WREATH PRODUCTS IN STREAM CIPHER DESIGN

VLADIMIR ANASHIN

ABSTRACT. The paper develops a novel approach to stream cipher design: Both the state update function and the output function of the corresponding pseudorandom generators are compositions of arithmetic and bitwise logical operations, which are standard instructions of modern microprocessors. Moreover, both the state update function and the output function are being modified dynamically during the encryption. Also, these compositions could be keyed, so the only information available to an attacker is that these functions belong to some exponentially large class.

The paper shows that under rather loose conditions the output sequence is uniformly distributed, achieves maximum period length and has high linear complexity and ℓ-error linear complexity. Ciphers of this kind are flexible: One could choose a suitable combination of instructions to obtain due performance without affecting the quality of the output sequence. Finally, some evidence is given that a key recovery problem for (reasonably designed) stream ciphers of this kind is intractable up to plausible conjectures.

1. INTRODUCTION

A classical stream cipher is usually thought of as a pseudorandom generator which produces a keystream, that is, a binary random-looking string. Encryption procedure is just a bitwise addition modulo 2 (also called XORing) of the keystream to a plaintext, which is represented as a binary string either. That is, a pseudorandom generator is an algorithm that takes a short random string (*a key*, or *a seed*) and expands it into a very long random-looking string, a keystream.

To make software implementations of these algorithms platform-independent as well as to achieve high performance, the algorithms must use only those instructions that are common for contemporary processors. These instructions are numerical operations (addition, multiplication, subtraction,..) and logical ones (bitwise exclusive *or*, XOR, bitwise *and*, AND, etc.).

All these numerical and bitwise logical operations, and whence, all their compositions, belong to a special class of mappings from n-bit words into n-bit words: Each i^{th} bit of the output word depends only on bits $0, 1, \ldots, i$ of input words.[1] This fact underlies a number of results that enable one to determine whether a function of this kind is one-to-one, i.e., induces a permutation on n-bit words, or whether this permutation is a single cycle, or whether the function is balanced; that is, for each n-bit word the number of all its preimages is exactly the same, etc. Systematical studies of these properties

1991 *Mathematics Subject Classification.* 65C10, 11K45, 94A60, 68P25.

Key words and phrases. Stream cipher, pseudorandom generator, counter-dependent generator, uniform distribution, linear complexity, ℓ-error linear complexity, period, wreath product, skew product.

[1] These mappings are well-known mathematical objects (however, under different names: Compatible mappings in algebra, determined functions in automata theory, triangle boolean mappings in the theory of Boolean functions, functions that satisfy Lipschitz condition with constant 1 in p-adic analysis) dating back to 1960^{th} [22], [24]. Usefulness of these mappings in cryptography has being directly pointed out since 1993 by V.S. Anashin [9], [3], [4], [5], [6], [7]. The name "T-functions" for these mappings was suggested by A. Klimov and A. Shamir in 2002 [17].

for the above mentioned mappings were started by [9] and [3] (see also [4]) followed by [19],[5],[6], [7],[8], as well as by later works [17], [16], and [15].

The main goal of the paper is to present a mathematical background for a novel approach to the design of stream ciphers.[2] In this design, recurrence laws that define the key-stream are combinations of the above mentioned numerical and logical operations; moreover, these laws are being dynamically modified during encryption. Nevertheless, under minor restrictions we are able to prove that the key-stream has the longest (of possible) period, uniform distribution, and high linear complexity as well as high ℓ-error linear complexity and high 2-adic span. To give an idea of how these algorithms look like, consider the following illustrative example.

Let $m \equiv 3 \pmod 4$, $3 \le m \le \frac{2^n}{n}$. Take m *arbitrary* compositions $v_0(x), \dots, v_{m-1}(x)$ of the above mentioned machine instructions (addition, multiplication, XOR, AND, etc.), then take another m *arbitrary* compositions $w_0(x), \dots, w_{m-1}(x)$ of this kind. Arrange two arrays V and W writing these $v_j(x)$ and $w_j(x)$ to memory in *arbitrary* order. Now choose an arbitrary $x_0 \in \{0, 1, \dots 2^n - 1\}$ as a seed. The generator calculates the recurrence sequence of states $x_{i+1} = (i \bmod m + x_i + 4 \cdot v_{i \bmod m}(x_i)) \bmod 2^n$ and outputs the sequence $z_i = (1 + \pi(x_i) + 4 \cdot w_{i \bmod m}(\pi(x_i))) \bmod 2^n$, where π is a bit order reverse permutation, which reads an n-bit number $z \in \{0, 1, \dots, 2^n - 1\}$ in a reverse bit order; e.g., $\pi(0) = 0, \pi(1) = 2^{n-1}, \pi(2) = 2^{n-2}, \pi(3) = 2^{n-2} + 2^{n-1}$, etc. Then the sequence $\{x_i\}$ of n-bit numbers is periodic; its shortest period is of length $2^n m$, and each number of $\{0, 1, \dots, 2^n - 1\}$ occurs at the period exactly m times. Moreover, replacing each number x_i in $\{x_i\}$ by an n-bit word that is a base-2 expansion of x_i, we obtain by concatenation of these n-bit words a binary counterpart of the sequence $\{x_i\}$, i.e., a binary sequence $\{x_i\}'$ with a period of length $2^n mn$. This period is random in the sense of [18, Section 3.5, Definition Q1] (see (4.3.1) further); each k-tuple $(0 < k \le n)$ occurs in this sequence $\{x_i\}'$ with frequency[3] $\frac{1}{2^k}$ exactly. The output sequence $\{z_i\}$ of numbers is also periodic; its shortest period is of length $2^n m$; each number of $\{0, 1, \dots, 2^n - 1\}$ occurs at the period exactly m times. Finally, length of the shortest period of every binary subsequence $\{\delta_s(z_i) \colon i = 0, 1, 2, \dots\}$ obtained by reading s^{th} bit $\delta_s(z_i)$ $(0 \le s \le n - 1)$ of each member of the sequence $\{z_i\}$ is a multiple of 2^n; linear complexity of this binary subsequence $\{\delta_s(z_i)\}$ (as well as linear complexity of binary counterparts $\{z_i\}'$ and $\{x_i\}'$) exceeds 2^{n-1}.

Ciphers of this kind are rather flexible. For instance, in the above example one can take $m = 2^k$ instead of odd $m \equiv 3 \pmod 4$ and replace $i \bmod m$ in the definition of the state transition functions by an arbitrary $c_i \in \{0, 1, \dots, 2^k - 1\}$. To guarantee the above declared properties both of the state sequence and of the output sequence one must only demand that $c_0 + c_1 + \cdots + c_{m-1} \equiv 1 \pmod 2$. Moreover, one can take instead of π an arbitrary permutation of bits that takes the leftmost bit to the rightmost position (for instance, a circular 1-bit rotation towards higher order bits, which is also a standard instruction in modern microprocessors). Also, one can replace the second $+$ in the definition of the state transition and/or output functions with $\oplus$ (i.e., with XOR), or take the third summand in the form $2 \cdot (w(\pi(x) + 1) - w(\pi(x)))$ (or $2 \cdot (w(\pi(x) + 1) + \mathsf{NOT}(w(\pi(x))))$ instead of $4 \cdot w(\pi(x_i))$, etc. Once again we emphasize that both v and w could be arbitrary compositions of the above mentioned

[2]This approach has been already resulted in a very fast and flexible stream cipher ABC v.2, see [10],[2].

[3]we count overlapping k-tuples either

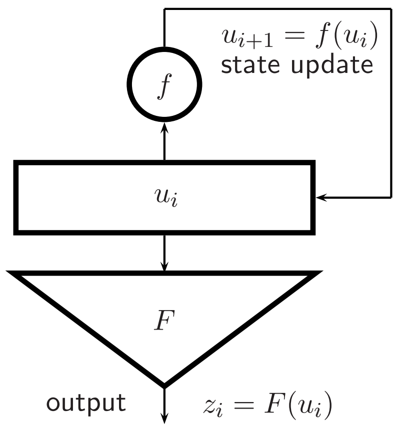

FIGURE 1. Ordinary PRNG

machine instructions (and derived ones); e.g., in the above example one might take[4]

$$v(x) = \left(1 + 2 \cdot \frac{(x \,\mathsf{AND}(x^2 + x^3)) \,\mathsf{OR}\, x^4}{3 + 4 \cdot (5 + 6x^5)^{x^6 \,\mathsf{XOR}\, x^7}}\right)^{7 + \frac{8x^8}{9+10x^9}}$$

We assume here and on that all the operands are non-negative integer rationals represented in their base-2 expansions; so, for instance, $2 = 1 \,\mathsf{XOR}\, 3 = 2 \,\mathsf{AND}\, 7 \equiv \mathsf{NOT}\, 13 \pmod 8$, $\frac{1}{3} \equiv 3^{-1} \equiv 11 \equiv -5 \pmod{16}$, $3^{-\frac{1}{3}} \equiv 3^{11} \equiv 3^{-5} \equiv 11 \pmod{16}$, etc. Up to this agreement the functions v and w are well defined. The performance of the whole scheme depends only on the ratio of 'fast' and 'slow' operations in these compositions; one may vary this ratio in a wide range to achieve desirable speed.

The paper is organized as follows. Section 2 concerns basic facts about functions we use as 'building blocks' of our generators, Section 3 describes how to construct a generator out of these blocks, Section 4 studies properties of output sequences of these generators, and Section 5 gives some reasoning why (some of) these generators could be provably secure. Due to the space constraints, no proofs are given.

