Sequences, DFT and Resistance against Fast Algebraic Attacks

Guang Gong

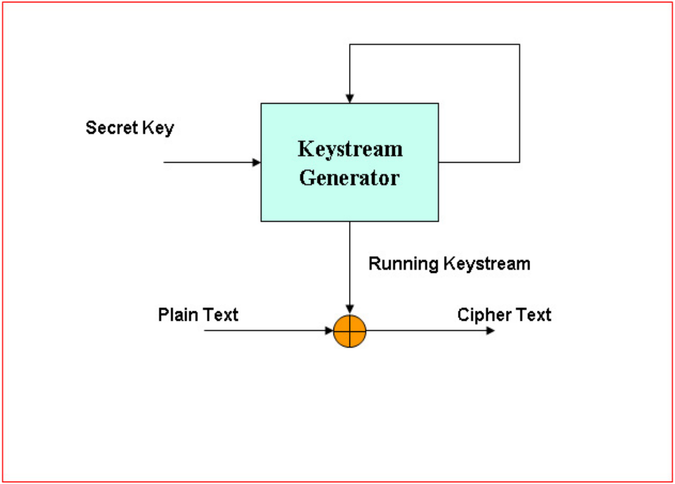
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario N2L 3G1, CANADA
Email. ggong@calliope.uwaterloo.ca

SETA 2008

Outline

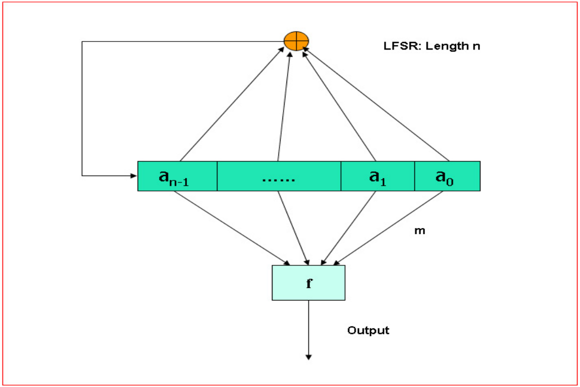
- **Historic Evolutions** of Attacks on Nonlinear Function Generators
- **Basic Definitions**, Properties, and (Discrete) Fourier Transform (DFT) of Sequences
- **Boolean** Bases and **Polynomial** Bases
- **Characterization** of Existence of Fast Algebraic Attacks
- **Resistance** against Fast Algebraic Attacks
- **Discussions**

A General Model of Stream Cipher

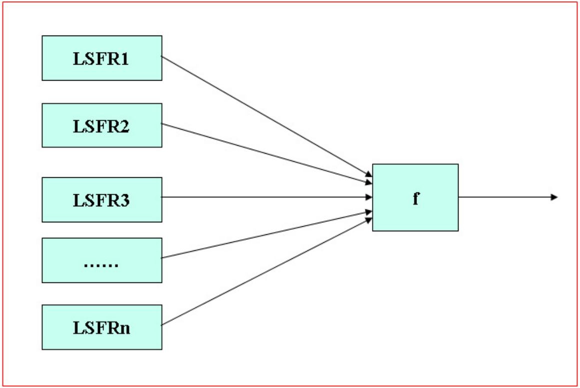


A keystream generator is implemented by a **pseudorandom sequence generator (PSG)**.

Filtering Sequence Generators



Combinatorial Sequence Generators



- Some of those LFSRs can be replaced by **nonlinear FSRs** and those FSRs could be **controlled by each other**, e.g., Grain in ECRYPT.

How to find unknown keys or seeds in a PRG or how to break a stream cipher?

- **A straightforward approach**: obtain equations with unknown keys.
- In a stream cipher model, a ciphertext is a bit stream given by

message bits:	$m_0, m_1, \cdots$	
keystream bits:	$s_0, s_1, \cdots$	
ciphertext bits:	$c_0, c_1, \cdots$	where $c_i = m_i + s_i, i = 0, 1, 2, ..$

- **Known plaintext attacks:** Assume that a certain plaintext is known. Then some bits of $\{s_t\}$ can be recovered.
- Solve the equations with unknown key or the seed from **those known bits** of $\{s_t\}$. If so, the rest of bits of the key stream, i.e., all bits of $\{s_t\}$, is reconstructed.

Known Plaintext Attack on Nonlinear Generators

- An **initial state** of an LFSR or a concatenation of the **initial states** of several LFSRs is a key, denoted by $K = (k_0, k_1, ..., k_{n-1}), k_i \in F_2$.

