Linear Codes in Constructing Resilient Functions with High Nonlinearity

Enes Pasalic[1] and Subhamoy Maitra[2]

[1] Department of Information Technology, Lund University,
P.O. Box 118, 221 00 Lund, Sweden
enes@it.lth.se
[2] Computer & Statistical Service Center, Indian Statistical Institute,
203, B.T. Road, Calcutta 700 035, India
subho@isical.ac.in

Abstract. In this paper we provide a new generalized construction method of highly nonlinear t-resilient functions, $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$. The construction is based on the use of linear error correcting codes together with multiple output bent functions. Given a linear $[u, m, t+1]$ code we show that it is possible to construct n-variable, m-output, t-resilient functions with nonlinearity $2^{n-1} - 2^{\lceil \frac{n+u-m-1}{2} \rceil}$ for $n \geq u + 3m$. The method provides currently best known nonlinearity results.

Keywords: Resilient functions, Nonlinearity, Correlation Immunity, Stream Ciphers, Linear Codes.

1 Introduction

A well known method for constructing a running key generator exploits several linear feedback shift registers (LFSR) combined by a nonlinear Boolean function. This method is used in design of stream cipher system where each key stream bit is added modulo two to each plaintext bit in order to produce the ciphertext bit. The Boolean function used in this scenario must satisfy certain properties to prevent the cipher from common attacks, such as Siegenthaler's correlation attack [18], linear synthesis attack by Berlekamp and Massey [14] and different kinds of approximation attacks [7]. If we use multiple output Boolean function instead of single output one, it is possible to get more than one bits at each clock and this increases the speed of the system. Such a multiple output function should possess high values in terms of order of resiliency, nonlinearity and algebraic degree.

Research on multiple output binary resilient functions has received attention from mid eighties [6,1,8,19,2,9,21,12,11,4,5]. The initial works on multiple output binary resilient functions were directed towards linear resilient functions. The concept of multiple output resilient functions was introduced independently by Chor et al [6] and Bennett et al [1]. A similar concept was introduced at the same time for single output Boolean functions by Siegenthaler [17]. Besides its importance in random sequence generation for stream cipher systems, these

S. Vaudenay and A. Youssef (Eds.): SAC 2001, LNCS 2259, pp. 60–74, 2001.

resilient functions have applications in quantum cryptographic key distribution, fault tolerant distributed computing, etc.

The nonlinearity issue for such multiple output resilient functions was first discussed in [20]. After that, serious attempts towards construction of nonlinear resilient functions have been taken in [21,12,11,5]. We here work in that direction and provide better results than the existing work. For given number of input variables n, number of output variables m, and order of resiliency t, we can construct functions $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ that achieve higher nonlinearity values than existing constructions for almost all choices of n, m and t.

The paper is organized as follows. Section 2 provides basic definitions and notations both for 1-output and m-output functions, $m > 1$. In Section 3, we review some important techniques and results used towards the new construction of t-resilient functions. Section 4 provides the new construction based on the use of linear error-correcting codes together with bent functions. Some numerical values for the constructed functions and comparison with previous constructions are presented in Section 5. Section 6 concludes this paper.

2 Preliminaries

For binary strings S_1, S_2 of the same length λ, we denote by $\#(S_1 = S_2)$ (respectively $\#(S_1 \neq S_2)$), the number of places where S_1 and S_2 are equal (respectively unequal). The *Hamming distance* between S_1, S_2 is denoted by $d(S_1, S_2)$, i.e.,

$$d(S_1, S_2) = \#(S_1 \neq S_2).$$

Also the *Hamming weight* or simply the weight of a binary string S is the number of ones in S. This is denoted by $wt(S)$.

By $\mathbb{F}_2^n$ we denote the vector space corresponding to the finite field $\mathbb{F}_{2^n}$. The addition operator over $\mathbb{F}_2$ is denoted by $\oplus$ (the XOR operation, which is basically addition modulo 2). By V_n we mean the set of all Boolean functions on n-variables, i.e., V_n corresponds to all possible mappings $\mathbb{F}_2^n \mapsto \mathbb{F}_2$. We interpret a Boolean function $f(x_1, \ldots, x_n)$ as the output column of its truth table, that is, a binary string of length 2^n,

$$[f(0, 0, \ldots, 0), f(1, 0, \ldots, 0), f(0, 1, \ldots, 0), \ldots, f(1, 1, \ldots, 1)].$$

An n-variable function f is said to be *balanced* if its output column in the truth table contains equal number of 0's and 1's (i.e., $wt(f) = 2^{n-1}$).

An n-variable Boolean function $f(x_1, \ldots, x_n)$ can be considered to be a multivariate polynomial over $\mathbb{F}_2$. This polynomial can be expressed as a sum of products representation of all distinct k-th order product terms ($0 \leq k \leq n$) of the variables. More precisely, $f(x_1, \ldots, x_n)$ can be written as

$$f(x_1, \ldots, x_n) = a_0 \oplus (\bigoplus_{i=1}^{i=n} a_i x_i) \oplus (\bigoplus_{1 \leq i \neq j \leq n} a_{ij} x_i x_j) \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n,$$

where the coefficients $a_0, a_{ij}, \ldots, a_{12\ldots n} \in \{0,1\}$. This representation of f is called the *algebraic normal form* (ANF) of f. The number of variables in the highest order product term with nonzero coefficient is called the *algebraic degree*, or simply degree of f.

Functions of degree at most one are called affine functions. An affine function with constant term equal to zero is called a linear function. The set of all n-variable affine (respectively linear) functions is denoted by A_n (respectively L_n). The *nonlinearity* of an n variable function f is

$$nl(f) = min_{g \in A_n}(d(f,g)),$$

i.e., the distance from the set of all n-variable affine functions.