2. Preliminaries

Basically, the generator we consider in the paper is a finite automaton $\mathfrak{A} = \langle N, M, f, F, u_0\rangle$ with a finite state set N, state transition function $f : N \to N$, finite output alphabet M, output function $F : N \to M$ and an initial state (seed) $u_0 \in N$. Thus, this generator (see Figure 1) produces a sequence

$$\mathcal{S} = \{F(u_0), F(f(u_0)), F(f^{(2)}(u_0)), \dots, F(f^{(j)}(u_0)), \dots\}$$

over the set M, where

$$f^{(j)}(u_0) = \underbrace{f(\dots f(}_{j \text{ times}} u_0) \dots) \quad (j = 1, 2, \dots); \quad f^{(0)}(u_0) = u_0.$$

[4]this example is of no practical value; it serves only to illustrate how 'crazy' the compositions could be

Automata of the form $\mathfrak{A}$ could be used either as pseudorandom generators per se, or as components of more complicated pseudorandom generators, the so called *counter-dependent generators* (see Figure 2); the latter produce sequences $\{z_0, z_1, z_2, \ldots\}$ over M according to the rule

$$z_0 = F_0(u_0), u_1 = f_0(u_0); \ldots z_i = F_i(u_i), u_{i+1} = f_i(u_i); \ldots \tag{2.0.1}$$

That is, at the $(i+1)^{\text{th}}$ step the automaton $\mathfrak{A}_i = \langle N, M, f_i, F_i, u_i\rangle$ is applied to the state $u_i \in N$, producing a new state $u_{i+1} = f_i(u_i) \in N$, and outputting a symbol $z_i = F_i(u_i) \in M$.

Now we give a more formal

2.1. **Definition.** Let $\mathfrak{A}_j = \langle N, M, f_j, F_j\rangle$ be a family of automata with the same state set N and the same output alphabet M indexed by elements of a non-empty (possibly, countably infinite) set J (members of the family need not be necessarily pairwise distinct). Let $T\colon J \to J$ be an arbitrary mapping. A *wreath product* of the family $\{\mathfrak{A}_j\}$ of automata with respect to the mapping T is an automaton with the state set $N \times J$, state transition function $\breve{f}(j, z) = (f_j(z), T(j))$ and output function $\breve{F}(j, z) = F_j(z)$. The state transition function $\breve{f}(j, z) = (f_j(z), T(j))$ is called a *wreath product of a family of mappings* $\{f_j\colon j \in J\}$ *with respect to the mapping* T [5]. We call f_j (resp., F_j) *clock* state update (resp., output) functions.

It worth notice here that if $J = \mathbb{N}_0$ and F_i does not depend on i, this construction gives us a number of examples of counter-dependent generators in the sense of [23, Definition 2.4], where the notion of a counter-dependent generator was originally introduced. However, we use this notion in a broader sense in comparison with that of [23]: In our counter-dependent generators not only the state transition function, but also the output function depends on i. Moreover, in [23] only a special case of counter-dependent generators is studied; namely, counter-assisted generators and their cascaded and two-step modifications. A state transition function of a counter-assisted generator is of the form $f_i(x) = i \star h(x)$, where $\star$ is a binary quasigroup operation (in particular, group operation, e.g., $+$ or XOR), and $h(x)$ does not depend on i. An output function of a counter-assisted generator does not depend on i either. Finally, our constructions provide long period, uniform distribution, and high linear complexity of output sequences; cf. [23], where only the diversity is guaranteed.

Throughout the paper we assume that $N = \mathbb{I}_n(p) = \{0, 1, \ldots, p^n - 1\}$, $M = \mathbb{I}_m(p)$, $m \le n$, where p is a prime. Moreover, mainly we are focused on the case $p = 2$ as the most suitable for computer implementations. It is convenient to think of elements $z \in \mathbb{I}_n(p)$ as base-p expansions of rational integers:

$$z = \delta_0^p(z) + \delta_1^p(z) \cdot p + \cdots + \delta_{n-1}^p(z) \cdot p^{n-1};$$

here $\delta_j^p(z) \in \{0, 1, \ldots, p-1\}$. For $p = 2$ we usually omit the superscript, when this does not lead to misunderstanding. Further we usually identify $\mathbb{I}_n(p)$ with the ring $\mathbb{Z}/p^n$ of residues modulo p^n.

As said above, we consider bitwise logical operators as functions defined on the set $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$ Machine instructions SHR_m and SHL_m — an m-bit right shift ($\cdot \Rsh m$, which is a multiplication by 2^m) and an m-bit left shift ($\cdot \Lsh m$, integer division by 2^m, i.e., $\lfloor \frac{\cdot}{2^m} \rfloor$, with $\lfloor \alpha \rfloor$ being the greatest rational integer that does not exceed α) are defined on $\mathbb{N}_0$ either. *Note that since this moment throughout the paper we represent integers i in reverse bit order — less significant bits left, according to their occurrences in 2-adic canonical representation of* $i = \delta_0(i) + \delta_1(i) \cdot 2 + \delta_2(i) \cdot 4 + \ldots$; so 0011 is 12, and not 3. Moreover, one may think about these logical and machine operators, as well as of numerical, i.e., arithmetic ones

[5] cf. *skew shift* in ergodic theory; cf. round function in the Feistel network. We are using a term from group theory.

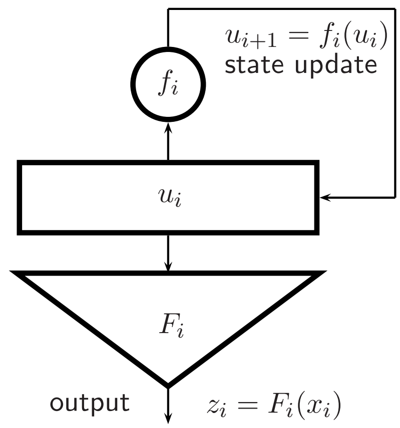

Figure 2. Counter-dependent PRNG

(addition, multiplication, etc.), as of functions that are defined on (and valuated in) the set $\mathbb{Z}_2$ of all 2-adic integers[6] (see [3, 5]), e.g., $x \,\mathsf{OR}\, y = (\delta_0(x) \vee \delta_0(y)) + (\delta_1(x) \vee \delta_1(y)) \cdot 2 + (\delta_2(x) \vee \delta_2(y)) \cdot 2^2 + \ldots$.

A common feature of the above mentioned operations is that they all, with exception of shifts towards less significant bits and circular rotations[7], are *compatible*, i.e., $\omega(u, v) \equiv \omega(u_1, v_1) \pmod{2^r}$ whenever both congruences $u \equiv u_1 \pmod{2^r}$ and $v \equiv v_1 \pmod{2^r}$ hold simultaneously. The notion of compatible mapping could be naturally generalized to multivariate mappings $(\mathbb{Z}/p^l)^t \to (\mathbb{Z}/p^l)^s$ and $(\mathbb{Z}_p)^t \to (\mathbb{Z}_p)^s$ over a residue ring modulo p^l (resp., the ring $\mathbb{Z}_p$ of p-adic integers). Obviously, a composition of compatible mappings is a compatible mapping. We list now some important examples of compatible operators $(\mathbb{Z}_p)^2 \to \mathbb{Z}_p$, p prime (see [5]). Part of them originates from arithmetic operations:

$$\begin{aligned}
&\text{multiplication, } \cdot: \ (u, v) \mapsto uv;\\
&\text{addition, } +: \ (u, v) \mapsto u + v;\\
&\text{subtraction, } -: \ (u, v) \mapsto u - v;\\
&\text{exponentiation, } \uparrow_p\!: \ (u, v) \mapsto u \uparrow_p v = (1 + pu)^v; \text{ in particular,}\\
&\text{raising to negative powers, } u \uparrow_p (-r) = (1 + pu)^{-r}, r \in \mathbb{N}; \text{ and}\\
&\text{division, } /_p : u/_p v = u \cdot (v \uparrow_p (-1)) = \frac{u}{1 + pv}.
\end{aligned} \tag{2.1.1}$$

[6]The latter ones within the context of this paper could be thought of as countable infinite binary sequences with members indexed by $0, 1, 2, \ldots$; $\mathbb{Z}_2$ is a metric space with respect to the 2-adic norm $\|\alpha\|_2 = 2^{-k}$, where k is the number of the first zero members of the sequence $\alpha \in \mathbb{Z}_2$: $\|0\| = \|000\ldots\|_2 = 0$, $\|1\| = \|100\ldots\|_2 = 1$, $\|2\| = \|010\ldots\|_2 = \frac{1}{2}$, etc.

[7]nevertheless, the both are used in further constructions

The other part originates from digitwise logical operations of p-valued logic:

$$\begin{aligned} &\text{digitwise multiplication } u \odot_p v : \delta_j(u \odot_p v) \equiv \delta_j(u)\delta_j(v) \pmod p; \\ &\text{digitwise addition } u \oplus_p v : \delta_j(u \oplus_p v) \equiv \delta_j(u) + \delta_j(v) \pmod p; \\ &\text{digitwise subtraction } u \ominus_p v : \delta_j(u \ominus_p v) \equiv \delta_j(u) - \delta_j(v) \pmod p. \end{aligned} \tag{2.1.2}$$

Here $\delta_j(z)$ $(j = 0, 1, 2, \ldots)$ stands for the j^{th} digit of z in its base-p expansion.

More compatible mappings could be derived from the above mentioned ones. For instance, a reduction modulo p^n, $n \in \mathbb{N}$, is $u \bmod p^n = u \odot_p \frac{p^n-1}{p-1}$, an l-step shift towards more significant digits is just a multiplication by p^l, etc. Obviously, $u \odot_2 v = u \,\mathsf{AND}\, v$, $u \oplus_2 v = u \,\mathsf{XOR}\, v$. Further in case $p = 2$ we omit subscripts of the corresponding operators.