- Then

$$s_t = f_t(k_0, k_1, ..., k_{n-1}), t = 0, 1, ..., \tag{1}$$

where

$$f_t(x_0, x_1, ..., x_{n-1}) = f(L^t(x_0, x_1, ..., x_{n-1}))$$

which is a system of nonlinear equations in n variables $k_0, \cdots, k_{n-1}$.

- If this system can be solved from a **set of known bits** of $\{s_t\}$, then the rest of bits of the key stream can be reconstructed.

Linearization (1970-)

- **Question**: How can the system be efficiently solved with a subset of known bits of the key stream?
- The system of the equations (1) can be **linearized** when each monomial in $k_{i_1} \cdots k_{i_s}$ is treated as a variable.
- The **number of unknowns** in the system (represented in a boolean form) is varied, but it is dominated by the degree of f.
- For **filtering function generators**, i.e., apply f on m tap positions of an LFSR of degree n, the number of unknowns in the system is $T_{\deg(f)}$ where $\deg(f)$ is the degree of f and T_j:

$$T_j = \sum_{i=0}^{j} \binom{n}{i}.$$

How to reduce the number of unknowns in the system of the linear equations?

Solution: Algebraic Attack (Courtois *et al.* 2003)

- The algebraic attack is to **multiply** f by a function g with a degree lower than f such that the product fg is zero. In other words, using g with $\deg(g) < \deg(f)$ such that $fg = 0$, we have the system of the linear equations as follows

$$s_t g_t(K) = 0, t = 0, 1, \cdots \tag{2}$$

- In this case, the number of the unknowns of (2) is now **dominated by the degree of g** instead of f .
- However,

$$T_{\deg(g)} < T_{\deg(f)}.$$

How to resistance to the algebraic attack (AA)?

- Let $\mathcal{B}_n$ be the set consisting of all boolean functions in n variables.
- The **algebraic immunity** (Meier, Pasalic and Carlet, 2004) of f is defined as the smallest degree $\deg(g)$ such that $fg = 0$ or $(1 + f)g = 0$, denoted by $AI(f)$, i.e.,

$$AI(f) = \min_{g \in Ann(f)} \deg(g),$$

where

$$Ann(f) = \{g \in \mathcal{B}_n \mid fg = 0 \text{ or } g(f + 1) = 0\}.$$

- In order to prevent the AA, the algebraic immunity of the boolean function employed in the system should be high.

Fast Algebraic Attack

Question: If the algebraic immunity (AI) is high, is it possible to obtain a system of the linear equations with the number of unknowns less than the number controlled by the AI?

- Originally, the fast algebraic attack (FAA) (Courtois, 2003) on stream ciphers is to **accelerate the algebraic attack** by introducing linear relations among the key stream bits.
- The **idea** is that if we can find some g such that $fg = h \neq 0$ where $\deg(g) < AI(f)$. In this way, one could further reduce the number of the unknowns in the linear equations.
- **The reason** is

$$h = fg \Rightarrow f(g+h) = 0 \Rightarrow g + h \in Ann(f).$$

Hence $\deg(g+h)$ may be greater than $AI(f)$.

Fast Algebraic Attack (cont.)

- FAA consists of **two steps**:
  - Given (d, e) and f where $d < AI(f)$ and $d < e$, **find a boolean function** g with $d = \deg(g) < \deg(f)$ such that the product $fg = h \neq 0$ with $e = \deg(h) > 0$;
  - **compute** $q(x)$ which is a characteristic polynomial of the output sequence or a factor of it, and **apply** $q(x) = \sum_{i=0}^{r} c_i x_i$ to $s_t g_t(K) = h_t(K)$ which results in

$$\sum_{i=0}^{r} c_i s_{i+t} g_{i+t}(K) = \sum_{i=0}^{r} c_i h_{i+t}(K). \tag{3}$$

- If $v_t = \sum_{i=0}^{r} c_i h_{i+t}(K), t = 0, 1, \ldots$ is equal to zero, then (3) is a system of linear equations in at most T_d variables which can be solved by known $T_d + T_e$ **consecutive bits** (Courtois, 2003).
- If we choose $h(x)$ such that $\{v_t\}$ **is a nonzero** sequence, then (3) can be solved by known T_e **consecutive bits**, which is less than the case that $\{v_t\}$ is a zero sequence (Armknecht and Krause, 2003, Armknecht and Ars, 2004, 2005).
- Using the **polynomial representation (or DFT)**, both the number of the unknowns in the linear equations and required known consecutive bits will be further reduced (Gong, *et al.*, 2008).