Let $x = (x_1, \ldots, x_n)$ and $\omega = (\omega_1, \ldots, \omega_n)$ both belong to $\mathbb{F}_2^n$. The *dot product* of x and ω is defined as

$$x \cdot \omega = x_1\omega_1 \oplus \ldots \oplus x_n\omega_n.$$

For a Boolean function $f \in V_n$ the *Walsh transform* of $f(x)$ is a real valued function over $\mathbb{F}_2^n$ that can be defined as

$$W_f(\omega) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus x \cdot \omega}.$$

Next we define *correlation immunity* in terms of the characterization provided in [10]. A function $f(x_1, \ldots, x_n)$ is m-th order correlation immune (CI) iff its Walsh transform W_f satisfies

$$W_f(\omega) = 0, \text{ for all } \omega \in \mathbb{F}_2^n \text{ s.t. } 1 \leq wt(\omega) \leq m.$$

If f is balanced then $W_f(\overline{0}) = 0$. Balanced m-th order correlation immune functions are called *m-resilient* functions. Thus, a function $f(x_1, \ldots, x_n)$ is m-resilient iff its Walsh transform W_f satisfies

$$W_f(\omega) = 0, \text{ for all } \omega \in \mathbb{F}_2^n \text{ s.t. } 0 \leq wt(\omega) \leq m.$$

Given all these definitions we now start the definitions with respect to the multiple output Boolean functions $\mathbb{F}_2^n \mapsto \mathbb{F}_2^m$. That is, in this case we provide the truth table of m different columns of length 2^n. Let us consider the function $F(x) : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ such that $F(x) = (f_1(x), \ldots, f_m(x))$. Then the nonlinearity of F is defined as,

$$nl(F) = \min_{\tau \in \mathbb{F}_2^{m*}} nl(\bigoplus_{j=1}^{m} \tau_j f_j(x)).$$

Here, $\mathbb{F}_2^{m*} = \mathbb{F}_2^m \backslash 0$ and $\tau = (\tau_1, \ldots, \tau_m)$. Similarly the algebraic degree of F is defined as,

$$deg(F) = \min_{\tau \in \mathbb{F}_2^{m*}} deg(\bigoplus_{j=1}^{m} \tau_j f_j(x)).$$

Now we define an n-variable, m-output, t-resilient function, denoted by (n, m, t), as follows. A function F is an (n, m, t) resilient function, iff $\bigoplus_{j=1}^{m} \tau_j f_j(x)$ is an $(n, 1, t)$ function (n variable, t-resilient Boolean function) for any choice of $\tau \in \mathbb{F}_2^{m*}$. Since we are also interested in nonlinearity, we provide the notation (n, m, t, w) for an (n, m, t) resilient function with nonlinearity w. In this paper we concentrate on the nonlinearity value. Thus, for given size of input parameters n, m, t, we construct the functions with currently best known nonlinearity.

3 Useful Techniques

In this section we will describe a few existing techniques that will be used later. First we recapitulate one result related to linear error correcting codes. The following lemma was proved in [11]. We will use it frequently in our construction, and therefore it is stated with the proof.

Proposition 1. *Let $c_0, \ldots, c_{m-1}$ be a basis of a binary $[u, m, t+1]$ linear code C. Let β be a primitive element in $\mathbb{F}_{2^m}$ and $(1, \beta, \ldots, \beta^{m-1})$ be a polynomial basis of $\mathbb{F}_{2^m}$. Define a bijection $\phi : \mathbb{F}_{2^m} \mapsto C$ by*

$$\phi(a_0 + a_1\beta + \cdots a_{m-1}\beta^{m-1}) = a_0c_0 + a_1c_1 + \cdots a_{m-1}c_{m-1}.$$

Consider the matrix

$$A^* = \begin{pmatrix} \phi(1) & \phi(\beta) & \ldots & \phi(\beta^{m-1}) \\ \phi(\beta) & \phi(\beta^2) & \ldots & \phi(\beta^m) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\beta^{2^m-2}) & \phi(1) & \ldots & \phi(\beta^{m-2}) \end{pmatrix}.$$

For any linear combination of columns (not all zero) of the matrix A^*, each nonzero codeword of C will appear exactly once.

Proof. Since ϕ is a bijection, it is enough to show that the matrix

$$\begin{pmatrix} 1 & \beta & \ldots & \beta^{m-1} \\ \beta & \beta^2 & \ldots & \beta^m \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{2^m-2} & 1 & \ldots & \beta^{m-2} \end{pmatrix}$$

has the property that each element in $\mathbb{F}_{2^m}^*$ will appear once in any nonzero linear combination of columns of the above matrix.

Any nonzero linear combination of columns can be written as

$$(c_0 + c_1\beta + \cdots + c_{m-1}\beta^{m-1}) \begin{pmatrix} 1 \\ \beta \\ \vdots \\ \beta^{2^m-2} \end{pmatrix},$$

for some $c_0, c_1, \ldots, c_{m-1} \in \mathbb{F}_2$, and this gives the proof. □

There are $2^m - 1$ rows in the matrix A^*. Let us only concentrate on the first 2^{m-1} rows of this matrix. That is, we consider each column to be of length 2^{m-1}. It is clear that for any nonzero linear combination of the columns, a nonzero codeword of C will appear exactly once in it. Hence, in the resulting column of length 2^{m-1}, no codeword will appear more than once. In this direction, we update Proposition 1 with the following result.

