform the analysis on the compound function *without* the central independence assumption. This initiates the generalized bias problem [21] with linearly-dependent inputs, which might have potentially large state space. Suppose the compound Boolean function $f_1(a_1) \oplus \cdots \oplus f_k(a_k)$, where f_i is derived[7] from F_i. Assume that the sum (modulo 2) of all inputs follows a known distribution D. By means of Walsh transform, the total bias of the compound function, subject to this input constraint, can be expressed in a simple form. It incorporates Piling-up lemma as one extreme case when D is a uniform distribution. Note that Kukorelly [15] showed that in the context of block ciphers, Piling-up lemma approximation can differ considerably from the real bias.

In this paper, we give fully extended results on [21]. Our new results can be summarized as follows. Firstly, deep insights on assumptions behind the main theorems (i.e., Theorem 1 and Theorem 2) are given in Theorem 3 and Theorem 4. We take an information-theoretic approach to show that our main theorems assume the setting of the maximum entropy for the input variables subject to the linearly-dependent input constraint. In particular, the joint entropy of input variables is maximized if and only if two input requirements are satisfied. Further, we show that under the assumptions of the main theorems, if D is a uniform distribution, the inputs are all independent, and vice versa.

[6] It becomes common practice that several core functions, which are not necessarily identical, are combined together (e.g., by block-wise simple operations), in order to construct a new function with large state space.

[7] The detail of how f_i is derived is not relevant in this paper (and f_i can be derived by just taking the inner product between a fixed binary vector and F_i).

Secondly, as application with identical f_i's and strongly biased D, we answer a long-standing open problem in correlation attacks on combiners with memory (cf. [19, 23, 24, 27]). This is inspired by the work of Molland and Helleseth [32]. They studied a special case for irregular clocked and filtered keystream generators; however, the underlying assumptions on the inputs were not explicitly given in [32]. In our work, we assume a model of generalized combiners with memory by Lu and Vaudenay [24]. Given the correlation between keystream outputs and LFSR[8] outputs, we give a closed-form exact solution for correlation involving the multiple polynomial of any weight *for the first time*. We also give Walsh analysis for numerical approximation. This allows to compare with the Piling-up lemma approximation with respect to the absolute values and the signs respectively. It is no surprise to see that Piling-up lemma approximation could give misleading results. Meanwhile, an interesting bias phenomenon is uncovered, i.e., for even and odd weight of the polynomial, the total correlation behaves differently, which is never the case under the independence assumption. As a practical example, an improved attack [1] on Bluetooth encryption E0 core is given. Due to recent coding theoretic technique [18], a slightly better attack strategy is possible. It leads to the best key-recovery attack results with preprocessing $O(2^{35})$ and runtime $O(2^{36})$ using data $O(2^{33})$.

Thirdly, based on Walsh analysis, our numerical approximation method is extended for a more general D. We introduce the notion of weakly biased distribution. We prove that for a weakly biased D, Piling-up lemma is still valid.

The rest of the paper is organized as follows. In Section 2, we give the basics of Walsh-Hadamard transform. In Section 3, we present the common application of Walsh-Hadamard transform in cryptanalysis, i.e., the regular bias computing problem. Our generalized bias computing problem is studied in Section 4. In Section 5, we give an application to answer an open problem in correlation attacks on combiners with memory. In Section 6, we propose the notion of weakly biased distribution, and give Walsh analysis for the bias approximation. We give concluding remarks in Section 7.

[8] LFSR stands for Linear Feedback Shift Registers, see [31, Sect. 6.2.1, P195-8] for introduction.

2 The Basics of Walsh-Hadamard Transform

Given a real-valued function $f : GF(2)^n \to \mathrm{R}$, which is defined on an n-bit vector, the Walsh-Hadamard transform of f, denoted by $\widehat{f}$, is another real-valued function defined as

$$\widehat{f}(x) = \sum_{y \in GF(2)^n} (-1)^{<x,y>} f(y), \tag{1}$$

for all $x \in GF(2)^n$, where $< x, y >$ denotes the inner product between two n-bit vectors x, y. Below, we give properties of Walsh-Hadamard transform, which will be used later. These properties can be derived from the definition in (1). The reader can refer to [39] for newly-found interesting properties, [20] for new interpretation on energy and power of general discrete statistical signals and [14, P32] for more.

Property 1.

$$\sum_{y \in GF(2)^n} f(y) = \widehat{f}(0) \tag{2}$$

$$\sum_{y \in GF(2)^n} \widehat{f}(y) = 2^n f(0) \tag{3}$$

For Property 1, note that (2) and (3) are duals.

As stated by the second property, Walsh-Hadamard transform can be considered as an involution if we ignore the multiplicative factor 2^n.

Property 2. $\widehat{\widehat{f}}(y) = 2^n f(y),$ for all $y \in GF(2)^n$.

Property 3 (Parseval's Theorem). Given $f : GF(2)^n \to \mathrm{R}$, we always have

$$\sum_{x \in GF(2)^n} \left(\widehat{f}(x)\right)^2 = 2^n \cdot \sum_{x \in GF(2)^n} f^2(x).$$

Given two arbitrary real-valued functions $f, g : GF(2)^n \to \mathrm{R}$, the convolution of f, g, denoted by $f \otimes g$, is another real-valued function defined by

$$f \otimes g(x) = \sum_{y \in GF(2)^n} f(y) \cdot g(x \oplus y), \tag{4}$$

for all $x \in GF(2)^n$. Note that the right side of (4) is equivalent to $\sum_{y \in GF(2)^n} g(y) \cdot f(x \oplus y)$ and convolution is symmetric in f, g, i.e., we have $f \otimes g = g \otimes f$ for any f, g.

Computing the convolution function alone needs operations $O(2^{2n})$ in time domain by definition. Here, we do not take into consideration the time to evaluate the underlying functions. Note that the issue of evaluation is a sampling problem [20] in signal processing.

The following property ensures that this can be done with three times of Walsh-Hadamard transforms, i.e., in time $O(3n \cdot 2^n)$ in transform domain.

Property 4.

$$2^n \cdot (f \otimes g)(x) = \widehat{\widehat{f} \cdot \widehat{g}}(x), \tag{5}$$

for all $x \in GF(2)^n$.

For convolution with three functions $f, g, h : GF(2)^n \to \mathrm{R}$, using the convolution property with two functions, we have $(\widehat{(f \otimes g) \otimes h})(x) = \widehat{(f \otimes g)}(x) \cdot \widehat{h}(x) = \widehat{f}(x) \cdot \widehat{g}(x) \cdot \widehat{h}(x)$ for all x. This can be extended to convolution with multiple functions $f_1, \ldots, f_k : GF(2)^n \to \mathrm{R}$,

$$\widehat{(f_1 \otimes \cdots \otimes f_k)}(x) = \widehat{f_1}(x) \cdot \widehat{f_2}(x) \cdots \widehat{f_k}(x), \tag{6}$$

for all $x \in GF(2)^n$.

3 Common Application in Bias Computing

In design of symmetric crypto-systems, Walsh-Hadamard transform has been a useful tool, which is often associated with bent function [38]. The subject of Walsh-Hadamard transform and bent function has stimulated long-term research efforts (e.g., [4,13,28,33–35]). In cryptanalysis, one of the most common applications of Walsh-Hadamard transform (cf. [3] for another application), which sometimes bears the name of Fourier transform[9], is given below.