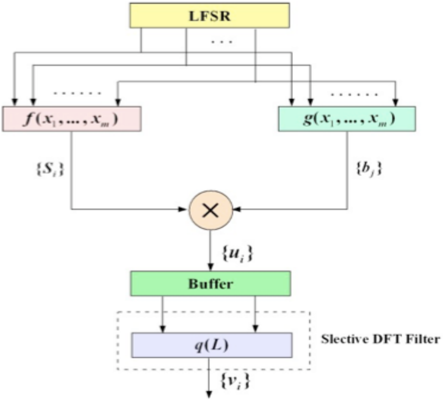


Figure: **Fast Algebraic Attacks on a Filtering Generator**

When is FAA Applicable?

- **Assertion A.** There exists a boolean function g in n variables such that $h = fg$ with $deg(g) \leq d$ **and** $deg(h) \leq e$ when

$d = \lceil \frac{n}{2} \rceil$ and $e = \lceil \frac{n+1}{2} \rceil$	Courtois and Meier, Eurocrypt'2003
$d + e \geq n$	Courtois, Crypto'2003

- For a given boolean function f in n variables and two positive integers d and e, we observed that **the sufficient condition** $d + e \geq n$ **cannot guarantee** the existence of such a function g with $\deg(g) \leq d$ such that $fg \neq 0$ with $\deg(fg) \leq e$.

Linear Feedback Shift Register (LFSR) Sequences

- Let $t(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + 1$ be a polynomial over $\mathbb{F}_2$. A sequence $\mathbf{a} = \{a_t\}$ is an **LFSR sequence** if it satisfies the following recursive relation

$$a_{n+k} = \sum_{i=0}^{n-1} c_i a_{k+i}, k = 0, 1, \cdots , \tag{4}$$

  $(a_0, \cdots , a_{n-1})$ is an **initial state** of $\mathbf{a}$.
- $t(x)$ is called a **characteristic polynomial of a** (the **reciprocal** of $t(x)$ is referred to as a **feedback polynomial of a**, and we also say that $\mathbf{a}$ is generated by $t(x)$. The polynomial with the smallest degree which generates $\mathbf{a}$ is referred to as **minimal polynomial of a**.
- We say that **a is an *m*-sequences** when $t(x)$ is primitive (Golomb, 1954).
- Example: $\mathbf{a} = 1001011$ is an m-sequence of period 7 generated by $t(x) = x^3 + x + 1$.

The Shift Operator

- The **(Left cyclically) shift operator** L:

$$\begin{aligned} L\mathbf{a} &= a_1, a_2, \cdots, \\ L^r\mathbf{a} &= a_r, a_{r+1}, \cdots . \end{aligned}$$

  If $\mathbf{b} = L^r\mathbf{a}$, then we say that they are **shift equivalent**. Otherwise, they are **shift distinct**.

- **Example**: let

$$\begin{aligned} \mathbf{a} &= 1001011 \\ \mathbf{b} &= 1011100 \\ \mathbf{c} &= 1110100 \end{aligned}$$

  then **a and b are shift equivalent**, and **a and c are shift distinct**.

The Decimation Operator

- The k-decimation $\mathbf{a}^{(k)}$ of $\mathbf{a}$ is

$$a_0, a_k, a_{2k}, \ldots$$

  If $\gcd(k, N) = 1$, where N is the (least) period of $\mathbf{a}$, then the (least) period of $\mathbf{a}^{(k)}$ is N.

- **Example**: let

$$\mathbf{a} = 1001011,$$

  then

$$\mathbf{a}^{(3)} = 1110100.$$

  It is of (least) period 7.

Polynomial Functions and Boolean Functions

- Notation:
  - $\mathcal{F}_n$, the set of functions from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$,
  - $\mathcal{B}_n$, the set of **boolean functions** in n variables,
  - $\{\alpha_0, \alpha_1, ..., \alpha_{n-1}\}$, a **basis** of $\mathbb{F}_{2^n}/\mathbb{F}_2$.
- For any function $f(x)$ from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$,

$$f(x) = f(x_0\alpha_0 + x_1\alpha_1 + ... + x_{n-1}\alpha_{n-1}) = g(x_0, x_1, ..., x_{n-1}).$$

  Then

$$\delta : f(x) \to g(x_0, x_1, ..., x_{n-1})$$

  is a **bijective map** from $\mathcal{F}_n$ to $\mathcal{B}_n$.