In case $p = 2$ compatible mappings could be characterized in terms of Boolean functions. Namely, each mapping $T\colon \mathbb{Z}/2^n \to \mathbb{Z}/2^n$ could be considered as an ensemble of n Boolean functions τ_i^T, $i = 0, 1, 2, \ldots, n-1$, in n Boolean variables $\chi_0, \ldots, \chi_{n-1}$ by assuming $\chi_i = \delta_i(u)$, $\tau_i^T(\chi_0, \ldots, \chi_{n-1}) = \delta_i(T(u))$ for u running from 0 to $2^n - 1$. The following proposition holds.

2.2. **Proposition.** ([3, Proposition 3.9]) *A mapping $T\colon \mathbb{Z}/2^n \to \mathbb{Z}/2^n$ (resp., a mapping $T\colon \mathbb{Z}_2 \to \mathbb{Z}_2$) is compatible iff each Boolean function $\tau_i^T(\chi_0, \chi_1, \ldots) = \delta_i(T(u))$, $i = 0, 1, 2, \ldots,$ does not depend on the variables $\chi_j = \delta_j(u)$ for $j > i$.*

Note. Mappings satisfying conditions of the proposition are also known in the theory of Boolean functions as *triangle* mappings; the term *T-functions* is used in [17], [16], [15] instead. For multivariate mappings theorem 2.2 holds either: A mapping $T = (t_1, \ldots, t_s)\colon (Z_2)^{(r)} \to (Z_2)^{(s)}$ is compatible iff each Boolean function $\tau_i^{t_j}(\chi_{1,0}, \chi_{1,1}, \ldots, \chi_{r,0}, \chi_{r,1}, \ldots) = \delta_i(t_k(u, \ldots, u_r))$ $(i \in \mathbb{N}_0,\ k = 0, 1, \ldots, s)$ does not depend on the variables $\chi_{\ell,j} = \delta_j(u_\ell)$ for $j > i$ $(\ell = 1, 2, \ldots, r)$.

Now, given a compatible mapping $T\colon \mathbb{Z}_2 \to \mathbb{Z}_2$, one can define an induced mapping $T \bmod 2^n\colon \mathbb{Z}/2^n \to \mathbb{Z}/2^n$ assuming $(T \bmod 2^n)(z) = T(z) \bmod 2^n = (T(z)) \,\mathsf{AND}(2^n-1)$ for $z = 0, 1, \ldots, 2^n-1$. Obviously, $T \bmod 2^n$ is also compatible. For odd prime p, as well as for multivariate case $T\colon (\mathbb{Z}_p)^s \to (\mathbb{Z}_p)^t$ an induced mapping $T \bmod p^n$ could be defined by analogy.

2.3. **Definition.** (See [5]). We call a compatible mapping $T\colon \mathbb{Z}_p \to \mathbb{Z}_p$ *bijective modulo* p^n iff the induced mapping $T \bmod p^n$ is a permutation on $\mathbb{Z}/p^n$; we call T *transitive modulo* p^n, iff $T \bmod p^n$ is a permutation with a single cycle. We say that T is *measure-preserving* (respectively, *ergodic*), iff T is bijective (respectively, transitive) modulo p^n for all $n \in \mathbb{N}$. We call a compatible mapping $T\colon (\mathbb{Z}_p)^s \to (\mathbb{Z}_p)^t$ *balanced modulo* p^n iff the induced mapping $T \bmod p^n$ maps $(\mathbb{Z}/p^n)^s$ onto $(\mathbb{Z}/p^n)^t$, and each element of $(\mathbb{Z}/p^n)^t$ has the same number of preimages in $(\mathbb{Z}/p^n)^s$. Also, the mapping $T\colon (\mathbb{Z}_p)^s \to (\mathbb{Z}_p)^t$ is called *measure-preserving* iff it is balanced modulo p^n for all $n \in \mathbb{N}$.[8]

Both transitive modulo p^n and balanced modulo p^n mappings could be used as building blocks of pseudorandom generators to provide both long period and uniform distribution of output sequences. The following obvious proposition holds.

[8]The terms measure-preserving and ergodic originate from the theory of dynamical systems. Namely, a mapping $T\colon \mathbb{Z}_p \to \mathbb{Z}_p$ is compatible iff it satisfies Lipschitz condition with a constant 1 with respect to the p-adic metric; T defines a dynamics on the measurable space $\mathbb{Z}_p$ with respect to the normalized Haar measure. The mapping T is, e.g., ergodic with respect to this measure (in the sense of the theory of dynamical systems) iff it satisfies 2.3, see [5] for details.

2.4. **Proposition.** *If the state transition function f of the automaton $\mathfrak{A}$ is transitive on the state set N, i.e., if f is a permutation with a single cycle of length $|N|$; if, further, $|M|$ is a factor of $|N|$, and if the output function $F: N \to M$ is balanced (i.e., $|F^{-1}(s)| = |F^{-1}(t)|$ for all $s, t \in M$), or, in particular, bijective, then the output sequence $\mathcal{S}$ of the automaton $\mathfrak{A}$ is purely periodic with a period of length $|N|$* (i.e., maximum possible)*, and each element of M occurs at the period the same number of times: $\frac{|N|}{|M|}$ exactly.* That is, the output sequence $\mathcal{S}$ is uniformly distributed.

2.5. **Definition.** Further in the paper we call a sequence $\mathcal{S} = \{s_i \in M\}$ over a finite set M purely periodic with a period of length t iff $s_{i+t} = s_i$ for all $i = 0, 1, 2, \ldots$. The sequence $\mathcal{S}$ is called *strictly uniformly distributed* iff it is purely periodic with a period of length t, and every element of M occurs at the period the same number of times, i.e., exactly $\frac{t}{|M|}$. A sequence $\{s_i \in \mathbb{Z}_p\}$ of p-adic integers is called *strictly uniformly distributed modulo p^k* iff the sequence $\{s_i \bmod p^k\}$ of residues modulo p^k is strictly uniformly distributed over a residue ring $\mathbb{Z}/p^k$.

Note. A sequence $\{s_i \in \mathbb{Z}_p : i = 0, 1, 2, \ldots\}$ of p-adic integers is uniformly distributed (with respect to the normalized Haar measure μ on $\mathbb{Z}_p$) [9] iff it is uniformly distributed modulo p^k for all $k = 1, 2, \ldots$; that is, for every $a \in \mathbb{Z}/p^k$ relative numbers of occurrences of a in the initial segment of length ℓ in the sequence $\{s_i \bmod p^k\}$ of residues modulo p^k are asymptotically equal, i.e., $\lim_{\ell\to\infty} \frac{A(a,\ell)}{\ell} = \frac{1}{p^k}$, where $A(a, \ell) = |\{s_i \equiv a \pmod{p^k} : i < \ell\}|$ (see [20] for details). So strictly uniformly distributed sequences are uniformly distributed in the common meaning of the theory of distribution of sequences.

Thus, assuming $N = \mathbb{Z}/2^n, M = \mathbb{Z}/2^m, n = km,\ f = \overline{f} = \widetilde{f} \bmod 2^n$ and $F = \overline{F} = \widetilde{F} \bmod 2^m$, where the function $\widetilde{f} : \mathbb{Z}_2 \to \mathbb{Z}_2$ is compatible and ergodic, and the function $\widetilde{F} : (\mathbb{Z}_2)^k \to \mathbb{Z}_2$ is compatible and measure-preserving, we obtain an automaton that generates a uniformly distributed periodic sequence, and length of a period of this sequence is 2^n. That is, each element of $\mathbb{Z}/2^m$ occurs at the period the same number of times (namely, 2^{n-m}). Obviously, the conclusion holds if one takes as F an arbitrary composition of the function $\overline{F} = \widetilde{F} \bmod 2^m$ with a measure-preserving function: For instance, one may put $F(i) = \overline{F}(\pi(i))$ or $F(i) = \delta_j(i)$, etc. Thus, proposition 2.4 makes it possible to vary both the state transition and the output functions (for instance, to make them key-dependent, or in order to achieve better performance[10]) *leaving the output sequence uniformly distributed.*

There exists an easy way to construct a measure preserving or ergodic mapping out of an arbitrary compatible mapping, i.e., out of an arbitrary composition of both arithmetic (2.1.1) and logical (2.1.2) operators.

2.6. **Proposition.** [5, Lemma 2.1 and Theorem 2.5]. *Let Δ be a difference operator, i.e., $\Delta g(x) = g(x+1) - g(x)$ by the definition. Let, further, p be a prime, let c be a coprime with p, $\gcd(c, p) = 1$, and let $g : \mathbb{Z}_p \to \mathbb{Z}_p$ be a compatible mapping. Then the mapping $z \mapsto c + z + p \cdot \Delta g(z)$ $(z \in \mathbb{Z}_p)$ is ergodic, and the mapping $z \mapsto d + cx + p \cdot g(x)$ preserves measure for an arbitrary d. Moreover, if $p = 2$, then the converse also holds: Each compatible and ergodic (respectively, each compatible and measure preserving) mapping $z \mapsto f(z)$ $(z \in \mathbb{Z}_2)$ could be represented as $f(x) = 1 + x + 2 \cdot \Delta g(x)$ (respectively, as $f(x) = d + x + 2 \cdot g(x)$) for suitable $d \in \mathbb{Z}_2$ and compatible $g : \mathbb{Z}_2 \to \mathbb{Z}_2$.*

---

[9] i.e., $\mu(a + p^k\mathbb{Z}_p) = p^{-k}$ for all $a \in \mathbb{Z}_p$ and all $k = 0, 1, 2. \ldots$

[10] e.g., in [17] there was introduced a fast generator of this kind: $f(x) = (x + (x^2 \text{ OR } C)) \bmod 2^{2n}$, $F(x) = \lfloor \frac{x}{2^n} \rfloor \bmod 2^n$

2.7. **Corollary.** *Let $p=2$, and let f be a compatible and ergodic mapping of $\mathbb{Z}_2$ onto itself. Then for each $n=1,2,\dots$ the state transition function $f \bmod 2^n$ could be represented as a finite composition of bitwise logical and arithmetic operators.*

For the sequel we need one more representation, in a Boolean form (see 2.2). The following theorem is just a restatement of a known result from the theory of Boolean functions, the so-called bijectivity/transitivity criterion for triangle Boolean mappings. However, the criterion belongs to the mathematical folklore; thus it is difficult to attribute it to somebody, yet a reader could find a proof in, e.g., [3, Lemma 4.8]. Recall that every Boolean function $\psi(\chi_0,\dots\chi_n)$ in the Boolean variables $\chi_0,\dots\chi_n$ admits a unique representation in the form

$$\psi(\chi_0,\dots\chi_n)\equiv\sum_{\varepsilon_0,\dots,\varepsilon_n\in\{0,1\}}\xi_{\varepsilon_0,\dots,\varepsilon_n}\chi_0^{\varepsilon_0}\cdots\chi_i^{\varepsilon_n}\pmod 2,$$

where $\xi_{\varepsilon_0,\dots,\varepsilon_n}\in\{0,1\}$; the sum in the right hand part is called an algebraic normal form (ANF) of the Boolean function ψ. The degree $\deg\psi$ is $\max\{\varepsilon_0+\dots+\varepsilon_n\colon \xi_{\varepsilon_0,\dots,\varepsilon_n}=1\}$.

