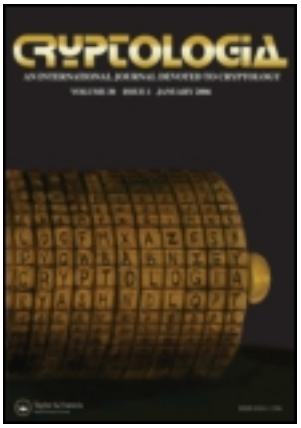

Cryptologia
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ucry20

Review of Cryptographic Boolean Functions and Applications by Thomas Cusick and Pantelimon Stănică

David Joyner

Published online: 01 Apr 2013.

To cite this article: David Joyner (2013) Review of Cryptographic Boolean Functions and Applications by Thomas Cusick and Pantelimon Stănică, Cryptologia, 37:2, 189-192, DOI: 10.1080/01611194.2013.767683

To link to this article: http://dx.doi.org/10.1080/01611194.2013.767683

Cryptologia, 37:189–192, 2013
ISSN: 0161-1194 print
DOI: 10.1080/01611194.2013.767683

Taylor & Francis
Taylor & Francis Group

Review of *Cryptographic Boolean Functions and Applications* by Thomas Cusick and Pantelimon Stănică

DAVID JOYNER

Cusick, Thomas and Stănică, Pantelimon. *Cryptographic Boolean Functions and Applications*. Academic Press, NY, 2009. 240 pages, Hardback, $62.00. ISBN-10: 0123748909, ISBN-13:978-0123748904.

> Information is the resolution of uncertainty.
>
> *Claude Shannon*

Boolean functions have applications to cooperative game theory (used to model economic and social behaviour) [10], electrical circuit design [16], theoretical computer science [16, 1], error-correcting codes [2, 3], and, as I will discuss below, cryptography. There are even some amusing applications to picture-hanging puzzles and to Brunnian links [5]. Often, in the game-theory or computer science literature, the Boolean functions considered are algebraic real-valued functions which send an n-tuple of ± 1's, that is an element of $\{\pm 1\}^n$, to an element of $\{\pm 1\}$. However, for the applications to error-correcting codes and cryptography, the Boolean functions considered are algebraic Boolean-valued functions which send an n-tuple of 0's and 1's, that is an element of $\{0, 1\}^n$, to an element of $\{0, 1\}$. In the latter case, it is convenient to think of $\{0, 1\}$ as a set with extra algebraic structure.[1]

Historically, the topic of Boolean functions can be traced back to English mathematician George Boole (1815–1864), who spent the majority of his academic life as a professor at what is now called University College Cork, Ireland. In books and papers, Boole initiated an algebraic binary approach to set theory and logic. Boole's ideas were extended and popularized by American logician Charles Sanders Pierce (1839–1914), who developed applications of Boole's ideas to logical circuits. Starting in his 1937 master's thesis, Claude Shannon (1916–2001) further promoted Boole's and Pierce's ideas in electrical circuit design [14].

The use of Boolean functions in cryptography may have arrived about the same time that the theory of linear feedback shift registers [7] was being developed, some

Address correspondence to David Joyner, Department of Mathematics, U.S. Naval Academy, Annapolis, MD 20402, USA. E-mail: wdj@usna.edu

[1]Namely, a finite Galois field with two elements, denoted $GF(2) = \{0, 1\}$, where one adds elements using exclusive or and one multiplies elements as usual.

time in the 1950s.[2] An article of historical significance is the survey by John Dillon [6].[3] Though written in 1972, it clearly shows that Boolean functions were investigated at least since the mid-1960s in (what was then called) the R41 group at NSA. For example, it references an Institute for Defense Analysis (IDA) technical report from 1966 by Oscar Rothaus[4] entitled "On Bent Functions." A paper by Rothaus with virtually the same title was published in the open literature ten years later [13]. It is a pioneering work in the application of Boolean functions to cryptography.

How are Boolean functions used in cryptography?