Let s be the n-bit output (sub-)string of a target function. Let $f(\cdot)$ be the probability distribution of s, assuming that the input to the target function is random and uniformly distributed. Then, for any n-bit $m \neq 0$, $\widehat{f}(m)$ is the bias of the bit $< m, s >$ and we usually call m the output mask. Here, the bias (also termed as *imbalance* [12] or *normalized correlation* [29]) of a binary random variable $\mathcal{A}$, is defined by $E[(-1)^{\mathcal{A}}]$. Note that $\mathcal{A}$ is called *balanced* if the bias is zero. This property is used

[9] In spite of the similarities and common properties between the two transforms, note that they are derived from two different topologic groups and are not interchangeable in general (see [36]).

as a routine to check for potential weakness of the core functions in a target cryptographic system. That is, it is used to check for existence of any (nonzero) biases for the target function. Once such a bias is found, it is then possible to perform further cryptanalysis to examine the security of the full system. As a matter of fact, trying to find a bias *as large as possible* constitutes one of the main foundations and challenges in linear cryptanalysis.

It is worth pointing out the computational advantage with Walsh-Hadamard transform here. With FWT, we get the biases for all masks (corresponding to all the Walsh coefficients) simultaneously; otherwise, we have to compute the bias for each mask (corresponding to each individual Walsh coefficient) one by one. In next section, we present application of Walsh-Hadamard transform to a class of generalized bias computing problems with linearly-dependent inputs due to Lu and Desmedt [21].

4 Our Generalized Bias Computing Problem

4.1 Our Problem

Given arbitrary $f_1, f_2 : GF(2)^n \to GF(2)$, consider this new target function $f_1(a) \oplus f_2(b)$. We start with the problem of computing its bias, assuming that the inputs a, b are random and independent with uniform distribution. Let d_1, d_2 be the bias of (the output bit of) f_1, f_2, assuming uniformly distributed inputs respectively. Due to independence of inputs, it is known that the bias of the target function is $d_1 \cdot d_2$, because the probability that the target function takes value 0 is $\frac{1+d_1}{2} \cdot \frac{1+d_2}{2} + \frac{1-d_1}{2} \cdot \frac{1-d_2}{2} = \frac{1}{2} + \frac{d_1 d_2}{2}$. It can be easily extended when the target function is composed of an arbitrary number of single functions, assuming that all the inputs are independent. This is the famous Piling-up Lemma [25]. Nevertheless, this is an *idealized* assumption to assume the inputs involved are all independent always.

In practice, it is often the case that the inputs, though random and uniformly distributed individually, are often jointly dependent in a rather complicated manner. In the recent work of Lu and Desmedt [21], an important step is taken to formally study the bias problem of this compound function for a simple form of input dependence, i.e., when the inputs are *linearly dependent*. More specifically, with the additional constraint on the inputs, i.e., the variable of their sum (modulo 2) follows a given distribution, a very simple form of expression can be obtained for the bias of the compound function. Our motivation for this bias problem with

linearly-dependent inputs was drawn from a long-standing open problem in correlation attacks on LFSR-based stream ciphers (see Section 5), i.e., to give a more precise correlation estimate associated with a given multiple polynomial. Note the conventional approach of Piling-up lemma approximation is based on the naive assumption that the inputs are all independent. Meanwhile, our model will also be useful to help analyze large building blocks of symmetric crypto-systems, which exhibit strong linear dependence on inputs.

Theorem 1 (Lu-Desmedt 2011, [21]). *Given $f_1, f_2 : GF(2)^n \to GF(2)$ and a distribution D over $GF(2)^n$, assume that the uniformly distributed n-bit a, b satisfy that 1) a and $a \oplus b$ are independent, and 2) $a \oplus b$ complies with the given distribution D. Then, the bias δ of $f_1(a) \oplus f_2(b)$ can be expressed by*

$$\delta = \frac{1}{2^{2n}} \sum_{x \in GF(2)^n} \widehat{g_1}(x) \cdot \widehat{g_2}(x) \cdot \widehat{D}(x),$$

where $g_1, g_2 : GF(2)^n \to \{1, -1\}$ are derived from f_1, f_2 respectively by $g_1(x) = (-1)^{f_1(x)}$, $g_2(x) = (-1)^{f_2(x)}$.

Theorem 2 extends Theorem 1 to an arbitrary number of Boolean functions over the same binary vector space. It means that we need time $O(kn \cdot 2^n)$ to compute the total bias if all f_i's are distinct, according to Property 4 in Sect. 2. The runtime grows linearly in k and is practical for modest n. In contrast, under the independence assumption, we need time $O(k \cdot 2^n)$ to compute the bias by Piling-up lemma.

Theorem 2 (Lu-Desmedt 2011, [21]). *Given $f_1, f_2, \ldots, f_k : GF(2)^n \to GF(2)$ and a distribution D over $GF(2)^n$, assume that the uniformly distributed n-bit $a_1, a_2, \ldots, a_k$ satisfy that 1) $a_1, a_2, \ldots, a_{k-1}$ and $(a_1 \oplus a_2 \oplus \cdots \oplus a_k)$ are all independent, and 2) $a_1 \oplus a_2 \oplus \cdots \oplus a_k$ complies with the given distribution D. Then, the bias δ of $f_1(a_1) \oplus f_2(a_2) \oplus \cdots \oplus f_k(a_k)$ can be expressed by*

$$\delta = \frac{1}{2^{kn}} \sum_{x \in GF(2)^n} \widehat{g_1}(x) \cdot \widehat{g_2}(x) \cdots \widehat{g_k}(x) \cdot \widehat{D}(x), \tag{7}$$

where $g_i : GF(2)^n \to \{1, -1\}$ is derived from f_i by $g_i(x) = (-1)^{f_i(x)}$ for $i = 1, 2, \ldots, k$.

Proof. For a Boolean function $F : GF(2)^n \to GF(2)$, let d be the bias of $F(x)$ with random and uniformly distributed x. It is easy to see that

$$2^n \cdot d = \sum_{x \in GF(2)^n} (-1)^{F(x)}. \tag{8}$$

From the independence assumption of $a_1, \ldots, a_{k-1}$ and $(a_1 \oplus \cdots \oplus a_k)$ and the uniform distribution assumption of the a_i's, we directly calculate the bias δ,

$$\begin{aligned} & 2^{(k-1)n} \cdot \delta \qquad (9) \\ &= \sum_{a_1} \cdots \sum_{a_{k-1}} \sum_{s} (-1)^{f_1(a_1) \oplus \cdots \oplus f_{k-1}(a_{k-1}) \oplus f_k(s \oplus a_1 \oplus \cdots \oplus a_{k-1})} \cdot D(s) \\ &= \sum_{a_1} \cdots \sum_{a_{k-1}} \sum_{s} g_1(a_1) \cdots g_{k-1}(a_{k-1}) \cdot g_k(s \oplus a_1 \oplus \cdots \oplus a_{k-1}) \cdot D(s) \end{aligned}$$

For any fixed $a_1, \ldots, a_{k-1}$, we know

$$\sum_{s} g_k(s \oplus a_1 \oplus \cdots \oplus a_{k-1}) \cdot D(s) = \sum_{a_k} g_k(a_k) \cdot D(a_1 \oplus \cdots \oplus a_k)$$

always holds. So, we rewrite (9) by

$$\begin{aligned} 2^{(k-1)n} \cdot \delta &= \sum_{a_1} \cdots \sum_{a_k} g_1(a_1) \cdots g_k(a_k) \cdot D(a_1 \oplus \cdots \oplus a_k) \\ &= \sum_{a_1} \cdots \sum_{a_{k-1}} g_1(a_1) \cdots g_{k-1}(a_{k-1}) \cdot (g_k \otimes D)(a_1 \oplus \cdots \oplus a_{k-1}) \\ &= (g_1 \otimes g_2 \otimes \cdots \otimes g_k \otimes D)(0) \qquad (10) \end{aligned}$$