Cyclotomic Coset

- A cyclotomic coset C_s **modulo $2^n - 1$** is defined by

$$C_s = \{s, s \cdot 2, \cdots, s \cdot 2^{n_s - 1}\},$$

  where n_s is the **smallest** positive integer such that $s \equiv s2^{n_s} \pmod{2^n - 1}$. The subscript s is chosen as the smallest integer in C_s, and s is called the **coset leader** of C_s.
- $\Gamma(n)$ represents the set consisting of all coset leaders modulo $2^n - 1$.
- **Example**: the cyclotomic cosets modulo 15 are:

$$\begin{aligned} C_0 &= \{0\}, \\ C_1 &= \{1, 2, 4, 8\}, \\ C_3 &= \{3, 6, 12, 9\}, \\ C_5 &= \{5, 10\}, \\ C_7 &= \{7, 14, 13, 11\} \end{aligned}$$

  where $\Gamma(4) = \{0, 1, 3, 5, 7\}$.

DFT of Sequences

- Let α be a primitive element in $\mathbb{F}_{2^n}$. We associate $f(x)$ with a binary sequence $\mathbf{a} = \{a_t\}$ whose elements are given by

$$a_t = f(\alpha^t), t = 0, 1, \cdots, 2^n - 2.$$

  Then the period of $\{a_t\}$ is a factor of $2^n - 1$.

- The DFT of $\mathbf{a}_f$ is defined by

$$A_k = \sum_{t=0}^{2^n-2} a_t \alpha^{-tk}, 0 \leq k < 2^n - 1,$$

  and the inverse DFT of $\mathbf{a}_f$ is given by

$$a_t = \sum_{k=0}^{2^n-2} A_k \alpha^{kt} = \sum_{k \in \Gamma(n)} Tr_1^{n_k}(A_k \alpha^{kt}), 0 \leq t < 2^n - 2.$$

- The above representation is called the trace representation of the sequence $\mathbf{a}$.

One-to-one Correspondence between Sequences, Polynomial Functions and Boolean Functions

- Let $\mathcal{S}_n$ be the set of binary sequences with period $N \mid 2^n - 1$, and $\mathcal{F}_n^-$ be the set of functions $f(x)$ from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$ with $f(0) = 0$. Then there exists a **one-to-one correspondence** between $\mathcal{S}_n$ and $\mathcal{F}_n^-$.

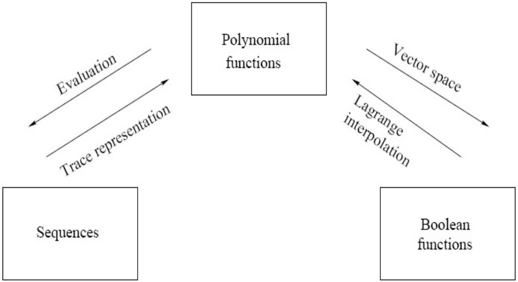


Figure: Correspondence among $\mathcal{S}_n$, $\mathcal{F}_n$, and $\mathcal{B}_n$.

Boolean Bases

- Let f be of a boolean representation. We list the elements in $\mathbb{F}_2^n$ in the same order as the truth table of f. Thus,

$$f(x_0, x_1, \cdots, x_{n-1}) = (f(\mathbf{t}_0), f(\mathbf{t}_1), \cdots, f(\mathbf{t}_{2^n-1})),$$

where $\mathbf{t}_i = (t_{i,0}, t_{i,1}, \cdots, t_{i,n-1}),\ t_{ij} \in \mathbb{F}_2$, and
$i = t_{i,0} + t_{i,1}2 + \cdots + t_{i,n-1}2^{n-1}, 0 \leq i < 2^n$.

- For $\mathbf{x} = (x_0, \cdots, x_{n-1})$ and $\mathbf{c} = (c_0, \cdots, c_{n-1})$ in $\mathbb{F}_2^n$, we denote $\mathbf{x}^{\mathbf{c}} = x_0^{c_0} x_1^{c_1} \cdots x_{n-1}^{c_{n-1}}$. Then, the basis Δ of $\mathcal{F}_n$, regarded as a linear space over $\mathbb{F}_2$, consists of all monomial terms:

$$\Delta = \{\mathbf{x}^{\mathbf{c}} \mid \mathbf{c} \in \mathbb{F}_2^n\}.$$

This basis is referred to as a **boolean basis** of $\mathcal{F}_n$.

