Design of Filter Functions for Key Stream Generators using Boolean Power Functions

Jong-Min Baek, Seok-Yong Jin and Hong-Yeop Song

{jm.baek, sy.jin, hysong}@yonsei.ac.kr
Coding and Information Theory Lab.
Yonsei University

November 4, 2006

Contents

Contents

Key Stream Generators

- Key part of entire stream cipher system
- Driven by initial key (initial state)
  - Not actual encryption key
  - Update states by internal logic: Finite state machine
- Generate periodic binary sequences correspond to states: **Key stream sequences**
  - Actually encrypt message stream at each clock
  - Should be shown as random sequence
  - Should be strong against cryptanalysis
- Use linear feedback shift registers (LFSR) for internal logic

Nonlinear Filter Functions

- Basically a Boolean function denoted by $f : {\mathbb{F}_2}^n \rightarrow \mathbb{F}_2$
- Applied to stages of LFSR to increase linear complexity of output sequence
- Easy to increase linear complexity, hard to have good statistics

Design Idea

- Modified component function of Rijndael S-box (Jin, *et. al*, 2006)
  - $g(x) = x^8 + x^4 + x^3 + x^2 + 1$, primitive polynomial
  - $h(x) = x^8 + x^4 + x^3 + x + 1$, irreducible but not primitive polynomial
  - A component function is obtained from S-box which is defined in $\mathbb{F}_2[x]/h(x)$
  - The realization of the component function is performed in $\mathbb{F}_2[x]/g(x)$
  - The resultant trace representation has a lot of trace terms
  - The resultant modified S-box is a permutation

Goal

- Apply the previous method to design of filter function
$\rightarrow$Make the key stream sequence with large linear complexity and good statistics

Contents

LFSR and Finite Field

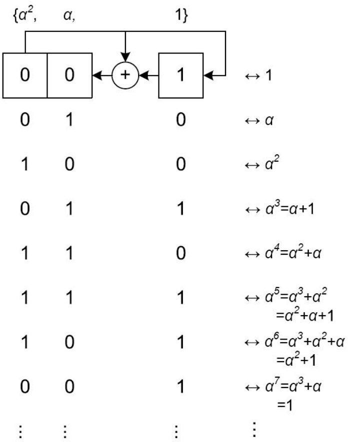


Figure: 3-Stage LFSR with $g(x)$ and Corresponding $\mathbb{F}_{2^n}$

- For realization of function, we use LFSRs with Galois configuration
  - Each states can be considered as a field element in $\mathbb{F}[x]/g(x)$
  - Denote a root of $g(x)$ be α
  - Mapping field elements to vector elements
    - ★ Let $\{1, \alpha, \cdots, \alpha^{n-1}\}$ be a basis
    - ★ $\alpha^i = x_1\alpha^{n-1} + x_2\alpha^{n-2} + \cdots + x_n \leftrightarrow (x_1, x_2, \cdots, x_n) = \underline{x}$
- Only primitive polynomials are used to achieve maximal period

Inversion Mapping $\mathrm{INV}(x)$

- Key part of Rijndael S-box
- Definition

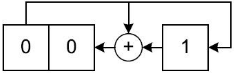
$$\mathrm{INV}(x) \triangleq \begin{cases} x^{-1} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

- $\mathrm{INV}(x)$ is a permutation function on $\mathbb{F}_{2^n}$
- Take an ith output bit from $\mathrm{INV}(x)$: Boolean power function $\mathrm{INV}_i(x)$

Proposed Filter Function

- Consider n-stage LFSR, $n \geq 3$
- Set LFSR connection to primitive polynomial $g(x)$
- Set $\mathbb{F}_{2^n}$ to be defined by another primitive polynomial $h(x)$, where $h(\beta) = 0$
- Define a function $P(x)$, $P : {\mathbb{F}_2}^n \to {\mathbb{F}_2}^n$
  - STEP 1: Convert a state vector $\underline{x} \leftrightarrow \alpha^i$ to a field element β^j
  - STEP 2: Calculate INV(β^j)
  - STEP 3: Convert INV(β^j) to output vector $P(x)$
- Take an ith output bit from $P(x)$: Proposed filter function $P_i(x)$

Comparing $\mathrm{INV}(x)$ and $P(x)$

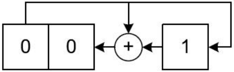


0	0	1	1
0	1	0	α
1	0	0	α^2
0	1	1	α^3
1	1	0	α^4
1	1	1	α^5
1	0	1	α^6

$g(x)=x^3+x+1$

INV(x) →

	$\mathrm{INV}_1(x)$	$\mathrm{INV}_2(x)$	$\mathrm{INV}_3(x)$
1	0	0	1
α^6	1	0	1
α^5	1	1	1
α^4	1	1	0
α^3	0	1	1
α^2	1	0	0
α	0	1	0