2.8. **Theorem.** *A mapping $T\colon\mathbb{Z}_2\to\mathbb{Z}_2$ is compatible and measure-preserving iff for each $i=0,1,\dots$ the ANF of the Boolean function $\tau_i^T=\delta_i(T)$ in Boolean variables $\chi_0,\dots,\chi_i$ could be represented as*

$$\tau_i^T(\chi_0,\dots,\chi_i)=\chi_i+\varphi_i^T(\chi_0,\dots,\chi_{i-1}),$$

where φ_i^T is a Boolean function. The mapping T is compatible and ergodic iff, additionally, the Boolean function φ_i^T is of odd weight, that is, takes value 1 exactly at the odd number of points $(\varepsilon_0,\dots,\varepsilon_{i-1})$, where $\varepsilon_j\in\{0,1\}$ for $j=0,1,\dots,i-1$. The latter holds if and only if $\varphi_0^T=1$ and degree of φ_i^T for $i\ge 1$ is exactly i, that is, the ANF of φ_i^T contains a monomial $\chi_0\cdots\chi_{i-1}$.

2.9. **Corollary.** *There are exactly 2^{2^n-n-1} compatible and transitive mappings of $\mathbb{Z}/2^n$ onto $\mathbb{Z}/2^n$.*

From theorem 2.8 follows an easy way to produce new ergodic functions out of given ones:

2.10. **Proposition.** *For any ergodic f and any compatible v the following functions are ergodic: $f(x+4\cdot v(x))$, $f(x\oplus(4\cdot v(x)))$, $f(x)+4\cdot v(x)$, and $f(x)\oplus(4\cdot v(x))$.*

With the use of theorem 2.8 one can determine whether a given compatible mapping f preserves measure (or is ergodic) assuming it is bijective (respectively, transitive) modulo 2^n and studying behaviour of the Boolean function $\delta_n(f)$. This approach is called a bit-slice analysis in [17], [16], and [15]. More 'analytic' techniques based on p-adic differential calculus and Mahler interpolation series were developed in [9], [3], and [5]; see also [21],[19] and [7] for various examples of compatible and ergodic functions, for instance:

- (see [9], [3]) The function $f(x)=a+a_1(x\oplus b_1)+\dots+a_k(x\oplus b_k)$ is ergodic iff it is transitive modulo 4;
- (see [9], [3]) The function $f(x)=a+a_0\cdot\delta_0(x)+a_1\cdot\delta_1(x)+\cdots$ is compatible and ergodic iff $a\equiv 1\pmod 2$, $a_0\equiv 1\pmod 4$, and $a_i\equiv 0\pmod{2^i}$, $a_i\not\equiv 0\pmod{2^{i+1}}$ for $i=1,2,\dots$;
- (see [19]) The function
$$f(x)=(\dots((((x+c_0)\oplus d_0)+c_1)\oplus d_1)+\dots+c_m)\oplus d_m,$$
is ergodic iff f is transitive modulo 4;
- (see [17]) The function $f(x)=x+((x^2)\,\mathsf{OR}\,c)$ is ergodic iff $c\equiv 5\pmod 8$ or $c\equiv 7\pmod 8$ (an equivalent statement — iff f is transitive modulo 8);

- (see [21]) The polynomial $f(x) = a_0 + a_1x + \cdots + a_dx^d$ with integral coefficients is ergodic iff the following congruences hold simultaneously:
$$a_3 + a_5 + a_7 + a_9 + \cdots \equiv 2a_2 \pmod 4;\; a_4 + a_6 + a_8 + \cdots \equiv a_1 + a_2 - 1 \pmod 4;$$
$$a_1 \equiv 1 \pmod 2;\; a_0 \equiv 1 \pmod 2$$
(an equivalent statement — iff f is transitive modulo 8);
- (see [5]) A polynomial of degree d with rational (and not necessarily integral) coefficients is integer-valued, compatible, and ergodic iff f takes integral values at the points
$$0, 1, \ldots, 2^{\lfloor \log_2(\deg f)\rfloor+3} - 1,$$
and the mapping
$$z \mapsto f(z) \bmod 2^{\lfloor \log_2(\deg f)\rfloor+3},$$
is compatible and transitive on the residue class ring $\mathbb{Z}/2^{\lfloor \log_2 d\rfloor+3}$ (i.e., modulo the biggest power of 2 not exceeding $8d$);
- (see [9], [3]) The entire function $f(x) = \frac{u(x)}{1+2\cdot v(x)}$, where $u(x), v(x)$ are polynomials with integral coefficients, is ergodic iff it is transitive modulo 8;
- (see [7, Example 3.6]) The function $f(x) = ax + a^x$ is ergodic iff a is odd (an equivalent statement — iff f is transitive modulo 2).

A multivariate case was studied in [15], [8]; see also [5, Theorem 3.11]. Multivariate ergodic mappings could be of use in order to produce longer periods out of shorter words operations: For instance, to obtain a period of length 2^{256} one may use either univariate ergodic functions (hence, 256-bit operands) or he may use 8-variate ergodic functions and work with 32-bit words. Multivariate ergodic mappings of [15] are conjugate to univariate ones (see [8]); hence *despite all further results are stated for a univariate case, they hold for these multivariate mappings as well*. Thus a designer could use further constructions either with longer words organized into 1-dimensional arrays, or with shorter words organized into arrays of bigger dimensions.

3. Constructions

In this section we introduce a method to construct counter dependent pseudorandom generators out of ergodic and measure-preserving mappings. The method guarantees that output sequences of these generators are always strictly uniformly distributed. Actually, all these constructions are wreath products of automata in the sense of 2.1; the following results give us conditions these automata should satisfy to produce a uniformly distributed output sequence. Our main technical tool is the following

3.1. **Theorem.** *Let $\mathcal{G} = \{g_0, \ldots, g_{m-1}\}$ be a finite sequence of compatible measure preserving mappings of $\mathbb{Z}_2$ onto itself such that*

(1) *the sequence $\{(g_{i \bmod m}(0)) \bmod 2 \colon i = 0, 1, 2, \ldots\}$ is purely periodic, its shortest period is of length m;*
(2) $\sum_{i=0}^{m-1} g_i(0) \equiv 1 \pmod 2$*;*
(3) $\sum_{j=0}^{m-1} \sum_{z=0}^{2^k-1} g_j(z) \equiv 2^k \pmod{2^{k+1}}$ *for all $k = 1, 2, \ldots$.*

Then the recurrence sequence $\mathcal{Z}$ defined by the relation $x_{i+1} = g_{i \bmod m}(x_i)$ is strictly uniformly distributed modulo 2^n for all $n = 1, 2, \ldots$: That is, modulo each 2^n the sequence $\mathcal{Z}$ is purely periodic, its shortest period is of length $2^n m$, and each element of $\mathbb{Z}/2^n$ occurs at the period exactly m times.

Note. In view of 2.8 condition (3) of theorem 3.1 could be replaced by the equivalent condition

$$\sum_{j=0}^{m-1} \mathrm{Coef}_{0,\ldots,k-1}(\varphi_k^j) \equiv 1 \pmod 2 \qquad (k = 1, 2, \ldots),$$

where $\mathrm{Coef}_{0,\ldots,k-1}(\varphi)$ is a coefficient of the monomial $\chi_0 \cdots \chi_{k-1}$ in the Boolean polynomial φ.