Polynomial Bases

- Let

$$\Pi_k = \{ Tr_1^{n_k}(\beta_k(\alpha^i x)^k) \,|\, i = 0, 1, \cdots, n_k - 1\}, \beta_k \in \mathbb{F}_{2^{n_k}}^*$$

  where n_k is the size of the **coset containing** k.

- Note that $\{\alpha^{ik} \mid i = 0, 1, \cdots, n_k - 1\}$ is a **basis** of $\mathbb{F}_{2^{n_k}}$ over $\mathbb{F}_2$, so is $\{c\alpha^{ik} \mid i = 0, 1, \cdots, n_k - 1\}$ for any nonzero $c \in \mathbb{F}_{2^{n_k}}$.

- For each **trace monomial term** $Tr_1^{n_k}(A_k x^k)$, since $A_k \in \mathbb{F}_{2^{n_k}}$, we have $A_k = \sum_{i=0}^{n_k-1} c_i \beta_k \alpha^{ik}$, $c_i \in \mathbb{F}_2$. Using the linear property of the trace function, we have

$$Tr_1^{n_k}(A_k x^k) = \sum_{i=0}^{n_k-1} Tr_1^{n_k}(c_i \beta_k (\alpha^i x)^k).$$

Polynomial Bases (cont.)

- Since any function in $\mathcal{F}_n$ can be represented as a **sum of the trace monomial terms**, the following set is a basis of $\mathcal{F}_n$:

$$\Pi = \bigcup_{k \in \Gamma(n)} \Pi_k.$$

  This basis is referred to as a **polynomial basis** of $\mathcal{F}_n$.

- Let $\mathbf{a}_k = \{a_{kt}\}_{t \geq 0}$. Then $\mathbf{a}_k$ is an LFSR sequence, generated by f_{α_k}, the **minimal polynomial of** α^k. Let $\{A_{k,j}\}$ be the DFT of $\mathbf{a}_k$. Then

$$A_{k,j} = \begin{cases} A_k & \textbf{if } j = k \\ 0 & \textbf{otherwise.} \end{cases}$$

- In other words, the trace representation of **a** can be considered as a **direct sum** of the LFSR sequences with irreducible minimal polynomials for which the DFT sequences of any two of them are **orthogonal**.

Efficient Computation of Polynomial Bases

- Let $\mathbf{a} = \{a_t\}$ be an **m-sequence of degree n**. Then we have $a_t = Tr_1^n(\beta\alpha^t)$ where $\beta \in \mathbb{F}_{2^n}^*$.
- We can assume that $\mathbf{a}^{(k)} \neq \mathbf{0}$, the k-decimation of $\mathbf{a}$, by a proper choice of the initial state of $\mathbf{a}$. Therefore,

$$a_{kt} = Tr_1^{n_k}(\beta_k\alpha^{kt}), t = 0, 1, \cdots,$$

where n_k is the **size** of the coset C_k.

Efficient Computation of Polynomial Bases (cont.)

- Denote

$$P_k = \begin{bmatrix} 0, & \mathbf{a}^{(k)} \\ 0, & L\mathbf{a}^{(k)} \\ \vdots & \vdots \\ 0, & L^{n_k-1}\mathbf{a}^{(k)} \end{bmatrix}$$

where $L^i\mathbf{a}^{(k)}$'s are regarded as binary vectors of dimension $2^n - 1$, and **each row** corresponds to **a function** in Π_k.

- For each coset leader k modulo $2^n - 1$, we only need to compute $\mathbf{a}^{(k)}$, the rest of the rows in P_k can be obtained by the **shift operator** which has no cost.

- For computing the polynomial basis of $\mathcal{F}_n$, one only needs to compute $|\Gamma(n)|$ **decimation sequences** from $\mathbf{a}$, approximately, $2^n/n$ decimation sequences from $\mathbf{a}$.