Proposition 2. *Let $c_0, \ldots, c_{m-1}$ be a basis of a binary $[u, m, t+1]$ linear code C. Let β be a primitive element in $\mathbb{F}_{2^m}$ and $(1, \beta, \ldots, \beta^{m-1})$ be a polynomial basis of $\mathbb{F}_{2^m}$. Define a bijection $\phi : \mathbb{F}_{2^m} \mapsto C$ by*

$$\phi(a_0 + a_1\beta + \cdots a_{m-1}\beta^{m-1}) = a_0c_0 + a_1c_1 + \cdots a_{m-1}c_{m-1}.$$

For $0 \leq q \leq m-1$, consider the matrix

$$D = \begin{pmatrix} \phi(1) & \phi(\beta) & \ldots & \phi(\beta^{m-1}) \\ \phi(\beta) & \phi(\beta^2) & \ldots & \phi(\beta^m) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\beta^{2^q-1}) & \phi(\beta^{2^q}) & \ldots & \phi(\beta^{2^q+m-2}) \end{pmatrix}.$$

For any linear combination of columns (not all zero) of the matrix D, each nonzero codeword of C will either appear exactly once or not appear at all.

Note that the entries of D are elements from $\mathbb{F}_2^u$. For convenience, we use a standard index notation to identify the elements of D. That is, $d_{i,j}$ denotes the element in i-th row and j-th column of D, for $i = 1, \ldots, 2^q$, and $j = 1, \ldots, m$.

Throughout the paper we consider C to be a binary linear $[u, m, t+1]$ code with a set of basis vectors $c_0, c_1, \ldots, c_{m-1}$. To each codeword $c_i \in C$, $i = 0, \ldots, 2^m - 1$, we can associate a linear function $l_{c_i} \in L_u$, where

$$l_{c_i} = c_i \cdot x = \bigoplus_{k=1}^{u} c_{i,k}x_k.$$

This linear function is uniquely determined by c_i. Since the minimum distance of C is $t+1$, any function l_{c_i} for $c_i \in C$ will be nondegenerate on at least $t+1$ variables, and hence t-resilient.

According to Proposition 1, any column of the matrix A^* can be seen as a column vector of $2^m - 1$ distinct t-resilient linear functions on u variables. In [11], it was proved that the existence of a set $\mathcal{C}$ of linear $[u, m, t+1]$ *nonintersecting codes* of cardinality $|\mathcal{C}| = \lceil 2^{n-u}/2^m - 1 \rceil$ was sufficient and necessary requirement in construction of an $(n, m, t, 2^{n-1} - 2^{u-1})$ function. A set of linear $[u, m, t+1]$ codes $\mathcal{C} = \{C_1, C_2, \ldots, C_s\}$ such that $C_i \cap C_j = \{0\}, 1 \leq i < j \leq s$, is called a set of linear $[u, m, t+1]$ nonintersecting codes.

The results in [11] were obtained using a computer search for the set $\mathcal{C}$. Good results could be obtained only for small size of $n \leq 20$, thus not providing a good construction for arbitrary n.

In this initiative our approach is different. We do not try to search for non-intersecting linear codes. We only consider a single linear code with given parameters and use a repetition of the codewords in a specific manner. If we look into the matrix D of Proposition 2, and consider each column as concatenation of 2^q $(0 \leq q \leq m-1)$ linear functions on u variables, then each column can be seen as a Boolean function on $u+q$ variables, i.e., $g_j \in V_{u+q}$, $j = 1, \ldots, m$. In the ANF notation the functions $g_j \in V_{u+q}$ will be given by,

$$g_j(y, x) = \bigoplus_{\tau \in \mathbb{F}_2^q} (y_1 \oplus \tau_1) \cdots (y_q \oplus \tau_q)(d_{[\tau]+1,j} \cdot x),$$

where $[\tau]$ denotes the integer representation of vector τ. Once again note that we have denoted the elements of D matrix as $d_{i,j}$, for $i = 1, \ldots, 2^q$, and $j = 1, \ldots, m$. Since each of the constituent linear functions is nondegenerate on $t+1$ variables, they are all t-resilient. Thus, each of the $(u+q)$-variable Boolean function g_j is t-resilient. Next we have the following result on nonlinearity.

Proposition 3. *Any nonzero linear combination of the functions $g_1, \ldots, g_m$ has the nonlinearity $2^{u+q-1} - 2^{u-1}$.*

Proof. From [16], we have, $nl(g_j) = 2^{u+q-1} - 2^{u-1}$ for $j = 1, \ldots, m$. Moreover, from Proposition 2, it is clear that any nonzero linear combination of these functions $g_1, \ldots, g_m$ will have the same property. □

Hence we get the following result related to multiple output functions.

Proposition 4. *Given a $[u, m, t+1]$ linear code, it is possible to construct $(u+q, m, t, 2^{u+q-1} - 2^{u-1})$ resilient functions, for $0 \leq q \leq m-1$.*

A simple consequence of Proposition 4 is that for given m and t our goal is to use a linear code of minimum length, i.e., u should be minimized, since the nonlinearity is maximized in that case. Throughout this paper the functions constructed by means of Proposition 4 will be denoted by g_j. We immediately get the following corollary concerning the construction of 1-resilient functions.

Corollary 1. *It is possible to construct an $(n = 2m, m, 1, nl(F) = 2^{n-1} - 2^{\frac{n}{2}})$ function $F(x)$.*

Proof. It is possible to construct $[m+1, m, 2]$ linear code. Putting $u = m+1$ and $q = m-1$, we get $(n, m, 1, 2^{n-1} - 2^m)$ resilient functions. □

Thus, using Corollary 1 with $m = 16$, we can construct 1-resilient function $F(x) : \mathbb{F}_2^{32} \mapsto \mathbb{F}_2^{16}$ with nonlinearity $N_F = 2^{n-1} - 2^{\frac{n}{2}} = 2^{31} - 2^{16}$. This function can be used in a stream cipher system where at each clock it is possible to get 2-byte output.

Next we look into a more involved technique. For this we need a set of m bent functions such that any nonzero linear combination of these bent functions will also be a bent function.

The following proposition is well known and therefore stated without proof (for proof see [16]).

Proposition 5. *Let $h(y) \in V_k$ and $g(x) \in V_{n_1}$. Then the nonlinearity of $f(y,x) = h(y) \oplus g(x)$ is given by, $nl(f) = 2^k nl(g) + 2^{n_1} nl(h) - 2nl(g)nl(h)$.*

Next we present the following Corollaries which will be useful in the sequel.

