

Post-processing functions for a biased physical random number generator

Patrick Lacharme

Université de Toulon

Abstract. A corrector is used to reduce or eliminate statistical weakness of a physical random number generator. A description of linear corrector generalizing post-processing described by M. Dichtl at FSE'07 [4] is introduced. A general formula for non linear corrector, determining the bias and the minimal entropy of the output of a function is given. Finally, a concrete and efficient construction of post-processing function, using resilient functions and cyclic codes, is proposed.

keywords : bias, linear correcting codes, Fourier transform, resilient functions, entropy.

1 Introduction

The scheme of a true random number generator consists of two different parts. The first one is a noise source using a physical non deterministic phenomenon producing a raw binary sequence. The second one is a corrector compressing this sequence in order to provide randomness extraction.¹ At FSE'07, M. Dichtl proposed several true random number generators designed to reduce the bias of the noise source and extract more entropy than known algorithms [4]. He considered that the physical source produces statistically independent bits with constant bias. In his conclusion, the author suggested to extend his work in many directions : compression rates, other input sizes and systematic construction of good post-processing functions.

In this paper, we study the output bias of a function. The same assumptions as in [4] are taken : the input bits of the function are independent and have the same bias. General constructions of functions achieving very good output bias are exposed. Furthermore, these functions are very efficiently implemented in smart-card applications. The output bias of a linear corrector is bounded in Section 2, using linear correcting codes. Section 3 presents the explicit calculation of the output bias of a function with its Fourier transform. Resilient functions are used in Section 4 to construct correctors and Section 5 proposes an estimation of minimal entropy of the output sequence. Section 6 describes a concrete implementation of a mathematical corrector.

¹ True random number generator should not be used for cryptographic purposes without a more complex structure as a pseudo random generator [2].

2 A linear corrector

We consider a physical noise source providing a raw binary sequence. The bits are independent and display a constant bias, equal to $e/2$, with $0 \leq e \leq 1$. The bias of any bit x is given by the formula $|P(x = 0) - 1/2|$. This assumption is taken in order to get a simple formula and to compare our correctors with the correctors proposed in [4] on the same hypothesis. Nevertheless Theorem 1 can be generalized with non constant bias assumption.

The linear corrector H proposed in [4], maps 16 bits to 8 bits. For $x = (x_0, \dots, x_{15})$ the input vector and $y = (y_0, \dots, y_7)$ the output vector, the corrector H is defined by the following relation

$$\forall i = 0, \dots, 7 \quad y_i = x_i + x_{i+1 \bmod 8} + x_{i+8 \bmod 2} .$$

The compression rate of H is 2, exactly the same rate as the xor corrector

$$y_i = x_{2i} + x_{2i+1} \bmod 2 .$$

If we note X_1 and X_2 the two input bytes of H , + the bitwise xor and $RL(X, i)$ the circular rotation of i bits, we can write H in pseudocode

$$H(X_1, X_2) = X_1 + RL(X_1, 1) + X_2 .$$

Two further improvements of H are proposed

$$H_2(X_1, X_2) = X_1 + RL(X_1, 1) + RL(X_1, 2) + X_2 ,$$

$$H_3(X_1, X_2) = X_1 + RL(X_1, 1) + RL(X_1, 2) + RL(X_1, 4) + X_2 .$$

The author says that if the bias of any input bits is e , then the lowest power of e in the bias of output bytes is 3 for H , 4 for H_2 and 5 for H_3 . His approach is to determine probability of every inputs, and to sum up the probability for all input leading to the same output of the corrector.

In order to give a simple mathematical proof of previous results, any linear corrector has to be represented by a matrix. For $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$, any linear binary corrector mapping n bits to m bits, is defined as the product of the vector x by the binary matrix $G = (g_{i,j})$:

$$\begin{pmatrix} g_{1,1} & \dots & g_{1,n} \\ \vdots & & \\ g_{m,1} & \dots & g_{m,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} .$$

Theorem 1. *Let G be a linear corrector mapping n bits to m bits and $e/2$ the bias of the input bits. Then the bias of any non zero linear combination of the output bits is less or equal than $e^d/2$, where d is the minimal distance of the linear code constructed by the generator matrix G .*

Proof. Firstly, recall that if n bits x_1, \dots, x_n have a bias $e/2$, then the bias of $x_1 + \dots + x_n \bmod 2$ is $e^n/2$ (the proof is a simple induction).