Using Property 1, we have

$$2^n \cdot (g_1 \otimes \cdots \otimes g_k \otimes D)(0) = \sum_{x} (g_1 \otimes \widehat{\cdots \otimes g_k} \otimes D)(x). \tag{11}$$

By convolution property for multiple functions in (6), we know

$$(g_1 \otimes \widehat{\cdots \otimes g_k} \otimes D)(x) = \widehat{g_1}(x) \cdots \widehat{g_k}(x) \cdot \widehat{D}(x), \tag{12}$$

for all x. So, we continue with (11)

$$2^n \cdot (g_1 \otimes \cdots \otimes g_k \otimes D)(0) = \sum_{x} \widehat{g_1}(x) \cdots \widehat{g_k}(x) \cdot \widehat{D}(x). \tag{13}$$

Finally, putting (10) and (13) together, we complete our proof. □

4.2 More Results on Our Assumptions

In this section, we give deep insights on assumptions behind our main theorems (Theorem 1 and Theorem 2) in Sect. 4.1.

Let the set $E = \{a_1, \ldots, a_k\}$ denote the set of all inputs (with fixed k). Let $E_i = E - \{a_i\}$ (for $i = 1, \ldots, k$), denote the subset of E of cardinality $(k-1)$ with only one element a_i absent. Clearly, following our proof, the assumption in Theorem 2 that $a_1, \ldots, a_{k-1}$ and $(a_1 \oplus \cdots \oplus a_k)$ are all independent, can be substituted by the assumption that all the elements of E_i (for fixed i) together with $(a_1 \oplus \cdots \oplus a_k)$ are all independent. However, given a biased D, we can show that $a_1, \ldots, a_k$ *are not all* independent by contradiction. Assume that they *were* all independent otherwise. By the assumption that a_i's are all uniformly distributed respectively, we deduce that $(a_1 \oplus \cdots \oplus a_k)$ is uniformly distributed, i.e., D is a uniform distribution, which leads to contradiction.

Here, we give a toy example to illustrate the assumption setting of our theorems on the inputs a_i's. We consider $k = 2$ and the two input random variables a_1, a_2 are always equal, while the a_i's are uniformly distributed respectively. Thus, the variable of the total sum $a_1 \oplus a_2 = 0$ is actually a constant, and we have $D(0) = 1$. As a result, a_1 and $a_1 \oplus a_2$ are independent, and so are a_2 and $a_1 \oplus a_2$, yet a_1 and a_2 are *not* independent.

As will be shown next, in our generalized bias problem with linearly dependent inputs in Sect. 4.1, our assumptions on the input variables a_i's can be considered *minimal*. Put other way, subject to the constraint that the variable of the total sum (modulo 2), i.e., $a_1 \oplus \cdots \oplus a_k$, follows a given distribution, we can ask to have 1) *the independence assumption:* any subset of $\{a_1, \ldots, a_k\}$ with cardinality $(k-1)$ together with the variable of the total sum are all independent; 2) *the uniform distribution assumption:* the a_i's are uniformly distributed respectively. Additionally, as we have just explained, we *cannot* expand the list of above independence assumption to the set $\{a_1, \ldots, a_k\}$ with(out) the variable of the total sum.

Before we translate the minimalism requirements of the input variables into an information-theoretic result, we first recall some basic definitions of Shannon entropy [8]. The entropy $H(X)$ of a discrete random variable X with alphabet $\mathcal{X}$ and probability mass function $p(x)$ is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x).$$

The joint entropy $H(X_1, \ldots, X_n)$ of a collection of discrete random variables $(X_1, \ldots, X_n)$ with a joint distribution $p(x_1, x_2, \ldots, x_n)$ is defined

by

$$H(X_1, \ldots, X_n) = -\sum_{x_1, x_2, \ldots, x_n} p(x_1, x_2, \ldots, x_n) \log_2 p(x_1, x_2, \ldots, x_n).$$

Theorem 3. *Given k and n, let $A_1, \ldots, A_k$ be n-bit random variables. Let the n-bit random variable S denote the sum (modulo 2) $A_1 \oplus A_2 \oplus \ldots \oplus A_k$ for short. Assume that S is associated with a given probability mass function D over support of n-bit vectors. We let H_D be the Shannon entropy $H(S)$ of S. Then, we have inequality*

$$H(A_1, \ldots, A_k) \leq n(k-1) + H_D,$$

with equality if and only if 1) $A_1, \ldots, A_{k-1}, S$ are all independent, and 2) A_i's are uniformly distributed respectively.

Proof. There exists a one-to-one relation between the two k-tuples a_1, $\ldots$,a_{k-1}, a_k and $a_1, \ldots, a_{k-1}, (a_1 \oplus \cdots \oplus a_k)$. So, by definition of entropy we deduce that

$$\begin{aligned} H(A_1, \ldots, A_k) &= H(A_1, \ldots, A_{k-1}, S) && (14) \\ &\leq H_D + \sum_{i=1}^{k-1} H(A_i) && (15) \\ &\leq H_D + n \cdot (k-1), && (16) \end{aligned}$$

equality in (15) holds if and only if $A_1, A_2, \ldots, A_{k-1}$ and S are all independent, and equality in (16) holds if and if $A_1, A_2, \ldots, A_{k-1}$ are all uniformly distributed. We are left to show that A_k is uniformly distributed to finish our proof. It can be done by replacing $H(A_1, \ldots, A_{k-1}, S)$ in (14) by $H(A_2, \ldots, A_k, S)$ and repeating above proof. □

Remark 1. In Theorem 3, each random variable A_i corresponds to the input a_i in Theorem 2. Theorem 3 tells that, subject to the linear dependency constraint on the input variables, the joint entropy of input variables is maximized if and only if the aforementioned two input requirements (i.e., the independence assumption and the uniform distribution assumption) are met. In the spirit of the maximum entropy principle[10], our generalized bias problem assumes the setting of the maximum entropy for the inputs, as shown in Theorem 1 and Theorem 2.

[10] It originated in statistical mechanics in the nineteenth century and has been advocated for use in a broader context (cf. [8, Chapter 12, P425]).

Below, we give further results on the input assumptions behind Theorem 1 and Theorem 2.

Theorem 4. *Given k and n, let $A_1, \ldots, A_k$ be n-bit random variables. Let the n-bit random variable S denote the sum (modulo 2) $A_1 \oplus A_2 \oplus \ldots \oplus A_k$ for short. Assume that 1) S is associated with a given probability mass function D over support of n-bit vectors, 2) $A_1, \ldots, A_{k-1}, S$ are all independent, and 3) A_i's are uniformly distributed respectively. Then, if D is a uniform distribution, we have that $A_1, \ldots, A_k$ are all independent, and vice versa.*

Proof. We prove the first part of the results. Assume that D is a uniform distribution. We have $H_D = n$. By Theorem 3, we deduce that $H(A_1, \ldots, A_k) = n(k-1) + H_D = nk$. As A_i's are uniformly distributed respectively, we conclude that $A_1, \ldots, A_k$ are all independent by property of entropy.

The equivalent of the opposite (i.e., if D is biased, then, $A_1, \ldots, A_k$ are not all independent), is proved in Sect. 4.2. □

Next, we examine two extreme cases for our generalized bias problem with special D.

4.3 Two Extreme Cases

Case One: D is a uniform distribution.