Corollary 2. *Let $h(y)$ be a bent function on V_k, $k = 2m$. Let $g(x) \in V_{n_1}$ with $nl(g) = 2^{n_1-1} - 2^{u-1}$, for $u \leq n_1$. Then the nonlinearity of $f(y,x) = h(y) \oplus g(x)$ is given by, $nl(f) = 2^{n_1+k-1} - 2^{\frac{k}{2}} 2^{u-1}$.*

Proof. Put $nl(h) = 2^{k-1} - 2^{\frac{k}{2}-1}$ in Proposition 5. □

Corollary 3. *Let $h'(y')$ be a bent functions on V_k, $k = 2r$, and let $h(y)$ be a function on V_{k+1} given by $h(y) = x_{k+1} \oplus h'(y')$. Let $g(x) \in V_{n_1}$ with $nl(g) = 2^{n_1-1} - 2^{u-1}$, for $u \leq n_1$. Then the nonlinearity of $f(y,x) = h(y) \oplus g(x)$ is given by, $nl(f) = 2^{n_1+k-1} - 2^{\frac{k+1}{2}} 2^{u-1}$.*

Proof. Put $nl(h) = 2^{k-1} - 2^{\frac{k+1}{2}-1}$ in Proposition 5. □

Corollary 4. *Let $h(y)$ be a constant function on V_k, $k > 0$. Let $g(x) \in V_{n_1}$ with $nl(g) = 2^{n_1-1} - 2^{u-1}$, for $u \leq n_1$. Then the nonlinearity of $f(y,x) = h(y) \oplus g(x)$ is given by, $nl(f) = 2^{n_1+k-1} - 2^k 2^{u-1}$.*

Proof. Put $nl(h) = 0$ in Proposition 5. □

Thus, using the composition of bent functions with resilient functions, one may construct highly nonlinear resilient Boolean functions on higher number of variables. The question is if we may use the same technique for construction of multiple output functions. In other words, we want to find a set of bent functions of cardinality $2^m - 1$, say $B = \{b_1, \ldots, b_{2^m-1}\}$, with basis $b_1, \ldots, b_m$, such that $\bigoplus_{j=1}^{m} \tau_j b_j \in B$, for $\tau \in \mathbb{F}_2^{m*}$.

Now we discuss the construction in more detail [15]. Let A be of size $2^m \times m$ given by $A = (\frac{\mathbf{0}}{A^*})$, where A^* is a matrix constructed by means of Proposition 1 using $c_0, \ldots, c_{m-1}$, that spans an $[m, m, 1]$ code C with the unity matrix I as the generator matrix. Now consider each column of the matrix A, which can be seen as concatenation of 2^m distinct linear functions on m variables. This is a Maiorana-McFarland type bent function in $2m$-variables. Also using Proposition 1, it is clear that any nonzero linear combination of these bent functions will provide a bent function. The algebraic degree of this class of bent functions is equal to m. Thus, we have the following result.

Proposition 6. *It is possible to get m distinct bent functions on $2m$-variables, say $b_1, \ldots, b_m$, such that any nonzero linear combination of these bent functions will provide a bent function. Also, $deg(\bigoplus_{i=1}^{m} \tau_i b_i) = m$, for $\tau \in \mathbb{F}_2^{m*}$.*

Example 1. Let $m = 2$ and $c_0 = (01)$, $c_1 = (10)$. We use an irreducible polynomial $p(z) = z^2 + z + 1$ to create the field $\mathbb{F}_{2^2}$. Then it can be shown that the matrix A is given by,

$$A = \begin{pmatrix} 0 & 0 \\ c_0 & c_1 \\ c_1 & c_0 + c_1 \\ c_0 + c_1 & c_0 \end{pmatrix}.$$

In the truth table notation, let us consider the 4-variable bent function $g_1(x)$ as the concatenation of the 2-variable linear functions $0, x_1, x_2, x_1 \oplus x_2$ and similarly, $g_2(x)$ as concatenation of $0, x_2, x_1 \oplus x_2, x_1$. Then the function $g_1(x) \oplus g_2(x)$ is also bent, which is a concatenation of $0, x_1 \oplus x_2, x_1, x_2$.

Also note the following updation of Proposition 6.

Proposition 7. *It is possible to get m distinct bent functions on $2p$-variables ($p \geq m$), say $b_1, \ldots, b_m$, such that any nonzero linear combination of these bent functions will provide a bent function. Also, $deg(\bigoplus_{i=1}^{m} \tau_i b_i) = p$, for $\tau \in \mathbb{F}_2^{m*}$.*

With these results we present our construction method in the following section.

4 New Construction

In this section we will first provide the general construction idea using a $[u, m, t+1]$ linear code and then we will use specific codes towards construction of resilient functions of specific orders. Let us first discuss the idea informally. We take the matrix D as described in Proposition 2. Now it is clear that each column of D can be seen as a $u+q$ variable function with order of resiliency t and nonlinearity $2^{u+q-1} - 2^{u-1}$. Let us name these functions as $g_1, \ldots g_m$. From Proposition 4, it is known that any nonzero linear combination of these functions will provide $u+q$ variable function g with order of resiliency t and nonlinearity $2^{u+q-1} - 2^{u-1}$.

Now we concentrate on n-variable functions. It is clear that the $(u + q)$-variable function need to be repeated 2^{n-u-q} times to make an n-variable function. We will thus use an $(n - u - q)$-variable function and XOR it with the $(u+q)$-variable function to get an n-variable function. Also to get the maximum possible nonlinearity in this method, the $(n - u - q)$-variable function must be of maximum possible nonlinearity. We will use m different functions $h_1, \ldots, h_m$ and use the compositions $f_1 = h_1 \oplus g_1, \ldots, f_m = h_m \oplus g_m$, to get m different n-variable functions. Thus any nonzero linear combination of $f_1, \ldots, f_m$ can be seen as the XOR of linear combinations of $h_1, \ldots, h_m$ and linear combinations of $g_1, \ldots, g_m$. In order to get a high nonlinearity of the vector output function we will need high nonlinearity of the functions $h_1, \ldots, h_m$ and also high nonlinearity for their linear combinations.

