The constructing of 3-resilient Boolean functions of 9 variables with nonlinearity 240.

Andrey Khalyavin

Mech. & Math. Department
Moscow State University
119992 Moscow, Russia
email: halyavin@gmail.com

Abstract

In this work we construct for the first time 3-resilient Boolean functions of 9 variables with nonlinearity 240. We find these functions by exhaustive search in the class of functions symmetric under cyclic shifts of the first seven variables. The exhaustive search was reduced significantly by using of special techniques and algorithms which can be helpful in other similar problems. By means of found functions we construct some new functions that attain the upper bound on nonlinearity of higher number of variables whose existence was an open problem.

Keywords: secret-key cryptography, Boolean functions, resiliency, nonlinearity, fast algorithms.

1 Introduction

Boolean functions with high nonlinearity and correlation immunity have an important significance in cryptography since these functions allow to construct ciphers resistant to various attacks. In this work by means of optimized computer exhaustive search we have constructed Boolean functions with extremal characteristics of nonlinearity and correlation immunity whose existence was not known before. More exactly, these previously unknown functions are $3 + 2i$-resilient functions of $9 + 3i$ variables with nonlinearity $2^{8+3i} - 2^{4+2i}$, $i = 0, \ldots, 6$.

2 Definitions and the formulation of main result

Boolean function f is called *m-resilient* if for any substitution of any m constants instead of any arguments, the fraction of vectors where the obtained subfunction takes the value 1 is equal to one-half. *The nonlinearity of a function f* is the distance between this function and the class of linear functions. As the distance between functions we take the Hamming distance: $d(f, g) = |\{x|f(x) \neq g(x)\}|$. For m-resilient functions of n variables it was proved the upper bound on nonlinearity $2^{n-1} - 2^{m+1}$ [5, 6, 8]. It follows the problem of a construction of functions that attain this upper bound. Below we denote by (n, m, nl) the class of m-resilient functions of n variables with nonlinearity nl. In the work [1] it were found all $(7, 2, 56)$ (totally 72), $(5, 1, 12)$ (totally 8) functions symmetric relatively cyclic shifts of variables (so named rotation symmetric functions). Also in [7] it were given direct constructions of $(n, m, 2^{n-1} - 2^{m+1})$ functions for $n - 2 \geq m \geq 0.6n - 1$. In this paper we succeed to construct $(9, 3, 240)$-functions the existence of which was the open problem.

Earlier these functions were looked in the class of rotation symmetric functions but in [2] it was proved that this class does not contain desired functions. We will look for these functions in larger (by cardinality) class of functions invariant under cyclic shifts of only the first 7 arguments. We have checked also the class of functions invariant relatively cyclic shifts of the first 8 arguments but it does not appear $(9, 3, 240)$ functions there.

There are exist 20 classes of an equivalence for 7-dimension Boolean vectors relatively a cyclic shift. The last 2 bits increase the total number of equivalence classes up to $20 \cdot 4 = 80$. Thus, our space of search has the size 2^{80}. The direct exhaustive search of such big number of functions is practically impossible in our time, therefore, we will use different methods based on some properties of Walsh coefficients in order to reduce the search space.

3 Walsh coefficients of desired functions

A Walsh coefficient $W_f(u)$ of a Boolean function f is called the value $\sum\limits_{x \in F_2^n} (-1)^{f(x)+<u,x>}$. It is easy to express the nonlinearity of a function f via Walsh coefficients: $nl(f) = 2^{n-1} - \frac{1}{2} \max\limits_{u \in F_2^n} |W_f(u)|$. The desired functions have the nonlinearity 240, so $|W_f(u)| \leq 32$ for each u. On the other hand, by Sarkar Identity ([3]) $\sum\limits_{u \in F_2^n, u \in w} W_f(u) = 2^n - 2^{|w|+1} wt(f_w)$ where $wt(f)$ is the weight of the function f, f_w is the function obtained from f by the substitution of ones instead of all variables at the positions of unit bits of w, $u \in w$ is the majorization relation. The right side is always divisible by 32 because of 3-resiliency of our function. Individual Walsh coefficients can be easily expressed via sums of the left sides, therefore, they are divisible by 32 too. Thus, the Walsh coefficients can take only values 0, 32 and -32. Moreover, if the weight of u is less or equal to 3, we can substitute constants instead of variables at the positions of unit bits of u and obtain that $W_f(u) = \sum_{x_{i_1}, x_{i_2}, \ldots x_{i_k}} (-1)^{x_{i_1} + \ldots + x_{i_k}} \sum_{x_{j_1}, \ldots x_{j_l}} (-1)^{f(x)} = 0$ where $i_1, \ldots, i_k$ — unit bits of u whereas $j_1, \ldots, j_l$ — zero bits of u.