Property 5. If D is a uniform distribution, then, we have

$$\delta = \frac{1}{2^{kn}} \widehat{g_1}(0) \cdot \widehat{g_2}(0) \cdots \widehat{g_k}(0). \tag{17}$$

Remark 2. Let δ_i denote the bias of f_i. By (8) and Property 1, we have $\delta_i = \frac{1}{2^n} \sum_x g_i(x) = \frac{1}{2^n} \widehat{g_i}(0)$. By (17), we deduce $\delta = \delta_1 \cdots \delta_k$. On the other hand, because D is a uniform distribution, we know that a_i's are all independent and uniformly distributed by Theorem 4. Piling-up lemma directly tells us that $\delta = \delta_1 \cdot \delta_2 \cdots \delta_k$. Consequently, Piling-up lemma is a very special case of our result when D is a uniform distribution.

Case Two: D is a delta function[11]. Here, we examine δ when the inputs are subject to the constraint of a $GF(2)$-linear relation[12], i.e., $a_1 \oplus a_2 \oplus \cdots \oplus a_k = constant$.

[11] The delta function is defined by $\delta(x-n) = 1$ if $x = n$ and $\delta(x-n) = 0$ otherwise, for the discrete x.

[12] Because of this special case, we refer to the general case as linearly-dependent inputs.

Property 6. If $D(a_0) = 1$ for a fixed n-bit a_0, then, we have

$$\delta = \frac{1}{2^{kn}}\Big(\sum_{\substack{x\in GF(2)^n:\\ <a_0,x>=0}} \widehat{g_1}(x)\cdots\widehat{g_k}(x) - \sum_{\substack{x\in GF(2)^n:\\ <a_0,x>=1}} \widehat{g_1}(x)\cdots\widehat{g_k}(x)\Big). \tag{18}$$

Further, if $a_0 = 0$, we have

$$\delta = \frac{1}{2^{kn}} \sum_{x\in GF(2)^n} \widehat{g_1}(x)\cdots\widehat{g_k}(x). \tag{19}$$

Note that the result of Molland and Helleseth [32] corresponds to $f_1 = f_2 = \cdots = f_k$ and $D(0) = 1$ here. In next section, we discuss the application of Theorem 2 with identical f_i's and a strongly biased D.

5 The Open Problem of Precise Correlation Estimation

We aim to answer a long-standing open problem in correlation attacks on LFSR-based stream ciphers[13], i.e., give a more precise correlation estimate for a given multiple polynomial. We refer to [30] for a recent review on correlation attacks on stream ciphers.

We assume a model of generalized combiners with memory by Lu and Vaudenay [24, Sect. 2]. The combiner consists of k regularly-clocked LFSRs with m-bit memory ($m \geq 0$). The keystream output of the combiner is generated by $z_t = y_t \oplus u_t$, for $t \geq 0$, where $y_t = x_t^1 \oplus \cdots \oplus x_t^k$ is the sum (modulo 2) of the outputs (denoted by x_t^i for $i = 1, \ldots, k$) of LFSRs, and u_t is one bit generated[14] by the internal state. Further, y_t can be produced by the output of a single equivalent LFSR (see [17, Theorem 6.57, P218]) and we denote its feedback polynomial by $g_0(x)$ with degree L. Without loss of generality, assume that there exists known correlation[15] (called bias in our context) δ_0 with mask $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_r)_2$ (in binary form) for

$$< \gamma, u_{t_0} u_{t_0+1} \ldots u_{t_0+r} >, \tag{20}$$

for all t_0. Let the normalized multiple polynomial of $g_0(x)$ of low weight w with degree d, be denoted by, $Q(x) = \sum_{i=1}^{w} x^{q_i}$ with $0 = q_1 < q_2 < \cdots < q_w = d$. We now state the open problem as follows.

[13] see [31, Sect. 6.3, P203-5] for a review on LFSR-based stream ciphers.

[14] The details of how u_t is generated are not relevant in our context and we omit here.

[15] In the context of correlation attacks on LFSR-based stream ciphers, we often say that there exists correlation δ_0 with mask γ between keystream outputs $\{z_t\}$ and the equivalent LFSR outputs $\{y_t\}$, i.e., $< \gamma, z_{t_0} z_{t_0+1} \ldots z_{t_0+r} > \oplus < \gamma, y_{t_0} y_{t_0+1} \ldots y_{t_0+r} >$, which is equal to $< \gamma, u_{t_0} u_{t_0+1} \ldots u_{t_0+r} >$, has bias δ_0.

Open Problem: subject to the constraint on the LFSR outputs,

$$\oplus_{i=1}^{w} x_{t+q_i}^{j}, \oplus_{i=1}^{w} x_{t+q_i+1}^{j}, \ldots, \oplus_{i=1}^{w} x_{t+q_i+r}^{j} = \mathbf{0}, \tag{21}$$

for all t and $j = 1, \ldots, k$, where $\mathbf{0}$ denotes the zero vector, what is the bias δ for

$$\begin{aligned} &\oplus_{i=1}^{w} < \gamma, z_{t_0+q_i} z_{t_0+q_i+1} \ldots z_{t_0+q_i+r} > \\ &= \oplus_{i=1}^{w} < \gamma, u_{t_0+q_i} u_{t_0+q_i+1} \ldots u_{t_0+q_i+r} >, \end{aligned} \tag{22}$$

given t_0?

Remark 3. Based on the convenient assumption that the w addends on the right side of (22) are all independent, Piling-up lemma yields the estimate

$$\delta \approx (\delta_0)^w. \tag{23}$$

Unfortunately, this independence assumption does not hold due to the effect of the multiple polynomial in (21).

5.1 Our Closed-Form Exact Solution

Fix t_0 and r, let F_i (for $i = 1, \ldots, w$) be the function that outputs the sequence $u_{t_0+q_i} u_{t_0+q_i+1} \ldots u_{t_0+q_i+r}$. The n-bit input (denoted by a_i) of F_i consists of the m-bit memory at time $t_0 + q_i$ and LFSRs outputs involved. For convenience, we let the least significant m bits of a_i be the memory bits. Given γ with length $r+1$, let $f_i(a_i) =< \gamma, F_i(a_i) >$. Thus, we see that (22) is equal to $f_1(a_1) \oplus \cdots \oplus f_w(a_w)$. Before we proceed to calculate the correlation δ for (22), we make a few comments. First, we always have $f_1 = \ldots = f_w$ (denoted by f). Second, for each f_i, the input a_i is uniformly distributed. Third, by (21), we deduce that the sum (modulo 2) of a_i's assumes a special distribution D that satisfies:

$$D(0) = D(1) = \ldots = D(2^m - 1) = \frac{1}{2^m}. \tag{24}$$

Let $LSB_m(x) = 0$ denote that the least significant m bits of x are all zeros. By (24), we have

$$\widehat{D}(x) = \begin{cases} 1 & \text{if } LSB_m(x) = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Fourthly, the naive independence assumption on $a_1, \ldots, a_w$ no longer holds due to (24); yet, we assume that $a_1, \ldots, a_{w-1}$ and the sum of a_i's

are all independent. Therefore, the open problem of correlation estimate fits well into our generalized bias problem. We apply Theorem 2 with $g_1 = \ldots = g_w = (-1)^f$ (denoted by g), and obtain a closed-form exact solution δ for (22):

$$\delta = \frac{1}{2^{wn}} \sum_{\substack{x \in GF(2)^n: \\ LSB_m(x)=0}} \Big(\widehat{g}(x)\Big)^w \tag{26}$$

