On the Way of Constructing $2n$-Bit Permutations from n-Bit Ones

Denis Fomin

National Research University Higher School of Economics, Russia
dfomin@hse.ru

June 4, 2019

- Permutation (or S-Box) is one of the basic components of modern symmetric key cryptography
- Permutation is a bijective (generally nonlinear) function over $\mathbb{F}_{2^n}$
- Using S-Boxes is one of the well studied ways to hide the connection between the key and plain text (or provide Shannon's confusion)
- S-Boxes are utilized to provide the only nonlinear part of the symmetric key cryptography

- The security of the symmetric key cryptography functions strongly depends on the properties of the used permutations
- Permutations should be carefully chosen to resist linear, differential and algebraic cryptanalysis
- Cryptographic properties of permutations affects their resistance towards known methods of cryptanalysis

Definition

The Walsh-Hadamard Transform (WHT) $W^S_{a,b}$ of a function S for fixed values $a \in \mathbb{F}_{2^n}$, $b \in \mathbb{F}_{2^m}$ is defined as follows: $W^S_{a,b} = \sum\limits_{x \in \mathbb{F}_{2^n}} (-1)^{\langle a,x \rangle \oplus \langle b,S(x) \rangle}$.

Definition

The linearity L_S of a S is defined as follows: $L_S = \frac{1}{2} \max\limits_{a,b \neq 0} |W_S(a,b)|$.

The nonlinearity of a function S is denoted by N_S and defined by: $N_S = 2^{n-1} - L_S$.

An S-Box with larger nonlinearity has better resistance against linear cryptanalysis.

Definition

The algebraic degree $\deg(S)$ *of a function S is the minimum among all maximum numbers of variables of the terms in the algebraic normal form (ANF) of* $\langle a, S(x)\rangle$ *for all possible values x and* $a \neq 0$*:* $\deg(S) = \min_{a \in \mathbb{F}_{2^m} \setminus 0} \deg\left(\langle a, S(x)\rangle\right)$.

Definition

For a given $a \in \mathbb{F}_{2^m}/0, b \in \mathbb{F}_{2^m}$ *we consider*

$$\delta_S(a,b) = \#\left\{x \in \mathbb{F}_{2_n} | S(x \oplus a) \oplus S(x) = b\right\}.$$

The differential uniformity of an S-Box S is $\delta_S = \max_{a \in \mathbb{F}_{2^m} \setminus 0, b} \delta_S(a,b)$.

An S-Box with smaller differential uniformity has the better resistance against differential cryptanalysis.

Software Implementation

- Precomputed tables (rather fast if they're small enough)
- Bit-sliced implementation (generally faster, secure against cache timing attacks)

Hardware Implementation

- FPGA and ASIC implementation (the smaller nonlinear part the better)

There are several well known ways of building S-Boxes $S : \mathbb{F}_{2^8} \mapsto \mathbb{F}_{2^8}$:

- **Pseudorandom generation**.
  Differential uniformity and nonlinearity $\delta_S \leq 8, N_S \leq 100$. But complex interpolation polynomial and a huge amount of such a permutation.

There are several well known ways of building S-Boxes $S : \mathbb{F}_{2^8} \mapsto \mathbb{F}_{2^8}$:

- **Pseudorandom generation**.
  Differential uniformity and nonlinearity $\delta_S \leq 8, N_S \leq 100$. But complex interpolation polynomial and a huge amount of such a permutation.
- **Heuristic methods**.
  Differential uniformity and nonlinearity are up to $\delta_S = 6, N_S = 104$.
  Complex interpolation polynomial, huge amount of such a permutation but hard to find.