4 Definition and properties of matrix A

It is obvious that under the permutation of variables the Walsh coefficients are permuted by the same way. Therefore it is possible to split them into the same 80 classes of an equivalence. We number these classes of an equivalence as $c_1, \ldots, c_{80}$.

Let c_i and c_j be two classes of an equivalence. We define the number $a_{ij} = \sum_{x \in c_i} (-1)^{<u,x>}$ where $u \in c_j$ (this expression does not depend on the choice of u since under the permutation of coordinates neither the inner product nor the set of vectors x are not changed). All such numbers form the matrix which we denote by A. This matrix allows to calculate easily Walsh coefficients via the values of a function. Let us represent the function f by the row v where in the position i it is written 1 if $f(x) = 0, x \in c_i$, and -1 if $f(x) = 1, x \in c_i$. Let us represent the Walsh coefficients of the function f by the column w where $w_i = W_f(u), u \in c_i$. Then using the definition of the matrix A and the definition of Walsh coefficients we obtain that $w = vA$.

Theorem 1 *Let c_i and c_j be the classes of an equivalence whose representatives have the 8th bit equal to 0. Let $c_{i'}$ and $c_{j'}$ be the classes of an equivalence obtained from c_i and c_j, correspondently, by reverting the 8th bit. Then $a_{ij} = a_{i'j} = a_{ij'} = -a_{i'j'}$.*

Proof. The sums for the numbers a_{ij}, $a_{i'j}$, $a_{ij'}$, $a_{i'j'}$ are distinguished only by the multiplier $(-1)^{u_8 x_8}$. This multiplier is different from 1 only in the case $u_8 = 1$ and $x_8 = 1$ which corresponds to classes $c_{i'}$ and $c_{j'}$. □

A similar statement is true for the 9th bit. An idea to use the similar symmetry of the matrix A was stated in the work [2] but we have used more simple way to decompose the matrix A into 2 parts above. In [2] the class c_i was mapped to $c_{\bar{i}}$ which is obtained from c_i by the inversion of the first 7 bits that gives one more way to decompose the matrix A into 2 parts. Thus, the choice of the family of functions invariant under cyclic shifts of some first variables generates a matrix with a rich family of symmetries that can also help in the solution of other problems.