Example 1

- For $n = 3$, we have $\Gamma(3) = \{0, 1, 3\}$. Let $\mathbf{a} = 1001011$ be an m-sequence of period 7 generated by with $x^3 + x + 1 = 0$.
- Then $\mathbf{a}^{(3)} = 1110100$.

$$P_0 = \begin{bmatrix} 11111111 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 0||\mathbf{a} \\ 0||L\mathbf{a} \\ 0||L^2\mathbf{a} \end{bmatrix} = \begin{bmatrix} 01001011 \\ 00010111 \\ 00101110 \end{bmatrix}.$$

$$P_3 = \begin{bmatrix} 0||\mathbf{a}^{(3)} \\ 0||L\mathbf{a}^{(3)} \\ 0||L^2\mathbf{a}^{(3)} \end{bmatrix} = \begin{bmatrix} 01110100 \\ 01101001 \\ 01010011 \end{bmatrix}.$$

We are ready to answer the question about when the FAA is applicable

- **Question.** For a given f, a boolean function in n variables, and a pair of positive integers (d, e) with $d < deg(f)$, what is the sufficient and necessary condition that there exists a function g such that

$$(I) \qquad deg(g) = d \textbf{ and } h = fg \textbf{ with } deg(h) = e?$$

- **Existing solution** (Courtois *et al.*): if $d + e \geq n$, then (I) is true.

Low Degree Multiplier

Definition

Given f a boolean function in n variables, and a pair of positive integers (d, e) with $d < deg(f)$, if there exists some function g with $deg(g) \leq d$ such that the product

$$h = fg \neq 0 \text{ or } h = (f+1)g \neq 0$$

with $deg(h) \leq e$, then g is said to be a **low degree multiplier** of f.

Some Notations

- Let
$$\mathcal{S}_d = \{ \textbf{row vectors of } P_k \mid k \in \Gamma(n), w(k) \leq d\}.$$
Then S_d can be considered as either the set consisting of all the functions in the **polynomial basis** with $w(k) \leq r$ or the set consisting of all **boolean monomial terms** of degrees less than or equal to d.
- **Notice** that any function in $\mathcal{F}_n$ of degree d is a **linear combination** of functions in S_d over $\mathbb{F}_2$.
- The sequence version of S_d, still denoted by S_d, is given by
$$S_d = \{L^i \mathbf{a}^{(k)} \mid 0 \leq i < n_k, w(k) \leq d\}$$
where n_k is the **size** of the coset C_k or the **degree** of the minimal polynomial of $\mathbf{a}^{(k)}$.
- For a given $f \in \mathcal{F}_n$, we denote
$$fS_d = \{fg \mid g \in S_d\}.$$

Observations

- **Degenerated Cases:** It is possible that $|fS_d| < |S_d|$ and the elements in fS_d are linearly dependent over $\mathbb{F}_2$.
- In this case, possibly, there is **no function** g with $\deg(g) \leq d$ such that $fg \neq 0$ and $\deg(fg) \leq e$.

Existence of Low Degree Multipliers

Theorem

For a given $f \in \mathcal{F}_n$ and a pair of positive integers (d, e) with $1 \leq d, e < n$, let D_d be a ***maximal linearly independent set*** *of fS_d. Then there exists a function $g \in \mathcal{F}_n$ with degree at most d such that $h = fg \neq 0$ with $deg(h) \leq e$* ***if and only if $D_d \cup S_e$ is linearly dependent over $\mathbb{F}_2$.***

Example 2

- Let $f(x) = Tr_1^3(\alpha^5 x + \alpha^6 x^3)$ be a function from $\mathbb{F}_{2^3}$ to $\mathbb{F}_2$ where α is a primitive element in $\mathbb{F}_{2^3}$ with $\alpha^3 + \alpha + 1 = 0$.
- Let $d = 2$ and $e = 1$, then $d + e = n$.
- The set S_2 contains the following **seven** functions shown before as P_0, P_1, and P_2, reproduced there:

$$\begin{array}{rl|rl}
\textbf{const. fun. c} & = 11111111 & & \\
Tr_1^3(x) & = 01001011 & Tr_1^3(x^3) & = 01110100 \\
Tr_1^3(\alpha x) & = 00010111 & Tr_1^3((\alpha x)^3) & = 01101001 \\
Tr_1^3(\alpha^2 x) & = 00101110 & Tr_1^3((\alpha^2 x)^3) & = 01010011
\end{array}$$

- From $f(x) = 00100001$, the elements of fS_2 are

$$\begin{array}{rl}
fTr_1^3(x) = fTr_1^3(\alpha x) = fTr_1^3((\alpha^2 x)^3) & = 00000001 \\
fTr_1^3(\alpha^2 x) = fTr_1^3(x^3) & = 00100000 \\
fc = fTr_1^3((\alpha x)^3) & = 00100001
\end{array}$$

Example 2 (Cont.)

