On an improved summation generator with 2-bit memory

Hoon Jae Lee\textsuperscript{a,\ast}, Sang Jae Moon\textsuperscript{b}

\textsuperscript{a}Department of Computer Engineering, Kyungwoon University, San 3-1 Induk-ri, Sandong-Myun, Kumi, Kyungbuk 730-850, South Korea
\textsuperscript{b}School of Electronic and Electrical Engineering, Kyungpook National University 1370, Sankyuk-Dong, Taegu 702-701, South Korea

Abstract

The summation generator is a real adder generator with a maximum period, near-maximum linear complexity and maximum order of correlation immunity. However it is neither secure against nor immune to correlation attack between its output sequences and carry sequences in special cases. A modified summation generator, secure against such an attack, has recently been proposed, but no proof is given about its period and linear complexity. In this paper, we propose a new modified summation generator immune to correlation attack. Moreover, we conclude that its period is maximum and that its linear complexity is the same as that of the original summation generator. © 2000 Elsevier Science B.V. All rights reserved.

1. Introduction

In cryptographic and spread-spectrum communication systems [1,14], the running-key generator
in the stream cipher needs to be secure. There are four main characteristics of a secure stream cipher: period, randomness, and linear complexity of the output sequences (according to Beker and Piper [1]), and correlation immunity (according to Siegenthaler [11]).

In general, a running-key generator, such as Goffe’s generator [11], consists of $N$ driving linear feedback shift registers (LFSRs) and a nonlinear combiner on $N$ output sequences in order to produce the running key. The most common running-key generator, the summation generator [8,9], is used in stream cipher (“pseudo-noise,” or PN sequence generators) and in spread-spectrum systems. The summation generator, when combined with two or more LFSR sequences using 1-bit memory, has some cryptographically good properties. It produces random binary sequences whose period is proved to be maximum and whose linear complexity is conjectured to be near to its period. Moreover, it is easily implemented by a hardware or software approach. For hardware implementation, little memory and few logical gates are required: $L_1 + L_2$ memory bits (D flip-flops), one full adder, a number of XOR gates in feedback taps, and one AND gate.

However, the summation generator is not yet entirely secure [2]. In generating long, consecutive-zero output sequences, it is subject to correlation attack: the correlation between the generator’s output sequences and carry sequences can easily be estimated by an outside party. A modified summation generator, which is secure against this correlation attack, has recently been proposed [2], but no proof has been given concerning its period and linear complexity.

In this paper, we propose a new modified summation generator, which is immune to correlation attack. Moreover, we prove that it has a maximum period and near-maximum linear complexity, the same as those of original generator.

### 2. Improved summation generator with 2-bit memory

**2.1. Analysis of the summation generator**

Rueppel’s summation generator [8,9] outputs $z_j$ and $c_j$ from each LFSR outputs $a_j$ and $b_j$ and previous carry $c_{j-1}$ as in Fig. 1.

$$z_j = a_j \oplus b_j \oplus c_{j-1},$$

$$c_j = a_jb_j \oplus (a_j \oplus b_j)c_{j-1}, \quad j = 0, 1, 2, \ldots$$

where $a$ is the output sequence of LFSR 1, $b$ is the output sequence of LFSR 2, $c$ is the carry sequence, with carry initialization value $c_{-1} = 0$.

**Theorem 1.** The cryptographical properties of Rueppel’s summation generator are as follows [8].

1. Period, $P = (2^{L_1} - 1)(2^{L_2} - 1)$.
2. Good randomness.
3. Linear complexity, $LC \leq P$.
4. The order of correlation immunity, $m = 1$.

**Theorem 2.** (Meier and Staffelbach [7]). (1) Suppose that the output of the basic summation combiner satisfies $z_{j+1} = z_{j+2} = \cdots = z_{j+s} = 0$ and $z_{j+s+1} = 1$. Then, for every $t$ with $1 \leq t \leq s$, the $s-t+2$ equations

\[
z_{j+t+1} = a_{j+t+1} + b_{j+t+1} + 1 = 0,
\]

\[
z_{j+t+2} = a_{j+t+2} + b_{j+t+2} + 1 = 0,
\]

\[\cdots\]

\[
z_{j+s+1} = a_{j+s+1} + b_{j+s+1} + 1 = 1,
\]

\[
z_{j+s+2} = a_{j+s+2} + b_{j+s+2} + a_{j+s+1}
\]

are simultaneously satisfied with probability at least $1 - 2^{-t}$.

