Constructions of Two Classes of Permutation Polynomials

Chenfan Huang[1,2], Shixiong Xia[1,*], Fengrong Zhang[1,2], Yong Zhou[1]

1. School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China. E-mail: xiasx@cumt.edu.cn, zhfl203@163.com
2. Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China.

Abstract. In this paper, we first investigate the constructions of permutation polynomials of the shape $G(X) \oplus \gamma Tr(H(X)))$ over F_{2^n}. A mapping function which transforms a Boolean function on n variables to a univariate function over F_{2^n} is provided. On basis of the mapping function, we put forward two methods for constructing two classes of univariate functions over F_{2^n}. Further, two classes of permutation polynomials of the shape $G(X) \oplus \gamma Tr(H(X)))$ can be obtained using the two classes of univariate functions. At last, based on the one-to-one correspondence between Boolean permutations and Maiorana-McFarland's (M-M) bent functions, we propose an algorithm to compute the algebraic normal form (ANF) of a $2k$-variable M-M bent function from its truth-table. The complexity of this algorithm is much smaller than that of the Butterfly algorithm which is directly used to compute the ANF of a $2k$-variable M-M bent function from its truth-table.

Keywords : Boolean function, bent function, linear structure, permutation polynomial, linearized polynomial, Trace

1 Introduction

Boolean permutations are used in various different areas and play an important role in the security of cryptosystems. Their most prominent cryptographic applications include the analysis and design of S-boxes in block ciphers. For example, the S-box used in the design of the Advanced Encryption Standard (AES) is a Boolean permutation on 8 variables. The researches on Boolean permutations are paid much attention [7–10]. Charpin and Kyureghyan [11] studied the permutation polynomials of the shape

$$F(X) = G(X) \oplus \gamma Tr(H(X))) \tag{1}$$

over F_{2^n}. They showed that the considered problem is related to finding Boolean functions with linear structures (in terms of linear structures, we can see [12].) and then presented some classes of permutation polynomials by using Boolean functions with linear structures. These were generalized in [13], where $F(X) \in F_{p^n}[X]$, p is any prime number. In addition, Charpin and Kyureghyan [13] used the univariate variables represent to characterize the functions assuming a linear structure. However, the characterization of linear structure of a function over the finite fields becomes difficult as soon as its the expression includes more than two terms. For some specific types of Boolean function,the study of permutation polynomials over F_{p^n} [4–6] has great helped. Recently, Charpin and Sarkar

[14] fully characterized the bilinear polynomial with linear structure and then presented a class of permutation polynomials of the type (1) over F_{2^n}. Moreover, they showed the relation between a Maiorana-McFarland's (M-M) bent function with an affine derivative and a polynomial with a linear structure.

Bent functions are the most famous Boolean functions since they achieve the upper bound on nonlinearity [15]. Bent functions play an important role in the design and analysis of stream ciphers [16] in that they resist linear attacks in the best manner [3, 15]. Although many concrete constructions of bent functions are known [1, 2, 17, 18], the general structure of bent functions is still unclear. In particular a complete classification of bent functions seems hopeless today and it can therefore be useful to focus on special families. When effective constructions are considered, there are two main classes of bent functions, the M-M class and the partial spreads (PS) class.

The M-M class of bent functions was first proposed by Maiorana and McFarland [19]. Based on Walsh-Hadamard matrices (Sylvester-type Hadamard matrices), Preneel *et al.* [20] have presented the truth-tables of all the $2^{2^k}(2^k!)$ M-M bent functions on $2k$ variables since 1990. However, the ANF of a $2k$-variable M-M bent function has not been simply obtained for large k in that *Butterfly algorithm* [21] to compute the ANF of an $2k$-variable Boolean function from its truth-table requires $O(2k2^{2k})$ time. Currently, Butterfly algorithm is still the best algorithm for computing the ANF of a Boolean function from its truth-table. We also know that the complexity of the ANF of a function is coherent with its algebraic complexity, i.e., its implementation with **and/xor** gates. In addition, the algebraic degree of a Boolean function can be directly characterized by its ANF. Hence, it is important to efficiently propose the ANFs of the M-M bent functions.

In this paper, we study the constructions of permutation polynomials of the shape $G(X) \oplus \gamma Tr(H(X)))$ over F_{2^n} and present a algorithm for computing the ANFs of M-M bent functions. Firstly, we present a mapping function which transforms a Boolean function on n variables to a univariate function over F_{2^n}. Moreover, based on the presented mapping function, we propose tow methods for constructing two classes of univariate functions with a linear structure. In addition, we show that

1. For n odd, $2^{2^{n-1}}$ permutation polynomials of type (1) over F_{2^n} can be obtained, where $G(X) = X(X \oplus 1)$. In addition, the permutation polynomials presented in [14, Proposition 5] belong to the set of the $2^{2^{n-1}}$ permutation polynomials.
2. If n is odd, then $2^{2^{n-1}} - 1$ permutation polynomials of type (1) over F_{2^n} can be obtained for any permutation polynomial $G(X)$; If n is even, then $2^{2^{n-1}+1} - 1$ permutation polynomials of type (1) over F_{2^n} can be obtained for any permutation polynomial $G(X)$.

At last, it is shown that the computational complexity of this algorithm is $O(k(k+1)2^k)$ which is much smaller than that of Butterfly algorithm.