There are several well known ways of building S-Boxes $S : \mathbb{F}_{2^8} \mapsto \mathbb{F}_{2^8}$:

- **Pseudorandom generation**.
  Differential uniformity and nonlinearity $\delta_S \leq 8, N_S \leq 100$. But complex interpolation polynomial and a huge amount of such a permutation.
- **Heuristic methods**.
  Differential uniformity and nonlinearity are up to $\delta_S = 6$, $N_S = 104$.
  Complex interpolation polynomial, huge amount of such a permutation but hard to find.
- **Algebraic constructions**.
  The best and well-known example – monomial permutations. Finite field inversion has best known differential uniformity and nonlinearity: $\delta_S = 4$, $N_S \leq 112$. Simple interpolation polynomial, not many permutations. But finite inversion has a weakens: there exists systems of quadratic equations
  (graph algebraic immunity is equal to 2).

What if we combine the ways?

There are a lot of well-studied ways to construct permutations using functions of smaller dimensions:

- Feistel network[1] (CRYPTON v0.5, Zorro)
- Misty network[1] (Mysty, Kasumi, Fantomas)
- Lai-Massey construction (Whirpool)
- The ones where the XORs have been replaced by finite field multiplications[2]
- SPN network (Iceberg, Khazard, Crypton v1.0)
- Some other constructions[3]

[1] *Construction of Lightweight S-Boxes using Feistel and MISTY structures (Full Version).* Anne Canteaut and Sébastien Duval and Gaëtan Leurent. `eprint.iacr.org/2015/711`

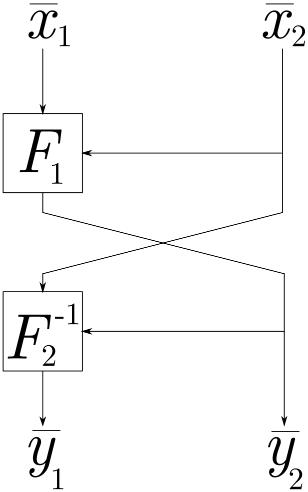
[2] *On some methods for constructing almost optimal S-Boxes and their resilience against side-channel attacks.* Reynier Antonio de la Cruz Jiménez. `eprint.iacr.org/2018/618`

[3] *Differentially 4-Uniform Permutations with the Best Known Nonlinearity from Butterflies.* Shihui Fu and Xiutao Feng and Baofeng Wu. `eprint.iacr.org/2017/449`

Trade-off!

- good software implementation with precomputed tables,
- better bit-sliced implementation and secure against cache timing attacks than those relying on general S-boxes, which require table lookups in memory
- implementation for lightweight cryptography with smaller tables or lower gate count,
- efficient masking in hardware,
- generally has cryptographic properties like random permutation has or better

Two Base Constructions

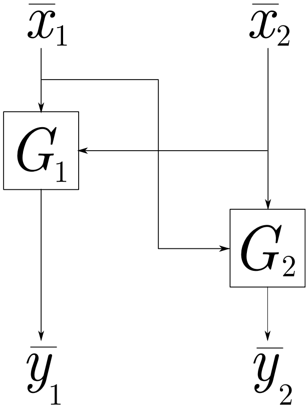


- "F-constructions" (Feistel-like constructions).
- Based on the so-called TU-decomposition.
- Let F be a mapping $\mathbb{F}_{2^m} \times \mathbb{F}_{2^m} \to \mathbb{F}_{2^m} \times \mathbb{F}_{2^m}$ and $F_1,\ F_2 : \mathbb{F}_{2^m} \times \mathbb{F}_{2^m} \to \mathbb{F}_{2^m}$ be the functions with the property: for any fixed value $\overline{v}_2$ the function $F_i(\overline{v}_1, \overline{v}_2)$, $i \in \overline{1,2}$ is a bijection.
- Then the definition $F_2^{-1}(\overline{x}_2, \overline{y}_2) = \overline{y}_1$ is correct.
- $F(\overline{x}_1, \overline{x}_2) = (\overline{y}_1, \overline{y}_2)$, where $\begin{cases} \overline{y}_2 = F_1(\overline{x}_1, \overline{x}_2) \\ \overline{x}_2 = F_2(\overline{y}_1, \overline{y}_2) \end{cases}$

Proposition

The amount of permutations that can be build by using the F-construction is equal to $(2^m!)^{2^{m+1}}$.