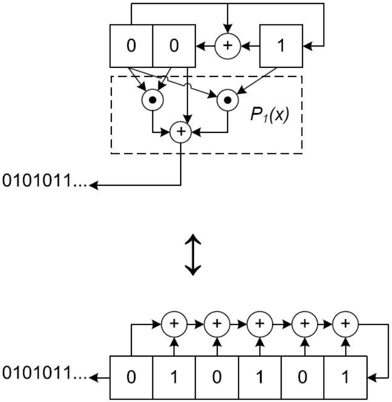
1	0	0	1	1
α	0	1	0	β
α^2	1	0	0	β^2
α^3	0	1	1	β^5
α^4	1	1	0	β^6
α^5	1	1	1	β^4
α^6	1	0	1	β^3

$h(x)=x^3+x^2+1$

INV(x) →

	$P_1(x)$	$P_2(x)$	$P_3(x)$	
1	0	0	1	1
β^6	1	1	0	α^4
β^5	0	1	1	α^3
β^2	1	0	0	α^2
β	0	1	0	α
β^3	1	0	1	α^6
β^4	1	1	1	α^5

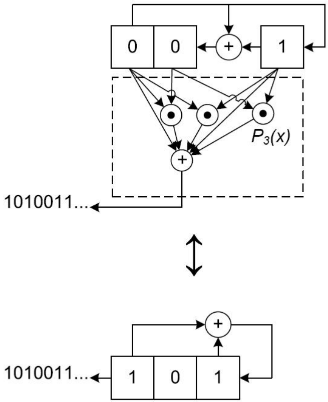
Example: $P_1(x)$



- Output sequence (0101011)
  - Maximal period
  - Balanced 0's and 1's
  - Increased linear complexity: Maximum value 6 (7 is achieved by complement)

Figure: Applying $P_1(x_1, x_2, x_3) = x_2 + x_1 x_2 + x_1 x_3$

Another Example: $P_3(x)$



- Output sequence (1010011)
  - Maximal period
  - Balanced 0's and 1's
  - Not Increased linear complexity→Degeneracy

Figure: Applying $P_3(x_1, x_2, x_3) = x_1 + x_3 + x_1x_2 + x_1x_3 + x_2x_3$

Contents

Maximal Period Property

Theorem

Consider an n-stage LFSR with primitive connection polynomial. If an arbitrary nonlinear filter function f is applied to the LFSR, then the resultant output key stream sequence has maximal period $2^n - 1$.

- Using k-tuple balance property of m-sequences to show all terms in ANF do not have subperiod
- Since the ANF of f can represent all $(2^n - 1)$-tuple vector, the result follows

Corollary

The key stream sequence obtained from $P_i(x)$ has maximal period for all i.

Balance Property

Theorem

The key stream sequence obtained from $P_i(x)$ satisfies balance property for all i.

- Although INV(x) is performed in different field structure, it is still a permutation
  $\rightarrow$The resultant key stream sequence is a bit-permutated m-sequence

Large Linear Complexity (1)

- For some numerical results, we investigate all possible sequences for n-stage LFSR
  - Set $g(x)$ and $h(x)$ differently as possible
  - For each pair $\big(g(x), h(x)\big)$, generate sequences from $P_i(x)$ for all i
- 3-Stage LFSR
  - Primitive polynomials: $x^3 + x + 1$ and $x^3 + x^2 + 1$
  - For each pair, 3 functions exist

Linear Complexity	3	6
frequency	1	5

Large Linear Complexity (2)

- The portion of maximum linear complexity

n	3*	4**	5*	6	7*	8	9	10	11**	12
%	83	87.5	86	49.4	88.65	54.4	69.3	56.2	91.6	37.8

- Maximum linear complexity vs. minimum linear complexity

n	3*	4**	5*	6	7*	8	9	10	11**	12
min	3	12	20	45	98	210	462	950	1969	3960
max	6	14	30	62	126	254	510	1022	2046	4094
ratio	0.5	0.86	0.67	0.73	0.78	0.83	0.91	0.93	0.96	0.97

  - *: $2^n - 1$ is prime
  - **: $2^n - 1$ is factorized into two distinct primes

Large Linear Complexity (3)

Conjecture

There exist $P_i(x)$ which achieve maximum linear complexity

Contents

Summary

- New design of filter function
  - Output sequence has maximal period
  - Output sequence satisfies balance property due to $\mathrm{INV}(x)$
  - Guarantees large linear complexity due to set $g(x) \neq h(x)$
- Future Works
  - Investigation for run distributions
  - Theoretical analysis for large linear complexity property
    - ★ The case of $2^n - 1$ is prime or factorized into two prime
    - ★ Proof of the conjecture on maximum linear complexity
  - More cryptographic analysis
    - ★ Correlation attack, algebraic attack, etc.