The matrix G is seen as a generator matrix of a $[n, k, d]$ linear code. By definition of the minimal distance of the code, any non zero linear combination of output bits is the sum of, at least, d input bits. We conclude that the bias of any non zero linear combination of output bits is less or equal than $e^d/2$. \square

This theorem gives an upper bound of the output bias for an arbitrary linear corrector. In particular, the matrix corresponding to H , H_2 and H_3 are respectively generator matrix of $[16, 8, 3]$, $[16, 8, 4]$ and $[16, 8, 5]$ linear codes. If all input bits have a bias $e/2$, then the bias of any linear combination of output bits is bounded, respectively by $e^3/2$, $e^4/2$ and $e^5/2$. Theorem 13 of Section 5 allows to conclude on Dichtl results on the lowest power of e in the output bytes bias.

Any linear $[n, m, d]$ -code provides a linear corrector with an estimation of its output bias. The compression rate of a corrector mapping n bits to m bits is defined by n/m . A table of linear codes gives good linear corrector with variable compression rates and input sizes [5]. The hardware implementation of linear corrector is efficiently achieved as a simple multiplication of an input vector by a constant matrix. A cyclic code provides a more compact implementation of the corrector and improves its realisation (Section 6).

There are no linear binary codes of length 16, dimension 8 with minimal distance greater than 5 [5]. In these conditions, to minimize output bias, we must search non linear correctors.

3 Non linear corrector

Let f be a corrector mapping n -bits to m -bits. A non zero linear combination of output bits of f is defined using a m -bits vector $u \neq 0$, by the Boolean function $\phi_u(x) = \sum_{i=1}^m u_i f_i(x) = u \cdot f(x)$. For an input bits bias e , the bias of this linear combination is

$$\Delta_u = \frac{P(\phi_u(x) = 1) - P(\phi_u(x) = 0)}{2}.$$

The bias Δ_u can be directly computed using the truth table of $\phi_u(x)$ and the input bias e by the formula

$$\begin{aligned} 2\Delta_u(e) &= \sum_{\substack{x \in \mathbf{F}_2^n \\ \phi_u(x)=1}} \left(\frac{1}{2} - e\right)^{n-w_h(x)} \left(\frac{1}{2} + e\right)^{w_h(x)} - \sum_{\substack{x \in \mathbf{F}_2^n \\ \phi_u(x)=0}} \left(\frac{1}{2} - e\right)^{n-w_h(x)} \left(\frac{1}{2} + e\right)^{w_h(x)} \\ &= \sum_{x \in \mathbf{F}_2^n} \left(\frac{1}{2} - e\right)^{n-w_h(x)} \left(\frac{1}{2} + e\right)^{w_h(x)} (-1)^{\phi_u(x)+1}. \end{aligned}$$

Therefore

$$\Delta_u(e) = -\frac{1}{2} \sum_{x \in \mathbf{F}_2^n} \left(\frac{1}{2} - e\right)^{n-w_h(x)} \left(\frac{1}{2} + e\right)^{w_h(x)} (-1)^{\phi_u(x)}. \quad (1)$$

For a function f , the Hamming weight $w_h(f)$ denotes the number of '1' in its truth table. The Fourier transform of f is :

$$\forall v \in \mathbf{F}_2^n \quad F_f(v) = \sum_{x \in \mathbf{F}_2^n} f(x)(-1)^{x.v} .$$

The Walsh transform of f is :

$$\forall v \in \mathbf{F}_2^n \quad \widehat{f}(v) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+v.x} .$$

The inverse Fourier transform is ([6], ch 14, sec 3) :

$$f(x) = \frac{1}{2^n} \sum_{v \in \mathbf{F}_2^n} (-1)^{u.x} F_f(v) . \quad (2)$$