Two Base Constructions



- "G-constructions" (Generalised constructions).
- Any permutation $G(\overline{x}_1, \overline{x}_2) = (\overline{y}_1, \overline{y}_2)$ can be represented using mappings G_1 and G_2:

$$\begin{cases} \overline{y}_1 = G_1(\overline{x}_1, \overline{x}_2) \\ \overline{y}_2 = G_2(\overline{y}_1, \overline{y}_2) \end{cases}$$

- Harder to build a permutation using this construction in compare with "F-constructions"

The core question is: “How to choose F_i and G_i?”.

In this work we will use Dobbertin-like functions:

$$s(x,y) = \begin{cases} s'(x,y), \ \pi(y) \neq 0; \\ \widehat{\pi}(x), \ \pi(y) = 0; \end{cases} \qquad ,$$

where $\widehat{\pi}_y(x)$ are permutations over $\mathbb{F}_{2^m}$ and s' is a is a permutation of $x \in \mathbb{F}_{2^m}$ for every fixed value $y \in \mathbb{F}_{2^m} \backslash \dot{y}$.

Value $\dot{y} = \pi^{-1}(0)$ we will call a punctured value of the function s'.

Proposition 1

Let $s(x,y) = \begin{cases} s'(x,y), & \pi(y) \neq 0; \\ \widehat{\pi}(x), & \pi(y) = 0; \end{cases}$ *, where* $\pi, \widehat{\pi} \in S\left(\mathbb{F}_{2^m}\right)$, $s'(x,y) : \mathbb{F}_{2^{2m}} \to \mathbb{F}_{2^m}$ *is a bijection for all* y, $\pi(y) \neq 0$. *Let* $\dot{y} = \pi^{-1}(0)$ *be the punctured value of the function* s *and* $s'\left(x, \dot{y}\right) = 0$. *Then the WHT of the function* $s(x,y)$ *can be calculated as follows:*

$$W^{s}_{\alpha\|\beta,\gamma} = \begin{cases} W^{s'}_{\alpha\|\beta,\gamma} + (-1)^{\langle\beta,\dot{y}\rangle} \cdot W^{\widehat{\pi}}_{\alpha,\gamma}, & \alpha \neq 0; \\ 0, & \alpha = 0, \gamma \neq 0; \\ W^{s'}_{0\|\beta,0}, & \alpha = 0, \gamma = 0. \end{cases}$$

Corollary 2

For chosen Dobbertin-like functions

$$L_s \leq L_{s'} + L_{\widehat{\pi}}$$

The smaller linearity of function s' and permutation π potentially lead to smaller linearity of the function s.

We can choose both functions to be equal the following two functions $s_1, s_2 : \mathbb{F}_{2^m} \times \mathbb{F}_{2^m} \mapsto \mathbb{F}_{2^m}$ and s_i has one punctured value that is defined by permutations π_i:

$$s_1(x,y) = \begin{cases} s_1'(x,y), & \pi_1(y) \neq 0; \\ \widehat{\pi}_1(x), & \pi_1(y) = 0; \end{cases},$$

$$s_2(x,y) = \begin{cases} s_2'\left(y, s_1(x,y)\right), & \pi_2\left(s_1(x,y)\right) \neq 0; \\ \widehat{\pi}_2(y), & \pi_2(s_1(x,y)) = 0; \end{cases},$$

where for all $i \in \overline{1,2}$ $\pi_i, \widehat{\pi}_i \in S\left(\mathbb{F}_{2^m}\right)$, $s_i'(x,y) : \mathbb{F}_{2^{2m}} \to \mathbb{F}_{2^m}$ is a bijection for all $y \neq \pi_i^{-1}(0)$.

