[245] RCP Marc Gysin and Jennifer Seberry, New normal sequences of length 25, *ACSC'95 - Australian Computer Science Conference*, Glenelg, January, 1995.

New Normal Sequences of Length 25

Marc Gysin and Jennifer Seberry *

The University of Wollongong,
Wollongong, NSW 2500
Australia

e-mail: marc@cs.uow.edu.au, j.seberry@cs.uow.edu.au

10 July 1994

Abstract

An introduction to binary sequences, combinatorial designs and how they are related to communication theory and computer security is given. An exhaustive search algorithm for normal sequences is presented. This is the first time that the lengths $n = 24$ and $n = 25$ have been searched through completely. No sequences of length 24 are found. It turns out that all the normal sequences of length 25 can be derived from Turyn sequences. This construction is subject to a new theorem that is given here.

Key words: Hadamard matrices, normal sequences, near–Yang sequences, Turyn sequences, exhaustive search algorithm.

1 Introduction

We start with a definition:

Definition 1
(Nonperiodic Autocorrelation Function)

Let $X = \{\{x_{11}, \ldots, x_{1n}\}, \{x_{21}, \ldots, x_{2n}\}, \ldots, \ldots, \{x_{m1}, \ldots, x_{mn}\}\}$ be a family of m sequences of elements 1, 0 and -1 and length n. The *nonperiodic autocorrelation function* of the family of sequences X, denoted by N_X, is a function defined by

$$N_X(s) = \sum_{i=1}^{n-s}(x_{1i}x_{1,i+s} + x_{2i}x_{2,i+s} + \ldots + x_{mi}x_{m,i+s}),$$

where s can range from 0 to $n-1$.

Initially people were interested in single sequence (that is m would be equal to 1 in the above definition) such as Barker sequences. A crucial criterion was the autocorrelation function being zero or one for each possible shift ($s \geq 1$) and being a comparatively large number when not shifted ($s = 0$). We observe that for $s = 0$ the autocorrelation function always returns the total number of nonzero elements in the sequence(s). Barker sequences were useful to measure long distances such as from the earth to the moon or to an aircraft. A single sequence was sent out to the remote object and overlapped with the reflected sequence. The autocorrelation function was then calculated. When the returned value increased from one or zero to a large number it was known that the sequences were no longer shifted against each other. Therefore, assuming that the velocity of the signal sent out was known, the distance to the object could be easily worked out.

Later the search turned to sets of sequences ($m > 1$) such as Golay, base and normal sequences. Golay sequences are two binary sequences with zero autocorrelation function (from now on we shall denote the nonperiodic zero autocorrelation function being zero for $s = 1, \ldots, n-1$ by zero autocorrelation function). Golay sequences could be used in spec-

*Supported by the ARC grants A49131885 and A9130102, The University of Wollongong and the Centre for Computer Security Research

trometry to cancel out all but one frequency of light and in sonar using a distance, say a submarine, between transmitters. Base sequences are four ternary sequences X_1, X_2, X_3, X_4 with X_1, X_2 having entries 1, -1 and X_3, X_4 both starting with 1, -1 and both ending with p zeros ($p = 0, \ldots, n-1$). Turyn sequences are a special set of base sequences with certain symmetries imposed on the sequences. A formal definition is given in the text below. Near–Yang sequences are a more generalized form of normal sequences.

We give the formal definitions of normal sequences and of Turyn sequences.

Definition 2 (Normal Sequences)
A triple $(F; G, H)$ of sequences is said to be a set of *normal sequences* of length n, denoted by $NS(n)$, if the following conditions are satisfied:

(i) $F = (f_k)$ is a sequence of length n with entries $1, -1$.

(ii) $G = (g_k)$ and $H = (h_k)$ are sequences of length n with entries $0, 1, -1$, such that $G + H = (g_k + h_k)$ is a $(1, -1)$ sequence of length n.