It turns out that the sequence $\mathcal{Z}$ of 3.1 is just the sequence $\mathcal{Y}$ of the following

3.2. **Lemma.** *Let $c_0, \ldots, c_{m-1}$ be a finite sequence of 2-adic integers, and let $g_0, \ldots, g_{m-1}$ be a finite sequence of compatible mappings of $\mathbb{Z}_2$ onto itself such that*

(i) $g_j(x) \equiv x + c_j \pmod 2$ *for* $j = 0, 1, \ldots, m-1$,
(ii) $\sum_{j=0}^{m-1} c_j \equiv 1 \pmod 2$,
(iii) *the sequence* $\{c_{i \bmod m} \bmod 2 \colon i = 0, 1, 2, \ldots\}$ *is purely periodic, its shortest period is of length* m,
(iv) $\delta_k(g_j(z)) \equiv \zeta_k + \varphi_k^j(\zeta_0, \ldots, \zeta_{k-1}) \pmod 2$, $k = 1, 2, \ldots$, *where* $\zeta_r = \delta_r(z)$, $r = 0, 1, 2, \ldots$,
(v) *for each* $k = 1, 2, \ldots$ *an odd number of Boolean polynomials* φ_k^j *in the Boolean variables* $\zeta_0, \ldots, \zeta_{k-1}$ *are of odd weight.*

Then the recurrence sequence $\mathcal{Y} = \{x_i \in \mathbb{Z}_2\}$ defined by the relation $x_{i+1} = g_{i \bmod m}(x_i)$ is strictly uniformly distributed: It is purely periodic modulo 2^k for all $k = 1, 2, \ldots$; its shortest period is of length $2^k m$; each element of $\mathbb{Z}/2^k$ occurs at the period exactly m times. Moreover,

(1) *the sequence* $\mathcal{D}_s = \{\delta_s(x_i) \colon i = 0, 1, 2, \ldots\}$ *is purely periodic; it has a period of length* $2^{s+1}m$,
(2) $\delta_s(x_{i+2^s m}) \equiv \delta_s(x_i) + 1 \pmod 2$ *for all* $s = 0, 1, \ldots, k-1$, $i = 0, 1, 2, \ldots$,
(3) *for each* $t = 1, 2, \ldots, k$ *and each* $r = 0, 1, 2, \ldots$ *the sequence*

$$x_r \bmod 2^t, x_{r+m} \bmod 2^t, x_{r+2m} \bmod 2^t, \ldots$$

is purely periodic, its shortest period is of length 2^t, each element of $\mathbb{Z}/2^t$ occurs at the period exactly once.

3.3. *Note.* Assuming $m = 1$ in 3.1 one obtains ergodicity criterion 2.8.

3.4. **Corollary.** *Let a finite sequence of mappings $\{g_0, \ldots, g_{m-1}\}$ of $\mathbb{Z}_2$ into itself satisfy conditions of theorem 3.1, and let $\{F_0, \ldots, F_{m-1}\}$ be an arbitrary finite sequence of balanced (and not necessarily compatible) mappings of $\mathbb{Z}/2^n$ $(n \geq 1)$ onto $\mathbb{Z}/2^k$, $1 \leq k \leq n$. Then the sequence $\mathcal{F} = \{F_{i \bmod m}(x_i) \colon i = 0, 1, 2 \ldots\}$, where $x_{i+1} = g_{i \bmod m}(x_i) \bmod 2^n$, is strictly uniformly distributed over $\mathbb{Z}/2^k$: It is purely periodic with a period of length $2^n m$, and each element of $\mathbb{Z}/2^k$ occurs at the period exactly $2^{n-k}m$ times.*

Theorem 3.1 and lemma 3.2 together with corollary 3.4 enables one to construct a counter-dependent generator out of the following components:

- A sequence $c_0, \ldots, c_{m-1}$ of integers, which we call a *control sequence*.
- A sequence $h_0, \ldots, h_{m-1}$ of compatible mappings, which is used to form a sequence of clock state update functions g_i (see e.g. examples 3.5).
- A sequence $H_0, \ldots, H_{m-1}$ of compatible mappings to produce clock output functions F_i (see e.g. proposition 4.9).

Note that ergodic functions that are needed to meet conditions of 4.9 or 3.5 (3) could be produced out of compatible ones with the use of 2.6 or 2.10. A control sequence could be produced by an external generator (which in turn could be a generator of the kind considered in this paper), or it could be just a queue the state update and output functions are called from a look-up table. The functions h_i and/or H_i could be either precomputed to arrange that look-up table, or they could be produced on-the-fly in a form that is determined by a control sequence. This form may also look 'crazy', e.g.,

$$h_i(x) = (\cdots((u_0(\delta_0(c_i)) \bigcirc_{\delta_1(c_i),\delta_2(c_i)} u_1(\delta_3(c_i))) \bigcirc_{\delta_4(c_i),\delta_5(c_i)} u_2(\delta_6(c_i)))\cdots, \tag{3.4.1}$$

where $u_j(0) = x$, the variable, and $u_j(1)$ is a constant (which is determined by c_i, or is read from a precomputed look-up table, etc.), while (say) $\bigcirc_{0,0} = +$, an integer addition, $\bigcirc_{1,0} = \cdot$, an integer multiplication, $\bigcirc_{0,1} = \mathsf{XOR}$, $\bigcirc_{1,1} = \mathsf{AND}$. There is absolutely no matter what these h_i and H_i look like or how they are obtained, *the above stated results give a general method to combine all the data together to produce a uniformly distributed output sequence of a maximum period length.*

3.5. *Examples.* These are obtained with the use of 3.2, 2.8, 2.10, and (5.0.2).

(1) A control sequence could be produced by the generator $\mathfrak{A} = \langle \mathbb{Z}/2^s, \mathbb{Z}/2^s, f, F, u_0\rangle$ (see Section 2) with ergodic state update function f and measure-preserving output function F. Then length of the shortest period of the control sequence is $m = 2^s$, see 2.4. Take m arbitrary ergodic functions $h_0, \ldots, h_{m-1}$ and arbitrary odd $k \in \{0, 1, \ldots, m-1\}$, and put $\breve{g}_0(x) = x \oplus (x+1) \oplus h_0(x), \ldots, \breve{g}_{k-1} = x \oplus (x+1) \oplus h_{k-1}(x)$, $\breve{g}_k = h_k, \ldots, \breve{g}_{m-1} = h_{m-1}$, $g_i = \breve{g}_{c_i \bmod m}$ for $i = 0, 1, 2, \ldots$. In other words, in this case the control sequence just define the queue the functions $\breve{g}_j$ are called, thus producing the output sequence
$$x_0, x_1 = \breve{g}_{c_0}(x_0) \bmod 2^n, x_2 = \breve{g}_{c_1}(x_1) \bmod 2^n, \ldots$$
Obviously, in this example a control sequence could be an arbitrary permutation of $0, 1, \ldots, 2^s - 1$, and not necessarily an output of the generator $\mathfrak{A}$.

(2) Now let $\{c_0, \ldots, c_{m-1}\}$ be an arbitrary sequence of length $m = 2^s$, i.e., $c_0, \ldots, c_{m-1}$ are not necessarily pairwise distinct. Let $\{h_0, \ldots, h_{m-1}\}$ be arbitrary compatible and ergodic mappings. For $0 \le j \le m-1$ put $g_j(x) = c_j + h_j(x)$. [11] These mappings g_j satisfy conditions of theorem 3.1 if and only if $\sum_{j=0}^{2^m-1} c_j \equiv 1 \pmod 2$.

[11] one may also put $g_j(x) = (c_j + x) \oplus (2 \cdot h_j(x))$.

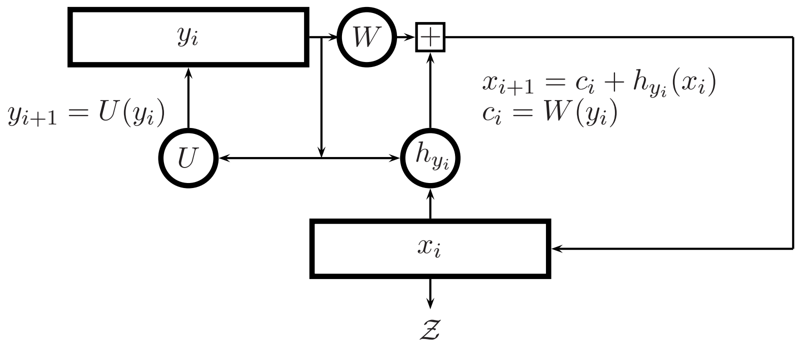

FIGURE 3. Wreath product basic circuit of Examples 3.5, (2)–(4).

(3) For $m > 1$ odd let $\{h_0, \dots, h_{m-1}\}$ be a finite sequence of compatible and ergodic mappings; let $\{c_0, \dots, c_{m-1}\}$ be a finite sequence of integers such that
- $\sum_{j=0}^{m-1} c_j \equiv 0 \pmod 2$, and
- the sequence $\{c_{i \bmod m} \bmod 2\colon i = 0, 1, 2, \dots\}$ is purely periodic with the shortest period of length m.

Put $g_j(x) = c_j \oplus h_j(x)$ (respectively, $g_j(x) = c_j + h_j(x)$). Then g_j satisfy conditions of 3.1.

(4) The conditions of (3) are satisfied in the case $m = 2^s - 1$ and $\{c_0, \dots, c_{m-1}\}$ is the output sequence of a maximum period linear feedback shift register over $\mathbb{Z}/2$ with s cells.

A basic circle illustrating these example wreath products is given at Figure 3. A number of counter dependent generators could be derived from 3.5 by taking explicit expressions for involved mappings. For instance, one can obtain the following result, which is a variation of theme of [16, Theorem 3]). Take odd $m > 1$ and consider a finite sequence $C_0, \dots, C_{m-1}$ of integers such that $\delta_0(C_j) = 1$ and $\delta_2(C_j) = 1$, $j = 0, 1, \dots, m-1$. Let a sequence $\{c_j\colon j = 0, 1, 2, \dots\}$ satisfy conditions of 3.5(3). Then the sequence $\{x_{i+1} = (x_i + c_i + (x_i^2 \,\mathsf{OR}\, C_i)) \bmod 2^n\colon i = 0, 1, 2, \dots\}$ is purely periodic modulo 2^k for all $k = 1, 2, \dots$ with the shortest period of length $2^k m$, and each element of $\mathbb{Z}/2^k$ occurs at the period exactly m times. This is a stronger claim in comparison with that of [16, Theorem 3]): Not only the sequence of pairs (y_i, x_i) defined by $y_{i+1} = (y_i + 1) \bmod m$; $x_{i+1} = (x_i + c_i + (x_i^2 \,\mathsf{OR}\, C_{y_i})) \bmod 2^n$ is periodic with a period of length $2^n m$, yet length of the shortest period of the sequence $\{x_i\}$ is $2^n m$. The latter could never be achieved under conditions of Theorem 3 of [16]: They imply that the length of the shortest period of the sequence $\{x_i \pmod 2\}$ is 2, and not $2m$.