2 Preliminaries

Let F_{2^n} be the finite field of 2^n elements. For any set E, we will denote $E\backslash\{0\}$ by E^* and the cardinality of E by $\|E\|$. Any polynomial $F(X) \in F_{2^n}[X]$ defines a function

$$\begin{aligned} F : F_{2^n} &\to F_{2^n} \\ x &\mapsto F(x) \end{aligned}$$

which is called the *associated function* of $F(X)$. Recall that any function of a finite field into itself is a given by a polynomial. Throughout the paper, we identify a polynomial with its associated function. The weight of an integer is the Hamming weight of the 2-adic expression of the integer. The degree of a polynomial $F(X)$ defined over F_{2^n} is the maximum of weights of the exponents of X in $F(X)$. In addition, a *permutation polynomial* over F_{2^n} defines a bijective function from F_{2^n} to itself.

For any k dividing n, the function $Tr_k^n : F_{2^n} \to F_{2^k}$ is defined as

$$Tr_k^n(x) = x \oplus x^{2^k} \oplus x^{2^{2k}} \oplus \ldots \oplus x^{2^{k(n/k-1)}}, x \in F_{2^n}.$$

it will be denoted by $Tr(x)$ when $k = 1$.

Definition 1 *Let m and n be positive integers. Let $F : F_{2^n} \to F_{2^m}$. For $a \in F_{2^n}$, the function D_aF given by*

$$D_aF(x) = F(x) \oplus F(x \oplus a)$$

is called the derivative of F in the direction of a. Further, $a \in F_{2^n}^*$ is said to be a linear structure *of F if the function D_aF is constant. $a \in F_{2^n}^*$ is said to be an* affine derivative *of F if the function D_aF is an affine function.*

By definition, it is clear that if $a \in F_{2^n}$ is a linear structure of F, then

$$F(x) \oplus F(x \oplus a) = F(0) \oplus F(a) = c, \text{ for all } \mathrm{x} \in \mathrm{F}_{2^n},$$

where $c \in F_{2^m}$. a is called *c-linear structure* of F.

Let F_2^n denote the vector space of 2^n binary n-tuples. The vector space F_2^n can easily be identified to the field F_{2^n}. This is done by choose a basis $\{\alpha_1, \ldots, \alpha_n\}$ for F_{2^n} over F_2. Then an element $x \in F_{2^n}$ can be described as $\bigoplus_{i=1}^n x_i\alpha_i$, i.e., we can identify x to the n-tuple

$$(x_1, x_2, \ldots, x_n) \in F_2^n.$$

The number of nonzero x_i's is the Hamming weight of $(x_1, x_2, \ldots, x_n)$, denote by $wt(x_1, x_2, \ldots, x_n)$, and any function $f : F_2^n \to F_2$ is an n-variable Boolean function. Let B_n be the set of all n-variable Boolean functions from F_2^n to F_2. The *Hamming weight* $wt(f)$ of a Boolean function $f \in B_n$ is the weight of its truth-table. The *Hamming distance* $d(f, g)$ between two Boolean functions f and g is the Hamming weight of their difference $f \oplus g$.

Any Boolean function on n variables has a unique representation as a multivariate polynomial over F_2, called the *algebraic normal form*(ANF), of the special form:

$$f(x_1, \ldots, x_n) = \bigoplus_{I \subseteq \{1,2,\ldots,n\}} a_I \prod_{i \in I} x_i$$

where the $a_I \in F_2$. The terms $\prod_{i \in I} x_i$ are called monomials. The *algebraic degree* $\deg(f)$ of a Boolean function f equals the maximum degree of those monomials whose coefficients are nonzero in its algebraic normal form. A Boolean function is affine if it has degree at most 1. The set of all affine functions is denoted by A_n. An affine function with constant term equal to 0 is called a linear function. Any linear function on F_2^n is denoted by $\omega \cdot (x_1, \ldots, x_n) = \omega_1 x_1 \oplus \ldots \oplus \omega_n x_n$ where $\omega = (\omega_1, \ldots, \omega_n) \in F_2^n$.

Definition 2 *Let $x = (x_1, \ldots, x_n) \in F_2^n$. An (n, n)-function $\phi(x)$=$(\phi_1(x), \phi_2(x), \ldots, \phi_n(x))$ is called a* Boolean permutation *if the number of solutions (x) of $\phi(x) = \boldsymbol{a}$ is exactly 1 for any $\boldsymbol{a} \in F_2^n$, where $\phi_1, \ldots, \phi_n$ are its coordinate Boolean function.*

It is well known that there exists a simple divide-and-conquer Butterfly algorithm [21] to compute the ANF of a Boolean function from its truth-table (or vice-versa). In what follows, we first introduce this algorithm.

