Linear Codes in Generalized Construction of Resilient Functions With Very High Nonlinearity

Enes Pasalic and Subhamoy Maitra

***Abstract*—In this paper, we provide a new generalized construction method for 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 highly nonlinear multiple output 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 very high nonlinearity for $n > u$. The method provides the currently best known nonlinearity results for most of the cases.**

***Index Terms*—Correlation immunity, linear codes, nonlinearity, resilient functions, stream ciphers.**

I. 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 the design of stream cipher systems where each key stream bit is added modulo two to each plain text bit in order to produce the cipher text 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 [22], the linear synthesis attack of Berlekamp and Massey [17], and different kinds of approximation attacks [9]. If we use a multiple-output Boolean function instead of a single-output one, it is possible to get more than one bit at each clock pulse 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 since the mid-1980s [8], [1], [10], [23], [2], [11], [25], [15], [14], [6], [7]. The initial works on multiple-output binary resilient functions were directed toward linear resilient functions. The concept of multiple-output resilient functions had been introduced independently by Chor *et al.* [8] and Bennett *et al.* [1]. A similar concept was introduced at the same time for single-output Boolean functions by Siegenthaler [21]. Besides its importance in random sequence generation for stream cipher systems, these resilient functions have applications in quantum cryptographic key distribution, fault-tolerant distributed computing, etc.

Manuscript received November 21, 2001; revised April 21, 2002. The material in this paper was presented in part at the Conference on Selected Areas in Cryptography, SAC 2001, Toronto, ON, Canada, August 16–17, 2001.

E. Pasalic is with the Department of Information Technology, Lund University, 221 00 Lund, Sweden (e-mail: enes@it.lth.se).

S. Maitra is with the Computer and Statistical Service Center, Indian Statistical Institute, Calcutta, Pin 700 108, India (e-mail: subho@isical.ac.in).

Communicated by N. I. Koblitz, Associate Editor for Complexity and Cryptography.

Publisher Item Identifier 10.1109/TIR.2002.800492.

The nonlinearity issue for such multiple-output resilient functions was first discussed in [24] and it has been shown that it is possible to construct infinite classes of nonlinear resilient functions from Kerdock and Preparata codes [16]. After that, serious attempts toward construction of nonlinear resilient functions have been taken in [25], [15], [14], [7].

The connection between resilient functions on the binary alphabet and a large set of orthogonal arrays was established in [23]. In [23], the relation between error-correcting codes and resilient functions has also been considered and codes with minimum distance $t+1$ had been used to construct t-resilient functions. In [1], [10], the following result has been noted. Let G be a generator matrix for an $(n, m, t+1)$ linear code. Then the function F: $\mathbb{F}_2^n \to \mathbb{F}_2^m$, given by $F(x) = xG^T$, is an n-input, m-output, t-resilient function. This result had also been proved in [23] using the orthogonal array characterization.

Note that, by $\mathbb{F}_q^n$ we denote the vector space corresponding to the finite field $\mathbb{F}_{q^n}$.

In [12], Gopalakrishnan and Stinson introduced three characterizations of nonbinary correlation-immune (CI) and resilient functions. They have considered tth-order CI and resilient functions over $\mathbb{F}_q$. The characterizations were in terms of i) structure of a certain associated matrix, ii) Fourier transform, and iii) large sets of orthogonal arrays. Later, in [4], Camion and Canteaut extended the results where they considered functions over any finite alphabet $\mathcal{A}$ endowed by the structure of an Abelian group, i.e., a mapping F: $\mathcal{A}^n \mapsto \mathcal{A}^m$. Moreover, they have identified some tradeoff between the degree of the algebraic normal form and the order of correlation immunity. Note that the concept of nonlinearity order mentioned in [4] is not the same as the way we consider the nonlinearity in this paper. The nonlinearity order of [4] is the algebraic degree, whereas we define nonlinearity as the minimum distance from the set of affine functions.

In [5], Carlet showed that applying a suitable modification to bent functions enables to construct CI and resilient functions over Galois fields, in some cases over Galois rings.

We here concentrate on multiple-output Boolean functions, i.e., we fix the alphabet $\mathcal{A} = \mathbb{F}_2$. Then, our construction provides better results than the existing works [25], [15], [14], [7]. Given the number of input variables n, the number of output variables m, and the order of resiliency t, we can construct functions F: $\mathbb{F}_2^n \to \mathbb{F}_2^m$ that achieve higher nonlinearity values than existing constructions for almost all choices of n, m, and t.

The paper is organized as follows. Section II provides basic definitions and notations both for one-output and m-output functions, $m > 1$. In Section III, we review some important techniques and results used toward the new construction of t-resilient functions. Section IV provides the new constructions

based on the use of linear error-correcting codes together with highly nonlinear functions. Some numerical values for the constructed functions and a comparison with previous constructions are presented in Section V. Section VI concludes this paper.