Proposition 3

Let $a_1, a_2, b_1, b_2 \in \mathbb{F}_{2^m}$, then the number of solutions of the following system of equations (number of pairs $x, y \in \mathbb{F}_{2^m}$):

$$\begin{cases} s_1(x,y) \oplus s_1(x \oplus a_1, y \oplus a_2) = b_1 \\ s_2(x,y) \oplus s_2(x \oplus a_1, y \oplus a_2) = b_2 \end{cases}$$

greater or equal to the number of solutions of the following system:

$$\begin{cases} \pi_1(y) \neq 0 \\ \pi_1(y \oplus a_2) \neq 0 \\ \pi_2(s_1'(x,y)) \neq 0 \\ \pi_2(s_1'(x \oplus a_1, y \oplus a_2)) \neq 0 \\ s_1'(x,y) \oplus s_1'(x \oplus a_1, y \oplus a_2) = b_1 \\ s_2'(y, s_1'(x,y)) \oplus s_2'(y \oplus a_2, s_1'(x \oplus a_1, y \oplus a_2)) = b_2 \end{cases}$$

Let us consider the algebraic degree of the function (14).

$$\langle a, s(x,y)\rangle = \left\langle a, s'(x,y)\cdot\overline{I_0}(y) + \pi(x)\cdot I_0(y)\right\rangle,$$

where $I_0(y)$ is a function that is equal to 1 only when $\pi(y) = 0$, and equal to 0 otherwise, and function $\overline{I_0}(y)$ is equal to 0 only when $\pi(y) = 0$ and 1 otherwise.

It's quite easy to show that $\deg(I_0) = m$ because $\pi(y)$ is a permutation. At the same time $1 \le \deg(\pi) \le m-1$.

In fact that $I_0(y)$ depends only on y, and $\pi(x)$ depends only on x and if $\deg(\pi) = m-1$ then $\deg(s) = 2m-1$. This property specifies the way of constructing functions with high algebraic degree.

In this work we will focus on the constructions that are similar to the well known Maiorana–McFarland construction: $s'(x, y) = \psi(x) \cdot \phi(y)$, where ψ, ϕ are the permutations over $\mathbb{F}_{2^m}$ and "$\cdot$" is a multiplicative operator of the finite field $\mathbb{F}_{2^m}$.

It's well known that if either ψ or ϕ is a linear permutation, then s' is a bent-function.

Our plan:

- study cryptographic properties but focus on the differential uniformity of the constructions;
- consider the monomial choice of some parameters to simplify the construction;
- find some parameters that provide a way to build permutation with better cryptographic properties in some special cases;
- focus on the most interesting way $m = 4$.

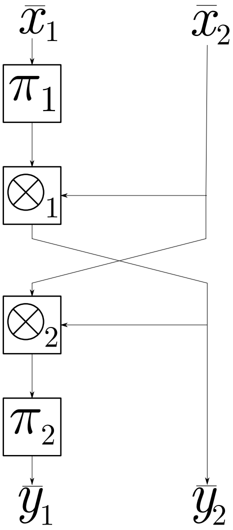
Let us consider the F-construction

$$\overline{y}_2 = F_1\left(\overline{x}_1, \overline{x_2}\right) = \begin{cases} \pi_1\left(\overline{x}_1\right) \cdot \overline{x}_2, \ \overline{x}_2 \neq 0; \\ \widehat{\pi}_1\left(\overline{x}_1\right), \ \overline{x}_2 = 0. \end{cases};$$

$$\overline{x}_2 = F_2\left(\overline{y}_1, \overline{y_2}\right) = \begin{cases} \pi_2\left(\overline{y}_1\right) \cdot \overline{y}_2, \ \overline{y}_2 \neq 0; \\ \widehat{\pi}_2\left(\overline{y}_1\right), \ \overline{y}_2 = 0. \end{cases}.$$

Both F_1 and F_2 are bent functions and could have rather high nonlinearity (with the proper choice of $\widehat{\pi}_i$).