Butterfly algorithm: For every $u = (u_1, u_2, \ldots, u_n) \in F_2^n$, the coefficient a_u of $\prod_{i \in u} x_i$ in the ANF of f equals

$$\begin{aligned} &\bigoplus_{(x_1,\ldots,x_{n-1}) \preceq (u_1,\ldots,u_{n-1})} [f(x_1, \ldots, x_{n-1}, 0)] \quad if\ u_n = 0 \quad and \\ &\bigoplus_{(x_1,\ldots,x_{n-1}) \preceq (u_1,\ldots,u_{n-1})} [f(x_1, \ldots, x_{n-1}, 0) \\ &\qquad \oplus f(x_1, \ldots, x_{n-1}, 1)] \quad if\ u_n = 1, \end{aligned}$$

where $(x_1, x_2, \ldots, x_n) \preceq u$ if and only if $\sup(x_1, x_2, \ldots, x_n) \subseteq \sup(u)$, $\sup(u) = \{i | u_i \neq 0\}$. Hence if, in the truth-table of f, the binary vectors are ordered in lexicographic order, with the bit of higher weight on the right (for instance), the table of the ANF equals the concatenation of those of the $(n-1)$-variable functions $f(x_1, \ldots, x_{n-1}, 0)$ and $f(x_1, \ldots, x_{n-1}, 0) \oplus f(x_1, \ldots, x_{n-1}, 1)$. We deduce the following recursive algorithm:

1. write the truth-table of f, in which the binary vectors of length n are in lexicographic order as described above;
2. let f_0 be the restriction of f to $F_2^{n-1} \times \{\mathbf{0}\}$ and f_1 the restriction of f to $F_2^{n-1} \times \{\mathbf{1}\}$; the truth-table of f_0 (resp. f_1) corresponds to the upper (resp. lower) half of the table of f; replace the values of f_1 by those of $f_0 \oplus f_1$;
3. apply recursively step 2, separately to the functions now obtained in the places of f_0 and f_1.

When the algorithm ends (i.e., when it arrives to functions on one variable each), the global table gives the values of the ANF of f. The computational complexity of this algorithm is $O(n2^n)$.

3 Permutation Polynomials With Linear Structure

The permutation polynomials of shape

$$F(X) = G(X) \oplus \gamma Tr(H(X)),$$

where $G(X), H(X) \in F_{2^n}[X]$ and $\gamma \in F_{2^n}$, have been studied in [11, 13, 14]. In this section we describe two classes of such permutation polynomials. Before that, we first present two theorems in the following.

Theorem 1 *Let $(y_1, \ldots, y_n) \in F_2^n$ and $x \in F_{2^n}$. Let ψ be a mapping which satisfies*

$$\psi\left(\bigoplus_{I\subseteq\{1,2,\ldots,n\}} a_I \prod_{i\in I} y_i\right) = \bigoplus_{I\subseteq\{1,2,\ldots,n\}} a_I \prod_{i\in I} x^{2^{i-1}}, \tag{2}$$

where the $a_I \in F_2$. Let $h(y_1, \ldots, y_n) \in B_n$ and $H(x) = \psi(h(y_1, \ldots, y_n))$. We have

$$h(y_1, \ldots, y_n) \oplus h((y_1, \ldots, y_n) \oplus \mathbf{1}) = c$$

for all $(y_1, \ldots, y_n) \in F_2^n$ (i.e., $\mathbf{1}$ is a c-linear structure of $h(y_1, \ldots, y_n)$) if and only if

$$H(x) \oplus H(x \oplus 1) = c$$

for all $x \in F_{2^n}$, where $c \in F_2$, $\mathbf{1} = (1, \ldots, 1) \in F_2^n$.

Proof. Without loss of generality, we set $h(y_1, \ldots, y_n) = \bigoplus_{I\subseteq\{1,2,\ldots,n\}} a_I \prod_{i\in I} y_i$. Thus, we have $H(x) = \bigoplus_{I\subseteq\{1,2,\ldots,n\}} a_I \prod_{i\in I} x^{2^{i-1}}$. Furthermore,

$$h((y_1, \ldots, y_n) \oplus \mathbf{1}) = \bigoplus_{I\subseteq\{1,2,\ldots,n\}} a_I \prod_{i\in I} (y_i \oplus 1), \tag{3}$$

and

$$H(x \oplus 1) = \bigoplus_{I\subseteq\{1,2,\ldots,n\}} a_I \prod_{i\in I} (x \oplus 1)^{2^{i-1}}. \tag{4}$$

We also know that F_{2^n} is a finite field with characteristic 2. Therefore, $(x \oplus 1)^{2^{i-1}} = x^{2^{i-1}} \oplus 1$. Moreover, the Equation (4) can be represented as follows:

$$H(x \oplus 1) = \bigoplus_{I\subseteq\{1,2,\ldots,n\}} a_I \prod_{i\in I} (x^{2^{i-1}} \oplus 1). \tag{5}$$

Combining Eqs. (3) and (5), we know that

$$H(x) \oplus H(x \oplus 1) = c$$

if $h(y_1, \ldots, y_n) \oplus h((y_1, \ldots, y_n) \oplus \mathbf{1}) = c$ for all $(y_1, \ldots, y_n) \in F_2^n$, and Vice Versa.

Next, we discuss the properties of the Boolean functions which have a nonzero linear structure.

Theorem 2 *Let $h(y_1, \ldots, y_n) \in B_n$ and $S = \{(y_1, \ldots, y_n) | h(y_1, \ldots, y_n) = 1\}$. Let $\widetilde{S} = \{(y_1, \ldots, y_n) \oplus \mathbf{1} | h(y_1, \ldots, y_n) = 1\}$. Then,*

1. *$S = \widetilde{S}$ if and only if the vector $\mathbf{1}$ is a 0-linear structure of $h(y_1, \ldots, y_n)$.*

2. $S \cup \widetilde{S} = F_2^n$ *if and only if the vector* $\mathbf{1}$ *is a* 1*-linear structure of* $h(y_1, \ldots, y_n)$.

Proof. Clearly, $\widetilde{S} = \{(y_1, \ldots, y_n) | h(y_1 \oplus 1, \ldots, y_n \oplus 1) = 1\}$.