(2) Suppose that the output of the basic summation combiner satisfies $z_{j+1} = z_{j+2} = \cdots = z_{j+s} = 1$ and $z_{j+s+1} = 0$. Then, for every $t$ with $1 \leq t \leq s$, the

![Fig. 1. The summation generator (SUM-BSG).](image-url)
s - t + 2 equations

\[ z_{j+t+1} = a_{j+t+1} + b_{j+t+1} = 1, \]
\[ z_{j+t+2} = a_{j+t+2} + b_{j+t+2} = 1, \]
\[ \cdots, \]
\[ z_{j+s+1} = a_{j+s+1} + b_{j+s+1} = 0, \]
\[ z_{j+s+2} = a_{j+s+2} + b_{j+s+2} + a_{j+s+1} \]
are simultaneously satisfied with probability at least \( 1 - 2^{-t} \).

Observe that Theorem 2, which states that (1) and (2) are simultaneously satisfied with a certain probability, is much stronger than the statement that these equations are individually satisfied with the same probability.

2.2. Dawson’s generator

Dawson’s generator [2] outputs \( z_j \) and \( c_j \) from each LFSR outputs \( a_j \) and \( b_j \) and previous carry \( c_{j-1} \):

\[ z_j = a_j \oplus b_j \oplus c_{j-1}, \]
\[ c_j = b_j \oplus (a_j \oplus b_j)c_{j-1}, \quad j = 0, 1, 2, \ldots, \]

where \( a \) is the output sequence of LFSR 1, \( b \) is the output sequence of LFSR 2, \( c \) is the carry sequence, with carry initialization value \( c_{-1} = 0 \).

However, this generator was not analyzed in terms of cryptographical security.

2.3. An improved summation generator with 2-bit memory

An improved summation generator with 2-bit memory outputs \( z_j, c_j \) and \( d_j \) from each of LFSR outputs \( a_j \) and \( b_j \), previous carry \( c_{j-1} \) and previous memory \( d_{j-1} \) as in Fig. 2.

\[ z_j = y_j \oplus d_{j-1}, \]
\[ d_j = f(a_j, b_j, d_{j-1}) = b_j \oplus (a_j \oplus b_j)d_{j-1}, \]
\[ j = 0, 1, 2 \ldots, \]

where \( y \) is the output sequence of summation generator, \( a \) is the output sequence of LFSR 1, \( b \) is the output sequence of LFSR 2, \( c \) is the carry sequence, carry initialization value \( c_{-1} = 0 \), \( d \) is memory sequences, memory initialization value \( d_{-1} = 0 \).

Because the summation generator has a probability of a carry-output correlation of \( 1/4 \), highly correlated, it can be vulnerable to correlation attack [2,7] when it outputs consecutive zeros or ones. But the proposed generator is secure by having an additional 0–1 balanced nonlinear memory function, exclusive-ored of the output of the summation generator.

2.3.1. Correlation properties

We refer to the input–output correlation and carry–output correlation in the summation generator in Table 1, which gives us an input-output correlation probability of \( \frac{1}{2} \), balanced, and a carry-output correlation probability of \( \frac{1}{4} \), not balanced. But in the proposed generator, input(\( a_j, b_j, c_{j-1} \) or \( d_{j-1} \))–output(\( z_j \)) correlation probability is \( \frac{1}{2} \), and memory(\( c_j \) or \( d_j \))–output(\( z_j \)) correlation probability is \( \frac{1}{2} \) too, as shown in Table 2. Therefore, the proposed generator is improved in terms of input–, carry–, or memory–output correlation probability. It is safe against correlation attack in a special cases(a long consecutive zero-output).

2.3.2. Security

We analyzed the proposed generator in terms of period, linear complexity, and the order of correlation immunity.
Theorem 3. In an improved summation generator with 2-bit memory which has two LFSRs of length $L_1$ and $L_2$, $\gcd(L_1, L_2) = 1$, the period of output sequences is $(2^{L_1} - 1)(2^{L_2} - 1)$, except null initial state of two LFSRs each, and two LFSRs make $d_j = 0$, only when initial state.