II. 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 1's in S. This is denoted by $\mathrm{wt}(S)$.

We have already mentioned that by $\mathbf{F}_2^n$ we denote the vector space corresponding to the finite field $\mathbf{F}_{2^n}$. The addition operator over $\mathbf{F}_2$ is denoted by $\oplus$, representing 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 $\mathbf{F}_2^n \mapsto \mathbf{F}_2$. 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 having the form

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

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

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

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

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 $\mathbf{F}_2^n$. The *dot product* of x and ω is defined as

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

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

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

Next we define *correlation immunity* in terms of the characterization provided in [13]. A function $f(x_1, \ldots, x_n)$ is tth-order CI iff its Walsh transform W_f satisfies

$$W_f(\omega) = 0, \qquad \text{for all } \omega \in \mathbf{F}_2^n \text{ such that } 1 \leq \mathrm{wt}(\omega) \leq t.$$

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

$$W_f(\omega) = 0. \qquad \text{for all } \omega \in \mathbf{F}_2^n \text{ such that } 0 \leq \mathrm{wt}(\omega) \leq t. \tag{2}$$

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

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

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

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

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 $\tau \in \mathbf{F}_2^{m^*}$. 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 and given the input parameters n, m, t, we construct the functions with currently best known nonlinearity.

III. 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 [14]. We will use it frequently in our construction, and, therefore, it is stated with the proof. Throughout the paper we consider C to be a binary linear $[n, m, t+1]$ code with a set of basis vectors $c_0, c_1, \ldots, c_{m-1}$.

Proposition 1 [14]: 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

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

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

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

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^*. For convenience, we use a standard index notation to identify the elements of A^*. That is, $a_{i,j}$ denotes the element in the ith row and jth column of A^*, for $i = 0, \ldots, 2^m - 2$ and $j = 0, \ldots, m-1$. Since each entry $a_{i,j}$ of A^* is a nonzero codeword of a linear $[u, m, t+1]$ linear code C, the corresponding linear function $a_{i,j}(x) = x \cdot a_{i,j}$ on L_u will be nondegenerate on at least $t-1$ variables. This linear function is t-resilient from the Walsh transform characterization in (2).

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 [14], it was proved that the existence of a set $\mathcal{C}$ of linear $[u, m, t+1]$ *nonintersecting codes* of cardinality $|\mathcal{C}| = \lceil 2^{u-s}/2^m - 1 \rceil$ was necessary and sufficient for the 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 $C_i \cap C_j = \{0\}$, $1 \le i < j \le s$, is called a set of linear $[u, m, t+1]$ nonintersecting codes.

The results in [14] were obtained using a computer search for the set $\mathcal{C}$. Good results could be obtained only for small size of n, thus not providing a good construction for arbitrary n.

In this initiative our approach is different. We do not try to search for nonintersecting linear codes. We only consider a single linear code with given parameters and use a repetition of the codewords in a specific manner.

Now we analyze the assertion of Proposition 1 even if some rows of the matrix A^* are discarded. Let us only concentrate on the first 2^q rows of A^* for $0 \le q \le m-1$. More formally, we introduce the matrix D with entries $d_{i,j} = a_{i,j}$, $i = 0, \ldots, 2^q - 1$, and $j = 0, \ldots, m-1$. Note that the entries of D are elements from $\mathbb{F}_2^u$ given by $d_{i,j} = \phi(\beta^{i+j})$. In view of Proposition 1, 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. Let the set $\{g_1, \ldots, g_m\}$ of Boolean functions on $u+q$ variables be defined as

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

where $[\tau]$ denotes the integer representation of vector τ, and $j = 0, \ldots, m-1$. That is, to the jth column of D we associate the function g_{j+1}.

Proposition 2: Any nonzero linear combination of the functions $g_1, \ldots, g_m$ is a t-resilient function.

Proof: Let

$$g(y, x) = \bigoplus_{j=1}^{m} \tau_j g_j$$

for some $\tau \in \mathbb{F}_2^m$. We have to prove that $W_g(\omega) = 0$ for any ω with $\mathrm{wt}(\omega) \le t$. Then, for any $(b, a) \in \mathbb{F}_2^q \times \mathbb{F}_2^u$ with $\mathrm{wt}(b, a) \le t$, we have

$$\begin{aligned} W_g(b, a) &= \sum_{y,x} (-1)^{g(y,x) \oplus (b,a)\cdot(y,x)} \\ &= \sum_{y,x} (-1)^{g(y,x)} (-1)^{(b,a)\cdot(y,x)} \\ &= \sum_{y} (-1)^{b \cdot y} \left(\sum_{x} (-1)^{g(y,x) \oplus x \cdot a} \right). \end{aligned} \tag{3}$$

For any fixed y, by Proposition 1, the function

$$g(y, x) = x \cdot \bigoplus_{j=0}^{m-1} c_j d_{[y],j}$$

is a linear function nondegenerate on at least $t+1$ variables. (Here, $c_j = \tau_{j+1}$ for $j = 0, \ldots, m-1$.) Now $\mathrm{wt}(b, a) \le t$ implies that $\mathrm{wt}(a) \le t$, and, consequently, the right-hand sum is zero, completing the proof. □

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 [20], we have $nl(g_j) = 2^{u+q-1} - 2^{u-1}$ for $j = 1, \ldots, m$. Moreover, from Proposition 1, 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 $0 \leq q \leq m-1$.