5 The algorithms of exhaustive search

The symmetric property of the matrix A allows to reduce an exhaustive search significantly. We split the classes of an equivalence into 2 groups. Put into G_0 all classes that have elements with the 9th bit is equal to 0, and put into G_1 all remained classes. We split vectors v and w that represent our function and Walsh coefficients, correspondently, by the same way. Then the matrix A takes the form $\begin{pmatrix} B & B \\ B & -B \end{pmatrix}$, and we obtain $w_0 = v_0 B + v_1 B$ and $w_1 = v_0 B - v_1 B$ where B is the minor 40×40 of the matrix A formed by rows and columns from G_0. All coordinates in w_0 and w_1 are divisible by 32, therefore all coordinates in $v_0 B$ and $v_1 B$ are divisible by 16. In order to find all vectors v_0 for which all coordinates $z_0 = v_0 B$ are divisible by 16 we split G_0 into 2 subgroups of $|G_0|/2 = 20$ elements (the way of a decomposition doesn't matter). We split the vector v_0 in the same way. Thus, we obtain $z_{00} = v_{00} C_0 + v_{01} C_1$ where the matrices C_0 and C_1 of size 20×40 are obtained after the decomposition of the matrix B into 2 parts according to the decomposition of G_0. Now we calculate the vectors $v_{00} C_0$ and $v_{01} C_1$ for all vectors v_{00} and v_{01}. We obtain two sets of 2^{20} vectors. For all vectors v_{00} and v_{01} we construct the vectors of residues $v_{00} C_0$ and $-v_{01} C_1$ by modulo 16. Now we sort vectors according to these vectors of residues (we compare vectors of residues lexicographically). After this it is possible to select in each set the groups of vectors with the same vectors of residues and to find all pairs of groups from different sets which have the same vectors of residues in linear time. For each such pair of groups we obtain all possible desired vectors v_0 combining all vectors from a group in the first set with all vectors from a group in the second set. Their number is appeared to be 8880903. Now for each vector v_0 we construct the vector of residues of components $v_0 B$ by modulo 32 (its components will be only 0 and 16). In order to a pair of vectors v_0 and v_1 gives the vector w with coordinates divisible by 32, these vectors of residues must coincide. Therefore, we sort all vectors v_0 according to their vectors of residues and find the groups of vectors with the same vectors of residues in linear time. Then for each pair of vectors from the same group we check the obtained vector w. If the vector w satisfies all conditions, we have found the required function since for 3-resiliency it is sufficient that all Walsh coefficients with the weight at most 3 are zeroes, and for the equality of the nonlinearity to 240 it is enough that all components of the vector w are upper bounded by 32. In fact, we check even more tight conditions. As a result, after 5 hours of calculations it were founded 423634 different 3-resilient functions with nonlinearity 240. The groups of vectors v_0 with the same vectors of residues were relatively large: from 100 until 30000 vectors, therefore for the speeding up of calculations we applied additional methods. At first, if in some position in the vectors $v_0 B$ and $v_1 B$ the values are equal by modulo 32 then either in $v_0 B + v_1 B$ or in $v_0 B - v_1 B$ we obtain the number with the absolute value 64 in this position. Thus, the final vector w will not satisfy to our conditions. Therefore, for each vector v_0 we constructed the mask in which for every position in the vector $v_0 B$ it was written 0

if the coordinate is less than 32 by the absolute value, and 1 otherwise. We deleted the positions with numbers comparable with 16 by modulo 32 from the mask since in these positions the result value 16 ± 16 of the vector's w component is always equal to allowable value 0, 32 or -32. Thus, the necessary condition that a pair of vectors is desired is the absence of digits with 1 in both masks. This condition can by quickly checked by computer using the bitwise "AND" operator. Besides, the task of the search of disjoint masks can appear in other problems in the theory of Boolean functions. In the next section we describe algorithms that can speed up calculations in this case. The first of these algorithms was used in our calculations that had allowed to speed up the calculations a little more.

6 The search of disjoint masks

Suppose that we have t-bit masks $m_1, \ldots, m_n$ that have uniform distribution over the Boolean cube B^t.

Theorem 2 *Denote by $k = n^2 \left(\frac{3}{4}\right)^t$ the average number of pairs of disjoint bit masks. There exists the algorithm of their finding using at most $O(n^\alpha + k)$ time in average where $\alpha = \log_2(1+\varphi) = 1.388...$, $\varphi = 1.618...$ is the golden section. (We assume that we can check if two given masks are intersected using one operation.)*

Proof. We will prove the bound by induction on n. Moreover, we suppose that the first masks are choosing from some set of masks A and the second masks are choosing from some set of masks B. Let us assume that for $n \le N$ we need $C(n^\alpha + k) + C_1 n \log(n)$ operations in order to find all pairs of disjoint masks for $|A| \le n, |B| \le n$, and $\varphi C(n^\alpha + k) + C_1 n \log(n)$ operations in the case $|A| \le n, |B| \le 2n$. Let us prove these bounds for $2N \ge n > N$.

Let $|A| \le n, |B| \le n$. If $t = 0$ then we simply output all possible pairs of masks. This requires Ck operations. In other case we split masks in A and B into 2 classes by the first bit. We put into A_0 all masks that begin with 0, and put into A_1 all masks that begin with 1. This requires $C_2 n$ operations. If the deviation $|A_1| - |A|/2$ is greater than $n^{0.51}$ then we simply check all pairs of masks from A and B. Since the probability of this event is decreasing exponentially when n grows, it requires $o(n)$ operations in average, therefore we can neglect this term. In the case of a small deviation, we start our algoritm recursively for the sets of masks A and B_0, and also A_0 and B_1. The total number of operations will be at most