Proof. Let $j \geq 0$, period of $a_j, P_a$, period of $b_j, P_b$, expected period of $z_j, P = \text{lcm}(P_a, P_b)$, then

$$d_j = b_j \oplus (a_j \oplus b_j) d_{j-1}$$

where $j \geq 0$.

$$d_j = b_j \oplus (a_j \oplus b_j)[b_{j-1} \oplus (a_{j-1} \oplus b_{j-1}) \oplus d_{j-2}]$$

and $d_0 = 0$ (in case of two LFSRs are all initial states), $b_{j+p} = b_j, a_{j+p} = a_j, \text{ and } d_{j+p} = d_j$ therefore, period of $d_{j-1}$ is $P_d = P = \text{lcm}(P_a, P_b)$.

On the other hand, the period of $y_j$ in summation generator is $P_y = \text{lcm}(P_a, P_b)$ by Rueppel [8,9] and $z_j = y_j \oplus d_{j-1}$, therefore, the period of $z_j$ is $P = \text{lcm}(P_y, P_d) = \text{lcm}(P_a, P_b) = (2^{L_1} - 1)(2^{L_2} - 1)$ in cases of $\gcd(L_1, L_2) = 1$.

Theorem 4. In an improved summation generator with 2-bit memory, linear complexity of output sequence $z_j$ is approximately equal to the period of $z_j$.

For a small size of $L_1$ and $L_2$, simulation examples of period and LC for the summation generator and the proposed generator are as shown in Table 3. By computing in the Berlekamp–Massey algorithm [5], linear complexities of output

```
Table 1
Correlation probability of the summation generator

<table>
<thead>
<tr>
<th>a_j</th>
<th>b_j</th>
<th>c_{j-1}</th>
<th>c_j</th>
<th>z_j</th>
<th>Correlation probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>* Input–output:</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$P[a_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$P[b_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$P[c_{j-1} = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>* Carry–output:</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$P[c_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$P[d_{j-1} = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$P[d_j = z_j] = \frac{1}{2}$</td>
</tr>
</tbody>
</table>
```

```
Table 2
Correlation probability of the proposed generator

<table>
<thead>
<tr>
<th>a_j</th>
<th>b_j</th>
<th>c_{j-1}</th>
<th>d_{j-1}</th>
<th>c_j</th>
<th>y_j</th>
<th>d_j</th>
<th>z_j</th>
<th>Correlation probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>* Input–output:</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$P[a_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$P[b_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$P[d_{j-1} = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$P[d_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>* Carry–output:</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$P[c_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$P[d_{j-1} = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$P[d_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>* Carry–output:</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$P[c_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$P[d_j = z_j] = \frac{1}{2}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$P[d_j = z_j] = \frac{1}{2}$</td>
</tr>
</tbody>
</table>
```
sequences in the two generators are close to their periods.

**Theorem 5.** In an improved summation generator with 2-bit memory, the order of correlation immunity is 1.

**Proof.** The result of computing in the Walsh transform [7] is the following table. In the table for all w except $w = 0000_B$ and $w = 1111_B$, $F(w) = 1$, and so it has the third order of correlation immunity containing a carry and a memory. On the other hand, the maximum order of correlation immunity is $m_{max} = N - 1 = 2 - 1 = 1$ by Meier and Staffelbach [7]; therefore, the order of correlation immunity is 1 by subtracting 2 (a carry and a memory) from 3 (Table 4).

2.3.3. Randomness test

Because the sequence of the proposed generator has large period in practice, we choose the method of local randomness [1]. After choosing the significance level = 0.05 and Chi-square distribution [1], the frequency test, serial test, generalized serial test [4], poker test and autocorrelation test [2] are adapted.

We sampled about 160,000 bits of the output sequences randomly for the following three examples (generators): the (31, 33), (63, 65) and (127, 131) stages of LFSR pairs.

**Example**

(1) $P_1(X) = X^{31} + X^3 + 1$

$\text{(IV} = 10101010 \ldots 101)$,

$Q_1(X) = X^{33} + X^{13} + 1$

$\text{(IV} = 11111111 \ldots 111)$.

(2) $P_2(X) = X^{63} + X^1 + 1$ (IV = random),

$Q_2(X) = X^{65} + X^{18} + 1$ (IV = random).

(3) $P_3(X) = X^{127} + X^1 + 1$ (IV = random).

$Q_3(X) = X^{131} + X^{13} + X^2 + X^1 + 1$

$\text{(IV} = \text{random})$.