1. According to the definitions of S and $\widetilde{S}$, it is obvious that $S = \widetilde{S}$ if and only if $h(y_1, \ldots, y_n) = h((y_1, \ldots, y_n) \oplus \mathbf{1})$ for all $x \in F_2^n$.
2. From the definitions of S and $\widetilde{S}$, we know that $\parallel S \parallel = \parallel \widetilde{S} \parallel$. If $S \cup \widetilde{S} = F_2^n$, then $\parallel S \parallel = \parallel \widetilde{S} \parallel = 2^{n-1}$ and $S \cap \widetilde{S} = \emptyset$. For any vector $\alpha \in F_2^n$, we have $\alpha \in S$ or $\alpha \in \widetilde{S}$, that is, $h(\alpha) \oplus h(\alpha \oplus \mathbf{1}) = 1$. Conversely, if the vector $\mathbf{1}$ is a 1-linear structure of $h(y_1, \ldots, y_n)$, i.e., $h(y_1, \ldots, y_n) \oplus h((y_1, \ldots, y_n) \oplus \mathbf{1}) = 1$. Further, we know that $h((y_1, \ldots, y_n) \oplus \mathbf{1}) = h(y_1, \ldots, y_n) \oplus 1$ and $wt(h(y_1, \ldots, y_n)) = wt(h((y_1, \ldots, y_n) \oplus \mathbf{1}))$. By the definitions of S and $\widetilde{S}$, we have $S \cup \widetilde{S} = F_2^n$.

3.1 Permutation Polynomials from Boolean Functions with a 1-linear Structure

Let $G(X) = L(X)$ be a linearized polynomials over F_{2^n}. In this subsection, we present a class of permutation polynomials over a finite field. In [11], a class of permutation polynomials was presented by Charpin and Kyureghyan.

Proposition 1 *[11, Lemma 4] Let* $L : F_{2^n} \to F_{2^n}$ *be a linear 2-to-1 mapping with kernel* $\{0, \alpha\}$ *and* $H : F_{2^n} \to F_{2^n}$. *If for some* $\gamma \in F_{2^n}$ *the mapping*

$$N(x) = L(x) \oplus \gamma Tr(H(x))$$

is a permutation of F_{2^n}, *then* γ *does not belong to the image set of* L. *Moreover, for such an element* γ *the mapping* $N(x)$ *is a permutation if and only if* α *is a 1-linear structure of* $Tr(H(x))$.

Based on Proposition 1, Charpin and Sarkar [14] presented a fact as follows.

Corollary 1 *[14] Let* $H : F_{2^n} \to F_{2^n}$ *be a mapping. If* $H(x)$ *has a linear structure* α, *then* α *is also a linear structure* $Tr(H(x))$. *Moreover, if* α *is a 1-linear structure of* $Tr(H(x))$, *then*

$$N(x) = x(x \oplus \alpha) \oplus \gamma Tr(H(x))$$

is a permutation with linear structure α, *where* $Tr(\gamma/\alpha^2) \neq 0$.

Note that γ does not belong to the image set of $x(x \oplus \alpha)$ in that $Tr(\gamma/\alpha^2) \neq 0$ (i.e., $x^2 \oplus \alpha x \oplus \gamma \neq 0$ for any $x \in F_{2^n}$).

The next result is a direct consequence of Proposition 1 and Corollary 1.

Corollary 2 *Let* n *be odd and* $H : F_{2^n} \to F_{2^n}$ *be a mapping. If* 1 *is a* 1*-linear structure of* $H(x)$, *then*

$$N(x) = x(x \oplus 1) \oplus \gamma Tr(H(x)) \tag{6}$$

is a permutation which has 1 as a γ*-linear structure, where* $Tr(\gamma) \neq 0$.

Proof. Clearly, 1 is a 1-linear structure of $Tr(H(x))$ in that n is odd. From Proposition 1 and Corollary 1, $N(x)$ is a permutation, that is, $N(X)$ is a permutation polynomial over F_{2^n}. In addition, $N(x) \oplus N(x \oplus 1) = \gamma(Tr(H(x) \oplus H(x \oplus 1)) = \gamma$, so 1 is a γ-linear structure of $N(x)$.

In the sequel, we put forward a method to construct a class of functions such that they satisfy a stringent condition given in Corollary 2.

From Theorem 1 and Theorem 2, the Boolean function $H(x)$ such that $H(x) \oplus H(x \oplus 1) = 1$ can be easy direct constructed as we show now.

Construction 3 *Let n be a positive integer.*

Step 1 *Set $i = 1$, $S = \emptyset$ and $M = \emptyset$;*
Step 2 *For $i = i + 1$, choose $y^{(i)}$ in $F_2^n \backslash M$;*
Step 3 *Set $M = M \cup \{y^{(i)}, y^{(i)} \oplus \mathbf{1}\}$, $S = S \cup \{y^{(i)}\}$;*
Step 4 *If $i < 2^{n-1}$, goto Step 2; otherwise goto Step 5;*
Step 5 *Let S be the support set of h (i.e., $S = \{(y_1, \ldots, y_n) | h(y_1, \ldots, y_n) = 1\}$). Compute the ANF of $h(y_1, \ldots, y_n)$ by using the Butterfly algorithm;*
Step 6 *Present the function $H(x)$ by using the mapping ϕ defined as in Theorem 1.*

At the end, we can construct a function $H(x)$ which has 1 as a 1-linear structure.