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: There exists an $[m+1, m, 2]$ 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 a 1-resilient function $F(x)$: $\mathbb{F}_2^{32} \mapsto \mathbb{F}_2^{16}$ with

$$nl(F) = 2^{n-1} - 2^{\frac{n}{2}} = 2^{31} - 2^{16}.$$

This function can be used in a stream cipher system where at each clock pulse it is possible to get 2-byte output.

Next we look into a more involved technique. For this, we present a set of technical results. The following proposition is well known [20] and therefore stated without proof.

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 a higher number of variables. The question is if we may use the same technique for the construction of multiple-output functions. In other words, we want to find a binary vector space of bent functions of dimension m.

We discuss this construction in more detail (see also [18]). Let A be of size $2^m \times m$ given by $A = \left(\frac{0}{A^*}\right)$, 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 each column of the matrix A can be seen as a 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 obtain a binary vector space of bent functions on $2m$ variables of dimension m. Also,

$$\deg\left(\bigoplus_{i=1}^{m} \tau_i b_i\right) = m$$

where $b_1, b_2, \ldots, b_m$ is the basis and $\tau \in \Gamma_2^{m^*}$.

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 Γ_{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 four-variable bent function $h_1(x)$ as the concatenation of the two-variable linear functions $0, x_1, x_2, x_1 \oplus x_2$ and, similarly, $h_2(x)$ as concatenation of $0, x_2, x_1 \oplus x_2, x_1$. Then the function $h_1(x) \oplus h_2(x)$ is also bent, which is a concatenation of $0, x_1 \oplus x_2, x_1, x_2$.

Also note the following update of Proposition 6.

Proposition 7: It is possible to obtain m distinct bent functions on $2p$ variables ($p \geq m$), say $h_1, \ldots, h_m$, such that any nonzero linear combination of these bent functions will provide a bent function. Also,

$$\deg\left(\bigoplus_{i=1}^{m} \tau_i h_i\right) = p. \qquad \text{for } \tau \in \mathbb{F}_2^{m^*}.$$

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

IV. 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 toward construction of resilient functions of specific orders. Let us first discuss the idea informally. We will use the matrix D as described earlier. 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 $2^{u+q-1} - 2^{u-1}$. As a consequence of the discussion in the previous section, these functions are referred to as $g_1, \ldots g_m$. From Proposition 4, it is known that any nonzero linear combination of these functions will provide a

$(u+q)$-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 needs 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 by 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 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 is not possible to get $2m$ bent functions with the desired property. We formalize the results in the following theorem. *For convenience, throughout this section we denote* $\pi = n-u-m+1$.

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

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

Proof: We consider different cases separately. We will use functions $g_1, \ldots, g_m$ on $(u+q)$ variables which are basically concatenations 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 $\tau \in \mathbf{F}_2^m$,

$$nl\left(\bigoplus_{j=1}^{m} \tau_j g_j\right) = 2^{u+q-1} - 2^{u-1}.$$

Next we consider m different functions $h_1, \ldots, h_m$ on $(n-u-q)$ variables. We will choose these functions in such a manner that for any $\tau \in \mathbf{F}_2^m$, $nl(\bigoplus_{j=1}^{m} \tau_j h_j)$ is high. Mostly, we will use bent functions as in Propositions 6 and 7. Now we construct the vector output function $F = (f_1, \ldots, f_m)$ where $f_j = h_j \oplus g_j$. For any $\tau \in \mathbf{F}_2^m$, $\bigoplus_{j=1}^{m} \tau_j f_j(x)$ 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 are $x_1, \ldots, x_{u+q}$ and the input variables of h_j are $x_{u-q+1}, \ldots, x_n$.

1. Here, $u \leq n < u+m$. By Proposition 4, we construct $(n = u+q, m, t, 2^{u-1} - 2^{u-1})$ function F.
2. Let $u+m \leq n < u+2m$. Here we take the first 2^{m-1} rows of A^* in Proposition 1, i.e., $q = m-1$. The functions g_j 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 of $(n-u-m+1)$ variables which are constant functions. We know that $nl(g_j) = 2^{u+m-2} - 2^{u-1}$. 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, the g_j's are of $u+q$ variables. We take m bent functions h_j, each of $2m$ variables as in Proposition 6. We know that
$$nl(g_j) = 2^{u+q-1} - 2^{u-1}$$
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 III is a special case of 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

$$2 \le \deg(F) \le n-u+1, \qquad u \le n < u+m \tag{1}$$

$$2 \le \deg(F) \le m, \qquad u+m \le n < u+2m \tag{2}$$

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

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 $(0 \le q < m)$ of u variables each. The exact calculation of the 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 $(0 \le q < m)$ of u variables each. The exact calculation of the 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 $\deg(f) \ge 2$. 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, $u+2m \le n < u+3m$, which gives $q < m$. Hence $\deg(f) = m$.
4. In this case
$$\deg(h) = \frac{n-u-m+1}{2}$$
(from Proposition 7) and
$$\deg(g) \le q+1 = m.$$
Here $n \ge u+3m-1$, i.e., $n-u-m+1 \ge 2m$, which gives $\frac{n-u-m+1}{2} \ge m$. Thus $\deg(f) = \frac{n-u-m+1}{2}$.
5. In this case
$$\deg(h) = \frac{n-u-m}{2}$$
and
$$\deg(g) \le q+1 = m.$$
Here $n \ge u+3m$, i.e., $n-u-m \ge 2m$, which gives $\frac{n-u-m}{2} \ge m$. Thus $\deg(f) = \frac{n-u-m}{2}$. □

Next we provide results toward further improvement of nonlinearity.