4. Properties of output sequences

Distribution of k-tuples. The output sequence $\mathcal{Z}$ of any wreath product of automata that satisfy 3.1 is strictly uniformly distributed as a sequence over $\mathbb{Z}/2^n$ for all n. That is, each sequence $\mathcal{Z}_n$ of residues modulo 2^n of members of the sequence $\mathcal{Z}$ is purely periodic, and each element of $\mathbb{Z}/2^n$ occurs at the period the same number of times. However, when this sequence $\mathcal{Z}_n$ is used as a key-stream, that is, as a binary sequence $\mathcal{Z}'_n$ obtained by a concatenation of successive n-bit words of $\mathcal{Z}$, it is important to know how n-tuples are distributed in this binary sequence. Yet strict uniform distribution of an arbitrary sequence $\mathcal{T}$ as a sequence over $\mathbb{Z}/2^n$ does not necessarily imply uniform distribution of n-tuples, if this sequence is considered as a binary sequence $\mathcal{T}'$.

For instance, let $\mathcal{T} = 023102310231\dots$. This sequence is strictly uniformly distributed over $\mathbb{Z}/4$; the length of its shortest period is 4. Its binary representation is $\mathcal{T}'_2 = 000111100001111000011110\dots$

Considering $\mathcal{T}$ as a sequence over $\mathbb{Z}/4$, each number of $\{0,1,2,3\}$ occurs in the sequence with the same frequency $\frac{1}{4}$. Yet if we consider $\mathcal{T}$ in its binary form $\mathcal{T}_2'$, then 00 (as well as 11) occurs in this sequence with frequency $\frac{3}{8}$, whereas 01 (as well as 10) occurs with frequency $\frac{1}{8}$.

In this subsection we show that such an effect does not take place for output sequences of automata described in 3.1, 3.2, and 3.5: *Considering any of these sequences in a binary form, a distribution of k-tuples is uniform, for all $k \le n$.* Now we state this property formally.

Consider a (binary) *n-cycle* $C = (\varepsilon_0\varepsilon_1\dots\varepsilon_{n-1})$, i.e., an oriented graph on vertices $\{a_0, a_1, \dots, a_{n-1}\}$ and edges

$$\{(a_0,a_1),(a_1,a_2),\dots,(a_{n-2},a_{n-1}),(a_{n-1},a_0)\},$$

where each vertex a_j is labelled with $\varepsilon_j \in \{0,1\}$, $j = 0,1,\dots,n-1$. (Note that then $(\varepsilon_0\varepsilon_1\dots\varepsilon_{n-1}) = (\varepsilon_{n-1}\varepsilon_0\dots\varepsilon_{n-2}) = \dots$, etc.). Clearly, each purely periodic sequence $\mathcal{S}$ over $\mathbb{Z}/2$ with period $\alpha_0\dots\alpha_{n-1}$ of length n could be related to a binary n-cycle $C(\mathcal{S}) = (\alpha_0\dots\alpha_{n-1})$. Conversely, to each binary n-cycle $(\alpha_0\dots\alpha_{n-1})$ we could relate n purely periodic binary sequences with periods of length n: Those are n shifted versions of the sequence

$$\alpha_0\dots\alpha_{n-1}\alpha_0\dots\alpha_{n-1}\dots.$$

Further, *a k-chain in a binary n-cycle C* is a binary string $\beta_0\dots\beta_{k-1}$, $k < n$, that satisfies the following condition: There exists $j \in \{0,1,\dots,n-1\}$ such that $\beta_i = \varepsilon_{(i+j) \bmod n}$ for $i = 0,1,\dots,k-1$. Thus, a k-chain is just a string of length k of labels that corresponds to a chain of length k in a graph C. We call a binary n-cycle C *k-full*, if each k-chain occurs in the graph C the same number $r > 0$ of times.

Clearly, if C is k-full, then $n = 2^k r$. For instance, a well-known De Bruijn sequence is an n-full 2^n-cycle. Clearly enough that a k-full n-cycle is $(k-1)$-full: Each $(k-1)$-chain occurs in C exactly $2r$ times, etc. Thus, if an n-cycle $C(\mathcal{S})$ is k-full, then each m-tuple (where $1 \le m \le k$) occurs in the sequence $\mathcal{S}$ with the same probability (limit frequency) $\frac{1}{2^m}$. That is, the sequence $\mathcal{S}$ is *k-distributed*, see [18, Section 3.5, Definition D].

4.1**. Definition.** A purely periodic binary sequence $\mathcal{S}$ with the shortest period of length N is said to be *strictly k-distributed* iff the corresponding N-cycle $C(\mathcal{S})$ is k-full.

Thus, if a sequence $\mathcal{S}$ is strictly k-distributed, then it is strictly s-distributed, for all positive $s \le k$.

4.2**. Theorem.** *For the sequence $\mathcal{Z}$ of theorem 3.1 each binary sequence $\mathcal{Z}_n'$ is strictly k-distributed for all $k = 1,2,\dots,n$.*

4.3. *Note.* Theorem 4.2 remains true for the sequence $\mathcal{F}$ of corollary 3.4, where $F_j(x) = \left\lfloor \frac{x}{2^{n-k}} \right\rfloor \bmod 2^k$, $j = 0,1,\dots,m-1$, a truncation of $(n-k)$ less significant bits. Namely, *a binary representation $\mathcal{F}_n'$ of the sequence $\mathcal{F}$ is a purely periodic strictly k-distributed binary sequence with a period of length $2^n mk$.*

Theorem 4.2 treats an output sequence of a counter-dependent automaton as an infinite (though, a periodic) binary sequence. However, in cryptography only a part of a period is used during encryption. So it is natural to ask how 'random' is a finite segment (namely, the period) of this infinite sequence. According to [18, Section 3.5, Definition Q1] a finite binary sequence $\varepsilon_0\varepsilon_1\dots\varepsilon_{N-1}$ of length N is said to be random, iff

$$\left|\frac{\nu(\beta_0\dots\beta_{k-1})}{N} - \frac{1}{2^k}\right| \le \frac{1}{\sqrt{N}} \tag{4.3.1}$$

for all $0 < k \le \log_2 N$, where $\nu(\beta_0 \dots \beta_{k-1})$ is the number of occurrences of a binary word $\beta_0 \dots \beta_{k-1}$ in a binary word $\varepsilon_0\varepsilon_1 \dots \varepsilon_{N-1}$. If a finite sequence is random in the sense of this Definition Q1 of [18], we shall say that this sequence *satisfies* Q1. We shall also say that an *infinite periodic sequence satisfy* Q1 iff its shortest period satisfies Q1. Note that, contrasting to the case of strict k-distribution, which implies strict $(k-1)$-distribution, it is not enough to demonstrate only that (4.3.1) holds for $k = \lfloor \log_2 N \rfloor$ to prove a finite sequence of length N satisfies Q1: For instance, the sequence 1111111100000111 satisfies (4.3.1) for $k = \lfloor \log_2 N \rfloor = 4$ and does not satisfy (4.3.1) for $k = 3$.

4.4. **Corollary.** *The sequence $\mathcal{Z}'_n$ of theorem 4.2 satisfies* Q1 *if $m \le \frac{2^n}{n}$. Moreover, in this case under the conditions of 4.3 the output binary sequence still satisfies* Q1 *if one truncates $0 \le k \le \frac{n}{2} - \log_2 \frac{n}{2}$ lower order bits* (that is, if one uses clock output functions F_j of 4.3).

We note here that according to 4.4 a control sequence of a counter-dependent automaton (see 3.1, 3.2, 3.4, and the text and examples thereafter) may not satisfy Q1 at all, yet nevertheless a corresponding output sequence necessarily satisfies Q1. Thus, *with the use of wreath product techniques one could stretch 'non-randomly looking' sequences to 'randomly looking' ones.*

Structure. A recurrence sequence could be 'very uniformly distributed', yet nevertheless could have some mathematical structure that might be used by an attacker to break the cipher. For instance, a clock sequence $x_i = i$ is uniformly distributed in $\mathbb{Z}_2$; moreover, its counterpart in the field $\mathbb{R}$ of real numbers, the so-called Van der Corput sequence $u_i = i \cdot 2^{-\lfloor \log_2 i \rfloor - 1}$, has the least (of the known) discrepancy, see [20]. We are going to study what structure could have sequences outputted by our counter-dependent generators.

Theorem 3.1 immediately implies that the j^{th} *coordinate sequence* $\delta_j(\mathcal{Z}) = \{\delta_j(x_i) \colon i = 0, 1, 2, \dots\}$ $(j = 0, 1, 2, \dots)$ *of the sequence* $\mathcal{Z}$, i.e., a sequence formed by all j^{th} bits of members of the sequence $\mathcal{Z}$, has a period not longer than $m \cdot 2^{j+1}$. Moreover, the following could be easily proved:

4.5. **Proposition.** (1) *The j^{th} coordinate sequence $\delta_j(\mathcal{Z})$ is a purely periodic binary sequence with a period of length $2^{j+1}m$, and* (2) *the second half of the period is a bitwise negation of the first half: $\delta_j(x_{i+2^j m}) \equiv \delta_j(x_i) + 1 \pmod 2$, $i = 0, 1, 2, \dots$*

This means that the j^{th} coordinate sequence of the sequence of states of a counter-dependent generator is completely determined by the first half of its period; so, intuitively, it is as 'complex' as the first half of its period. Thus we ought to understand what sequences of length $2^j m$ occur as the first half of the period of the j^{th} coordinate sequence.