$$C((n/2+n^{0.51})^\alpha + k) + C_1(n/2+n^{0.51})\log(n/2+n^{0.51}) + \varphi C((n/2+n^{0.51})^\alpha + k) +$$

$$C_1(n/2+n^{0.51})\log(n/2+n^{0.51}) + C_2 n =$$

$$(1+\varphi)C((n/2)^\alpha + k + o(n)) + o(n) + C_1 n\log(n) + C_2 n - C_1\log(2)n \le$$

$$C\frac{1+\varphi}{2^\alpha}(n^\alpha + k) + C_1 n\log(n) = Cn^\alpha + k + C_1 n\log(n).$$

In the second inequality we use the fact that we can choose C_1 as large as possible, in particular, greater than $C_2/\log(2)$. We have proved the first sentence of the theorem. For the second sentence ($|A| \le n, |B| \le 2n$) we act by the same way. The number of operations is at most

$$C((n+n^{0.51})^\alpha + k) + C_1(n/2+n^{0.51})\log(n/2+n^{0.51}) + \varphi C((n/2+n^{0.51})^\alpha + k) +$$

$$C_1(n/2+n^{0.51})\log(n/2+n^{0.51}) + C_2 n =$$

$$C(1+\frac{\varphi}{2^\alpha})(n^\alpha+k)+o(n)+C_1 n\log(n)+(C_2 n-C_1\log(2)n)\le$$
$$C(1+\frac{\varphi}{1+\varphi})(n^\alpha+k)+C_1 n\log(n)=$$
$$C\varphi(n^\alpha+k)+C_1 n\log(n).$$

□

This algorithm was used in calculations. In reality the distribution of masks is not uniform and the algorithm works worse. It is possible to modify the algorithm for the case of unequal probabilities for the appearance of zero and one — we should just switch sets so that $|A|<|B|$ (in the recursion call the sets A and B are used non symmetrically!). However, in this case it is hard to obtain a tight bound for the number of operations since the ratio of cardinalities of A and B can take an infinite set of values.

It is possible to decrease the exponent α in the bound $O(n^\alpha+n^2\beta^t)$ by increasing β. This kind of algorithms is effective when the number of solutions is very small (that is why they were not used in our case).

Theorem 3 *For any s there exists an algorithm of the finding of all pairs of disjoint masks which works for $O(n^\alpha+n^2\beta^{t/s})$ operations in average (in assumption that all masks are equiprobable) where $\beta=1-2\cdot 2^{-s}+3\cdot 2^{-2s}, \alpha=1+1/s, |A|\le n, |B|\le n$.*

Proof. If $t<s$, we simply check all pairs of masks that demands $Cn^2=O(n^2\beta^t)$ operations. In the opposite case we split all masks in $|A|$ and $|B|$ into 3 groups: we put into A_0 and B_0 all masks where the first s bits are equal to 1, we put into A_1 and B_1 all masks where the first s bits are equal to 0, and we put into A_2 and B_2 all remained masks. Then we solve subproblems for the pairs of sets (A_0,B_1), (A_1,B_0), $(A_1\cup A_2, B_1\cup B_2)$. We will prove the bound on the number of operations by induction. By the same way as in the previous theorem it is possible to neglect the nonuniformity of the distribution of sets into parts and the linear number of operations for the fulfillment of this decomposition. Then we can estimate the number of operations by the value

$$C(n/2^s)^\alpha+Cn^2\beta^{t/s-1}/2^{2s}+C(n/2^s)^\alpha+Cn^2\beta^{t/s-1}/2^{2s}+$$
$$+C(n(1-1/2^s))^\alpha+Cn^2\beta^{t/s-1}(1-1/2^s)^2=$$
$$Cn^\alpha(2(1/2^s)^\alpha+(1-1/2^s)^\alpha)+Cn^2\beta^{t/s}(2/2^{2s}+(1-1/2^s)^2)/\beta=$$
$$Cn^\alpha(2/2^{s+1}+(1-1/2^s)^\alpha)+Cn^2\beta^{t/s}<$$
$$Cn^\alpha(1/2^s+(1-1/2^s))+Cn^2\beta^{t/s}=Cn^\alpha+Cn^2\beta^{t/s}.$$