A. Further Improvements

In this subsection, we like to point out that the results of items 2 and 3 in Theorem 1 can be improved further. First we concentrate on the item 2. In the construction, we have considered all the h_j's as $(n-u-m+1)$-variable constant functions, thus without getting any nonlinearity for the h_j's and the linear combinations of them. We explain this with an example. Consider the construction of a $(9, 3, 1)$-resilient function. We start with $[4, 3, 2]$ linear code. Thus, we land into item 2 of Theorem 1 since

$$u+m = 4+3 \le n = 9 < u+2m = 4+6.$$

Hence we get the nonlinearity $2^{9-1} - 2^{9-3} = 192$. Thus we can construct a $(9, 3, 1, 192)$ function. This is because we consider the functions h_1, h_2, h_3 as constant functions on $n-u-m+1 = 3$ variables. The functions g_1, g_2, g_3 are on $u+m-1 = 4+3-1 = 6$ variables and the nonlinearity of any linear combination of them is

$$2^{u+m-2} - 2^{u-1} = 2^5 - 2^3 = 24.$$

However, we can very well use the functions

$$h_1(y_1, y_2, y_3) = y_1y_2 \oplus y_3$$
$$h_2(y_1, y_2, y_3) = y_2y_3 \oplus y_1$$

and

$$h_3(y_1, y_2, y_3) = y_3y_1 \oplus y_2$$

instead of the constant functions. Note that any nonzero linear combination of these three functions will provide nonlinearity 2. Hence using Proposition 5, we get the nonlinearity

$$2^3 \cdot 24 + 2^6 \cdot 2 - 2 \cdot 24 \cdot 2 = 224$$

for any linear combination of f_1, f_2, f_3. This provides $(9, 3, 1, 224)$ functions. This example makes it clear that there is room to improve item 2 of Theorem 1.

Similarly, in item 3 of Theorem 1, we select the value of q such that $n-u-q = 2m$ without choosing $q = m-1$. Let us consider the construction of a $(36, 8, 5)$ function using a $[17, 8, 6]$ linear code. This falls under item 3 of Theorem 1 since

$$u+2m = 17+2 \cdot 8 \le n = 36 < u+3m = 17+3 \cdot 8.$$

The nonlinearity in this case is $2^{n-1} - 2^{u+m-2} = 2^{35} - 2^{24}$. In this case

$$q = n-u-2m = 36-17-2 \cdot 8 = 3.$$

That is, we have used the functions h_j of $2m = 16$ variables and the functions g_j of $u+q = 20$ variables.

Let us now consider the following construction. We will use $q = m-1$, i.e., we will be using the functions g_j on $u+m-1 = 24$ variables and the functions h_j on $n-u-m-1 = 12$ variables. According to Proposition 3, any nonzero linear combination of g_j's has the nonlinearity $2^{u+q-1} - 2^{u-1}$. Now consider the mapping H': $v \to v^{-1}$ where $v \in \Gamma_2^p$, and p is even. It is known that the nonlinearity of the function H' is $2^{p-1} - 2^{\frac{p}{2}}$ [19]. Thus, it is clear that we can construct a function H: $\mathbb{F}_2^p \mapsto \mathbb{F}_2^r$ for even p and $r \le p$ with nonlinearity $2^{p-1} - 2^{\frac{p}{2}}$. In this case, we need to consider eight-output functions on 12 variables. Thus, it is possible to get m functions $h_1, \ldots, h_m$ on $n-u-m-1$ variables such that any nonzero linear combination

of the functions h_j has the nonlinearity $2^{n-u-m}-2^{\frac{n-u-m+1}{2}}$. Using Proposition 5, we get that the nonlinearity of any linear combination of the functions $f_j = h_j \oplus g_j$ is $2^{n-1}-2^{\frac{n+u-m+1}{2}}$. Hence we get a $(36, 8, 5)$ function with nonlinearity $2^{35}-2^{23}$ which improves on Theorem 1.

For odd p, we first consider a function $h: \mathbb{F}_2^{p-1} \mapsto \mathbb{F}_2^r$ with $r \le p-1$. Since $p-1$ is even, we get $nl(h) = 2^{p-2}-2^{\frac{p-1}{2}}$ [19]. Let us denote the r outputs of the function h as $h_1, \ldots, h_r$. Now we take the function $H: \mathbb{F}_2^p \mapsto \mathbb{F}_2^r$ with r output columns as $x_p \oplus h_1, x_p \oplus h_2, \ldots, x_p \oplus h_r$. It is easy to see that $nl(H) = 2^{p-1}-2^{\frac{p-1}{2}}$. Thus, we can summarize the following technical result.

Proposition 8: It is possible to construct a function $H: \mathbb{F}_2^p \to \mathbb{F}_2^r$ for $r \le p$ (respectively, $r \le p-1$) with nonlinearity $2^{p-1}-2^{\frac{p}{2}}$ (respectively, $2^{p-1}-2^{\frac{p+1}{2}}$) for p even (respectively, odd).