Theorem 4 *Let n be odd. Then we are able to obtain $2^{2^{n-1}}$ permutation polynomials of type (1) over F_{2^n}*

Proof. We know that $F_2^n = \bigcup_{i=1,2,\ldots,2^{n-1}} \{y^{(i)}, y^{(i)} \oplus \mathbf{1}\}$, where $y^{(i)} \neq y^{(j)}$ and $y^{(i)} \oplus \mathbf{1} \neq y^{(j)}$ if $i \neq j$. Therefore, there are $2^{2^{n-1}}$ different sets S such that $S \cup \widetilde{S} = F_2^n$ since there are two possibilities for any pairs $\{y^{(i)}, y^{(i)} \oplus \mathbf{1}\}$. That is to say, based on Theorem 2 and Construction 3, we are able to construct $2^{2^{n-1}}$ functions over F_{2^n}, which have a 1-linear structure. By Corollary 2, we are able to obtain $2^{2^{n-1}}$ permutations of type (6) over F_{2^n} for n odd. Then, $2^{2^{n-1}}$ permutation polynomials of type (1) over F_{2^n} can be obtained.

Remark 1 *For n odd, there are 2^n affine functions on n variables such that $\mathbf{1}$ is their 1-linear structure. Thus, among the constructed $2^{2^{n-1}}$ functions over F_{2^n}, there are $2^{2^{n-1}} - 2^n$ functions which are not affine functions over F_{2^n} and satisfy a stringent condition given in Corollary 2.*

In [14, Proposition 5], we know $N(x) = x(x \oplus 1) \oplus \gamma Tr(H(x))$ is a permutation, where $H(x) = x^s \oplus x^{2^{n-1}}(x^s \oplus (x \oplus 1)^s \oplus 1)$, γ satisfies $Tr(\gamma) \neq 0$, and $1 \leq s \leq 2^n - 2$. From Theorem 1, it is obvious that the permutations presented in [14, Proposition 5] are particular cases of permutations in Theorem 4.

Example 1 *Let $(y_1, y_2, \ldots, y_5) \in F_2^5$. According to Construction 3, we obtain a set $S =$* $\{(0)_2, (3)_2, (5)_2, (6)_2, (8)_2, (14)_2, (15)_2, (18)_2, (19)_2, (20)_2, (21)_2, (22)_2, (24)_2, (27)_2, (29)_2, (30)_2,$

where $(l)_2$ *denotes the binary expression of integer* $l < 32$ *(i.e.,* $(5)_2 = (0,0,1,0,1)$*). By using Butterfly algorithm, we get the ANF of* $h(y_1,\ldots,y_5)$ *as follows:*

$$\begin{aligned} h(y_1,\ldots,y_5) =& y_5y_4y_3y_2 \oplus y_5y_4y_3y_1 \oplus y_5y_4y_2y_1 \oplus y_5y_4y_1 \\ & \oplus y_5y_3y_2y_1 \oplus y_5y_3y_2 \oplus y_4y_3y_2y_1 \oplus y_4y_3y_1 \\ & \oplus y_4y_2y_1 \oplus y_5y_4 \oplus y_5y_1 \oplus y_5 \oplus y_3 \oplus y_2 \oplus y_1 \oplus 1. \end{aligned}$$

Further, by using the mapping ϕ *defined as in Theorem 1, we have*

$$\begin{aligned} H(x) =& x^{30} \oplus x^{29} \oplus x^{27} \oplus x^{25} \oplus x^{23} \oplus x^{22} \oplus x^{24} \oplus x^{17} \\ & \oplus x^{16} \oplus x^{15} \oplus x^{13} \oplus x^{11} \oplus x^4 \oplus x^2 \oplus x^1 \oplus 1. \end{aligned}$$

Thus, we present a permutation polynomial $G(X) = X(X \oplus 1) \oplus \gamma Tr(H(X))$ *over* F_{2^n}*, where*

$$\begin{aligned} H(X) =& X^{30} \oplus X^{29} \oplus X^{27} \oplus X^{25} \oplus X^{23} \oplus X^{22} \oplus X^{24} \oplus X^{17} \\ & \oplus X^{16} \oplus X^{15} \oplus X^{13} \oplus X^{11} \oplus X^4 \oplus X^2 \oplus X^1 \oplus 1. \end{aligned}$$

3.2 Permutation Polynomials from Boolean Functions with a 0-linear Structure

Let $G(X)$ be a permutation polynomial over F_{2^n}. In [11], a class of permutation polynomials over F_{2^n} was presented as follows.

Proposition 2 *[11, Theorem 2] Let* $G(X), H(X) \in F_{2^n}[X]$*,* $\gamma, x \in F_{2^n}$ *and* $G(X)$ *be a permutation polynomial. Then*

$$F(X) = G(X) \oplus \gamma Tr(H(X))$$

is a permutation polynomial over F_{2^n} *if and only if* $H(X) = R(G(X))$*, where* $R(X) \in F_{2^n}[X]$ *and* γ *is a* 0*-linear structure of the Boolean function* $Tr(R(x))$*.*

Charpin and Kyureghyan [11] presented two classes of permutation polynomials of type (1). From Proposition 2, it follows that a new permutation polynomial of type (1) is obtained by substituting $G(X)$ into a permutation polynomial of shape $X \oplus \gamma Tr(R(X))$. Thus, for a given permutation polynomial $G(X)$, a new permutation polynomial $F(X)$ over F_{2^n} can be obtained if we construct a new polynomial $R(X)$ over F_{2^n}.