If $(n - u - q)$ is even, we can use bent functions $h_1, \ldots, h_m$. Importantly, we require m different bent functions (as in Proposition 6) such that the nonzero linear combinations will also produce bent functions. For this we need $n-u-q \geq$

$2m$ (see Proposition 7). If $(n-u-q)$ is odd, we can use bent functions b_j of $(n-u-q-1)$ variables and take $h_j = x_n \oplus b_j$. This requires the condition $n-u-q-1 \geq 2m$ to get m distinct bent functions as in Proposition 7.

It may very well happen that the value of $n-u-q$ may be less than $2m$ and in such a scenario it may not be possible to get $2m$ bent functions with desired property. In such a situation we may not get very good nonlinearity. We formalize the results in the following theorem.

Theorem 1. *Given a linear $[u, m, t+1]$ code, for $n \geq u$ it is possible to construct $(n, m, t, nl(F))$ function $F = (f_1, \ldots, f_m)$, where*

$$nl(F) = \begin{cases} 2^{n-1} - 2^{u-1}, & u \leq n < u+m; \quad (1) \\ 2^{n-1} - 2^{n-m}, & u+m \leq n < u+2m; \quad (2) \\ 2^{n-1} - 2^{u+m-1}, & u+2m \leq n < u+3m; \quad (3) \\ 2^{n-1} - 2^{\frac{n+u-m-1}{2}}, & n \geq u+3m-1,\ n-u-m+1 \ \textit{even}; \quad (4) \\ 2^{n-1} - 2^{\frac{n+u-m}{2}}, & n \geq u+3m,\ n-u-m+1 \ \textit{odd}. \quad (5) \end{cases}$$

Proof. We consider different cases separately. We will use functions $g_1, \ldots, g_m$ on $u+q$ variables which are basically concatenation of q distinct linear functions on u variables. These linear functions are nondegenerate on at least $t+1$ variables. From Proposition 3, we get that for any $\tau \in \mathbb{F}_2^{m*}$, $nl(\bigoplus_{j=1}^m \tau_j g_j) = 2^{u+q-1} - 2^{u-1}$. Next we consider m different functions $h_1, \ldots, h_m$ on $(n-u-q)$ variables. We will choose those functions in such a manner so that, for any $\tau \in \mathbb{F}_2^{m*}$, $nl(\bigoplus_{j=1}^m \tau_j h_j)$ is high. Mostly we will use bent functions as in Proposition 6 and Proposition 7 in our construction. Now we construct the vector output function $F = (f_1, \ldots, f_m)$ where, $f_j = h_j \oplus g_j$. For any $\tau \in \mathbb{F}_2^{m*}$, $\bigoplus_{j=1}^m \tau_j f_j(x)$ can be written as $\bigoplus_{j=1}^m \tau_j h_j \oplus \bigoplus_{j=1}^m \tau_j g_j$. This can be done since the set of variables are distinct. The input variables of g_j's are $x_1, \ldots, x_{u+q}$ and the input variables of h_j's are $x_{u+q+1}, \ldots, x_n$.

1. Here, $u \leq n < u+m$. By Proposition 4, we construct $(n = u+q, m, t, 2^{n-1} - 2^{u-1})$ function F.
2. Let $u+m \leq n < u+2m$. Here we take $q = m-1$ in Proposition 2. The functions g_j's are of $u+m-1$ variables. Thus we need to repeat each function $\frac{2^n}{2^{u+m-1}}$ times. We will use functions h_j's of $(n-u-m+1)$ variables which are constant functions. We know, $nl(g_j) = 2^{u+m-2} - 2^{u-1}$. Hence, $nl(f_j) = 2^{n-u-m+1}(2^{u+m-2} - 2^{u-1}) = 2^{n-1} - 2^{n-m}$ as in Corollary 4.
3. Let $u+2m \leq n < u+3m$. We take q such that $n-u-q = 2m$. In this case g_j's are of $u+q$ variables. We take m bent functions h_j's, each of $2m$-variables as in Proposition 6. We know, $nl(g_j) = 2^{u+q-1} - 2^{u-1}$ and $nl(h_j) = 2^{2m-1} - 2^{m-1}$. Thus, if we consider the function $F = (f_1, \ldots, f_m)$, we get, $nl(F) = 2^{n-1} - 2^{u+m-1}$ as in Corollary 2.
4. For $n \geq u+3m-1$, $n-u-m+1$ even, we use $q = m-1$ and a set of bent functions on $n-u-m+1$ variables. Note that in this case $n-u-m+1 \geq 2m$. Thus we will get a set of m bent functions as in Proposition 7. Here, $nl(g_j) = 2^{u+m-1} - 2^{u-1}$ and $nl(h_j) = 2^{(n-u-m+1)-1} - 2^{\frac{n-u-m+1}{2}-1}$. Thus we get, $nl(F) = 2^{n-1} - 2^{\frac{n+u-m-1}{2}}$ as in Corollary 2.

5. For $n \geq u + 3m$, $n - u - m + 1$ odd, we use $q = m - 1$ and a set of bent functions on $n - u - m$ variables, say $b_1, \ldots, b_m$ as in Proposition 7. Note that in this case, $n - u - m \geq 2m$. We construct $h_j = x_n \oplus b_j$. Thus we get, $nl(g_j) = 2^{u+m-1} - 2^{u-1}$ and $nl(h_j) = 2^{(n-u-m+1)-1} - 2^{\frac{(n-u-m+1)-1}{2}}$. In this case, the nonlinearity is $nl(F) = 2^{n-1} - 2^{\frac{n+u-m}{2}}$ as in Corollary 3. □

Note that Corollary 1 in Section 3 is a special case of the item 1 in the above theorem. Next we consider the algebraic degree of functions constructed by means of Theorem 1.