The following lemma can be found in [6], ch 5, sec 2 :

Lemma 2. *Let g be the map defined on \mathbf{F}_2^n , by*

$$g(x) = X^{n-w_h(x)} Y^{w_h(x)} .$$

Then the Fourier transform F_g is defined by

$$F_g(v) = (X + Y)^{n-w_h(v)} (X - Y)^{w_h(v)} .$$

Theorem 3 presents a complete description of the bias of any non zero linear combination $\phi_u(x) = u.f(x)$ of a vectorial function f relatively to the the input bias e and the coefficients of the Walsh transform of ϕ_u :

Theorem 3. *Let f be a function which maps n bits to m bits and e the input bit bias. Then the bias Δ_u of ϕ_u is*

$$\Delta_u(e) = \frac{1}{2^{n+1}} \sum_{v \in \mathbf{F}_2^n} (2e)^{w_h(v)} (-1)^{w_h(v)+1} \widehat{\phi_u}(v) . \quad (3)$$

Proof. The bias Δ_u is obtained by formula (1) :

$$\Delta_u(e) = -\frac{1}{2} \sum_{x \in \mathbf{F}_2^n} \left(\frac{1}{2} - e\right)^{n-w_h(x)} \left(\frac{1}{2} + e\right)^{w_h(x)} (-1)^{\phi_u(x)} .$$

Let $g(x)$ be the function

$$\left(\frac{1}{2} - e\right)^{n-w_h(x)} \left(\frac{1}{2} + e\right)^{w_h(x)} .$$

By Lemma 2, we have

$$F_g(v) = \left(\frac{1}{2} - e + \frac{1}{2} + e\right)^{n-w_h(v)} \left(\frac{1}{2} - e - \frac{1}{2} - e\right)^{w_h(v)} = (2e)^{w_h(v)} (-1)^{w_h(v)} .$$

Then, with the inverse Fourier transform (2) of g :

$$\Delta_u(e) = -\frac{1}{2} \sum_{x \in \mathbf{F}_2^n} \frac{1}{2^n} \sum_{v \in \mathbf{F}_2^n} F_g(v) (-1)^{x \cdot v} (-1)^{\phi_u(x)} .$$

Therefore

$$\Delta_u(e) = \frac{1}{2^{n+1}} \sum_{v \in \mathbf{F}_2^n} (2e)^{w_h(v)} (-1)^{w_h(v)+1} \widehat{\phi}_u(v) .$$

□

For example, let f be the Boolean function defined by

$$f(x) = f(x_1, x_2, x_3) = x_2 + x_3 + x_1x_2 + x_2x_3 \pmod{2} ,$$

where the truth table and the Walsh coefficients are

x	$f(x)$	$\widehat{f}(x)$
000	0	0
001	1	4
010	1	0
100	0	-4
011	1	4
101	1	0
110	0	4
111	0	0

The probability $P(f(x) = 0) = \frac{1}{2} - e$ can be directly computed using the truth table of f :

$$\begin{aligned} P(f(x) = 0) &= \left(\frac{1}{2} - e\right)^3 + \left(\frac{1}{2} - e\right)^2 \left(\frac{1}{2} + e\right) + \left(\frac{1}{2} - e\right) \left(\frac{1}{2} + e\right)^2 + \left(\frac{1}{2} + e\right)^3 \\ &= \frac{1}{2} + 2e^2 . \end{aligned}$$

The output bias computed with Theorem 3 gives (with $u = 1$) :

$$\begin{aligned} \Delta_1(e) &= \frac{1}{16} (\widehat{f}(000) + 2e\widehat{f}(001) + 2e\widehat{f}(010) + 2e\widehat{f}(100) \\ &\quad - 4e^2\widehat{f}(011) - 4e^2\widehat{f}(101) - 4e^2\widehat{f}(110) + 8e^3\widehat{f}(111)) \\ &= -2e^2 . \end{aligned}$$

Definition 4. Let P be a polynomial of degree d , defined by

$$P(X) = \sum_{i=0}^d a_i X^i .$$

The valuation of P is the minimal $i > 0$ such that $a_i \neq 0$.