(iii)

$$\begin{aligned} g_j + g_{n-j+1} &\equiv 0 \pmod 2 \\ & \quad j = 1, ..., [\tfrac{n}{2}] \\ h_j + h_{n-j+1} &\equiv 0 \pmod 2 \end{aligned}$$

(iv) $N_F(s) + N_G(s) + N_H(s) = 0$, $s = 1, \ldots, n-1$.

It can be shown that the quasi–symmetry of the sequences, that is, condition (iii) is implied by all the other conditions. A proof is given in [Gysin93] or [1KKSYY91].

The second definition arose from the work of C.H. Yang [1Yang82], [2Yang83], [3Yang83] and [4Yang89].

Definition 3 (Turyn Sequences)
A quadruple A, B, C and D of sequences with entries $\{1, -1\}$ of length $n+1$, $n+1$, n, n are called *Turyn sequences*, denoted by $TS(2n+1)$, if they have zero autocorrelation function. That is, if $N_A(s)+N_B(s)+N_C(s)+N_D(s) = 0$ for $s = 1, \ldots, n-1$. Furthermore, their structure has to satisfy certain symmetry conditions.

If n is odd their structure is:

$$\begin{aligned} A &= \{1, a_1, a_2, \ldots, a_m, -a_m, \ldots, -a_2, -a_1, -1\} \\ B &= \{1, b_1, b_2, \ldots, b_m, -b_m, \ldots, -b_2, -b_1, 1\} \\ C &= \{c_0, c_1, \ldots, c_{m-1}, c_m, c_{m-1}, \ldots, c_1, c_0\} \\ D &= \{d_0, d_1, \ldots, d_{m-1}, d_m, d_{m-1}, \ldots, d_1, d_0\} \end{aligned}$$

where $n = 2m + 1$.
If n is even their structure is:

$$\begin{aligned} A &= \{1, a_1, a_2, \ldots, a_m, a_{m+1}, a_m, \ldots, a_2, a_1, 1\} \\ B &= \{1, b_1, b_2, \ldots, b_m, b_{m+1}, b_m, \ldots, b_2, b_1, -1\} \\ C &= \{c_0, c_1, \ldots, c_m, -c_m, \ldots, -c_1, -c_0\} \\ D &= \{d_0, d_1, \ldots, d_m, -d_m, \ldots, -d_1, -d_0\} \end{aligned}$$

where $n = 2m$.

Many sequences can be constructed from each other. Golay sequences, for example, are always normal sequences by definition. But there is also a special method concerning the symmetric and skew–symmetric part of the Golay sequences which also leads to normal sequences. We can get base sequences directly from normal sequences. Turyn sequences lead to special sets of normal sequences as shown below.

On the other hand, the sequences are also strongly related to other combinatorial designs such as orthogonal designs, Hadamard matrices, difference sets and bent functions. Some combinatorial designs can be gained from the sequences in a straightforward manner, others are subject to quite sophisticated theorems. Hadamard matrices are studied widely and they can also be used in communication theory. Bent functions play an important role in the design of Substitution Boxes which are of great importance to many cryptographic algorithms such as DES (Data Encryption Standard), LOKI and GOST. These functions are also useful in hashing algorithms such as HAVAL which produce digital fingerprints of messages.

2 New Normal Sequences Derived from Turyn Sequences

The following theorem is well known and a proof is given in [Gysin93].

Theorem 1
Let A, B be sequences of length n with entries $1, 0, -1$, where A is skew ($a_k = -a_{n-k+1}$) and B is symmetric ($b_k = b_{n-k+1}$) and $a_{[\frac{n+1}{2}]} = 0$ for odd n. Let $A + B$ and $A - B$ be $1, 0, -1$ sequences of length n and let $C = A+B$. Then $N_C(s) = N_A(s) + N_B(s)$, $s = 1, \ldots, n-1$.

[1KKSYY91] construct normal sequences from Turyn sequences:

Theorem 2 [1KKSYY91]
Let A, B, C, D be Turyn sequences $TS(2n+1)$. Let

$$\begin{aligned} F &= A/C = \{a_1, c_1, a_2, c_2, \ldots, a_n, c_n, a_{n+1}\} \\ G &= B/0_n = \{b_1, 0, b_2, 0, \ldots, b_n, 0, b_{n+1}\} \\ H &= 0_{n+1}/D = \{0, d_1, 0, d_2, \ldots, 0, d_n, 0\} \end{aligned}$$

where 0_n and 0_{n+1} are sequences of n and $n+1$ zeros.