Basically, they are used to provided better ways to generate pseudo-random number sequences. Linear feedback sequences were constructed in the 1960s to generate pseudo-random binary sequences. For example, the sequence $\{a_n\}$ determined by

$$a_n = a_{n-1} + a_{n-2}, \quad a_0 = 1, a_1 = 0,$$

is a linear feedback sequence. In the 1970s, it was discovered that these have some security weaknesses which limit their use [11]. In other words, if one is given a "short" substream of such a sequence, one can recover the entire stream relatively easily. One of the main uses of Boolean functions is to "filter" a linear feedback shift register sequence, hopefully resulting in a more secure pseudo-random binary sequence. For example, if one lets f: $GF(2)^3 \to GF(2)$ be defined by $f(x_0, x_1, x_2) = x_0x_1 + x_2$, then the sequence $\{b_n\}$ determined by

$$b_n = f(a_n, a_{n-1}, a_{n-2}), \quad b_0 = 1, b_1 = 0, b_2 = 1,$$

is a nonlinear feedback sequence "filtered" by f. Another way to use f as a filter is to take three linear feedback sequences, say $\{a_n\}$, $\{a'_n\}$, $\{a''_n\}$, and then define the filtered sequence $\{b_n\}$ by

$$b_n = f(a_n, a'_n, a''_n).$$

In either case, the new sequence is nonlinear since f is, and this in itself improves the security. A good, short introduction to algebraic feedback shift registers, with many related algorithms implemented in Sage [15], is Celerier [4].

Regarded as a function f: $GF(2)^n \to GF(2)$, one may identify a Boolean function f with a vector of length 2^n as follows. Fix some ordering of the elements of $GF(2)^n$, so one has a function b: $\{0, 1, \ldots, 2^n - 1\} \to GF(2)^n$, and one can identify f with the vector $f = (f(b(0)), f(b(1)), \ldots, f(b(2^n - 1))$. One defines the Hamming distance between two such functions f, g to be the number of coordinates where they differ:

$$d(f, g) = \mathrm{wt}(\vec{f} + \vec{g}).$$

[2]Another *Cryptologia* review of the Goresky-Klapper book [8] discusses linear feedback shift registers in more detail, see [9].

[3]John F. Dillon received his BS degree from Villanova in 1963 and his PhD from the Mathematics Department at University of Maryland College Park in 1974. He has worked as a mathematician at the National Security Agency for over 40 years.

[4]Oscar S. Rothaus was born in Baltimore, Maryland in 1928, and got his undergraduate and graduate degrees from Princeton University. His 1958 PhD thesis was written under the direction of Salomon Bochner. In 1960, he joined the IDA, an NSA subcontractor in Princeton, working there until 1966. Rothaus took an academic position at Cornell University in 1966, where he stayed until his retirement [12].

This measures how many different values the functions f and g have and gives a good geometric notion for how far apart they are. f is *balanced* if $d(f, 0) = 2^{n-1}$. In other words, half the values of f are 0 and half are 1. Let A_n denote the collection of all affine functions ℓ: $GF(2)^n \to GF(2)$. The affine functions are the Boolean functions which, when represented as a polynomial of least degree, are of degree 1 or less. Define the *nonlinearity* of f to be

$$N_f = \min_{\ell \in A_n} d(f, \ell).$$

What are the ways to "measure" how good a Boolean function f: $GF(2)^n \to GF(2)$ is, from the cryptological perspective?

It should have "low" *autocorrelation*:

$$r_f(a) = \sum_{v \in GF(2)^n} (-1)^{f(v)+f(v+a)}.$$

This gives a measure of how often $f(v) = f(v + a)$ occurs, and therefore a quantitative measure for how nonlinear and how balanced the sequence is.

It should satisfy the *strict avalanche condition* (SAC)[5]: For each vector $a \in GF(2)^n$ of weight one, the values of the function $\Delta_{f,\,a}(x) = f(x) + f(x + a)$ are balanced. This condition implies that a "small" change in the input will result in a "big" change in the output, thus suggesting that the values of the function f are somewhat "chaotic." Basically, it should be difficult for someone who knows a few values of f to be able to predict the other values.

f is *correlation immune of order* 1, if for each x_i,

$$\text{Prob}(x_i = 1 \,|\, f(x_0, x_1, \ldots, x_{n-1}) = 0) = \frac{1}{2},$$
$$\text{Prob}(x_i = 1 \,|\, f(x_0, x_1, \ldots, x_{n-1}) = 1) = \frac{1}{2}.$$

In other words, the values of f do not give any information about the values of the individual variables occurring in f. These conditions[6] give a measurement of how unpredicable and balanced the function f is.

In some sense, the best measurement is to determine whether the Boolean function is "bent" or not. A bent function is, roughly speaking, one that is as nonlinear as possible. In other words, its nonlinearity is maximal in some sense. It turns out that this is equivalent to saying that the "derivative" $\Delta_{f,a}(x) = f(x) + f(x + a)$ is balanced for each fixed $a \in GF(2)^n$. This is the class of functions that Rothaus discovered in the 1966 paper of his referenced in Dillon [6].