According to Proposition 2, we have a corollary in the following.

Corollary 3 *Let* $R(X) \in F_{2^n}[X]$*. Then*

$$F(X) = X \oplus Tr(R(X)) \tag{7}$$

is a permutation polynomial over F_{2^n} *if and only if* 1 *is a* 0*-linear structure of the Boolean function* $Tr(R(x))$*.*

Remark 2 *Based on Construction 3, we can obtain* $2^{2^{n-1}}$ *functions* $R(x)$ *with* 1*-linear structure over* F_{2^n}*. Thus,* $2^{2^{n-1}}$ *Boolean functions* $Tr(R(x))$ *with* 0*-linear structure on* n *variables can be presented for even* n*, that is,* $2^{2^{n-1}}$ *permutation polynomials of type (7) over* F_{2^n} *can be proposed for* n *even. Therefore, while* n *is even, we are able to obtain* $2^{2^{n-1}}$ *new permutation polynomials of type (1) over* F_{2^n} *for a given permutation polynomial* $G(X)$ *by using Construction 3.*

Clearly, the permutation polynomial $F(X)$ in Corollary 3 has 1-linear structure. Next we present a construction of $R(x)$ as follows.

Construction 5 *Let n be a positive integer. Let $P < 2^{n-1}$ be a positive integer as well. Let $S = \emptyset$.*

Step 1 *Set $i = 1$;*
Step 2 *For $i = i + 1$, choose $y^{(i)}$ in $F_2^n \backslash S$;*
Step 3 *Set $S = S \cup \{y^{(i)}, y^{(i)} \oplus \mathbf{1}\}$;*
Step 4 *If $i < P$, goto Step 2; otherwise goto Step 5;*
Step 5 *Let M be the support set of $h(y_1, \ldots, y_n)$. Compute the ANF of $h(y_1, \ldots, y_n)$ by using the Butterfly algorithm;*
Step 6 *Present the function $R(x)$ by using the mapping ϕ defined as in Theorem 1.*

For a given P, $\binom{2^{n-1}}{P}$ sets S, such that $S = \widetilde{S}$, can be obtained by using Construction 5.

Theorem 6 *For n odd, we are able to obtain $2^{2^{n-1}} - 1$ permutation polynomials of type (7) over F_{2^n}. For n even, we are able to obtain $2^{2^{n-1}+1} - 1$ permutation polynomials of type (7) over F_{2^n}.*

Proof. From Construction 5, we know that $2^{2^{n-1}} - 1 = \sum_{p=1}^{2^{n-1}} \binom{2^{n-1}}{P}$ sets $S(\subseteq F_2^n)$ such that $S = \widetilde{S}$ can be constructed. That is, $2^{2^{n-1}} - 1$ functions with a 0-linear structure can be constructed. Thus, combining Corollary 3 and Construction 5, we can obtain $2^{2^{n-1}} - 1$ permutation polynomials of type (7) over F_{2^n} for any n.

According to Remark 2, we know that $2^{2^{n-1}}$ permutation polynomials of type (7) over F_{2^n} can be obtained, where n is even. Combining Construction 3 and 5, we are able to construct

$$2^{2^{n-1}+1} - 1 = 2^{2^{n-1}} - 1 + 2^{2^{n-1}}$$

permutation polynomials of type (7) over F_{2^n} for n even.

Remark 3 *From Proposition 2, we know that a new permutation polynomial of type (1) is obtained by substituting $G(X)$ into a permutation polynomial of shape $F(X) = X \oplus Tr(R(X))$. Thus, for any permutation polynomial $G(X)$, $2^{2^{n-1}} - 1$ permutation polynomials of type (1) over F_{2^n} are able to be obtained for n odd, and $2^{2^{n-1}+1} - 1$ permutation polynomials of type (1) over F_{2^n} can be obtained for n even.*

4 Fast Algorithm for Computing Algebraic Normal Form of Maiorana-McFarland's bent Functions

In this section, we exhibit a fast algorithm for computing the ANFs of the M-M bent functions. From now on, we assume that $n = 2k$ and $x = (x_1, \ldots, x_k) \in F_2^k, y = (y_1, \ldots, y_k) \in F_2^k$. In addition, we denote by $\bar{l}$ the 2-adic expression of the integer l (i.e., $\overline{3} = (1, 1, 0, \ldots, 0) \in F_2^k$).

The *nonlinearity* of $f \in B_n$ is its distance from the set of all n-variable affine functions, i.e.,

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

Boolean functions used in cryptographic systems must have high nonlinearity to withstand linear and correlation attacks. It is upper bounded by $2^{n-1} - 2^{n/2-1}$ because of the so-called Parseval's equation [22] $\sum_{\omega \in F_2^n} (W_f(\omega))^2 = 2^{2n}$.

A Boolean function is called **bent function** if its nonlinearity equals $2^{n-1} - 2^{n/2-1}$, where n is even [15].

Bent functions have been widely investigated since the 80s of the last century. The original Maiorana-McFarland class [23] is the set of all the (bent) Boolean functions on $F_2^{2k} = \{(x, y), x, y \in F_2^k\}$ of the form:

$$f(y, x) = \phi(y) \cdot x \oplus g(y)$$

where $\phi(y) = (\phi_1(y), \phi_2(y), \ldots, \phi_k(y))$ is any permutation on F_2^k and $g(y)$ is any Boolean function on F_2^k. In 2004, Carlet [23] indicated that there existed a one-to-one correspondence between Boolean permutations and the original M-M class of bent functions.

