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

December 4, 2006
Contents

1 Motivation

2 Proposed Design of Filter Function

3 Some Properties of Proposed Filter Function

4 Concluding Remarks
Contents

1 Motivation

2 Proposed Design of Filter Function

3 Some Properties of Proposed Filter Function

4 Concluding Remarks
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
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
  → Make the key stream sequence with large linear complexity and good statistics
Contents

1 Motivation

2 Proposed Design of Filter Function

3 Some Properties of Proposed Filter Function

4 Concluding Remarks
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 $\alpha$
- Mapping field elements to vector elements
  - Let $\{1, \alpha, \cdots, \alpha^{n-1}\}$ be a basis
  - $\alpha^i = x_1 \alpha^{n-1} + x_2 \alpha^{n-2} + \cdots + x_n \leftrightarrow (x_1, x_2, \cdots, x_n) = \overline{x}$

- Only primitive polynomials are used to achieve maximal period
Inversion Mapping $\text{INV}(x)$

- Key part of Rijndael S-box
- Definition
  \[
  \text{INV}(x) \triangleq \begin{cases} 
  x^{-1} & \text{if } x \neq 0 \\
  0 & \text{if } x = 0
  \end{cases}
  \]
- $\text{INV}(x)$ is a permutation function on $\mathbb{F}_{2^n}$
- Take an $i$th output bit from $\text{INV}(x)$: Boolean power function $\text{INV}_i(x)$
Consider $n$-stage LFSR, $n \geq 3$

Let a primitive polynomial $g(x)$ be LFSR connection

Let $\mathbb{F}_{2^n}$ be defined by another primitive polynomial $h(x)$, where $h(\beta) = 0$

Define a function $P(x), P : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$

- STEP 1: Convert a state vector $x \leftrightarrow \alpha^i$ to a field element $\beta^j$
- STEP 2: Calculate $\text{INV}(\beta^j)$
- STEP 3: Convert $\text{INV}(\beta^j)$ to output vector $P(x)$

Take an $i$th output bit from $P(x)$: Proposed filter function $P_i(x)$
Comparing $\text{INV}(x)$ and $P(x)$

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

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

$\text{INV}(x)$

$\text{INV}_1(x)$

$\text{INV}_2(x)$

$\text{INV}_3(x)$

$P_1(x)$

$P_2(x)$

$P_3(x)$
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 $P_3(x_1, x_2, x_3) = x_1 + x_3 + x_1x_2 + x_1x_3 + x_2x_3$
1 Motivation

2 Proposed Design of Filter Function

3 Some Properties of Proposed Filter Function

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

Although $\text{INV}(x)$ is performed in different field structure, it is still a permutation.

→The resultant key stream sequence is a bit-permutated m-sequence
Maximal Period Property

**Theorem**

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

- Using the previous result, we know that the number of 1 is $2^{n-1}$
  - Subperiod cannot exist
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 $(g(x), h(x))$, generate sequences from $P_i(x)$ for all $i$

**Example: 3-Stage LFSR**

- Primitive polynomials: $x^3 + x + 1$ and $x^3 + x^2 + 1$
- For each pair, 3 functions exist

<table>
<thead>
<tr>
<th>Linear Complexity</th>
<th>3</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>frequency</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>
Large Linear Complexity (2)

- The portion of maximum linear complexity

<table>
<thead>
<tr>
<th>$n$</th>
<th>3*</th>
<th>4**</th>
<th>5*</th>
<th>6</th>
<th>7*</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11**</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>83</td>
<td>87.5</td>
<td>86</td>
<td>49.4</td>
<td>88.65</td>
<td>54.4</td>
<td>69.3</td>
<td>56.2</td>
<td>91.6</td>
<td>37.8</td>
</tr>
</tbody>
</table>

- Maximum linear complexity vs. minimum linear complexity

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

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

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

2 Proposed Design of Filter Function

3 Some Properties of Proposed Filter Function

4 Concluding Remarks
Summary

- New design of filter function
  - Output sequence has maximal period
  - Output sequence satisfies balance property due to 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.