The three kinds of sampled data display a good randomness, as shown in Table 5, in terms of the result of the frequency test, serial test, generalized serial test ($t = 3, 4, 5$), poker test ($m = 3, 4, 5$), and autocorrelation test.

2.3.4. Comparison

Because the summation generator had a carry–output correlation probability of $1/4$, not uncorrelated, it was broken by correlation attack [2,7] when the output had many consecutive zeros or ones. But the proposed generator has an input($a_j$, $b_j$, $c_j$) or $d_j$–output($z_j$) correlation probability of $1/2$, and a memory($c_j$ or $d_j$)–output($z_j$) correlation probability of $1/4$, and so it is secure and will not incur carry bits from the output of a long sequence of consecutive zeros and ones. In other words, it is secure by adding one bit of memory in nonlinear combine function based on the summation generator (Table 6).

---

### Table 3
Simulation examples of the period and LC for ISUM-BSG

<table>
<thead>
<tr>
<th>Stage of LFSRs</th>
<th>SUM-BSG</th>
<th>ISUM-BSG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>$L_2$</td>
<td>$P$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>105</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>217</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>465</td>
</tr>
</tbody>
</table>

### Table 4
The result of the Walsh transform for ISUM-BSG function

<table>
<thead>
<tr>
<th>$w$</th>
<th>0</th>
<th>1</th>
<th>2</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>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F(w)$</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$-8$</td>
</tr>
</tbody>
</table>
Table 5
The result of random tests for ISUM-BSG

<table>
<thead>
<tr>
<th>Items</th>
<th>Threshold</th>
<th>Test results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sample 1</td>
</tr>
<tr>
<td>(1) Frequency test</td>
<td>3.84</td>
<td>0.027</td>
</tr>
<tr>
<td>(2) Serial test</td>
<td>5.99</td>
<td>1.390</td>
</tr>
<tr>
<td>(3) Generalized $r$-serial test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t = 3$</td>
<td>9.48</td>
<td>6.919</td>
</tr>
<tr>
<td>$t = 4$</td>
<td>15.50</td>
<td>12.459</td>
</tr>
<tr>
<td>$t = 5$</td>
<td>29.29</td>
<td>23.057</td>
</tr>
<tr>
<td>(4) Poker test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m = 3$</td>
<td>14.067</td>
<td>6.185</td>
</tr>
<tr>
<td>$m = 4$</td>
<td>24.996</td>
<td>17.287</td>
</tr>
<tr>
<td>$m = 5$</td>
<td>44.654</td>
<td>37.304</td>
</tr>
<tr>
<td>(5) Autocorrelation test</td>
<td>max. $\leq 0.05$</td>
<td>max = 0.0060</td>
</tr>
</tbody>
</table>

Table 6
Comparison of similar summation generators

<table>
<thead>
<tr>
<th>Items</th>
<th>SUM-BSG</th>
<th>ISUM-BSG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>$P = (2^{L-1} - 1)(2^{L-2} - 1)$</td>
<td>$P = (2^{L-1} - 1)(2^{L-2} - 1)$</td>
</tr>
<tr>
<td>Randomness</td>
<td>Random</td>
<td>Random</td>
</tr>
<tr>
<td>Linear complexity</td>
<td>LC $\leq P$</td>
<td>LC $\leq P$</td>
</tr>
<tr>
<td>Correlation immunity</td>
<td>CI = 1</td>
<td>CI = 1</td>
</tr>
<tr>
<td>Correlation attack</td>
<td>Correlation breakable (consecutive “0” or “1” output)</td>
<td>Secure</td>
</tr>
</tbody>
</table>

3. Conclusion

In this paper, we have proposed an improved summation generator with 2-bit memory and have analyzed it. Because the original summation generator had a carry–output correlation probability of $\frac{1}{2}$ not uncorrelated, it was broken by correlation attack when it generated many consecutive zeros or ones. But the proposed generator has an input $(a_j, b_j, c_{j-1}$ or $d_{j-1})$–output($z_j$) correlation probability of $\frac{1}{2}$ and a memory $(c_j$ or $d_j)$–output($z_j$) correlation probability of $\frac{1}{2}$, so it is secure and will not incur carry bits from the output of a long sequence of consecutive zeros and ones. Therefore, it is secure by adding one bit of memory in nonlinear combine function based on the summation generator.

4. Further reading

The following references are also of interest to the reader: [3]; [6]; [10]; [12]; [13]; [15].

References