Corollary 5 is a consequence of Theorem 3 :

Corollary 5. *Let f be a function mapping n bits to m bits and e the input bias of the function. For any vector u , we define for all w , with $0 \leq w \leq n$*

$$B_w = \sum_{\substack{v \in \mathbf{F}_2^n \\ w_h(v)=w}} \widehat{\phi_u}(v) .$$

Then the bias of $\phi_u(x)$ is a polynomial of valuation W , with

$$W = \min\{w \mid B_w \neq 0\} .$$

Formula (3) gives a complete description of the bias and coefficients of the polynomial $\Delta_u(e)$ are determined by B_w .

In particular, if we consider the linear Boolean function which is the sum of d variables, then $B_w = 0$ for all $w \neq d$.

4 A resilient corrector

A (n, m, t) -resilient function is a function mapping n bits to m bits such that if t input bits are fixed, there is no influence on the output :

Definition 6. [3] A (n, m, t) -resilient function is a function f mapping \mathbf{F}_2^n to \mathbf{F}_2^m such that for any coordinates i_1, \dots, i_t and for any binary constant c_1, \dots, c_t and for all $y \in \mathbf{F}_2^m$, we have

$$P(f(x) = y \mid x_{i_1} = c_1, \dots, x_{i_t} = c_t) = 2^{-m},$$

where x_i with $i \notin \{i_1, \dots, i_t\}$ verify $P(x_i = 1) = P(x_i = 0) = 0.5$.

A (n, m, t) -linear resilient function is a linear corrector² and Theorem 8 shows the relation between resilience degree of a linear function and output bias :

Lemma 7. [3] *A $(n \times m)$ binary matrix M is a generator matrix of a linear $[n, m, d]$ -code if and only if the function*

$$x \mapsto M \cdot x$$

is a linear $(n, m, d - 1)$ -resilient function.

Theorem 8. *Let f be a linear (n, m, t) -resilient function and $e/2$ the input bias. Then the bias of any non zero linear combination of the output bits is less or equal than $e^{t+1}/2$.*

Proof. From Lemma 7, any linear (n, m, t) -resilient function provides a generator matrix of a $[n, m, t + 1]$ -linear code. The theorem follows with Theorem 1. \square

² In [7], Stinson, Martin and Sunar have proposed a true random number generator using a linear resilient function for the post-processing.

If the (n, m, t) -resilient function is non linear, Theorem 11 evaluates the output bias, using the resilience degree :

Lemma 9. [3] *Let f be a (n, m, t) -resilient function and u a non zero vector in \mathbf{F}_2^m . Then, any non zero linear combination $u.f(x)$ of f is a $(n, 1, t)$ -resilient Boolean function.*

Lemma 10. [9] *Let f be a $(n, 1, t)$ -resilient Boolean function. Then for all vector v in \mathbf{F}_2^n with $w_h(v) \leq t$, we have $\widehat{f}(v) = 0$.*

Theorem 11. *Let f be (n, m, t) -resilient function and all input bits have a bias e . Then the bias of any non zero linear combination of output bits is a polynomial in e of valuation greater than $t + 1$.*

Proof. Let $\phi_u(x) = u.f(x)$ be a linear combination of output bits. By Lemma 9 ϕ_u is a $(n, 1, t)$ -resilient Boolean function. So, all Walsh coefficients $\widehat{\phi_u}(v)$ are null for all vector v of Hamming weight less or equal than t (Lemma 10). Using Theorem 3, we obtain

$$\Delta_u(e) = \frac{1}{2^{n+1}} \sum_{\substack{v \in \mathbf{F}_2^n \\ w_h(v) > t}} (2e)^{w_h(v)} (-1)^{w_h(v)+1} \widehat{\phi_u}(v) .$$

□

In the non linear case, the resilient property is not a necessary condition to reduce the bias. The Boolean function of the previous example

$$f(x) = x_2 + x_3 + x_1x_3 + x_2x_3 \bmod 2 ,$$

is not resilient, but the output bias is reduced. Indeed, the Walsh coefficients of (001) and (100) are not null, but the sum of both is null.