Lemma 1 *[23] Let $x \in F_2^k, y \in F_2^k$, $\phi_i(y)$ with $1 \leq i \leq k$ be a k-variable Boolean function, and $g(y)$ be any k-variable Boolean function. A $2k$-variable Boolean function $f(y, x) = \phi(y) \cdot x \oplus g(y)$ is a bent function if and only if*

$$\phi(y) = (\phi_1(y), \phi_2(y), \ldots, \phi_k(y))$$

is a Boolean permutation.

Let $H_n = [h_{ij}]_{2^n \times 2^n}$ be the Walsh-Hadamard matrix that can be recursively defined as

$$H_n = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes H_{n-1}, \quad H_0 = [1].$$

Here $\otimes$ denotes the Kronecker product between matrices. It is easily seen that $H_n^2 = 2^n I_{2^n}$, where I_{2^n} denotes the unit matrix of size 2^n.

Let the matrix $A_n = [a_{ij}]_{2^n \times 2^n}$ be the associated matrix of H_n, where $a_{ij} = \frac{1-h_{ij}}{2}$. That is, if $h_{ij} = 1$ (*resp.* $h_{ij} = 0$), then $a_{ij} = 0$ (*resp.* $a_{ij} = 1$).

As early as in 1990, Preneel *et al.* [20] presented the truth-tables of all the $2^{2^k}(2^k!)$ M-M bent functions on $2k$ variables by using Walsh-Hadamard matrixes.

Lemma 2 *[20] Let k be an integer. Consider the rows of the matrix A_k. The concatenation of the 2^k rows or their complement in arbitrary order results in $2^{2^k}(2^k!)$ different bent functions on $2k$ variables.*

By using Lemma 2, we can obtain the the truth-tables of all the $2^{2^k}(2^k!)$ M-M bent functions on $2k$ variables.

We present an algorithm for computing the ANF of $f(x, y)$, which is a M-M bent function. Before that, we first present a theorem.

Theorem 7 *Let $x, y \in F_2^k$ and $f(x,y) = \phi(y) \cdot x \oplus g(y)$ be a $2k$-variable Boolean function. Let $[f(\bar{l})]$ be the truth-table of $f(x,y)$, where $l = 0, 1, \ldots, 2^{2k}-1$. Then*

$$[f(\overline{0}), f(\overline{2^k}), f(\overline{2 \cdot 2^k}), f(\overline{3 \cdot 2^k}), \ldots, f(\overline{(2^k-1)2^k})]$$

is the truth-table of $g(y)$. Furthermore,

$$[f(\overline{2^{i-1}}), f(\overline{2^k + 2^{i-1}}), \ldots, f(\overline{(2^k-1)2^k + 2^{i-1}})]$$

is the truth-table of $\phi^{(i)}(y) \oplus g(y)$, where $i = 1, 2, \ldots, k$.

Proof. Since $f(x,y) = \phi(y) \cdot x \oplus g(y)$. For $x = (0, \ldots, 0) \in F_2^k$, we have $f(\mathbf{0}, y) = g(y)$. Clearly, $[f(\overline{0}), f(\overline{2^k}), \ldots, f(\overline{(2^k-1)2^k})]$ is the truth-table of $g(y)$.

For $x = e^{(i)} \in F_2^k$, we have $f(e^{(i)}, y) = \phi^{(i)}(y) \oplus g(y)$, where $e^{(i)}$ represents a vector with the ith entry 1 and others 0. It is also clear that

$$[f(\overline{2^{i-1}}), f(\overline{2^k + 2^{i-1}}), \ldots, f(\overline{(2^k-1)2^k + 2^{i-1}})]$$

is the truth-table of $\phi^{(i)}(y) \oplus g(y)$, where $i = 1, 2, \ldots, k$.

According to Lemma 1, we can obtain a Boolean permutation for arbitrary M-M bent function. From Theorem 7, if we have truth-table of a M-M bent function $f(x,y)$ on $2k$ variables, then the truth-table of the k-variable Boolean permutation $\phi(y)$ (which corresponds to $f(x,y)$) can be easily obtained.

By Theorem 7 and Butterfly algorithm, we present an fast algorithm for computing the ANF of a M-M bent function in the following.

Algorithm 1 *Let $x, y \in F_2^k$. Let $[f(\bar{l})]$ be the truth-table of the M-M bent $f(x,y) \in B_{2k}$, where $l = 0, 1, \ldots, 2^{2k}-1$. Thus, the truth tables of the k-variable Boolean permutation $\phi(y)$ (which corresponds to $f(x,y)$) can be obtained. Based on Butterfly Algorithm, we deduce the following algorithm:*

Step 1 *Write the truth-table of $f(\boldsymbol{0}, y) = g(y)$, in which the binary vectors of length k are in lexicographic order as described Algorithm 2;*
Step 2 *Apply the Butterfly algorithm to present the ANF of $g(y)$. Set $i = 1$;*
Step 3 *Exhibit the truth-table of $\phi_i(y) \oplus g(y)$;*
Step 4 *Apply the Butterfly algorithm to present the ANF of $\phi_i(y) \oplus g(y)$, $i = i + 1$;*
Step 5 *If $i \leq k$, goto Step 3; else goto Step 6;*
Step 6 *Obtain the ANF of $f(x,y) = \phi(y) \cdot x \oplus g(y)$.*

When the algorithm ends, the global table gives the values of the ANF of f.