Theorem 2. *In reference to Theorem 1, the algebraic degree of the function F is given by,*

$$2 \leq deg(F) \leq n - u + 1, \quad u \leq n < u + m; \tag{1}$$

$$2 \leq deg(F) \leq m, \quad u + m \leq n < u + 2m; \tag{2}$$

$$deg(F) = \begin{cases} m, & u + 2m \leq n < u + 3m; & (3) \\ \frac{n-u-m+1}{2}, & n \geq u + 3m - 1,\ n - u - m + 1 \quad even; & (4) \\ \frac{n-u-m}{2}, & n \geq u + 3m,\ n - u - m + 1 \quad odd. & (5) \end{cases}$$

Proof. Let us consider any nonzero linear combination f of $(f_1, \ldots, f_m)$. Also we denote any nonzero linear combination of h_j's as h and that of g_j's as g. It is clear that $deg(F) = deg(f) = \max(deg(h), deg(g))$, as h, g are functions on distinct set of input variables.

1. Here f can be seen as the concatenation of 2^q linear functions ($0 \leq q < m$) of u variables each. The exact calculation of algebraic degree will depend in a complicated way on the choice of the codewords from C. However, it is clear that the function is always nonlinear and hence the algebraic degree must be ≥ 2. Also the function f will have degree at most $q + 1$. Here $q = n - u$, which gives the result.
2. In this case $q = m - 1$. Now f can be seen as the 2^{n-u-q} times repetition of function g, where g is the concatenation of 2^q linear functions ($0 \leq q < m$) of u variables each. The exact calculation of algebraic degree will depend in a complicated way on the choice of the codewords from C. However, it is clear that the function is always nonlinear and hence $deg(f) \geq 2$. Furthermore, the function g will have degree at most $q + 1$. Thus the result.
3. In this case $deg(f) = \max(deg(h), deg(g))$. Now, $deg(h) = m$ as we consider $2m$ variable bent functions with property as described in Proposition 6. Also, $deg(g)$ is at most $q + 1$. Now, $u + 2m \leq n < u + 3m$, which gives $q < m$. Hence $deg(f) = m$.
4. In this case $deg(h) = \frac{n-u-m+1}{2}$ (from Proposition 7) and $deg(g) \leq q+1 = m$. Here $n \geq u + 3m - 1$, i.e., $n - u - m + 1 \geq 2m$, which gives $\frac{n-u-m+1}{2} \geq m$. Thus $deg(f) = \frac{n-u-m+1}{2}$.
5. In this case $deg(h) = \frac{n-u-m}{2}$ and $deg(g) \leq q + 1 = m$. Here $n \geq u + 3m$, i.e., $n - u - m \geq 2m$, which gives $\frac{n-u-m}{2} \geq m$. Thus $deg(f) = \frac{n-u-m}{2}$. □

At this point, let us comment on construction of resilient functions of order 1 and 2. First we concentrate on 1-resilient functions. Let C_1 be an $[m+1, m, 2]$ linear code in systematic form, i.e., $C_1 = (I|\mathbf{1})$, where I is an identity matrix of size $m \times m$, and $\mathbf{1}$ is a column vector of all ones. In this case, we have $u = m+1$. Then we can apply Theorem 1 on this $[m+1, m, 2]$ code.

Next we look into the construction of 2-resilient functions. From the theory of error correcting codes we know that for any $l \geq 3$ there exists a linear $[u = 2^l - 1, m = 2^l - l - 1, 3]$ Hamming code. The codewords from such a code provide the construction of $(n, m, 2, nl(F))$ nonlinear resilient functions F. Also, given l, it is possible to obtain a sequence of linear codes of different length and dimension. In other words, given a linear $[2^l - 1, 2^l - l - 1, 3]$ Hamming code the generated sequence of codes is $[2^l - 1 - j, 2^l - l - 1 - j, 3]$, for $j = 0, 1, \ldots, 2^{l-1} - 1$. This code with Theorem 1 can be used to construct 2-resilient functions with high nonlinearity. Note that this construction of 2-resilient functions is not the best using this technique due to the existence of better linear $[n, m, 3]$ codes than those provided by the Hamming design.

The construction of resilient functions using simplex code has been discussed in [5]. A simplex code [13] is a $[2^m - 1, m, 2^{m-1}]$ linear code, whose minimal distance is maximal. By concatenating each codeword v times, one can get a $[v(2^m - 1), m, v2^{m-1}]$ linear code. Given Theorem 1, one can use such codes for construction of functions with order of resiliency $v2^{m-1} - 1$.

Given a linear $[u, m, t+1]$ code, where fixing u, m the maximum possible $t+1$ value can be achieved, will obviously be the most well suited for our construction as this will maximize the order of resiliency. Such table for $u, m \leq 127$ is available in [3].

5 Results and Comparison

In this section we compare the results obtained using the techniques presented in the previous section with the existing results. It was demonstrated that for a low order of resiliency and a moderate number of input variables the construction in [11] was superior to the other constructions, namely the constructions in [12,21]. However, the main disadvantage of the construction in [11] is the necessity of finding a set of nonintersecting linear codes of certain dimension. This may cause a large complexity for the search programs, since there is no theoretical basis for finding such a set. Next we show that our results are superior in comparison to [21,12,5]. Note that the construction of [12] gives higher nonlinearity than [21], whereas the construction of [21] provides larger order of resiliency than [12].