For example, M. Dichtl proposed a non linear corrector mapping 16 bits to 8 bits such that all e powers up to the fifth are gone in the output bias formula [4]. This corrector was found by exhaustive search and the hardware implementation requires a considerable amount of chip area.

The calculation of syndrome of the non linear (16, 256, 6) Nordstrom-Robinson code provides a (16, 8, 5)-resilient function [8]. Theorem 11, applied to this function, gives a corrector with a valuation of $\Delta_u(e)$ equal to 6 and with a possible implementation for smart-card applications.

5 Bias and minimal entropy

In order to evaluate the random quantity in a binary sequence, the minimal entropy is an appropriate notion for random number generation in cryptography [2]. In this part, we prove that if the bias of any non zero linear combination of output bits is bounded, then the minimal entropy of the output can be estimated. Theorem 13 is a particular case of a theorem giving the relation between biased sample space and almost k -wise independent random variables [1].

Lemma 12. Let f be a function mapping n bits to m bits and Δ_u the bias of $\phi_u(x) = u.f(x)$:

$$\Delta_u = \left| P(\phi_u(x) = 1) - \frac{1}{2} \right| .$$

For all $y \in \mathbf{F}_2^m$, the function g is defined by $g(y) = P(f(x) = y)$. Then we have the following relation between Δ_u and g :

$$\Delta_u = \frac{1}{2} |F_g(u)| .$$

Proof. By definition,

$$P(f(x).u = 1) = \sum_{y \in \mathbf{F}_2^m} (y.u) P(f(x) = y) ,$$

where the inner product $(y.u)$ is a Boolean. Hence

$$P(f(x).u = 1) = \frac{1}{2} \underbrace{\sum_{y \in \mathbf{F}_2^m} g(y)}_1 - \frac{1}{2} \sum_{y \in \mathbf{F}_2^m} (-1)^{y.u} g(y) .$$

Therefore

$$\Delta_u = \frac{1}{2} |F_g(u)| .$$

□

Theorem 13. With the same notations as Lemma 12, if for all $u \in \mathbf{F}_2^m - \{(0 \dots 0)\}$, we have $|\Delta_u| \leq e$, then for all $y \in \mathbf{F}_2^m$,

$$|P(f(x) = y) - 2^{-m}| \leq 2e .$$

Proof. The inverse Fourier transform (2) of the function g is

$$g(y) = 2^{-m} \sum_{u \in \mathbf{F}_2^m} (-1)^{u.y} F_g(u) .$$

Using that $|F_g(0)| = 1$, for all $y \in \mathbf{F}_2^m$,

$$|g(y) - 2^{-m}| = 2^{-m} \left| \sum_{u \neq (0 \dots 0)} (-1)^{u.y} F_g(u) \right| .$$

By hypothesis, with Lemma 12, for all vector $u \in \mathbf{F}_2^m - \{(0 \dots 0)\}$, $|F_g(u)| \leq 2e$. Then, for all $y \in \mathbf{F}_2^m$,

$$\begin{aligned} |g(y) - 2^{-m}| &\leq 2e \times 2^{-m} \underbrace{\sum_{u \neq (0 \dots 0)} |(-1)^{u.y}|}_{2^m - 1} \\ &\leq 2e(1 - 2^{-m}) \\ &\leq 2e . \end{aligned}$$

□

In other terms, the bias of any m -tuple is less or equal than

$$2 \max_{u \in \mathbf{F}_2^m} |\Delta_u| .$$

Definition 14. Let X be a discrete random variable on $\{0, 1\}^n$. The minimal entropy of X is the maximal number k such that

$$\forall x \in X, \quad P(X = x) \leq 2^{-k} .$$

Theorem 13 is a good tool to evaluate minimal entropy of the output. Indeed, we suppose that a (n, m, t) -resilient function is used, with an input bias e . Then, with Theorems 11 and 13, the bias of any output m -tuple is a polynomial of valuation greater than $t + 1$. If e^{t+1} is negligible compared to 2^{-m} , then the minimal entropy of the output is very close to m .