5.2 Detailed Numerical Analysis

Define $\alpha = \max_{LSB_m(x)=0} \frac{|\widehat{g}(x)|}{2^n}$. Define the disjoint set S_+, S_- by

$$S_+ = \{x : LSB_m(x) = 0, \text{ and } \widehat{g}(x) = +\alpha \cdot 2^n\} \tag{27}$$
$$S_- = \{x : LSB_m(x) = 0, \text{ and } \widehat{g}(x) = -\alpha \cdot 2^n\} \tag{28}$$

It is clear that $(|S_+| + |S_-|) \leq 2^{n-m}$. According to (7), we approximate δ by δ_{appx},

$$\delta \approx \delta_{appx} = \begin{cases} (|S_+| + |S_-|) \cdot \alpha^w, & \text{(for even } w\text{)} \\ (|S_+| - |S_-|) \cdot \alpha^w, & \text{(for odd } w\text{)} \end{cases} \tag{29}$$

We now compare the the bias δ with the Piling-up lemma estimate δ', which satisfies $\delta' = (\delta_0)^w = \big(\frac{\widehat{g}(0)}{2^n}\big)^w$. In general, we have $\alpha \neq \frac{|\widehat{g}(0)|}{2^n}$, that is, $\alpha > \frac{|\widehat{g}(0)|}{2^n}$. Thus, it results in $\alpha^w > |\delta'|$. So, if w is even, we have $|\delta| > |\delta'|$ generally; if w is odd and $|S_+| \neq |S_-|$, we have $|\delta| > |\delta'|$ generally. Additionally, for odd w and $|S_+| = |S_-|$, we have $\delta = 0 < |\delta'|$, which indicates that sometimes it is too optimistic to use Piling-up lemma.

With respect to the signs of δ, δ', clearly, both are non-negative for even w. For odd w, the signs of $\delta', \widehat{g}(0)$ are the same. By (29), dependent on the sign of $|S_+| - |S_-|$, the signs of δ, α are not necessarily the same, and the signs of $\delta, |S_+| - |S_-|$ are the same actually. This implies that for odd w, the signs of δ, δ' may not be the same, and it is possible that they have distinct signs. Here, we observe an interesting phenomenon on δ, δ' (with respect to the magnitudes and/or the signs) for odd w, i.e., Piling-up lemma approximation could give *misleading* results sometimes.

From our comparison, we see that for even w, δ is stable and $|\delta| \geq |\delta'|$, i.e., Piling-up lemma underestimates the result. For odd w, it is possible that $|\delta| < |\delta'|$ (e.g., $\delta = 0$), and the signs of δ, δ' are not related. Consequently, even w is desirable for the purpose of finding a larger bias δ.

5.3 Practical Example: Bluetooth E0 Combiner

Our analysis technique is applied to Bluetooth E0 combiner with 4-bit memory [1]. It is known (cf. [19, 23, 24]) that of all mask γ's up to 26 bits, the maximum of the bias $|\delta_0|$ for (20), is $2^{-3.3}$, which is obtained with two choices of γ, i.e., $(11111)_2$ and $(100001)_2$. Note that the binary function g as well as f are associated with a fixed γ. For each mask γ of up to 8 bits, we compute $\widehat{g}(x)$ subject to the constraint $LSB_4(x) = 0$. Interestingly, our computations find that of all these γ's, the maximum of α, is also $2^{-3.3}$, and it is achieved with four choices of γ, i.e., $(11111)_2$, $(100001)_2$, $(10111)_2$, $(110001)_2$. In Table 1, for each of the four γ's, we give the detailed analysis results on $\widehat{g}(\cdot)$, where '-' denotes bias 0. For the two known masks, i.e., $(11111)_2$, $(100001)_2$, we have equality $\alpha = |\delta_0|$, which is not typical in general as we have just mentioned. For the other two new masks, we see that $\alpha \gg |\delta_0|$; in particular, for $\gamma = (110001)_2$, we have $\delta_0 = 0$, yet it is remarkable to have $\alpha = 2^{-3.3}$, which is among one of the four largest. Also, subject to the constraint $LSB_4(x) = 0$, we have the sum $|S_+| + |S_-| = 8$ for each of the four masks.

Table 1. Analysis results on $\widehat{g}(\cdot)$ with $\gamma = (11111)_2$, $(100001)_2$, $(10111)_2$, $(110001)_2$

γ	$\delta_0 = \frac{\widehat{g}(0)}{2^n}$	α	$\lvert S_+\rvert$	$\lvert S_-\rvert$
$(11111)_2$	$-2^{-3.3}$	$2^{-3.3}$	6	2
$(100001)_2$	$2^{-3.3}$	$2^{-3.3}$	2	6
$(10111)_2$	-2^{-6}	$2^{-3.3}$	4	4
$(110001)_2$	-	$2^{-3.3}$	4	4

Table 2 to Table 5 compare the exact bias δ_{real} as calculated by Theorem 2, the approximated bias δ_{appx} by (29), and Piling-up lemma approximation δ'. As reference, we give the exact bias for the single function f associated with γ in the column $w = 1$.

With the two known masks, we see that $\delta_{\text{real}} \doteq \delta_{\text{appx}}$ for $w \geq 3$ in Table 2 ($w \geq 4$ in Table 3 resp.); for small w, $|\delta_{\text{real}}|$ is slightly greater than $|\delta_{\text{appx}}|$, because the sum of those addends $\big(\widehat{g}(x)\big)^w$ in (7), which all satisfy $LSB_4(x) = 0$ and $\frac{|\widehat{g}(x)|}{2^n} < \alpha$, is not ignorable. The signs of $\delta_{\text{real}}, \delta_{\text{appx}}$ are always the same. Regarding δ', we notice from Table 2 and Table 3 that

δ' does give the erroneous sign for odd w as we mentioned in Sect. 5.2. Further, as we have $\alpha = |\delta_0|$ here, from the values of $|S_+|, |S_-|$ in Table 1, we can check that $|\delta_{\text{appx}}| = 8 \cdot |\delta'|$ for even w and $|\delta_{\text{appx}}| = 4 \cdot |\delta'|$ for odd w, as shown in Table 2 and Table 3.

For the new mask $(10111)_2$, in Table 4, it is interesting to notice the following new bias phenomenon that is associated with even (or odd) w as discussed in Sect. 5.2. For odd w, the exact bias δ_{real} all vanishes, and our approximation yields the correct estimate because $|S_+| = |S_-|$ by Table 1; for even w, $|\delta_{\text{real}}|$ is not small and is almost the same as for the two known masks with the same w. Similarly as the two known masks, we find that $\delta_{\text{real}} \doteq \delta_{\text{appx}}$ by Table 4. But, unlike the case of the two known masks, as $\alpha \gg |\delta_0|$, it is no surprise to see that $\delta_{\text{real}}, \delta'$ differ significantly.

For the other new mask $(110001)_2$, we observe similar bias phenomenon that the exact bias δ_{real} behaves differently for even w and odd w respectively. That is, for even w, δ_{real} behaves almost the same as in the case of above new mask; for odd w, our approximation estimates that the bias should vanish (i.e., $\delta_{\text{appx}} = 0$) as $|S_+| = |S_-|$, while δ_{real} is not strictly zero, but it decreases much more quickly than in the case of even w. Finally, regardless of the value of $|\delta_0|$, it is remarkable to note that for even w, δ_{real} could be one of the largest, which is counter-intuitive.