But: $\overline{y}_1 = \pi_2^{-1}\left(\pi_1\left(\overline{x}_1\right)^{-1}\right)$ – depends only on $\overline{x}_1$.



Let us consider the F-construction and $\overline{x}_1, \overline{x}_2 \in \mathbb{F}_{2^m}$ then the permutation $S_A = (\overline{y}_1, \overline{y}_2)$, where

$$\overline{y}_1 = \begin{cases} \pi_2\left(\left(\overline{x}_2\right)^2 \cdot \pi_1\left(\overline{x}_1\right)\right), & \overline{x}_1 \neq 0; \\ \widehat{\pi}_2\left(\overline{x}_2\right), & \overline{x}_1 = 0. \end{cases}$$

$$\overline{y}_2 = \begin{cases} \pi_1\left(\overline{x}_1\right) \cdot \overline{x}_2, & \overline{x}_2 \neq 0; \\ \widehat{\pi}_1\left(\overline{x}_1\right), & \overline{x}_2 = 0. \end{cases}$$

we will call “A”-type permutation.

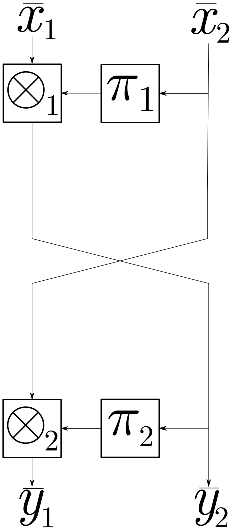
Proposition 4

Let the permutation π_2 is a linear permutation. Then it has differential uniformity larger than $2^m - 2$.

If we suppose that π_1 and π_2 are monomial permutations $\pi_1(x) = x^\alpha$, $\pi_2(x) = x^\beta$ and $m = 4$ then $\alpha \in \{1, 2, 4, 7, 8, 11, 13, 14\}$ and $\beta \in \{7, 11, 13, 14\}$.

S_A with the right choice of $\widehat{\pi}_i$:

- $L_{S_A} = 20$,
- $\delta_{S_A} = 6$,
- $\deg(S_A) = 7$.



Let us consider the F-construction and $\bar{x}_1, \bar{x}_2 \in \mathbb{F}_{2^m}$ then the permutation $S_B = (\bar{y}_1, \bar{y}_2)$, where

$$\bar{y}_1 = \begin{cases} \bar{x}_2 \cdot \pi_2(\bar{y}_2), & \pi_2(\bar{y}_2) \neq 0; \\ \widehat{\pi}_2(\bar{x}_2), & \pi_2(\bar{y}_2) = 0. \end{cases};$$

$$\bar{y}_2 = \begin{cases} \bar{x}_1 \cdot \pi_1(\bar{x}_2), & \pi_1(\bar{x}_2) \neq 0; \\ \widehat{\pi}_1(\bar{x}_1), & \pi_1(\bar{x}_2) = 0. \end{cases}.$$

we will call “B”-type permutation.

Proposition 5

Let $H < S(V_m)$ — be the group of linear permutations. Than if $\pi_2 \in H$ or $\pi_1 \in x^{-1}H$ then $\delta^{S_B} \geq 2^m - 2$.

If we suppose that π_1 and π_2 are monomial permutations $\pi_1(x) = x^\alpha$, $\pi_2(x) = x^\beta$ and $m = 4$.

Proposition 6

Let $m = 4$ and $\pi_1 = x^\alpha$, $\pi_2 = x^\beta$ where α, β: $GCD\left(\alpha, 2^4 - 2\right) = 1$, $GCD\left(\beta, 2^4 - 2\right) = 1$. Than if $\alpha\beta + 1 \neq 14$ then $\delta_{S_B} \geq 2^m - 2$.

The proposition 6 gives us only 4 possible constructions:

1. $\pi_1(x) = x$, $\pi_2(x) = x^{13}$,
2. $\pi_1(x) = x^2$, $\pi_2(x) = x^{14}$,
3. $\pi_1(x) = x^4$, $\pi_2(x) = x^7$,
4. $\pi_1(x) = x^8$, $\pi_2(x) = x^{11}$.