With a linear (n, m, t) -resilient function and an input bias $e/2$, we have

$$P(f(x) = y) \leq 2^{-m} + e^{t+1} ,$$

then the minimal entropy of the output is greater than

$$m - \log_2(1 + e^{t+1}2^m) .$$

6 Efficient construction of linear corrector

For a linear cyclic code, a syndrome is computed with a modular polynomial reduction, which is realized by using a linear feedback shift register. Lemma 7 explains how getting linear resilient function by calculating a syndrome. Let C a $[n, k, d]$ linear code, H its check matrix and d' its dual distance, then the function $x \mapsto H \cdot x$ is a $(n, n - k, d' - 1)$ -resilient function.

An example of a linear corrector efficient in term of logic gates will illustrate the previous results. Let C the $[255, 21, 111]$ BCH code, D the $[256, 234, 6]$ dual code of C , with generator polynomial

$$H(X) = X^{21} + X^{19} + X^{14} + X^{10} + X^7 + X^2 + 1 .$$

The input 255-tuple (m_1, \dots, m_{255}) is coded by a binary polynomial $m(X) = \sum_{i=1}^{255} m_i X^i$. Therefore the function f mapping \mathbf{F}_2^{255} to \mathbf{F}_2^{21} , defined by

$$m(X) \mapsto m(X) \bmod H(X)$$

is a $(255, 21, 110)$ -resilient function. This polynomial reduction is implemented by a shift register of length 21 with only seven xor gates.

In this case, with an important input bias $e/2 = 0.25$, Theorems 8 and 13 give an output bias of :

$$\forall y \in \mathbf{F}_2^{21} \quad |P(f(X) = y) - 2^{-21}| \leq 2^{-111} .$$

Therefore, the minimal entropy of the output is very close to 21.

7 Conclusion

In this work we present general constructions of good post-processing functions. We have shown that linear correcting codes and resilient functions provide many correctors achieving good bias reduction with variable input sizes. Linear feedback shift register are suitable for an hardware implementation where the chip area is limited.

Acknowledgment

The author would like to thank Philippe Langevin for helpful discussions.

References

1. N. Alon, O. Goldreich, J. Hastad and R. Peralta : Simple Constructions of Almost k -wise Independent Random Variables. IEEE Symposium on Foundations of Computer Science, 544–553. <http://citeseer.ist.psu.edu/alon92simple.html>
2. E. Barker and J. Kelsey : Recommendation for random number generation using deterministic random bit generators (revised). NIST Special publication 800-90, March 2007. http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf
3. B. Chor, O. Goldreich, J. Hastad, J. Freidmann, S. Rudich, and R. Smolensky : The bit extraction problem or t -resilient functions. Proc. 26 th IEEE Symposium on Foundations of Computer Sciences (1985), 396–407. <http://citeseer.ist.psu.edu/chor85bit.html>
4. M.Dichtl : Bad and good ways of post-processing biased physical random numbers. Proceedings of FSE'07, LNCS 4593, (2007) 127–152.
5. M. Grassl : Code table : bounds on the parameters of various types of codes. <http://www.codestables.de>
6. F.J. Mac Williams and N.J.A. Sloane : The theory of error correcting codes. North Holland, 1977.
7. W.J. Martin, B. Sunar and D.R. Stinson : A provably secure true random number generator with built-in tolerance to active attacks. IEEE Transactions on computers, vol 56, N 1, (2007), 109–119. http://ece.wpi.edu/~sunar/preprints/resilient_trng.pdf
8. D.R. Stinson and J. Massey : An infinite class of counterexamples to a conjecture concerning non linear resilient functions. Journal of cryptology, vol 8, N 3 (1995) 167–173. <http://citeseer.ist.psu.edu/629195.html>
9. G. Xiao et J. Massey : A spectral Characterization of correlation immune functions. IEEE Transactions on information theory, V 34, (1988) 569–571.