Now we update items 2 and 3 of Theorem 1 in the following theorem. Once again note that $\pi = n-u-m+1$.

Theorem 3: Given a linear $[u, m, t+1]$ code, it is possible to construct an $(n, m, t, nl(F))$ function $F = (f_1, \ldots, f_m)$, where

$$nl(F) = \begin{cases} 2^{n-1}-2^{n-m}, & \\ \quad u+m \le n < u+2m-1 & \text{i} \\ 2^{n-1}-2^{\frac{n+u-m-1}{2}}, \ \pi \text{ even}, & \\ \quad u+2m-1 \le n < u+3m-3 & \text{ii} \\ 2^{n-1}-2^{\frac{n+u-m-2}{2}}, \ \pi \text{ odd}, & \\ \quad u+2m \le n < u+3m-3 & \text{iii} \\ 2^{n-1}-2^{u+m-1}, & \\ \quad u+3m-3 \le n < u+3m. & \text{iv} \end{cases}$$

Proof: For the cases i and iv, we keep the same result as in Theorem 1. For case ii, it is clear that $m \le n-u-m+1$. In this case, we use $q = m-1$. The g_j's are on $u+m-1$ variables and h_j's are on $n-u-m-1$ variables. Then, using Proposition 8 we get the result. The result is similar for case iii. Note that we could use the same strategy as in items ii and iii in item iv. However, for the range of n in item iv , the nonlinearity $2^{n-1}-2^{u+m-1}$ supersedes the nonlinearity achievable using the approach of items ii and iii. □

We further like to concentrate on item i of Theorem 3. In this case, $u+m \le n < u+2m-1$, i.e., $1 \le n-u-m+1 < m$. If we like to use the strategy as in items ii and iii of Theorem 3, we need to construct some function $H: \mathbb{F}_2^p \mapsto \mathbb{F}_2^r$ for $r > p$ with some nonlinearity. As far as we know, there is no general strategy to construct such a function. Also, it is clear that for the cases $n-u-m+1 = 1, 2$ there is no possibility to get any nonlinearity. So we can update item i of Theorem 3 as follows.

Proposition 9: Given a linear $[u, m, t+1]$ code, it is possible to construct an $(n, m, t, nl(F))$ function $F = (f_1, \ldots, f_m)$, where

$$nl(F) = \begin{cases} 2^{n-1}-2^{n-m}, & \\ \quad u+m \le n < u+m+2 & \text{I} \\ 2^{n-1}-2^{n-m}+2^u\nu(n-u-m+1, m), & \\ \quad u+m+2 \le n < u+2m-1. & \text{II} \end{cases}$$

Here $\nu(p, r)$ is the maximum possible nonlinearity of a p-input r-output function with $3 \le p < r$.

Proof: Item I is same as item i of Theorem 3. In item II, we consider that we will get $(n-u-m+1)$-variable functions $h_1, \ldots, h_m$ such that any linear combination of them will provide nonlinearity at least $\nu(n-u-m+1, m)$. □

Next we summarize our results combining Theorem 1, Theorem 3, and Proposition 9.

Theorem 4: Given a linear $[u, m, t+1]$ code, it is possible to construct an $(n, m, t, nl(F))$ function $F = (f_1, \ldots, f_m)$, where

$$nl(F) = \begin{cases} 2^{n-1}-2^{u-1}, & \\ \quad u \le n < u+m & 1 \\ 2^{n-1}-2^{n-m}, & \\ \quad u \mid m \le n < u \mid m \mid 2 & 2 \\ 2^{n-1}-2^{n-m}+2^u\nu(n-u-m+1, m), & \\ \quad u+m+2 \le n < u+2m-1 & 3 \\ 2^{n-1}-2^{\frac{n+u+m-1}{2}}, & \\ \quad u+2m-1 \le n < u+3m-3, \ \pi \text{ even} & 4 \\ 2^{n-1}-2^{\frac{n+u+m-2}{2}}, & \\ \quad u+2m \le n < u+3m-3, \ \pi \text{ odd} & 5 \\ 2^{n-1}-2^{u+m-1}, & \\ \quad u+3m-3 \le n < u+3m & 6 \\ 2^{n-1}-2^{\frac{n+u+m-1}{2}}, & \\ \quad n \ge u+3m-1, \ \pi \text{ even} & 7 \\ 2^{n-1}-2^{\frac{n+u+m}{2}}, & \\ \quad n \ge u+3m, \ \pi \text{ odd}. & 8 \end{cases}$$

V. Results and Comparison

First let us 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. Thus, we get the following corollary.

Corollary 5: It is possible to construct an $(n, m, 1, nl(F))$ function $F = (f_1, \ldots, f_m)$, where

$$nl(F) = \begin{cases} 2^{n-1}-2^m, & m+1 \le n < 2m+1 & 1 \\ 2^{n-1}-2^{n-m}, & 2m+1 \le n < 3m+1 & 2 \\ 2^{n-1}-2^{2m}, & 3m+1 \le n < 4m+1 & 3 \\ 2^{n-1}-2^{\frac{n}{2}}, & n \ge 4m, \ n \text{ even} & 4 \\ 2^{n-1}-2^{\frac{n+1}{2}}, & n \ge 4m+1, n \text{ odd}. & 5 \end{cases}$$

We now provide some examples with respect to Corollary 5 and then compare our results with [14]. In Table I, we present the nonlinearity of m-output, 1-resilient functions for $n = 9, 10, 11, 12$. Beside the nonlinearity, we refer to the item number of Corollary 5 in parentheses which is used to calculate the nonlinearity. Also in some entries of the table we provide an improved value of nonlinearity after the / sign. This we explain as follows.