Then F, G, H are normal sequences of length $2n+1$.

We found the following new method to construct normal sequences from Turyn sequences:

Theorem 3
Let F, G, H be normal sequences $NS(2n+1)$ derived from Turyn sequences according to Theorem 2. Then the following sequences

$$\begin{aligned} F_2 &= F \\ G_2 &= \{0, g_2 + h_2, g_3 + h_3, \ldots, g_{2n} + h_{2n}, 0\} \\ H_2 &= \{g_1, 0, \ldots, 0, g_{2n+1}\} \end{aligned}$$

are normal sequences of length $2n+1$.

Proof. We prove that G and H have the same autocorrelation function as G_2 and H_2. We have to distinguish two cases, one for even n and one for odd n.

For even n, the sequences involved have a structure as follows:

$$\begin{aligned} G &= \{1, 0, b_1, \ldots, 0, b_{m+1}, 0, \ldots, b_1, 0, -1\} \\ H &= \{0, c_0, 0, \ldots, c_m, 0, -c_m, \ldots, 0, -c_0, 0\} \end{aligned}$$

$$G_2 = \{0, c_0, b_1, \ldots, c_m, b_{m+1}, -c_m, \ldots, b_1, -c_0, 0\}$$
$$H_2 = \{1, 0, \ldots, 0, -1\}.$$

We define

$$\bar{G} = G - H_2 = \{0, 0, b_1, \ldots, 0, b_{m+1}, 0, \ldots, b_1, 0, 0\}.$$

We note that

$$G_2 = \bar{G} + H.$$

Now by using Theorem 1:

$$N_{G_2}(s) = N_{\bar{G}}(s) + N_H(s), \; s = 1, \ldots, n-1 \tag{1}$$

and

$$N_G(s) = N_{\bar{G}}(s) + N_{H_2}(s), \; s = 1, \ldots, n-1. \tag{2}$$

We write (2) as

$$N_{H_2}(s) = N_G(s) - N_{\bar{G}}(s), \; s = 1, \ldots, n-1 \tag{3}$$

and add (1) and (3) to obtain

$$N_{G_2}(s) + N_{H_2}(s) = N_G(s) + N_H(s).$$

For odd n, the proof works exactly the same except that sequences which were skew are now symmetric and vice versa. □

Therefore, if there is any triple F_2, G_2 and H_2 derived from Turyn sequences according to Theorem 3 is an $NS(2n+1)$, there is always a triple satisfying the structure of F, G and H as in Theorem 2 and vice versa.

3 The Algorithm

We briefly outline the problem and the algorithm. A full description of the algorithm together with the program-code is given in [Gysin93].

The Problem

Given a set of sequences, it is easy to test if they are normal sequences or not. The problem is to search for normal sequences. We note that the search-space grows exponentially in some manner according to the definition of the search-space itself. While the search for smaller lengths n is only a matter of seconds,

finding longer sequences can take months of CPU–Time. This is because one always faces the problem of the combinatorial explosion.

A first simple algorithm may be implemented in the following manner. Treat each sequence as a binary or ternary number. Given a length n run through all the $2^n \times 3^n \times 3^n$ possible combinations of the numbers. Decode each combination of numbers into sequences and check if normal sequences are obtained.

Needless to say that this first algorithm runs out of CPU–Time very soon. For lengths $n \geq 12$ CPU-Time invested exceeded one day.

A better Definition of the Search–Space

By looking at Condition (ii) and (iii) from Definition 2 we first observe that we can redefine the search–space. Let us examine triples of sequences which "look" like normal sequences. That is, they fulfill Condition (i) to (iii) from Definition 2, but they may not have zero autocorrelation function. This drastically cuts down the search–space.