A number of natural questions arise. For example, how does one construct bent functions? Are there other ways to determine if a Boolean function is bent or not? How good are bent functions at constructing secure steam ciphers? These questions and more are explored in the excellent book by Cusick and Stanica. It turns out that bent functions have beautiful and fascinating connections with other areas of

[5]There are generalizations of this condition, the "SAC of order k" and the "propagation criteria of order k," which give more refined measurements. See the book under review for more details.

[6]This idea can be generalized to a criteria called "correlation immune of order k." Roughly speaking, it says that no information on the values of any k of the variables is given by knowing $f(x) = 0$.

combinatorics, such as graph theory and difference sets. These are explored in this fine book as well.

As to the contents of the book: After the introductory first chapter, chapter 2 is on ''Fourier Analysis of Boolean Functions,'' chapter 3 is on ''Avalanche and Propagation Criteria,'' chapter 4 is on ''Correlation Immune and Resilient Boolean Functions,'' chapter 5 is on ''Bent Boolean Functions,'' chapter 6 is on ''Stream Cipher Design,'' chapter 7 is on ''Block Ciphers,'' and the last chapter is on ''Boolean Cayley Graphs.'' The book has a large list of references and a helpful index. The book could serve as a text for the graduate student or advanced undergraduate interested in stream ciphers. It can also be used as a good reference book.

About the Reviewer

David Joyner is a professer at the U.S. Naval Academy. His website is at http://www.wdjoyner.org

References

1. Bourgain, J. 2002. ''On the Distribution of the Spectrum of Boolean Functions,'' *Israel J. Math.*, 151:269–276.
2. Carlet, C. 2006. *Boolean Functions for Cryptography and Error-Correcting Codes*. Preprint. www1.spms.ntu.edu.sg/~kkhoongm/chap-fcts-Bool.pdf.
3. Carlet, C., P. Gaborit, J.-L. Kim, and P. Solé. 2012. *A New Class of Codes for Boolean Masking of Cryptographic Computations*. Preprint. http://arxiv.org/abs/1110.1193.
4. Celerier, Charles. 2012. *Feedback with Carry Shift Registers and Bent Sequences*. Honors Thesis, USNA, Annapolis, MD, http://www.usna.edu/Users/math/wdj/celerier/.
5. Demaine, E., M. Demaine, Y. Minsky, J. Mitchell, R. Rivest, and M. Patrascu. 2012. *Picture-Hanging Puzzles*. Preprint. http://arxiv.org/abs/1203.3602.
6. Dillon, J. 1972. ''A Survey of Bent Functions,'' *NSA Tech. Journal*, Special issue NSAL-S-203, 191–215.
7. Golomb, S. 1967. *Shift Register Sequences*. Laguna Hills, CA: Aegean Park Press.
8. Klapper, A. and M. Goresky. 2012. *Algebraic Shift Register Sequences*. Cambridge University Press.
9. Joyner, D. 2013. Review of *Algebraic Shift Register Sequences* by Mark Goresky and Andrew Klapper, *Cryptologia*, 37(2):174–182.
10. Jukna, Stasys. 2012. *Boolean Function Complexity: Advances and Frontiers*. Vol. 27. Springer-Verlag, Series: Algorithms and Combinatorics, http://www.thi.informatik.uni-frankfurt.de/~jukna/boolean/index.html.
11. Massey, James L. Jan 1969. ''Shift-Register Synthesis and BCH Decoding,'' *IEEE Trans. on Information Theory*, 15(1):122–127.
12. Rothaus, O. 1976. ''On 'Bent' Functions,'' *J. Combin. Theory Ser. A*, 20:300–305.
13. Obituary for O. Rothaus. 2003. http://www.nytimes.com/2003/06/08/nyregion/oscar-rothaus-75-a-creator-of-a-math-tool.html.
14. Sloane N.J.A. and A.D. Wyner. Biography of Claude Elwood Shannon, http://www2.research.att.com/~njas/doc/shannonbio.html.
15. Stein William and the Sage Development Team. Sage: *Mathematical Software*, version 5.1. http://www.sagemath.org/ (Accessed 15 Febraury 2013).
16. Wegener, Ingo. 1991. *The Complexity of Boolean Functions*. Wiley, NY, http://www.karlin.mff.cuni.cz/~krajicek/wegener.ps.