- Thus $|fS_2| = 3$. However,

$$00100001 = 00000001 + 00100000.$$

  Thus the elements of fS_2 are **linearly dependent**. The maximal linear independent set in fS_2 is given by

$$D = \{00000001, 00100001\}.$$

- $D \cup S_1$ is a **linearly independent** set. According to Theorem, there is **no function** g with degree 2 such that $h = fg \neq 0$ with $\deg(h) = 1$. However, we can also directly verify this!
- This is a **counter example** to the existing result since $d + e = 2 + 1 = 3 = n$.

Algorithm for Finding a Low Degree Multiplier

Input: f, a function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$; $1 \leq d, e < n$; and $t(x) = x^n + \sum_{i=0}^{n-1} t_i x^i$, $t_i \in \mathbb{F}_2$, a primitive polynomial over $\mathbb{F}_2$ of degree n.

Output: g with $deg(g) \leq d$ and $h = fg$ with $h \neq 0$ and $deg(h) \leq e$ if there exist such g and h. Otherwise, outputs $g = 0$ and $h = 0$.

- Randomly select an initial state $(a_0, a_1, \cdots, a_{n-1})$, $a_i \in \mathbb{F}_2$, and compute

$$a_{n+i} = \sum_{j=0}^{n-1} t_j a_{j+i}, i = 0, 1, \cdots, 2^n - 1 - n.$$

- Compute k, each **coset leader** modulo $2^n - 1$, and n_k, the **size** of C_k.

Algorithm for Finding a Low Degree Multiplier (cont.)

- Let $m = max\{d, e\}$. Establish S_m as follows:

$$P_0 = (1, 1, \cdots, 1)$$

  - **for** $0 \neq k$ in $\Gamma(n)$ with $w(k) \leq m$ **do**
    Compute $\mathbf{a}^{(k)} = (a_0, a_k, \cdots, a_{k(2^n-2)})$, then apply the shift operator to the decimated sequence, and **establish** P_k.
  - **for** k in $\Gamma(n)$ with $0 \leq w(k) \leq m$ **do**
    Load P_k as an $n_k \times 2^n$ **sub-matrix** of S_m.

- Using the Gauss elimination, find the **rank** of fS_d, represented as an $|fS_d| \times 2^n$ **matrix**, and find D_d, a maximal linearly independent set of fS_d.

Algorithm for Finding a Low Degree Multiplier (cont.)

- Apply the Gauss elimination to the following matrix

$$\begin{bmatrix} D_d \\ S_e \end{bmatrix}.$$

If the **rank** of the above matrix is equal to $t+s$ where $|D_d| = t$ and $|S_e| = s$, set $g = 0$ and $h = 0$, then return g and h. Otherwise, find $c_i, i = 1, \cdots, t$ such that

$$\sum_{i=1}^{t} c_i f g_i + \sum_{i=1}^{s} c_{t+i} h_i = 0.$$

Set $g = \sum_{i=1}^{t} c_i g_i$ and $h = \sum_{i=1}^{s} c_{t+i} h_i$. Return g and h.

Resistance against FAA

- In order to **launch an efficient FAA attack**, one needs to find a multiplier g with $deg(g) \leq d$ such that $h = fg \neq 0$ with $deg(h) \leq e$ for some pair (d, e) which is in favor of FAA.

Definition

For a given function f and an integer pair (d, e), assume that there exists some function g with $deg(g) = d < AI(f)$ such that $h = fg \neq 0$ with $deg(h) = e$. Then we say that the pair (d, e) is an **enable pair** of f.

Definition

For a given $f \in \mathcal{F}_n$, and a pair of positive integers (d, e), f is said to be **(d, e)-resistance** against FAA if and only if for all g with $deg(g) \leq d$ and $h = fg \neq 0$ with $deg(h) \geq e$. If the assertion is true for every $d : 1 \leq d < AI(f)$, then f is said to be **e-resistance** against FAA.

Example 3

*Let $\mathbb{F}_{2^4}$ be defined by a primitive polynomial $t(x) = x^4 + x + 1$ and α a root of $t(x)$. Let $f(x) = Tr(\alpha x^3)$. (Note that $f(x)$ is a bent function.) Then, $f(x)$ is **2-resistance** against FAA.*

- Any function $g(x) \in \mathcal{F}_n$ with $g(0) = 0$ can be written as

$$g(x) = Tr(bx) + Tr(cx^3) + Tr_1^2(dx^5) + Tr(ex^7) + wx^{15},$$

where $b, c, e \in \mathbb{F}_{2^4}, d \in \mathbb{F}_{2^2}, w \in \mathbb{F}_2$, and $Tr_1^2(x) = x + x^2$ is the trace function from $\mathbb{F}_{2^2}$ to $\mathbb{F}_2$.