Moving through the Search–Space

By looking at the autocorrelation function we see that for $s = n - 1$ the equation is

$$f_1 f_n + g_1 g_n + h_1 h_n = 0,$$

and for $s = n - 2$

$$f_1 f_{n-1} + f_2 f_n + g_1 g_{n-1} + g_2 g_n + h_1 h_{n-1} + h_2 h_n = 0.$$

Note that these equations are only influenced by the outermost pairs of elements of the sequences. The innermost pairs make no contribution to these equations at all.

The search is now performed in the following way.

1. First, try to find all possible combinations f_1, f_n, g_1, g_n, h_1 and h_n which fulfill the "last" equation, that is, for $s = n - 1$.
2. Move to the next equation, $s = n - 2$, and try for each previous successful combination f_1, f_n, g_1, g_n, h_1, h_n to find all the new combinations f_2, f_{n-1}, g_2, g_{n-1}, h_2, h_{n-1} which satisfy this new equation.
3. Continue until all the elements of each sequence are determined. That is, the partial sequences F_{part}, G_{part} and H_{part} become complete sequences F, G and H.
4. Test the remaining equations from the autocorrelation function for each combination in order to find all possible normal sequences F, G and H.

We observe:

- The search is performed from the outermost to the innermost triple of pairs of elements.
- If we add a new triple of pairs, we have 32 possibilities to do so for normal sequences (in accordance with the new search–space described above).
- The combinations of the first triple of pairs, that is, f_1, f_n, g_1, g_n, h_1 and h_n which satisfy the last equation do not depend on n. For a large enough n, the same could be said for the next equation and so on. (Consider for example the last two equations from the autocorrelation function for $n = 20$ and $n = 21$: they are exactly the same, and therefore the triples of pairs which fulfill these equations are also the same.)

The last statement is very important. It tells us that we can store triples of pairs and use them again for larger lengths n, and therefore the time–consuming testing of the autocorrelation function only has to be performed once.

Storing, Compressing and Reusing Triples

Each triple of pairs of elements is assigned a number (from 0 to 31) and consecutive triples of pairs of elements that passed the corresponding equations from the autocorrelation functions are stored in a file. Further compression can be done by the observation that due to the nature of the tree–search algorithm, many successful configurations start with the same triple of pairs of elements. Therefore, we only store the triples of elements which have changed since the last configuration.

Example:

Uncompressed triples	Compressed triples
⋮	⋮
17 8 3 2	17 8 3 2 "end"
17 8 3 4	4 "end"
17 8 3 10	10 "end"
17 8 4 2	4 2 "end"
17 8 4 39	29 "end"
19 2 3 1	19 2 3 1 "end"
⋮	⋮

These triples the can be reused to start the search for longer lengths n as the equations from the autocorrelation function are the same assuming that the elements do not overlap each other.

The Search–Tree and Cutting Branches of the Tree

The triples of pairs of elements represent a search–tree with branching factor 32. Each node of the search tree corresponds to one triple of pairs of elements of the sequences. At each node one equation from the autocorrelation function is considered. Downwards steps are performed when the corresponding equation from the autocorrelation function is fulfilled for this particular configuration. If a certain triple of pairs fails its corresponding equation from the autocorrelation function, the next node on the same level of the search–tree is considered. If there are no more nodes on the same level, an upwards step is performed. If the algorithm reaches the bottom of the tree (that is a leaf of the tree is examined), all the remaining equations from the autocorrelation function are tested. If positive, the configuration under consideration is a triple of normal sequences.

Testing an equation from the autocorrelation function at each step means cutting branches of the tree when appropriate. In fact, there are further cutting mechanisms. These are described in [Gysin93].

4 Computational Results

We were able to carry out the search up to length $n = 25$. The lengths 24 and 25 have been searched through exhaustively for the first time.