Theorem 3. [21, Corollary 6] *If there exists a linear (n, m, t) resilient function, then there exists a nonlinear $(n, m, t, 2^{n-1} - 2^{n-\frac{m}{2}})$ whose algebraic degree is $m - 1$.*

Note that given any $[u, m, t+1]$ code, it is easy to construct a linear (u, m, t) function. Thus, using the method of [21] it is possible to construct a nonlinear

(u, m, t) function also. Consequently, for $n = u$, the result of [21] provides the presently best known parameters. Note that there are some cases (when the value of n is very close to u, which falls under item 1 of Theorem 1) where the results of [21] are better than ours. This is when $u - 1 > n - \frac{m}{2}$, i.e., $n < u + \frac{m}{2} - 1$. However, if we fix the values of m, t, then for the values of n that falls under items 2, 3, 4 and 5 of Theorem 1 (and also under item 1 when $n \geq u + \frac{m}{2} - 1$), our nonlinearity supersedes that of [21]. Hence, as we choose n comparatively larger than u, $n \geq u + \frac{m}{2} - 1$, the advantage of [21] decreases and our method provides better result. Moreover, the items 3, 4, 5 of Theorem 2 show that the algebraic degree of our construction is better than $(m - 1)$ given in [21]. We present an example here for the comparison.

We know the existence of a $[36, 8, 16]$ linear code. Hence, it is easy to get a linear (36, 8, 15) resilient function. Using the method of [21] it is possible to get a $(36, 8, 15, 2^{36-1} - 2^{36-\frac{8}{2}} = 2^{35} - 2^{32})$ function. Moreover, it has been mentioned in [12, Proposition 19] how to get a $(36, 8, 15, 2^{35} - 2^{31})$ function using the technique of [21]. Our method can not provide a function with these parameters. Let us now construct a function on larger number of input variables, say $n = 43$, for same m and t. For $n = 43$ and $t = 15$ the best known linear code have the parameters $[43, 12, 16]$. Then, with construction in [21], it is possible to construct a $(43, 12, 15, 2^{42} - 2^{37})$ and consequently a $(43, 8, 15, 2^{42} - 2^{37})$ function using less number of output columns. In our construction we start with a $[36, 8, 16]$ code and applying item 1 of Theorem 1 we obtain a $(43, 8, 15, 2^{42} - 2^{35})$ function which provides better nonlinearity.

Theorem 4. [12, Theorem 18] *For any even l such that $l \geq 2m$, if there exists an $(n - l, m, t)$ function $\Phi(x)$, then there exists an $(n, m, t, 2^{n-1} - 2^{n-\frac{l}{2}-1})$ resilient function.*

Note that if there exists a linear $[u = n - l, m, t + 1]$ code, then by the above theorem [12] it is possible to get the nonlinearity $2^{n-1} - 2^{n-\frac{n-u}{2}-1} = 2^{n-1} - 2^{\frac{n+u}{2}-1}$. Items 4 and 5 of our Theorem 1 provide better nonlinearity than [12]. Also a closer look reveals that our construction outperforms the result of [12] for any $n > u$, with same quality result for $n = u + 2m$.

Next we compare our result with a very recent work [5].

Theorem 5. [5, Theorem 5] *Given a linear $[u, m, t + 1]$ code $(0 < m \leq u)$, for any nonnegative integer Δ, there exists a $(u + \Delta + 1, m, t)$ resilient function with algebraic degree Δ, whose nonlinearity is greater than or equal to $2^{u+\Delta} - 2^u \lfloor \sqrt{2^{u+\Delta+1}} \rfloor + 2^{u-1}$.*

Thus it is clear that given a linear $[u, m, t+1]$ code, the above construction provides $(n, m, t, 2^{n-1} - 2^{\frac{n+2u}{2}} + 2^{u-1})$ resilient function. Note that the construction provides some nonlinearity only when $n - 1 \geq \frac{n+2u}{2}$, i.e., $n \geq 2u + 2$. It is very clear that our construction of $(n, m, t, 2^{n-1} - 2^{\lfloor \frac{n+u-m-1}{2} \rfloor})$ resilient functions for $n \geq u + 3m$ presents much better nonlinearity than that of [5]. However, comparing our result in Theorem 2 with [5, Theorem 5], it is clear that in terms of algebraic degree the result of [5] is superior to our result. It will be of interest to

construct functions with nonlinearity as good as our results with better algebraic degree as given in [5].

5.1 Examples

Next we compare the results with specific examples. Let us start with the construction of a $(24, 4, 2, nl(F))$ function $F(x)$. Given $m = 4$, it is possible to construct a nonlinear function $F(x)$ using the technique in [21] with $nl(F) \geq 2^{23} - 2^{22}$. We know the existence of $[7, 4, 3]$ linear Hamming code [13]. This gives $(7, 4, 2)$ resilient function. Using the construction in [12], we obtain a function $F(x)$ with $nl(F) > 2^{23} - 2^{15}$.

In our notation, $u = 7, m = 4, t = 2$. In this case, $n-u-m+1 = 24-7-4+1 = 14$ and $n = 24 \geq u+3m-1 = 18$. Thus from Theorem 1, we get the nonlinearity $2^{23} - 2^{13}$. Thus, our technique provides the currently best known nonlinearity.

Starting with a $[7, 4, 3]$ code, if we use the construction of [5], we will get $(24, 4, 2, 2^{23}-2^{19}+2^6)$ resilient function. To obtain the same value of nonlinearity using the construction in [11], one is forced to find $|\mathcal{C}| = \lceil 2^{n-u'}/(2^m - 1) \rceil = \lceil 2^{10}/15 \rceil$ nonintersecting linear $[14, 4, 3]$ codes, and this is computationally an extremely hard problem to solve.