For $j = 0$ (and $m > 1$) the answer immediately follows from 3.1 and 3.2 — any binary sequence $c_0, \dots, c_{m-1}$ such that $\sum_{j=0}^{m-1} c_j \equiv 1 \pmod 2$ does. It turns out that for $j > 0$ *any binary sequence could be produced as the first half of the period of the j^{th} coordinate sequence independently of other coordinate sequences.*

More formally, to each sequence $\mathcal{Z}$ described by theorem 3.1 we associate a sequence $\Gamma(\mathcal{Z}) = \{\gamma_1, \gamma_2, \dots\}$ of non-negative rational integers γ_j such that $0 \le \gamma_j \le 2^{2^j m} - 1$ and the base-2 expansion of γ_j agrees with the first half of the period of the j^{th} coordinate sequence $\delta_j(\mathcal{Z})$ for all $j = 1, 2, \dots$; that is

$$\gamma_j = \delta_j(x_0) + 2 \cdot \delta_j(x_1) + 4 \cdot \delta_j(x_2) + \dots + 2^{2^j m - 1} \cdot \delta_j(x_{2^j m - 1}),$$

where x_0 is an initial state; $x_{i+1} = g_{i \bmod m}(x_i)$, $i = 0, 1, 2, \dots$. Now we take an arbitrary sequence $\Gamma(\mathcal{Z}) = \{\gamma_1, \gamma_2, \dots\}$ of non-negative rational integers γ_j such that $0 \le \gamma_j \le 2^{2^j m} - 1$ and wonder whether this sequence could be so associated to some sequence $\mathcal{Z}$ described by theorem 3.1.

The answer is *yes*. Namely, the following theorem holds.

4.6. **Theorem.** *Let $m>1$ be a rational integer, and let $\Gamma=\{\gamma_1,\gamma_2,\dots\}$ be an arbitrary sequence over $\mathbb{N}_0$ such that $\gamma_j\in\{1,2,\dots,2^{2^j m}-1\}$ for all $j=1,2,\dots$. Then there exist a finite sequence $\mathcal{G}=\{g_0,\dots,g_{m-1}\}$ of compatible measure preserving mappings of $\mathbb{Z}_2$ onto itself and a 2-adic integer $x_0=z\in\mathbb{Z}_2$ such that $\mathcal{G}$ satisfies conditions of theorem 3.1, and the base-2 expansion of γ_j agrees with the first $2^j m$ terms of the sequence $\delta_j(\mathcal{Z})$ for all $j=1,2,\dots$, where the recurrence sequence $\mathcal{Z}=\{x_0,x_1,\dots\in\mathbb{Z}_2\}$ is defined by the recurrence relation $x_{i+1}=g_{i \bmod m}(x_i)$, $(i=0,1,2,\dots)$. In the case $m=1$ the assertion holds for an arbitrary $\Gamma=\{\gamma_0,\gamma_1,\dots\}$, where $\gamma_j\in\{1,2,\dots,2^{2^j}-1\}$, $j=0,1,2,\dots$.*

Linear complexity. The latter is an important cryptographic measure of complexity of a binary sequence; being a number of cells of the shortest linear feedback shift register (LFSR) that outputs the given sequence[12] it estimates dimensions of a linear system an attacker must solve to obtain initial state.

4.7. **Theorem.** *For $\mathcal{Z}$ and m of theorem 3.1 let $\mathcal{Z}_j=\delta_j(\mathcal{Z})$, $j>0$, be the j^{th} coordinate sequence. Represent $m=2^k r$, where r is odd. Then length of the shortest period of $\mathcal{Z}_j$ is $2^{k+j+1}s$ for some $s\in\{1,2,\dots,r\}$, and both extreme cases $s=1$ and $s=r$ occur: For every sequence $s_1,s_2,\dots$ over a set $\{1,r\}$ there exists a sequence $\mathcal{Z}$ of theorem 3.1 such that length of the shortest period of $\mathcal{Z}_j$ is $2^{k+j+1}s_j$, $(j=1,2,\dots)$. Moreover, linear complexity $\Psi_2(\mathcal{Z}_j)$ of the sequence $\mathcal{Z}_j$ satisfies the following inequality:*

$$2^{k+j}+1\le\Psi_2(\mathcal{Z}_j)\le 2^{k+j}r+1.$$

Both these bounds are sharp: For every sequence $t_1,t_2,\dots$ over a set $\{1,r\}$ there exists a sequence $\mathcal{Z}$ of theorem 3.1 such that linear complexity of $\mathcal{Z}_j$ is exactly $2^{k+j}t_j+1$, $(j=1,2,\dots)$.

Note. Somewhat similar estimates hold for 2-adic span (see definition in [14]), one more cryptographic measure of complexity of a sequence. We have to omit exact statements due to space limitations.

Whereas the linear complexity of a binary sequence $\mathcal{X}$ is the length of the shortest LFSR that produces $\mathcal{X}$, the *ℓ-error linear complexity* is the length of the shortest LFSR that produces a sequence with almost the same (with the exception of not more than ℓ members) period as that of $\mathcal{X}$; that is, the two periods coincide everywhere but at $t\le\ell$ places. Obviously, a random sequence of length L coincides with a sequence that has a period of length L approximately at $\frac{L}{2}$ places. That is, the ℓ-error linear complexity makes sense only for $\ell<\frac{L}{2}$. The following proposition holds.

4.8. **Proposition.** *Let $\mathcal{Z}$ be a sequence of Theorem 3.1, and let $m=2^s>1$. Then for ℓ less than the half of the length of the shortest period of the j-th coordinate sequence $\delta_j(\mathcal{Z})$, the ℓ-error linear complexity of $\delta_j(\mathcal{Z})$ exceeds 2^{j+m-1}, the half of the length of its shortest period.*

From 4.7 it follows that the less is j, the shorter is a period (and the smaller is linear complexity) of the coordinate sequence $\mathcal{Z}_j$. This could be improved by truncation of less significant bits (see 4.4) or, if necessary, with the use of clock output functions of special kind:

4.9. **Proposition.** *Let $H_i\colon\mathbb{Z}_2\to\mathbb{Z}_2$ $(i=0,1,2,\dots,m-1)$ be compatible and ergodic mappings. For $x\in\{0,1,\dots,2^n-1\}$ let $F_i(x)=(H_i(\pi(x)))\bmod 2^n$, where π is a permutation of bits of $x\in\mathbb{Z}/2^n$ such that $\delta_0(\pi(x))=\delta_{n-1}(x)$. Consider a sequence $\mathcal{F}$ of 3.4. Then the shortest period of the j^{th} coordinate*

[12] i.e., degree of the minimal polynomial over $Z/2$ of the given sequence

sequence $\mathcal{F}_j = \delta_j(\mathcal{F})$ $(j = 0, 1, 2, \dots, n-1)$ is of length $2^n k_j$ for a suitable $1 \le k_j \le m$. Moreover, linear complexity of the sequence $\mathcal{F}_j$ exceeds 2^{n-1}.

Note. In view of Note 3.3, all the results of Section 4 remain true for compatible mappings $T\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ (i.e., for T-functions) either.

5. Security issues

The paper introduces design techniques that guarantees in advance that the so constructed generator, which dynamically modifies itself during encryption, will meet certain important cryptographic properties; namely, long period, uniform distribution and high linear complexity of the output sequence. The techniques can not guarantee per se that every such cipher will be secure — obvious degenerative cases exist. On the other hand, if clock state update functions g_i are chosen arbitrarily under the conditions of 3.1, and clock output functions F_i just truncate k low order bits, $k \approx \frac{n}{2}$ (see 4.4), theorem 4.6 leaves no chance to an attacker to break such a scheme. Yet in practice we can not choose g_i arbitrarily; restrictions are determined by concrete implementations, which are not discussed here.

In this section we are going to give some evidence that with the use of the techniques described above it might be possible to design stream ciphers such that the problem of their key recovery is intractable up to the following conjecture: Choose (randomly and independently) $k \le n$ ANF's ψ_i in n Boolean variables $\chi_0, \dots, \chi_{n-1}$ from the class of ANF's with polynomially restricted number of monomials. Consider a mapping $F\colon \mathbb{Z}/2^n \to \mathbb{Z}/2^k$:

$$F(x) = F(\chi_0, \dots, \chi_{n-1}) = \psi_0(\chi_0, \dots, \chi_{n-1}) \oplus \psi_1(\chi_0, \dots, \chi_{n-1}) \cdot 2 \oplus \cdots \oplus \psi_{k-1}(\chi_0, \dots, \chi_{n-1}) \cdot 2^{k-1},$$

where $\chi_j = \delta_j(x)$ for $x \in \mathbb{Z}/2^n$. We conjecture that *this function F is one-way*, that is, one could invert it (i.e., could find an F-preimage in case it exists) only with a negligible in n probability. Note that to find any F-preimage, i.e., to solve an equation $F(x) = y$ in unknown x one has to solve a system of k Boolean equations in n variables. Yet *to determine whether k ANF have common zero is an NP-complete problem*, see e.g. [13, Appendix A, Section A7.2, Problem ANT-9].

Of course, it is not sufficient to conjecture F is one-way in case we only know that the problem of whether F-preimage exists is NP-complete; it must be hard in average to invert F. However, to our best knowledge, no polynomial-time algorithms that solve random systems of k Boolean equations in n variables for so restricted k are known. The best known results are polynomial-time algorithms that solve so-called overdefined Boolean systems of degree not more than 2, i.e., systems where the number of equations is greater than the number of unknowns and where each ANF is at most quadratic, see [11], [12].