Table 2. Comparison of $\delta_{\text{real}}, \delta_{\text{appx}}, \delta'$ for $w = 2, \ldots, 6$ with $\gamma = (11111)_2$

	Ref. Value					
w	(1)	2	3	4	5	6
δ_{real}	$-2^{-3.3}$	2^{-3}	2^{-8}	$2^{-10.5}$	$2^{-14.7}$	2^{-17}
δ_{appx}		$2^{-3.7}$	2^{-8}	$2^{-10.4}$	$2^{-14.7}$	2^{-17}
$\delta' = (\delta_0)^w$		$2^{-6.7}$	-2^{-10}	$2^{-13.4}$	$-2^{-16.7}$	2^{-20}

Based on our detailed analysis, an improved key-recovery attack on E0 core with pre-processing, time and data complexities $O(2^{37})$ was obtained in [21]. Further, using the recent coding theoretic technique [18], the complexities of finding the multiple polynomial of weight 4 can be improved, compared with using the generalized birthday problem [40]. This allows to use a slightly different strategy to improve the attack results, i.e., to recover the 31-bit R_2 first, rather than the shortest 25-bit R_1 as usual (see Appendix for details). Table 6 gives the new results to recover

Table 3. Comparison of $\delta_{\text{real}}, \delta_{\text{appx}}, \delta'$ for $w = 2, \ldots, 6$ with $\gamma = (100001)_2$

	Ref. Value					
w	(1)	2	3	4	5	6
δ_{real}	$2^{-3.3}$	$2^{-2.6}$	-2^{-7}	$2^{-10.4}$	$-2^{-14.7}$	2^{-17}
δ_{appx}		$2^{-3.7}$	-2^{-8}	$2^{-10.4}$	$-2^{-14.7}$	2^{-17}
$\delta' = (\delta_0)^w$		$2^{-6.7}$	2^{-10}	$2^{-13.4}$	$2^{-16.7}$	2^{-20}

Table 4. Comparison of $\delta_{\text{real}}, \delta_{\text{appx}}, \delta'$ for $w = 2, \ldots, 6$ with $\gamma = (10111)_2$

	Ref. Value					
w	(1)	2	3	4	5	6
δ_{real}	-2^{-6}	2^{-3}	–	$2^{-10.2}$	–	2^{-17}
δ_{appx}		$2^{-3.7}$	–	$2^{-10.4}$	–	2^{-17}
$\delta' = (\delta_0)^w$		2^{-12}	-2^{-18}	2^{-24}	-2^{-30}	2^{-36}

the full key, i.e., the 128-bit initial state of the LFSRs (and we omit the ignorable complexities of recovering R_3, R_4 at the last step). We compare the new results with the best previous attacks [21, 23] in Table 7. We comment that the time cost 2^{36} is *optimal* in the sense that the Walsh-Hadamard transform technique [23] gives the lower time bound $\ell \cdot 2^{\ell}$. And it seems that our new results approach the near-optimum bounds on the real security strength of E0 core.

6 Further Discussions

In this section, we extend our approximation idea for δ with special D (in Sect. 5) by Walsh analysis technique to a more general D. We first introduce the concept of *weakly biased distribution* according to the largest Walsh coefficient(s) of the distribution. Given D, define

$$\beta = \max_{x \neq 0} |\widehat{D}(x)|,$$

to be the largest (nontrivial) Walsh coefficient of D and we always have $0 \leq \beta \leq 1$. We assume that $\beta < 1$ holds for a general D throughout

Table 5. Comparison of $\delta_{\text{real}}, \delta_{\text{appx}}, \delta'$ for $w = 2, \ldots, 6$ with $\gamma = (110001)_2$

	Ref. Value					
w	(1)	2	3	4	5	6
δ_{real}	–	$2^{-2.6}$	$2^{-12.1}$	$2^{-10.2}$	$2^{-22.7}$	2^{-17}
δ_{appx}		$2^{-3.7}$	–	$2^{-10.4}$	–	2^{-17}
$\delta' = (\delta_0)^w$		–	–	–	–	–

Table 6. The new key-recovery attack complexities on E0 core

	weight	degree	# effective bits	data $\max(d, n)$	pre-proc.	time	space
R_2	4	2^{33}	2^{27}	2^{33}	2^{36}	2^{36}	2^{33}
R_1	4	2^{24}	2^{27}	2^{27}	2^{27}	2^{30}	2^{27}
total	-	-	-	2^{33}	2^{36}	2^{36}	2^{33}

the rest of this paper (unless otherwise mentioned). We call D a weakly biased distribution, if the following are satisfied:

$$\beta^2 \cdot |\{x \neq 0 : \widehat{D}(x) = \pm\beta\}| \ll 1, \tag{30}$$

$$\sum_{-\beta < \widehat{D}(x) < \beta} \left(\widehat{D}(x)\right)^2 \approx 0. \tag{31}$$

Note that the special D defined by (24) in Sect. 5, is not weakly biased, because the left side of (30) equals 2^{n-m} ($\gg 1$) by (25).

By Theorem 2, we have

$$\delta = \frac{1}{2^{kn}}\Big(\widehat{g_1}(0) \cdots \widehat{g_k}(0) + \sum_{\substack{x \neq 0: \\ |\widehat{D}(x)| > 0}} \widehat{g_1}(x) \cdots \widehat{g_k}(x) \cdot \widehat{D}(x)\Big). \tag{32}$$

Generally speaking, we can approximate δ by the first addend in (32), i.e.,

$$\delta \approx \frac{1}{2^{kn}} \widehat{g_1}(0) \cdots \widehat{g_k}(0) = \delta_1 \cdots \delta_k. \tag{33}$$

From (33), we see that for a weakly biased D, Piling-up lemma approximation is still a valid estimate for our bias problem with linearly-dependent inputs. Below, we give a formal proof for (33).

Table 7. Comparison of our new results with the best previous attacks [21, 23]

attack	pre-proc.	data	time
[23]	2^{37}	2^{39}	2^{39}
[21]	2^{37}	2^{37}	2^{37}
this paper	2^{36}	2^{33}	2^{36}

Proof. Given a distribution D over support of n-bit vectors, we use this result

$$\sum_{x \in GF(2)^n} \Big(D(x)\Big)^2 \geq \frac{1}{2^n}, \tag{34}$$

with equality if and only if D is a uniform distribution. We show this by induction. Let y_i denote $D(i)$ for all n-bit vector i. For $n = 1$, it is trivial to see $(y_0 + y_1)^2 \leq 2(y_0^2 + y_1^2)$, with equality if and only if $y_0 = y_1$. For $n = 2$, we have

$$\Big((y_0 + y_1) + (y_2 + y_3)\Big)^2 \leq 2\Big((y_0 + y_1)^2 + (y_2 + y_3)^2\Big) \leq 4(y_0^2 + \ldots + y_3^2),$$

with equality if and only if y_i's are all equal. Similarly, for arbitrary n, we have

$$1 = \Big(\sum_{i=0}^{2^n-1} y_i\Big)^2 \leq 2^n \sum_{i=0}^{2^n-1} y_i^2,$$

with equality if and only if y_i's are equal. Thus, it leads to (34).

Next, combining Parseval's theorem (i.e., Property 3) and (34), we have

$$\sum_{x \in GF(2)^n} \Big(\widehat{D}(x)\Big)^2 \geq 1,$$

with equality if and only if D is a uniform distribution. On the other hand, we have

$$\sum \Big(\widehat{D}(x)\Big)^2 \approx 1 + b \cdot \beta^2 \approx 1,$$

by (30) and (31), where $b = |\{x \neq 0 : \widehat{D}(x) = \pm\beta\}|$. And we deduce that D is approximately a uniform distribution. □

Remark 4. The concept of weakly biased distribution implies that the sum of squares of (nontrivial) Walsh coefficients can be approximated by

considering the largest Walsh coefficient(s) only, whose sum is small. In this case, the inputs a_i's in our main theorems can be assumed to be all independent, and we can use Piling-up lemma to approximate the real bias δ. If D is not weakly biased, it is not appropriate to estimate δ by Piling-up lemma.