Clearly, using Algorithm 1, the ANF of a M-M bent functions can be computed with a computational complexity $O((k+1)k2^k)$. In terms of the M-M bent functions, the computational complexity of Algorithm 1 is much smaller than the computational complexity $(O(2k \cdot 2^{2k}))$ of the Butterfly algorithm. Thus, for $k < 40$, we can quickly compute the ANF of a M-M bent function on $2k$ variables by using Algorithm 1.

5 Conclusion

In this paper, we presented two classes of permutation polynomials over a finite field. We firstly proposed a mapping which transforms a Boolean function to a univariate function over a finite field. Further, we introduced two effective methods to construct two classes of univariate functions with a linear structure. Based on the two classes of functions, we proposed many permutation polynomials of type (1) over a finite field. At last, we put forward a method for computing the ANF of a M-M bent function. The problem of how to determine whether a given polynomial has a linear structure is a difficult problem that we would like to address in the future.

References

1. Zhang, F., Wei, Y., Pasalic, E., & Xia, S. (2018). Large sets of disjoint spectra plateaued functions inequivalent to partially linear functions. IEEE Transactions on Information Theory, 64(4), 2987-2999.
2. Pasalic, E., Hodi, S., Zhang, F., & Wei, Y. (2018). Bent functions from nonlinear permutations and conversely. Cryptography and Communications, 1-19.
3. Wei, Y., Pasalic, E., Zhang, F., & Hodi, S. (2017). Efficient probabilistic algorithm for estimating the algebraic properties of Boolean functions for large n. Information Sciences, 402, 91-104.
4. Zha, Z., Hu, L., & Zhang, Z. (2018). New results on permutation polynomials of the form (x p m x+) s+ x p m+ x over p 2m. Cryptography and Communications, 10(3), 567-578.
5. Xu, X., Li, C., Zeng, X., & Helleseth, T. (2018). Constructions of complete permutation polynomials. Designs, Codes and Cryptography, 86(12), 2869-2892.
6. Wang, Y., Zha, Z., & Zhang, W. (2018). Six new classes of permutation trinomials over $\mathbb{F}_{3^{3k}}$. Applicable Algebra in Engineering, Communication and Computing, 29(6), 479-499.
7. Zhang, F., Hu, Y., Xie, M., Gao, J., & Wang, Q. (2012). Constructions of cryptographically significant boolean permutations. Appl. Math, 6(1), 117-123.
8. Zha, Z., & Hu, L. (2012). Two classes of permutation polynomials over finite fields. Finite Fields and Their Applications, 18(4), 781-790.
9. Li, N., Helleseth, T., & Tang, X. (2013). Further results on a class of permutation polynomials over finite fields. Finite Fields and Their Applications, 22, 16-23.
10. Tu, Z., Zeng, X., & Hu, L. (2014). Several classes of complete permutation polynomials. Finite Fields and Their Applications, 25, 182-193.
11. Charpin, P., & Kyureghyan, G. M. (2008, September). On a Class of Permutation Polynomials over $\mathbb{F}_{2^n}$. In International Conference on Sequences and Their Applications (pp. 368-376). Springer, Berlin, Heidelberg.
12. Dubuc, S. (2001). Characterization of linear structures. Designs, Codes and Cryptography, 22(1), 33-45.
13. Charpin, P., & Kyureghyan, G. (2009). When does G (x)+ Tr (H (x)) permute Fpn?. Finite Fields and Their Applications, 15(5), 615-632.
14. Charpin, P., & Sarkar, S. (2011). Polynomials with linear structure and MaioranaMcFarland construction. IEEE Transactions on Information Theory, 57(6), 3796-3804.
15. Rothaus, O. S. (1976). On bent functions. Journal of Combinatorial Theory, Series A, 20(3), 300-305.
16. Carlet, C. (2010). Boolean functions for cryptography and error correcting codes. Boolean models and methods in mathematics, computer science, and engineering, 2, 257-397.
17. Carlet, C., & Mesnager, S. (2011). On Dillons class H of bent functions, Niho bent functions and o-polynomials. Journal of Combinatorial Theory, Series A, 118(8), 2392-2410.
18. Meng, Q., Chen, L., & Fu, F. W. (2010). On homogeneous rotation symmetric bent functions. Discrete Applied Mathematics, 158(10), 1111-1117.
19. McFarland, R. L. (1973). A family of difference sets in non-cyclic groups. Journal of Combinatorial Theory, Series A, 15(1), 1-10.

20. Preneel, B., Van Leekwijck, W., Van Linden, L., Govaerts, R., & Vandewalle, J. (1990, May). Propagation characteristics of Boolean functions. In Workshop on the Theory and Application of of Cryptographic Techniques (pp. 161-173). Springer, Berlin, Heidelberg.
21. Jansen, C. J. A. (1989). Investigations on nonlinear streamcipher systems: construction and evaluation methods.
22. MacWilliams, F. J., & Sloane, N. J. A. (1977). The theory of error-correcting codes (Vol. 16). Elsevier.
23. Carlet, C. (2004). On the confusion and diffusion properties of MaioranaMcFarland's and extended MaioranaMcFarland's functions. Journal of complexity, 20(2-3), 182-204.