CryptMT3 Stream Cipher

Makoto Matsumoto\(^1\), Mutsuo Saito\(^2\), Takuji Nishimura\(^3\), and Mariko Hagita\(^4\)

\(^1\) Dept. of Math., Hiroshima University, m-mat@math.sci.hiroshima-u.ac.jp
\(^2\) Dept. of Math., Hiroshima University, saito@math.sci.hiroshima-u.ac.jp
\(^3\) Dept. of Math. Sci., Yamagata University, nisimura@sci.kj.yamagata-u.ac.jp
\(^4\) Dept. of Info. Sci., Ochanomizu University, hagita@is.ocha.ac.jp

Abstract. CryptMT version 3 (CryptMT3) is a stream cipher obtained by combining a large LFSR and a nonlinear filter with memory using integer multiplication. Its period is proved to be no less than \(2^{19937}-1\), and the 8-bit output sequence is at least 1241-dimensionally equidistributed. It is one of the fastest stream ciphers on a CPU with SIMD operations, such as Intel Core 2 Duo.

1 Introduction

In this article, we discuss pseudorandom number generators (PRNGs) for stream ciphers. We assume that the PRNG is implemented in software, and the platform is a 32-bit CPU with enough memory and fast integer multiplication.

Our proposal \[^5][^6\)] is to combine a huge state linear generator (called the mother generator) and a filter with memory, as shown in Figure 1.

![Fig. 1. Combined generator = linear generator + filter with memory](image.png)

[^5][^6]: CryptMT is proposed to eSTREAM Project http://www.ecrypt.eu.org/stream/. The reference codes are available there. This work is supported in part by JSPS Grant-In-Aid #16204002, #18654021, #18740044, #19204002 and JSPS Core-to-Core Program No.18005.
Definition 1. (Generator with a filter with memory.) Let $X$ be a finite set (typically the set of the word-size integers). The mother generator $G$ generates a sequence $x_0,x_1,x_2,\ldots \in X$. Let $Y$ be a finite set, which is the set of the possible states of the memory in the filter. We take a $y_0 \in Y$. Let $f: Y \times X \to Y$ be the state transition function of the memory of the filter, that is, the content $y_i$ of the memory is changed by the recursion

$$y_{i+1} := f(y_i, x_i).$$

The output at the $i$-th step is given by $g(y_i)$, where $g: Y \to O$ is the output function which converts the content of the memory to an output symbol in $O$.

In a previous manuscript [5], we chose the mother generator to be Mersenne Twister [4], which generates a sequence of 32-bit integers by an $F_2$-linear recursion. The filter is given by

$$f(y, x) := y \times (x|1) \mod 2^{32}, \quad g(y) := 8 \text{ MSBs of } y$$

where $(x|1)$ denotes $x$ with LSB set to 1, and 8 MSBs mean the 8 most significant bits of the integer $y$. Initially, the memory is set to an odd integer $y_0$. This is CryptMT version 1 (CryptMT1). There has been no attacks reported to this generator (even non-practical attacks). We introduced CryptMT version 2 [7] and version 3 [8], not to improve the security, but to improve the speed of initialization and generation. This manuscript is based on [8]. Theoretical analysis of this type of generators is developed in [9], where the quasigroup property of the filter plays the role of “balanced filter”.

2 CryptMT3: A New Variant Based on 128-Bit Operations

Modern CPUs often have single-instruction-multiple-data (SIMD) operations. Typically, a quadruple of 32-bit registers is considered as a single 128-bit register. CryptMT3 proposed here is a modification of the CryptMT version 1, so that it fits to the high-speed SIMD operations.

2.1 Notation

Let us fix the notations for 128-bit integers. A bold italic letter $x$ denotes a 128-bit integer. It is a concatenation of four 32-bit registers, each of which is denoted by $x[3], x[2], x[1], x[0]$, respectively, from MSB to LSB.

The notation $x[3][2]$ denotes the 64-bit integer obtained by concatenating the two 32-bit integers $x[3]$ and $x[2]$, in this order. Similarly, $x[0][3][2][1]$ denotes the 128-bit integer obtained by permuting (actually rotating) the four 32-bit integers in $x$. Thus, for example, $x = x[3][2][1][0]$ holds.

An operation on 128-bit registers that is executed for each 32-bit integer is denoted with the subscript 32. For example,

$$x +_{32} y := [(x[3] + y[3]), (x[2] + y[2]), (x[1] + y[1]), (x[0] + y[0])],$$