As a practical example, our generalized bias problem with the weakly biased distribution can be best illuminated by a recent synchronous stream cipher Shannon [37]. It has been designed by Qualcomm according to Profile 1A of ECRYPT call for stream cipher primitives [9]. The internal state uses a single nonlinear feedback shift register. This shift register state at time $t \geq 0$ consists of 16 elements s_{t+i} of 32 bits for $i = 0, \ldots, 15$. The critical observable variable $v \in GF(2)^{32}$ can be summarized by the sum of three independent addends in the following form (cf. [21]),

$$\begin{aligned} v = & \Big(\underbrace{f_1(s_{t+21} \oplus s_{t+22} \oplus K) \oplus f_1(s_{t+25} \oplus s_{t+26} \oplus K)}_{\text{distribution of sum (modulo 2) of inputs } \sim D} \Big) \oplus \\ & \Big(\underbrace{f_2((s_{t+11} \oplus s_{t+24}) \lll 1) \oplus f_2((s_{t+15} \oplus s_{t+28}) \lll 1)}_{\text{distribution of sum (modulo 2) of inputs } \sim D'} \Big) \oplus \\ & \Big(\underbrace{f_2((s_{t+3} \oplus s_{t+16}) \lll 1) \oplus f_2(s_{t+19} \oplus s_{t+32})}_{\text{distribution of sum (modulo 2) of inputs } \sim D''} \Big). \end{aligned} \tag{35}$$

Herein, $f_1, f_2 : GF(2)^{32} \to GF(2)^{32}$ are defined in [37], K is a 32-bit secret constant, and $\widehat{D}, \widehat{D'}, \widehat{D''}$ are defined in [21]. Assuming that each of the six addends in (35) uses independently and uniformly distributed input (i.e., D, D', D'' all *were* uniform distributions), one can perform Walsh-Hadamard transform $\widehat{f_1}, \widehat{f_2}$ and compute $\max_{m \neq 0}(2\widehat{f_1}(m) + 4\widehat{f_2}(m))$. As done in [11], this allows to find out the best output mask(s) m such that the bias δ' for the bit $< m, v >$ in (35) is the largest, i.e., $\delta' = 2^{-56}$ with $m = 0x410a4a1$ in hexadecimal form. With our proposed notion of weakly biased distributions, we have computed $\widehat{D}, \widehat{D'}, \widehat{D''}$ separately. We confirm that D, D', D'' can be considered weakly biased. Consequently, we conclude that Piling-up lemma would produce a fairly good estimate for the total combined bias δ (i.e., $\delta \approx 2^{-56}$) and so the complexity estimate of [11] is valid. And our result is consistent with [21], which directly uses Theorem 1 to calculate the exact bias δ given the mask $m = 0x410a4a1$. We refer to [21] for analysis details on Shannon cipher and Shannon cipher variant based on above critical variable v in (35).

7 Concluding Remarks

We study the generalized bias problem for a broad class of compound functions by the Walsh analysis technique. The compound functions are in the form of the sum (modulo 2) of an arbitrary number of Boolean functions over the same binary vector space. Assume that the input sum follows a given distribution D. We show that in the setting of the maximum input entropy, the bias of the compound function can be expressed in a simple form, due to Walsh-Hadamard transform. We give deep insights on assumptions behind our bias problem. Notably, two extreme cases of the problem are already known. As application, we answer a long-standing open problem in correlation attacks on combiners with memory. Based on Walsh analysis, we uncover a new bias phenomenon. Meanwhile, we also study the bias approximation for a more general case by Walsh analysis. We introduce the concept of weakly biased distribution. It allows to formally show that if D is weakly biased, the Piling-up lemma is still valid.

As Piling-up lemma has been used *almost* exclusively in linear cryptanalysis, it is interesting and useful to compare the real bias δ of our generalized bias problem with Piling-up lemma estimate δ'. We note that when D is not weakly biased, δ can differ significantly from δ' with respect to the magnitudes and/or the signs. First, if $\delta_i = 0$ for some $i \in \{1, \ldots, k\}$, or equivalently f_i is balanced, then, $\delta' = 0$. And we always have $|\delta| \geq |\delta'|$. Secondly, if $\delta_i \neq 0$ for all $i = 1, \ldots, k$, i.e., $\delta' \neq 0$, then, it is possible to have $|\delta| < |\delta'|$. This implies that the independence assumption, which is so often used for convenience, sometimes would *over-estimate* the real bias. This is somehow counter-intuitive. Thirdly, for identical f_i's and even k, we always have $\delta' \geq 0$; in contrast, it is possible to have $\delta < 0$. Fourthly, δ could behave differently for odd and even k respectively (e.g., $\delta = 0$ for odd k and δ is the largest for even k), while we know that it is never the case for δ'. As practical examples with strongly biased D and weakly biased D, our technique has been successfully demonstrated for E0 and Shannon cipher respectively.

Obviously, input dependency can serve as a measure to increase the security of the crypto-systems from complexity-theoretic approach. Our work to consider the linearly-dependent input constraint sheds new light on practical bias analysis. On the other hand, the simplicity of Walsh transform, has stimulated growing research efforts in cryptanalytic optimization techniques (e.g., [6, 7, 22, 23]). Our work shows that Walsh analysis is very useful and effective to a broad class of cryptanalysis prob-

lems. Currently, we are working on practical large dimensional Walsh-Hadamard transform.

Acknowledgements

We are indebted to Prof. Serge Vaudenay for his detailed comments and suggestions to improve the paper.

References

1. Bluetooth^{TM}, Bluetooth Specification (version 2.0 + EDR), `http://www.bluetooth.org`.
2. A. Canteaut, M. Trabbia, Improved fast correlation attacks using parity-check equations of weight 4 and 5, EUROCRYPT 2000, LNCS vol. 1807, pp. 573-588, Springer-Verlag, 2000.
3. F. Chabaud, S. Vaudenay, Links between differential and linear cryptanalysis, EUROCRYPT 1994, LNCS vol. 950, pp. 356-365, Springer-Verlag, 1995.
4. P. Charpin, E. Pasalic, C. Tavernier, On bent and semi-bent quadratic Boolean functions, IEEE Transactions on Information Theory, vol. 51, No. 12, pp. 4286 - 4298, Dec. 2005.
5. X. Chen, D. Guo, Robust Sublinear Complexity Walsh-Hadamard Transform with Arbitrary Sparse Support, IEEE Int. Symp. Information Theory, pp. 2573 - 2577, 2015.
6. B. Collard, F. -X. Standaert, Jean-Jacques Quisquater, Improving the time complexity of Matsui's linear cryptanalysis, ICISC 2007, LNCS vol. 4817, pp. 77-88, Springer-Verlag, 2007.
7. P. Chose, A. Joux, M. Mitton, Fast correlation attacks: an algorithmic point of view, EUROCRYPT 2002, LNCS vol. 2332, pp. 209-221, Springer-Verlag, 2002.
8. T. M. Cover, J. A. Thomas, Elements of Information Theory, John Wiley & Sons, Second Edition (2006)
9. eSTREAM: ECRYPT stream cipher project, `http://www.ecrypt.eu.org/stream/`.
10. S. W. Golomb, G. Gong, Signal Design With Good Correlation: For Wireless Communications, Cryptography and Radar Applications, Cambridge University Press, Cambridge (2005)
11. R. M. Hakala, K. Nyberg, Linear distinguishing attack on Shannon, ACISP 2008, LNCS vol. 5107, pp. 297-305, Springer-Verlag, 2008.
12. C. Harpes, J. L. Massey, Partitioning cryptanalysis, FSE 1997, LNCS vol. 1267, pp. 13-27, Springer-Verlag, 1997.
13. T. Helleseth and A. Kholosha, On generalized bent functions, ITA 2010, pp. 178-183, IEEE, 2010.
14. K. J. Horadam, Hadamard Matrices and Their Applications, Princeton University Press, Princeton and Oxford (2007)
15. Z. Kukorelly, The Piling-up lemma and dependent random variables, IMA 1999, LNCS vol. 1746, pp. 186-190, Springer-Verlag, 1999.
16. X. Li, J. K. Bradley, S. Pawar, K. Ramchandran, SPRIGHT: A Fast and Robust Framework for Sparse Walsh-Hadamard Transform, arXiv:1508.06336, 2015.