Note that the linear code used for the construction of the $(n, m, 1, nl(F))$ is an $[m+1, m, 2]$ code.

Let us take a closer look at Table I. Consider the construction of 9-variable, 3-output, 1-resilient function. If we construct the function using the linear code $[4, 3, 2]$, then using item 2 of Corollary 5 we get the nonlinearity 192. On the other hand,

TABLE I
NONLINEARITY OF 1-RESILIENT FUNCTIONS

m	$n=9$	$n=10$	$n=11$	$n=12$
2	224 (5)	480 (4)	960 (5)	1984 (4)
3	192 (2)/224	448 (3)/480	960 (3)	1984 (4)
4	224 (2)	448 (2)/480	896 (2)/960	1792 (2)/1984
5	224 (1)	480 (1)	960 (2)	1920 (2)/1984
6	192 (1)	448 (1)	960 (1)	1984 (2)

if we use the linear code $[5, 4, 2]$, we can construct a 9-variable, 4-output, 1-resilient function with nonlinearity 224 taking into consideration item 2 of Corollary 5. Now we can discard an output of a $(9, 4, 1, 224)$ function to get a $(9, 3, 1, 224)$ function. Hence we should use the $[5, 4, 2]$ linear code instead of the $[4, 3, 2]$ linear code to construct a 3-output function. That is, if we get an (n, m, t)-resilient function, then just by discarding some output columns of the function it is possible to get an (n, m_1, t) function with the same nonlinearity for $m_1 < m$. Hence, we present the improved nonlinearity for some entries in Table I after the / sign.

It was demonstrated that for a low order of resiliency and a moderate number of input variables the construction in [14] was superior to the other constructions, namely, the constructions in [15], [25]. However, the main disadvantage of the construction in [14] 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. In Tables II–IV, we compare our results with those of [14] for small values of n.

In the cases marked with ^ (in Tables II–IV), the results of [14] are better than our results, whereas in the cases marked with # our results are better. In the other cases, the quality of the results are the same.

The comparison for 1-resilient functions is presented in Table II. 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 $(u, m, 2, nl(F))$ 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

$$[2^l - 1 - j, 2^l - l - 1 - j, 3], \quad \text{for } j = 0, 1, \ldots, 2^{l-3} - 1.$$

This code can be used with Theorem 1 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.

Further we compare our results for 2-resilient functions with the results of [14] in Table III. For this purpose we use the linear codes $[5, 2, 3]$, $[6, 3, 3]$, $[7, 4, 3]$, $[9, 5, 3]$, and $[10, 6, 3]$.

Now we provide the comparison for 3-resilient functions in Table IV. We use the linear codes $[6, 2, 4]$, $[7, 3, 4]$, $[8, 4, 4]$, $[10, 5, 4]$, and $[10, 6, 4]$.

Given Tables II–IV, it is clear that our results are not as good as the results of [14] in some cases. However, we like to mention once again that the main problem of [14] is the need to obtain a set of nonintersecting linear codes of a certain dimension. So far, to the best of our knowledge, there is no general algorithm for finding such a set in low time complexity. For this reason, the only strategy is to use search programs, which is not feasible as n increases. On the other hand, our method provides a deterministic technique in this direction. For moderate-to-large values of n, the technique of [14] will not work, whereas our method will provide functions with very high nonlinearity. Moreover, even for a small number of variables, we get better results than those of [14] in some cases.

TABLE II
COMPARISON OF RESULTS FOR 1-RESILIENT FUNCTIONS

m	$n=9$		$n=10$		$n=11$		$n=12$	
	Our	[14]	Our	[14]	Our	[14]	Our	[14]
2	224	240^	480	480	960	992^	1984	1984
3	224	224	480	480	960	992^	1984	1984
4	224	224	480#	448	960	960	1984#	1920
5	224	224	480#	448	960	960	1984#	1920
6	192	192	448	448	960	960	1984#	1920

TABLE III
COMPARISON OF RESULTS FOR 2-RESILIENT FUNCTIONS

m	$n=9$		$n=10$		$n=11$		$n=12$	
	Our	[14]	Our	[14]	Our	[14]	Our	[14]
2	192	240^	448	480^	896	992^	1920	1984^
3	192	192	448	448	896	960^	1792	1984^
4	192#	128	448#	384	896	896	1792	1920^
5	0	0	256	256	768	768	1792	1792
6	0	0	0	0	512	512	1536	1536

TABLE IV
COMPARISON OF RESULTS FOR 3-RESILIENT FUNCTIONS

m	$n=9$		$n=10$		$n=11$		$n=12$	
	Our	[14]	Our	[14]	Our	[14]	Our	[14]
2	192	192	384	448^	896	960^	1792	1984^
3	192	192	384	384	896	896	1792	1920^
4	128	128	384#	256	896#	768	1792	1792
5	0	0	256	256	512	512	1536	1536
6	0	0	0	0	0	512^	1024	1024

Next we show that our results are superior in comparison to the generalized constructions provided in [25], [15], and [7]. Note that the construction of [15] gives higher nonlinearity than that in [25], whereas the construction of [25] provides a higher order of resiliency than that of [15]. For the comparison with [25], [15], [7] we use Theorem 1 and show that our results are better.