Proceeding with the above plausible conjecture, to each ANF ψ_i, $i = 0, 1, 2, \dots, k-1$ we relate a mapping $\Psi_i\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ in the following way: $\Psi_i(x) = \psi_i(\delta_0(x), \dots, \delta_{n-1}(x)) \in \{0, 1\} \subset \mathbb{Z}_2$. Now to each above mapping F we relate a mapping

$$f_F(x) = (1 + x) \oplus 2^{n+1} \cdot F(x) = (1 + x) \oplus 2^{n+1} \cdot \Psi_0(x) \oplus 2^{n+2} \cdot \Psi_1(x) \oplus \cdots \oplus 2^{n+k} \cdot \Psi_{k-1}(x)$$

of $\mathbb{Z}_2$ onto itself. Clearly,

$$\delta_j(f_F(x)) = \begin{cases} 1 \oplus \delta_0(x), & \text{if } j = 0; \\ \delta_j(x) \oplus \delta_0(x) \cdots \delta_{j-1}(x), & \text{if } 0 < j \le n; \\ \delta_j(x) \oplus \delta_0(x) \cdots \delta_{j-1}(x) \oplus \psi_{j-n-1}(\delta_0(x), \dots, \delta_{n-1}(x)), & \text{if } n + 1 \le j \le n + k. \end{cases}$$

In view of 2.8 the mapping $f_F\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ is compatible and ergodic for any choice of ANF's $\psi_0, \dots, \psi_{k-1}$.

Now for $m = 2^n$ and $i = 0, 1, 2, \dots, m-1$ choose arbitrarily and independently mappings $F_i\colon \mathbb{Z}/2^n \to \mathbb{Z}/2^k$ of the above kind. Put $d_0 = \dots = d_{2^n-3} = 0$, $d_{2^n-2} = d_{2^n-1} = 1$, and consider a recurrence sequence of states $x_{i+1} = d_{i \bmod m} \oplus f_{F_{i \bmod m}}(x_i)$ and a corresponding output sequence $g(x_0), g(x_1), \dots$ over $\mathbb{Z}/2^k$, where $g(x) = \lfloor \frac{x}{2^{n+1}} \rfloor \bmod 2^k$, a truncation. In view of 3.5 the output sequence satisfy 3.4.

We shall always take a key $z \in \{0, 1, \dots, 2^n - 1\}$ as an initial state x_0. Let z be the only information that is not known to an attacker, let everything else, i.e., n, k, f_{F_i}, d_i, and g, as well as the first s members of the output sequence $\{y_i\}$, be known to him. Since $\delta_0(x) \cdots \delta_{j-1}(x) = 1$ iff $x \equiv -1 \pmod{2^j}$, with probability $1-\epsilon$ (where ϵ is negligible if s is a polynomial in n) he obtains a sequence[13]:

$$y_0 = F_0(z), y_0 \oplus y_1 = F_1(z+1), \dots, y_{s-2} \oplus y_{s-1} = F_{s-1}(z+s-1) \tag{5.0.1}$$

To find z the attacker may try to solve any of these equations; he could do it with a negligible advantage, since F_i is one-way. Of course, the attacker may try to express $z+i$ as a collection of ANF's $\delta_0(z+i), \dots, \delta_{n-1}(z+i)$ in the variables $\chi_0 = \delta_0(z), \dots, \chi_{n-1} = \delta_{n-1}(z)$, then substitute these ANF's for the variables into the ANF's that define mappings F_i, to obtain an overdefined system (5.0.1) in unknowns $\chi_0, \dots, \chi_{n-1}$. However, the known formula (see e.g. [1] and fix an obvious misprint there)

$$\delta_j(z+i) \equiv \chi_j + \delta_j(i) + \sum_{r=0}^{j-1} \delta_r(i) \cdot \chi_r \prod_{t=r+1}^{j-1} (\delta_t(i) + \chi_t) \pmod 2; \tag{5.0.2}$$

implies that the number of monomials in the equations of the obtained system will be, generally speaking, exponential in n; to say nothing of that the number of operations to make these substitutions and then to collect similar terms is also exponential in n, unless the degree of all ANF's that define all F_i is bounded by a constant (the latter is not a case according to our assumptions).

Finally, our assumption that the attacker knows all F_i seems to be too strong: It is more practical to assume that he does not know F_i in 5.0.1, since given clock output (and/or clock state update) functions as explicit compositions of arithmetical and bitwise logical operators, 'normally' it is infeasible to express these functions in the Boolean form 2.2: Corresponding ANF's 'as a rule' are sums of exponential in n number of monomials, cf. (5.0.2). Moreover, if these clock output functions F_i and/or clock state update functions f_i are determined by a key-dependent control sequence (say, which is produced by a generator with unknown initial state), see Section 3, then the explicit forms of the mentioned compositions are also unknown. So in general an attacker has to find an initial state u_0 having only a segment $z_j, z_{j+1}, \dots$ of the output sequence formed according to the rule (2.0.1), where both f_i and F_i are not known to him. An 'algebraic' way to do this by guessing f_i and F_i and solving corresponding systems of equations seems to be hopeless in view of 2.9 and the above discussion. The results of preceding sections[14] give us reasons to conjecture that under common tests the sequence $z_j, z_{j+1}, \dots$ behaves like a random one, so 'statistical' methods of breaking such (reasonably designed) ciphers seem to be ineffective as well.

---

[13] which is pseudorandom even if $F = F_0 = F_1 = \dots$, under additional conjecture (how plausible is it?) that the function F constructed above is a pseudorandom function

[14] as well as computer experiments: Output sequences of explicit generators of the kind considered in the paper passed both DIEHARD and NIST test suites

References

[1] R. C. Alperin. p-adic binomial coefficients mod$p$. *The Amer. Math. Month.*, 92(8):576–578, 1985. 17

[2] V. Anashin, A. Bogdanov, and I. Kizhvatov. ABC: A New Fast Flexible Stream Cipher, Version 2. Available from http://crypto.rsuh.ru/papers/abc-spec-v2.pdf, 2005. 2

[3] V. S. Anashin. Uniformly distributed sequences of p-adic integers. *Mathematical Notes*, 55(2):109–133, 1994. 1, 2, 5, 6, 8, 9

[4] V. S. Anashin. Uniformly distributed sequences in computer algebra, or how to constuct program generators of random numbers. *J. Math. Sci.*, 89(4):1355–1390, 1998. 1, 2

[5] V. S. Anashin. Uniformly distributed sequences of p-adic integers, II. *Discrete Math. Appl.*, 12(6):527–590, 2002. A preprint available from http://arXiv.org/math.NT/0209407. 1, 2, 5, 6, 7, 8, 9

[6] V. S. Anashin. On finite pseudorandom sequences. In *Kolmogorov and contemporary mathematics.*, pages 382–383, Moscow, June 2003. Russian Academy of Sciences, Moscow State University. Abstracts of the Int'l Conference. 1, 2

[7] V. S. Anashin. Pseudorandom number generation by p-adic ergodic transformations. Available from http://arxiv.org/abs/cs.CR/0401030, January 2004. 1, 2, 8, 9

[8] V. S. Anashin. Pseudorandom number generation by p-adic ergodic transformations: An addendum. Available from http://arxiv.org/abs/cs.CR/0402060, February 2004. 2, 9

[9] Vladimir Anashin. Uniformly distributed sequences over p-adic integers. In I. Shparlinsky A. J. van der Poorten and H. G. Zimmer, editors, *Number theoretic and algebraic methods in computer science. Proceedings of the Int'l Conference (Moscow, June–July, 1993)*, pages 1–18. World Scientific, 1995. 1, 2, 8, 9

[10] Vladimir Anashin, Andrey Bogdanov, and Ilya Kizhvatov. Increasing the ABC Stream Cipher Period. Technical report, ECRYPT, July 2005. http://www.ecrypt.eu.org/stream/papersdir/050.pdf. 2

[11] M. Bardet, J.-C. Faugère, and B. Salvy. Complexity of Gröbner basis computation for semi-regular overdetermined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$. Available from http://www.inria.fr/rrrt/rr-5049.html, 2004. 16

[12] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Eurocrypt 2000*, volume 1807 of *Lect. Notes Comp. Sci.*, pages 392–407. Springer-Verlag, 2000. 16

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman and Co., 1979. 16

[14] A. Klapper and M. Goresky. Feedback shift registers, 2-adic span, and combiners with memory. *J. Cryptology*, 10:111–147, 1997. 15

[15] A. Klimov and A.Shamir. New cryptographic primitives based on multiword T-functions. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption: 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004. Revised Papers*, pages 1 – 15. Springer-Verlag GmbH, 2004. 2, 6, 8, 9

[16] A. Klimov and A. Shamir. Cryptographic applications of T-functions. In *Selected Areas in Cryptography -2003*, 2003. 2, 6, 8, 12

[17] A. Klimov and A. Shamir. A new class of invertible mappings. In B.S.Kaliski Jr.et al., editor, *Cryptographic Hardware and Embedded Systems 2002*, volume 2523 of *Lect. Notes in Comp. Sci*, pages 470–483. Springer-Verlag, 2003. 1, 2, 6, 7, 8

[18] D. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley, Third edition, 1998. 2, 13, 14

[19] L. Kotomina. Fast nonlinear congruential generators. Diploma Thesis, Russian State University for the Humanities, Moscow, 1999. (in Russian). 2, 8

[20] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences*. John Wiley & Sons, N.Y. etc., 1974. 7, 14

[21] M. V. Larin. Transitive polynomial transformations of residue class rings. *Discrete Mathematics and Applications*, 12(2):141–154, 2002. 8, 9

[22] Hans Lausch and Wilfried Nöbauer. *Algebra of Polynomials*. North-Holl. Publ. Co, American Elsevier Publ. Co, 1973. 1

[23] A. Shamir and B. Tsaban. Guaranteeing the diversity of number generators. *Information and Computation*, 171:350–363, 2001. Available from http://arXiv.org/abs/cs.CR/0112014. 4

[24] S. V. Yablonsky. Basic notions of cybernetics. In *Problems of Cybernetics*. Fizmatgiz, 1959. (in Russian). 1

Faculty of Information Security, Russian State University for the Humanities,, Kirovogradskaya Str., 25/2, Moscow 113534, Russia
E-mail address: anashin@rsuh.ru, vladimir@anashin.msk.su, vs-anashin@yandex.ru