17. R. Lidl, H. Niederreiter, Introduction to Finite Fields and Their Applications, Cambridge University Press, Cambridge (1986)
18. C. Löndahl, T. Johansson, Improved algorithms for finding low-weight polynomial multiples in $F_2[x]$ and some cryptographic applications, Designs, Codes and Cryptography, vol. 73, pp. 625-640, Springer (2014)
19. Y. Lu, Applied stream ciphers in mobile communications, Ph.D. Thesis, EPFL, `http://dx.doi.org/10.5075/epfl-thesis-3491` (2006)
20. Y. Lu, Walsh Sampling with Incomplete Noisy Signals, arXiv:1602.00095, 2016.
21. Y. Lu, Y. Desmedt, Bias analysis of a certain problem with applications to E0 and Shannon cipher, ICISC 2010, LNCS vol. 6829, pp. 16-28, Springer-Verlag, 2011.
22. Y. Lu, Y. Desmedt, Improved Davies-Murphy's attack on DES revisited, FPS 2013, LNCS vol. 8352, Springer-Verlag, pp. 264-271, 2014.
23. Y. Lu, S. Vaudenay, Faster correlation attack on Bluetooth keystream generator E0, CRYPTO 2004, LNCS vol. 3152, pp. 407-425, Springer-Verlag, 2004.
24. Y. Lu, S. Vaudenay, Cryptanalysis of an E0-like combiner with memory, Journal of Cryptology, vol. 21, pp. 430-457, Springer (2008)
25. M. Matsui, Linear cryptanalysis method for DES cipher, EUROCRYPT 1993, LNCS vol. 765, pp. 386-397, Springer-Verlag, 1994.
26. A. Maximov, T. Johansson, Fast computation of large distributions and its cryptographic applications, ASIACRYPT 2005, LNCS vol. 3788, pp. 313-332. Springer-Verlag, 2005.
27. W. Meier, O. Staffelbach, Fast correlation attacks on certain stream ciphers, Journal of Cryptology, vol. 1, pp. 159-176, Springer (1989)
28. W. Meier, O. Staffelbach, Nonlinearity criteria for cryptographic functions, EUROCRYPT 1989, LNCS vol. 434, pp. 549-562, Springer-Verlag, 1990.
29. W. Meier, O. Staffelbach, Correlation properties of combiners with memory in stream ciphers, Journal of Cryptology, vol. 5, pp. 67-86, Springer (1992)
30. W. Meier, Fast correlation attacks: methods and countermeasures, FSE 2011, LNCS vol. 6733, pp. 55-67, Springer-Verlag, 2011.
31. A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, Handbook of Applied Cryptography, CRC Press (1996)
32. H. Molland, T. Helleseth, An improved correlation attack against irregular clocked and filtered keystream generators, CRYPTO 2004, LNCS vol. 3152, pp. 373-389, Springer-Verlag, 2004.
33. K. Nyberg, Perfect nonlinear S-boxes, EUROCRYPT 1991, LNCS vol. 547, pp. 378-386, Springer-Verlag, 1991.
34. K. Nyberg, Constructions of Bent functions and difference sets, EUROCRYPT 1990, LNCS vol. 473, pp. 151-160, Springer-Verlag, 1991.
35. J. Olsen, R. Scholtz, L. Welch, Bent-function sequences, IEEE Transactions on Information Theory, IT-28 (6): 858-864, Nov. 1982.
36. J. Pearl, Application of Walsh transform to statistical analysis, IEEE Transactions on Systems, Man, and Cybernetics, SMC-1 (2): 111-119, Apr. 1971.
37. G. Rose, P. Hawkes, M. Paddon, C. McDonald, M. Vries, Design and Primitive Specification for Shannon, Symmetric Cryptography, 2007.
38. O. S. Rothaus, On "Bent" functions, Journal of Combinatorial Theory, Series A 20 (3), pp. 300-305 (1976)
39. R. Scheibler, S. Haghighatshoar, M. Vetterli, A Fast Hadamard Transform for Signals With Sublinear Sparsity in the Transform Domain, IEEE Transactions on Information Theory, vol. 61, No. 4, pp. 2115 - 2132, 2015.

40. D. Wagner, A generalized birthday problem, CRYPTO 2002, LNCS vol. 2442, pp. 288-304, Springer-Verlag, 2002.
41. L. P. Yaroslavsky, Digital Picture Processing - An Introduction, Springer-Verlag, Berlin (1985)
42. B. Zhang, C. Xu, W. Meier, Fast correlation attacks over extension fields, large-unit linear approximation and cryptanalysis of SNOW 2.0, CRYPTO 2015, LNCS vol. 9215, pp. 643-662, Springer, 2015.

Appendix: Intermediate Attack Results on E0 Core

Let $p_i(x)$ be the feedback polynomial of R_i (for $i = 1, \ldots, 4$) with degree $L_1 = 25$, $L_2 = 31$, $L_3 = 33$, $L_4 = 39$ respectively. We use the unusual attack strategy to recover the 31-bit R_2 first, rather than recover the shortest 25-bit R_1. The main reason is that we want to find the multiple polynomial of $p_1(x)p_3(x)p_4(x)$ (which has lower degree $25+33+39 = 97$) with weight $w = 4$, rather than find the multiple polynomial of $p_2(x)p_3(x)p_4(x)$ (which has relatively higher degree $31 + 33 + 39 = 103$) as done in usual. By the recent coding theoretic technique [18], the complexities of finding the multiple polynomial of weight 4 can be improved, compared with using the generalized birthday problem [40]. We thus expect to find the multiple polynomial with minimal degree $2^{97/3} \approx 2^{33}$ with estimated time 2^{36}.

For the data complexity, based on one largest bias $|\delta_0| = 2^{-3.3}$ with $\gamma = (100001)_2$, the basic distinguisher works with the exact bias $\delta = 2^{-10.4}$ when using the multiple polynomial of $p_1(x)p_3(x)p_4(x)$ with weight $w = 4$ by Table 3. Thus, the basic distinguisher needs a total number $n = (4L_2 \ln 2) \cdot \delta^2 \approx 2^{27}$ of effective bits to successfully recover R_2.

After recovering R_2, we aim to reconstruct R_1. We want to find the multiple polynomial of $p_3(x)p_4(x)$ (which has degree $33 + 39 = 72$) with weight $w = 4$. By [18], we expect to find the multiple polynomial with minimal degree $2^{72/3} = 2^{24}$ with estimated effort 2^{27}. Again, the basic distinguisher works with the same bias $\delta = 2^{-10.4}$ when using the multiple polynomial with weight 4. It needs a total number $n = (4L_1 \ln 2) \cdot \delta^2 \approx 2^{27}$ of effective bits to successfully recover R_1. Table 6 summarizes these results to recover R_1, R_2.