Theorem 5 [25, Corollary 6]: If there exists a linear (n, m, t)-resilient function, then there exists a nonlinear $(n, m, t, 2^{n-1} - 2^{n-\frac{m}{2}})$ function 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 [25] it is possible to construct a nonlinear (u, m, t) function as well. Consequently, for $n = u$, the result of [25] provides the currently 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 [25] 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 n, $n \geq u + \frac{m}{2} - 1$, our nonlinearity in Theorem 1 supersedes that of [25]. Hence, as we

choose n significantly larger than u, $n \geq u + \frac{m}{2} - 1$, the advantage of [25] decreases and our method provides better result. Moreover, items 3–5 of Theorem 2 show that the algebraic degree of our construction is better than $(m-1)$ given in [25]. We present an example here for comparison.

We know of 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 [25] 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 [15, Proposition 19] how to get a $(36, 8, 15, 2^{35} - 2^{31})$ function using the technique of [25]. Our method cannot provide a function with these parameters. Let us now construct a function on a 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 [25], 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 a smaller 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 6 [15, 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 [15] 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 [15]. Also, a closer look reveals that our construction outperforms the result of [15] for any $n > u$, with same quality result for $n = u + 2m$.

Next we compare our result with a very recent work [7].

Theorem 7 [7, 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 an $(n, m, t, 2^{n-1} - 2^{\frac{n-2u}{2}} \cdot 2^u - 1)$-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 a much better nonlinearity than that of [7]. However, comparing our result in Theorem 2 with [7, Theorem 5], it is clear that in terms of algebraic degree the result of [7] is superior to our result. It will be of interest to construct functions with nonlinearity as good as our results with better algebraic degree than that in [7].

Construction of resilient functions using simplex codes has been discussed in [7]. A simplex code [16] 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 the construction of functions with order of resiliency $v2^{m-2} - 1$. Also, in this case, our method will provide better nonlinearity than that of [7].

Let us consider some linear $[u, m, t+1]$ code, where u, m are fixed. Then we wish to maximize $t+1$, since this in turn will maximize the order of resiliency of the constructed function. A table with maximum value of $t+1$ for $u, m \leq 127$ is available in [3], which we will use in the next subsection.

TABLE V
NONLINEARITY OF $(36, 8, t)$-RESILIENT FUNCTIONS

(A) Order of resiliency t	7	6	5
(B) Nonlinearity of [15]	$2^{35} - 2^{27}$	–	$2^{35} - 2^{26}$
(C) Nonlinearity of [14]	$2^{35} - 2^{22}$	–	$2^{35} - 2^{23}$
(D) Our Nonlinearity	$2^{35} - 2^{25}$	$2^{35} - 2^{24}$	$2^{35} - 2^{23}$
(E) The codes	$[20, 8, 8]$	$[19, 8, 7]$	$[17, 8, 6]$

(A)	4	3	2	1
(B)	$2^{36} - 2^{26}$	$2^{36} - 2^{24}$	$2^{36} - 2^{23}$	$2^{36} - 2^{22}$
(C)	$2^{35} - 2^{22}$	$2^{35} - 2^{22}$	$2^{35} - 2^{21}$	$2^{35} - 2^{21}$
(D)	$2^{35} - 2^{23}$	$2^{35} - 2^{20}$	$2^{35} - 2^{19}$	$2^{35} - 2^{18}$
(E)	$[16, 8, 5]$	$[13, 8, 4]$	$[12, 8, 3]$	$[9, 8, 2]$

A. 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 [25] with $nl(F) \geq 2^{23} - 2^{22}$. We know the existence of a $[7, 4, 3]$ linear Hamming code [16]. This gives a $(7, 4, 2)$ resilient function. Using the construction in [15], 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 item 4 of Theorem 1, we get the nonlinearity $2^{23} - 2^{15}$. Thus, our technique provides the currently best known nonlinearity.

Starting with a $[7, 4, 3]$ code, if we use the construction of [7], we will get a $(24, 4, 2, 2^{23} - 2^{19} + 2^6)$ resilient function. To obtain the same value of nonlinearity using the construction in [14], 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 [15], 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 a $(36, 8, 5, nl(F))$ function, where $nl(F) \geq 2^{35} - 2^{26}$. We use a linear $[17, 8, 6]$ code (see [3]) and item ii of Theorem 3 to construct a $(36, 8, 5, 2^{35} - 2^{23})$ function. Using the same linear code, we can obtain a $(40, 8, 5, 2^{39} - 2^{24})$ function (item 4, Theorem 1).