S_B with the right choice of $\widehat{\pi}_i$:

- $L_{S_B} = 20$,
- $\delta_{S_B} = 6$,
- $\deg(S_B) = 7$.

Let’s consider “G”-construction:

$$G_1(\overline{x}_1, \overline{x}_2) = \overline{y}_1 = \begin{cases} \overline{x}_1^{\alpha} \cdot \overline{x}_2^{\beta}, & \overline{x}_2 \neq 0; \\ \widehat{\pi}_1(\overline{x}_1), & \overline{x}_2 = 0. \end{cases}$$

$$G_2(\overline{x}_1, \overline{x}_2) = \overline{y}_2 = \begin{cases} \overline{x}_1^{\gamma} \cdot \overline{x}_2^{\delta}, & \overline{x}_1 \neq 0; \\ \widehat{\pi}_2(\overline{x}_2), & \overline{x}_1 = 0. \end{cases}$$

The equation above defined a permutation iff

$$\begin{cases} G_1(\overline{x}_1, \overline{x}_2) = a_1 \\ G_2(\overline{x}_1, \overline{x}_2) = a_2 \end{cases}$$

has a solution for any $a_1, a_2 \in \mathbb{F}_{2^m}$.

Let’s consider the most interesting case $m = 4$. There are 8^4 sets of $(\alpha, \beta, \gamma, \delta)$ but using equation we can cut this list to 748 possible constructions.

It’s easy to show that set $(\alpha, \beta, \gamma, \delta)$ is linear equivalent to the following sets:

- $(\alpha \cdot d \pmod{2^m - 1}, \beta \cdot d \pmod{2^m - 1}, \gamma \cdot d \pmod{2^m - 1}, \delta \cdot d \pmod{2^m - 1})$ for any $d \in \{1, 2, 4, 8\}$;
- $(\alpha, \beta, \gamma, \delta)$, $(\gamma, \delta, \alpha, \beta)$, $(\beta, \alpha, \delta, \gamma)$, $(\delta, \gamma, \beta, \alpha)$.

Such permutations with the right choice of $\widehat{\pi}_i$:

- $L_G = 20$,
- $\delta_G = 6$,
- $\deg(G) = 7$.

48 classes of permutations:

α	β	γ	δ	α	β	γ	δ	α	β	γ	δ	α	β	γ	δ
1	1	7	11	1	4	7	11	1	11	7	13	1	14	7	7
1	1	7	14	1	4	7	14	1	11	11	14	1	14	11	11
1	1	11	13	1	4	11	7	1	11	13	7	1	14	13	13
1	1	13	14	1	4	13	11	1	11	14	11	1	14	14	14
1	2	7	7	1	7	7	2	1	13	7	8	7	7	7	11
1	2	7	13	1	7	7	11	1	13	7	14	7	7	7	14
1	2	11	11	1	7	11	1	1	13	11	4	7	7	11	13
1	2	11	14	1	7	11	13	1	13	11	7	7	7	13	14
1	2	13	7	1	7	13	8	1	13	13	2	7	11	7	13
1	2	13	13	1	7	13	14	1	13	13	11	7	11	11	14
1	2	14	11	1	7	14	4	1	13	14	1	7	11	13	7
1	2	14	14	1	7	14	7	1	13	14	13	7	11	14	11

- To implement S_A or S_B permutation it is necessary to implement two finite field multipliers (can be a linear function for FPGA) and up to 4 permutations (3 minimum).
- To implement some of G-type permutation it is necessary to implement two finite field multipliers and up to 6 permutations (2 minimum).

We’ve shown some ways to construct permutations that could be a trade-off for security and impelmetation rewirements.

But! It’s still too many questions that should be solved.

Thank you for your attention!

Questions?