For length $n = 24$ no sequences were found. The algorithm found 24 sets of inequivalent normal sequences of length 25 in approximately 2500 hours CPU–time on a Sun Sparc–4 computer. We considered two triples of sequences to be equivalent if one triple of sequences can be changed into the other triple of sequences by reversing and/or negating one or more sequences of the triple. We found that all of the normal sequences of length 25 can be derived from 12 inequivalent sets of Turyn sequences using Theorem 2 and 3. An example is given in the following table:

$$F_1 =$$
$$+-+-++-++++--++-+---+++++$$
$$G_1 =$$
$$0+0+0+0+0-0+0-0+0-0-0-0-0$$
$$H_1 =$$
$$+0+0+0-0-0+0-0+0-0-0+0+0-$$
$$F_2 =$$
$$+-+-++-++++--++-+---+++++$$
$$G_2 =$$
$$0+++++-+--++--++----+-+-0$$
$$H_2 =$$
$$+0000000000000000000000-.$$

The Turyn sequences constructing these normal sequences would be:

$$\begin{aligned} A &= +++-++-++-+++ \\ B &= +++--+-+--++- \\ C &= --+++-+---++ \\ D &= ++++-+-+----. \end{aligned}$$

5 Summary and Conclusions

We presented a tree–search algorithm for searching for normal sequences. The algorithm is quite simple and straightforward to implement. Previously calculated results can be reused for the search for longer lengths n.

We were able to carry out the search up to length $n = 25$. The last two lengths were new and results were previously unknown. Length 25 leaded to a new theorem. The tree–search algorithm is not only limited to normal sequences: it can be adapted for many groups of sequences. Although this algorithm is very fast we still have to deal with the problem of the combinatorial explosion.

Acknowledgement

We wish to thank the Center for Communication and Information Science, University of Nebraska, for letting us use thousands of CPU-hours on their machine "ramoth".

References

[Edmondson91] Genet M. Edmondson, More non–existent Turyn sequences, M.Sc. Thesis, The University of New South Wales, Australian Defence Force Academy, Canberra, 1991.

[EdmSebAnd92] Genet M. Edmondson, Jennifer Seberry, Malcolm R. Anderson, On the existence of Turyn sequences of lengths less than 43, *Mathematics of Computation*, **62**, 205, 351–362, 1994.

[Gysin93] M. Gysin, Algorithms for Searching for Normal and Near–Yang Sequences, M.Sc. Thesis, The University of Wollongong, 1993.

[GysSeb93] M. Gysin and J. Seberry, New Results with Near-Yang Sequences,. *Utilitas Mathematica*, to appear.

[1KKSYY91] C. Koukouvinos, S. Kounias, J. Seberry, C.H. Yang and J. Yang, On sequences with zero autocorrelation, *Codes and Cryptography*, to appear.

[2KKSYY91] C. Koukouvinos, S. Kounias, J. Seberry, C.H. Yang and J. Yang, Multiplication of sequences with zero autocorrelation, *Australasian Journal of Combinatorics*, to appear.

[KouKouSeb90] C. Koukouvinos, S. Kounias and J. Seberry, Further results on base sequences, disjoint complementary sequences, $OD(4t;t,t,t,t)$ and the excess of Hadamard matrices, *Ars Combinatoria*, **30**, 241–256, 1990.

[KouSeb91] C. Koukouvinos and J. Seberry, Addendum to Further results on base sequences, disjoint complementary sequences, $OD(4t;t,t,t,t)$ and the excess of Hadamard matrices, *Congressus Numerantium*, **82**, 97–103, 1991.

[1Yang82] C.H. Yang, Hadamard matrices and δ-codes of length $3n$, *Proc. Amer. Math. Soc.*, **85**, 480–482, 1982.

[2Yang83] C.H. Yang, A composition theorem for δ-codes, *Proc. Amer. Math. Soc.*, **89**, 375–378, 1983.

[3Yang83] C.H. Yang, Lagrange identity for polynomials and δ-codes of lengths $7t$ and $13t$, *Proc. Amer. Math. Soc.*, **88**, 746–750, 1983.

[4Yang89] C.H. Yang, On composition of four-symbol δ–codes and Hadamard matrices, *Proc. Amer. Math. Soc.*, **107**, 763–776, 1989.