The nonlinearity of $(36, 8, t)$ resilient functions has been used as important examples in [15], [14]. In Table V, we compare our results with existing ones. The values from [15] are the existing best known construction results and our techniques

clearly supersede these [15]. The results of [14] are not construction results. They show that resilient functions with such parameters exist. However, the construction of functions with such parameters are not available in [14]. Note that for resiliency of orders 3, 2, and 1, our construction provides better results than the existential bound in [14]. For resiliency of order 5 we could construct functions achieving the existential bound of [14]. For resiliency of orders 4 and 7, we could not achieve the existential bound of [14]. However, it should be noted that in all cases we provide the construction with currently best known nonlinearity. In the last row of Table V, we describe the linear codes (see [3]) that we use for our construction.

Remark 1: From the discussion in this subsection, it is clear that except in very few specific cases, our construction provides the best known nonlinearity in general. The result of [25, Theorem 5] provides better result than ours for a small range when $n \leq n + \frac{m}{2} - 1$. There are also a few cases for low values of n, when the results based on exhaustive search in [14] are better than ours. Apart from these specific cases, our technique clearly presents the best possible construction results.

VI. Conclusion

In this paper, we consider multiple-output Boolean functions. A new generalized construction of highly nonlinear resilient functions has been provided. The construction is based on the use of linear codes together with a specific set of highly nonlinear 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 in our method or to show that some of our constructions provide optimal nonlinearity which cannot be improved further.

Acknowledgment

The authors wish to thank Prof. T. Johansson, Department of Information Technology, Lund University, Sweden, for providing them with clear insight into this problem. Also, detailed comments from the anonymous referees significantly improved the technical quality and presentation of the paper.

References

[1] C. H. Bennet, G. Brassard, and J. M. Robert, "Privacy amplification by public discussion," *SIAM J. Comput.*, vol. 17, pp. 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 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1994, vol. 839, pp. 247–256.

[3] A. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 662–677, Mar. 1993.

[4] P. Camion and A. Canteaut, "Correlation-immune and resilient functions over a finite alphabet and their applications in cryptography," *Des., Codes Cryptogr.*, vol. 16, pp. 103–116, 1999.

[5] C. Carlet, "More correlation-immune and resilient functions over Galois fields and Galois rings," in *Advances in Cryptology—CRYPTO'97 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1997, vol. 435, pp. 422–433.

[6] J. H. Cheon and S. Chee, "Elliptic curves and resilient functions," in *Proc. ICISC 2000 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 2000, vol. 2015, pp. 64–72.

[7] J. H. Cheon, "Nonlinear vector resilient functions," in *Advances in Cryptology—CRYPTO 2001 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer Verlag, 2001.

[8] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolensky, "The bit extraction problem or t-resilient functions," in *Proc. 26th IEEE Symp. Foundations of Computer Science*, 1985, pp. 396–407.

[9] C. Ding, G. Xiao, and W. Shan, *The Stability Theory of Stream Ciphers (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1991, vol. 561.

[10] J. Friedman, "On the bit extraction problem," in *Proc. 33rd IEEE Symp. Foundations of Computer Science*, 1982, pp. 314–319.

[11] K. Gopalakrishnan, "A study of correlation-immune, resilient and related cryptographic functions," Ph.D. dissertation, Univ. Nebraska, Lincoln, 1994.

[12] K. Gopalakrishnan and D. R. Stinson, "Three characterization of nonbinary correlation immune and resilient functions," *Des., Codes Cryptogr.*, vol. 5, no. 3, pp. 241–251, 1995.

[13] X. Guo-Zhen and J. Massey, "A spectral characterization of correlation immune combining functions," *IEEE Trans. Inform. Theory*, vol. 34, pp. 569–571, May 1988.

[14] T. Johansson and E. Pasalic, "A construction of resilient functions with high nonlinearity," in *Proc. IEEE Int. Symp. Information Theory*, Sorrento, Italy, June 2000, full version available at Cryptology ePrint Archive, no. 2000/053 eprint.iacr.org.

[15] K. Kurosawa, T. Satoh, and K. Yamamoto, "Highly nonlinear t-resilient functions," *J. Univ. Comput. Sci.*, vol. 3, no. 6, pp. 721–729, 1997.

[16] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.

[17] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC, 1997.

[18] K. Nyberg, "Constructions of bent functions and difference sets," in *Advances in Cryptology—EUROCRYPT 1990 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1991, vol. 473, pp. 151–160.

[19] ——, "Differentially uniform mappings for cryptography," in *Advances in Cryptology—EUROCRYPT 1993 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer Verlag, 1994, vol. 765, pp. 55–64.

[20] P. Sarkar and S. Maitra, "Construction of nonlinear Boolean functions with important cryptographic properties," in *Advances in Cryptology—EUROCRYPT 2000 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer Verlag, 2000, vol. 1807, pp. 485–506.

[21] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 776–780, Sept. 1984.

[22] ——, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. Comput.*, vol. C-34, pp. 81–85, 1985.

[23] D. R. Stinson, "Resilient functions and large sets of orthogonal arrays," *Congressus Numerantium*, vol. 92, pp. 105–110, 1993.

[24] D. R. Stinson and J. L. Massey, "An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions," *J. Cryptol.*, vol. 8, no. 3, pp. 168–173, 1995.

[25] X. M. Zhang and Y. Zheng, "Cryptographically resilient functions," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1740–1747, Sept. 1997.