This proves the inductive step (the base is obvious since we can take as large constant C as we want). □

7 The analysis of constructed functions

For each of 423634 functions we have considered 4 subfunctions of 7 variables (decomposing by the last 2 variables) and calculated the degree of their resiliency. It is appears that in 400594 functions all subfunctions are only 1-resilient and in 23040 functions all subfunctions are 2-resilient. It is not appeared functions in which the part of subfunctions are 1-resilient and the part of subfunctions are 2-resilient. Moreover, it is appeared that the nonlinearity of these 2-resilient subfunctions is equal either 48 or 56, and also the number of subfunctions with nonlinearity 56 can be equal to

either 0 (4608 functions) or 2 (18432 functions). The example of a function without subfunctions with nonlinearity 56:

```
791C92A7 1EE2C659 9867C768 A5693996 6B5499A9 349EC636 D0AB1E65 2FC1D269
1AECE525 65936A99 CB3116DA E14E3C96 9C2762D9 C3E2971C A55A6768 3C5998A7
```

The example of a function with two subfunctions with nonlinearity 56:

```
F5191A96 1A6C9CE3 38C6C9A7 96A76750 8CA76E51 C36539B8 A7D0619B 721E5C8E
635A91EC B49AC707 5C2B3E64 8DD1A279 619A9E66 9E656169 1EE5E119 C338699E
```

Also it were found 3840 functions which are decomposed into two 3-resilient subfunctions of 8 variables with nonlinearity 112 (we decompose by the last variable). It allows (see [4]) to construct functions of the form $(9+3i, 3+2i, 2^{8+3i}-2^{4+2i})$. For $i=0,\ldots,6$, it gives functions that was not known before. The example of a function that admits such decomposition:

```
96C307DA AA71B54C 664EF138 5C3989A7 E5919C3A 1C8F7266 32AD8E55 5BE0C369
619A9E66 9E656169 1EE5E119 C338699E E51B1AD4 926C9DA3 3CC269A7 96976658
```

8 Acknowledgement

The author is deeply grateful to his scientific supervisor Prof. Yuriy Tarannikov for the formulation of the problem, attention to the work and valuable advices.

References

[1] P. Stanica, S. Maitra. Rotation symmetric Boolean functions — count and cryptographic properties. In *Advances in Cryptology - Crypto 2000*, pages 515-532, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science Volume 1880.

[2] A. Maximove, M. Hell, S. Maitra. Plateaued rotation symmetric Boolean functions on odd number of variables. Available at IACR eprint server, eprint.iacr.org, no. 2004/144, 25 June 2004, 2004.

[3] P. Sarkar. Spectral domain analysis of correlation immune and resilient Boolean functions. Cryptology ePrint archive (http://eprint.iacr.org). Reprint 2000/049, September 2000, 19p.

[4] E. Pasalic, S. Maitra, T. Johansson, P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bounds on nonlinearity, Workshop on Coding and Cryptography - WCC 2001, Paris, January 8–12, 2001, Electronic Notes in Discrete Mathematics, Volume 6, Elsevier Science, 2001.

[5] P. Sarkar, S. Maitra, Nonlinearity bounds and constructions of resilient Boolean functions, In Advanced in Cryptology: Crypto 2000, Proceedings, Lecture Notes in Computer Science, V. 1880, 2000, pp. 515–532.

[6] Yu. Tarannikov, On resilient Boolean functions with maximal possible nonlinearity, Proceedings of Indocrypt 2000, Lecture Notes in Computer Science, V. 1977, pp. 19–30, Springer-Verlag, 2000.

[7] Yu. Tarannikov, New constructions of resilient Boolean functions with maximal nonlinearity, 8th Fast Software Encryption Workshop, FSE 2001, Yokohama, Japan, April 2–4, 2001. Revised Papers, Lecture Notes in Computer Science, V. 2355, pp. 66–77, Springer-Verlag, 2002.

[8] Y. Zheng, X.-M. Zhang, Improved upper bound on the nonlinearity of high order correlation immune functions, Selected Areas in Cryptography, 7th Annual International Workshop, SAC2000, Lecture Notes in Computer Science, V. 2012, pp. 264–274, Springer-Verlag, 2001.