- In the following, we only show the case for $w = 0$.

Example 3 (cont.)

- We have the expansion of $f(x)g(x)$ as follows
$$f(x)g(x) = Tr(Ax + Bx^3 + Dx^7) + Tr_1^2(Cx^5) + E$$
where
$$\begin{aligned} A &= b^4\alpha^4 + d^2\alpha^2 + e^4\alpha + e\alpha^8 \\ B &= c^2\alpha^4 + c^4\alpha^2 + c^8\alpha^8 \\ D &= b\alpha^2 + b^4\alpha + d^2\alpha^4 + e^4\alpha^8 \\ C &= b^2\alpha + b^8\alpha^4 + e^2\alpha^2 + e^8\alpha^8 \\ E &= Tr(c\alpha^4). \end{aligned}$$
- Considering that $fg \neq 0$, then $deg(fg) = 1$ if and only if
$$B = C = D = 0.$$
It can be verified that the system of those equations has **no solutions** for any choices of b, c, d and e.
- Thus $deg(fg) \geq 2 \Longrightarrow f$ is 2-resistance to FAA.

Example 4

(a) Hyper-bent functions: degree $n/2$.

- Let $n = 8$, and α be a primitive element in $\mathbb{F}_{2^8}$ defined by $x^8 + x^4 + x^3 + x^2 + 1$.
- Let $f(x) = Tr(\alpha^{95}x^{15} + \alpha^{115}x^{45})$. Then the corresponding sequence: $\mathbf{a} = 000100010001111111$. This is a hyper-bent function.
- Let $g(x) = Tr(\alpha^{145}x^5)$. Then $h = fg = Tr(\alpha^{230}x^5 + \alpha^{215}x^{25})$.
- By computation, the algebraic immunity of f is

$$AI(f) = 3.$$

- We have $(d, e) = (2, 3)$. **Thus the FAA exists.**
- There are a total of **24310 hyper-bent functions** in 8 variables. All of them have the enable pairs $(2, 3)$.

Example 4 (cont.)

(b) Inverse functions: $Tr(x^{-1})$ with degree $n-1$

Enable Pairs and Algebraic Immunity

n	(d,e)	AI
9	$(3,4)$	4
10	$(3,5)$	5
11	$(3,5)$	5
12	$(3,5)$	5
13	$(3,6)$	6
14	$(5,6)$	6
15	$(5,6)$	6
16	$(4,6)$	6

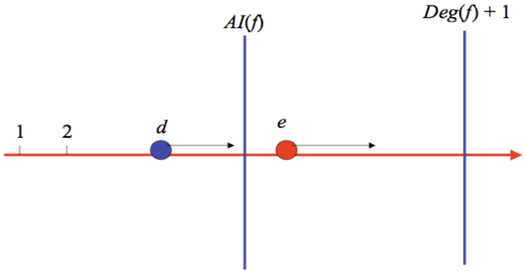
Research Problems on Resistance against FAA

- From the known experimental data, let $d = AI(f) - \delta$ where $\delta = 1, 2$ or very small compared with $AI(f)$. Then we can find $e = AI(f)$ **such that** (d, e) **is an enable pair** of f.
- In other words, for any $d = AI(f) - \delta$, experimentally, we always can find g such that $deg(g) = d$ **and** $h = fg$ **with degree** $deg(h) = AI(f)$**.**
- Do there exist some functions where FAA do not have any benefit?

Research Problems on Resistance against FAA (cont.)

- We believe if a function f can be **resistant against FAA**, then any enable paris should satisfy the following conditions.
  - For any enable pair (d, e) of f, d **should be close to** $AI(f)$, and $e > AI(f)$ and e should be "far" from $AI(f)$. In this case, Attacker only can obtain a system of linear equations of T_d unknowns which requires T_e consecutive bits to form this system.
  - If d is close to 1, then e **should be close to** $deg(f) + 1$. Otherwise, the attacker can solve a system of linear equations with very small number of unknowns at a modest cost of known bits.

Research Problems on Resistance against FAA (cont.)



- Do those functions exist? (**No examples so far!**)

- However, a **better approach** to investigate the functions who can be resistance against FAA is to investigate how many respective component sequences do the three sequences have, or equivalently, the numbers of **nonzero DFT coefficients** of these three functions, instead of **degrees of those functions**, i.e., the **selective DFT approach** (Gong-Sonjom-Helleseth-Hu, 2008).