In [12] the construction of a $(36, 8, 5, nl(F))$ function was discussed. Using a linear $[18, 8, 6]$ code the authors proved the existence of $(36, 8, 5, nl(F))$ function, where $nl(F) \geq 2^{35} - 2^{26}$. We use a linear $[17, 8, 6]$ code [3] to construct a $(36, 8, 5, 2^{35} - 2^{24})$ function (here $n \geq u + 2m$) by means of Theorem 1. Using the same linear code, we can obtain a $(40, 8, 5, 2^{39} - 2^{24})$ function (here $n \geq u + 3m - 1$).

Nonlinearity of $(36, 8, t)$ resilient functions has been used as important examples in [12,11]. We here compare our results with existing ones.

In this table the results of [12] are the existing best known construction results and our results clearly supersede these [12]. The results of [11] are not the construction results. They show that resilient functions with such parameters exist. However, the construction of functions with such parameters are not available in [11]. Note that, for resiliency of orders 3, 2 and 1 our construction provides better results than the existential bound in [11]. In the last row of Table 1, we describe the linear codes [3] which we use for our construction.

Table 1. Nonlinearity of $(36, 8, t)$ resilient functions.

Order of resiliency t	7	5	4	3	2	1
Nonlinearity of [12]	$2^{35} - 2^{27}$	$2^{35} - 2^{26}$	$2^{35} - 2^{25}$	$2^{35} - 2^{24}$	$2^{35} - 2^{23}$	$2^{35} - 2^{22}$
Nonlinearity of [11]	$2^{35} - 2^{22}$	$2^{35} - 2^{23}$	$2^{35} - 2^{22}$	$2^{35} - 2^{22}$	$2^{35} - 2^{21}$	$2^{35} - 2^{21}$
Our nonlinearity	$2^{35} - 2^{27}$	$2^{35} - 2^{24}$	$2^{35} - 2^{23}$	$2^{35} - 2^{20}$	$2^{35} - 2^{19}$	$2^{35} - 2^{18}$
The codes	$[20, 8, 8]$	$[17, 8, 6]$	$[16, 8, 5]$	$[13, 8, 4]$	$[12, 8, 3]$	$[9, 8, 2]$

6 Conclusion

A new generalized construction of highly nonlinear resilient multiple output functions has been provided. The construction is based on the use of linear codes together with a specific set of bent functions. We show that our construction outperforms all previous constructions for almost all choices of input parameters n, m, t. Many examples are provided demonstrating the better nonlinearity attained using this new construction in comparison to the previous ones. It will be of interest to construct functions with better nonlinearity than our method or to show that some of our constructions provide optimized nonlinearity which can not be improved further.

References

1. C. H. Bennet, G. Brassard, and J. M. Robert. Privacy amplification by by public discussion. *SIAM Journal on Computing*, 17:210–229, 1988.
2. J. Bierbrauer, K. Gopalakrishnan, and D. R. Stinson. Bounds on resilient functions and orthogonal arrays. In *Advances in Cryptology - CRYPTO'94*, number 839 in Lecture Notes in Computer Science, pages 247–256. Springer Verlag, 1994.
3. A. Brouwer and T. Verhoeff. An updated table of minimum-distance bounds for binary linear codes. *IEEE Transactions on Information Theory*, 39(2):662–677, 1993.
4. J. H. Cheon and S. Chee. Elliptic Curves and Resilient Functions. In *ICISC 2000*, number 2015 in Lecture Notes in Computer Science, pages 64–72. Springer Verlag, 2000.
5. J. H. Cheon. Nonlinear Vector Resilient Functions. In *Advances in Cryptology - CRYPTO 2001*, Lecture Notes in Computer Science. Springer Verlag, 2001.
6. B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions. In *26th IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.
7. C. Ding, G. Xiao, and W. Shan, The stability theory of stream ciphers, *Number 561, Lecture Notes in Computer Science*, Springer-Verlag, 1991.
8. J. Friedman. On the bit extraction problem. In *33rd IEEE Symposium on Foundations of Computer Science*, pages 314–319, 1982.
9. K. Gopalakrishnan. A study of Correlation-immune, resilient and related cryptographic functions. *PhD thesis, University of Nebraska*, 1994.
10. X. Guo-Zhen and J. Massey. A spectral characterization of correlation immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.
11. T. Johansson and E. Pasalic, A construction of resilient functions with high nonlinearity, In *IEEE International Symposium on Information Theory*, ISIT, June 2000, full version available at *Cryptology ePrint Archive, eprint.iacr.org, No.2000/053*.
12. K. Kurosawa, T. Satoh, and K. Yamamoto Highly nonlinear t-Resilient functions. *Journal of Universal Computer Science*, vol. 3, no. 6, pp. 721–729, Springer Publishing Company, 1997.
13. F. J. MacWillams and N. J. A. Sloane. *The Theory of Error Correcting Codes.* North Holland, 1977.
14. A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of applied cryptography,* CRC Press, 1997.

15. K. Nyberg. Constructions of bent functions and difference sets. In *Advances in Cryptology - EUROCRYPT 1990*, number 473 in Lecture Notes in Computer Science, pages 151–160. Springer Verlag, 1991.
16. P. Sarkar and S. Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. In *Advances in Cryptology - EUROCRYPT 2000*, number 1807 in Lecture Notes in Computer Science, pages 485–506. Springer Verlag, 2000.
17. T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, IT-30(5):776–780, September 1984.
18. T. Siegenthaler, Decrypting a class of stream ciphers using ciphertext only. *IEEE Trans. Comput.*, vol. C-34, pp. 81–85, 1985.
19. D. R. Stinson. Resilient functions and large sets of orthogonal arrays. *Congressus Numerantium*, 92:105–110, 1993.
20. D. R. Stinson and J. L. Massey. An infinite class of counterexamples to a conjecture concerning non-linear resilient functions. *Journal of Cryptology*, 8(3):167–173, 1995.
21. X. M. Zhang and Y. Zheng. Cryptographically resilient functions. *IEEE Transactions on Information Theory*, 43(5):1740–1747, 1997.