On a bias of Rabbit

Jean-Philippe Aumasson⋆

FHNW, 5210 Windisch, Switzerland

Abstract. This paper demonstrates the existence of a non-null bias in the keystream generated by the synchronous stream cipher Rabbit, a candidate to the eSTREAM project, from the observation that the core function is strongly unbalanced. The keystream bias is greater than $2^{-124.5}$ for certain bits, and this leads to a distinguisher requiring about 2^{247} 128-bit samples of keystream derived from random keys and random IV's, which remains much higher than the cost of exhaustive key search.

Rabbit is a candidate [10] to the eSTREAM project [1], first presented at FSE 2003 [11], targeting both hardware and software environments, having 128-bit key, 64-bit IV, and 513-bit internal state. Its authors made the effort of a deep security analysis – in the reference article [10] and a series of white papers [4, 8, 6, 7, 5, 3], they presented convincing arguments in favor of their cipher's resistance to the well-known algebraic, correlation, differential, guess-and-determine, and statistical attacks – and until now no other publication tackled Rabbit. In this paper, after a brief presentation of the cipher in Section 1, we state in Section 2 some properties of the function $G_n : x \rightarrow x^2 \oplus (x^2 \gg n) \mod 2^n$, which is essential in the keystream generation algorithm. In particular, we observe that the images by G_{32}, as bit words, have in average significantly less zeros than ones at each offset. From this property, we give a method to compute the bias of the keystream bits in Section 3, and illustrate it with the example of the least significant bit of certain keystream subblocks, which is null with probability greater than $\frac{1}{2} + 2^{-124.5}$. This leads to a distinguisher requiring about 2^{247} 128-bit samples of keystream derived from random keys and random IV's.

1 Preliminaries

1.1 Notations

The following symbols are used in the paper.

- $\oplus$: XOR operator, either between single bits or bit words.
- $\ll, \gg$: bitwise shift operators.
- $\lll, \ggg$: bitwise rotation operators.
- ||: bitwise concatenation operator.
- $|x|$: length of x in bits.
- base-16 notation, in sans serif font ($\mathsf{7B} = 123$).
- base-2 notation, in italic font ($1111011 = 123$).

The *bias* of some binary random variable X is defined as the value $\frac{1}{2} - P(x = 1) = P(X = 0) - \frac{1}{2}$, and so its expectation is simply $P(X = 1)$. Thus if X_1 and X_2 have bias 2^{-a} and 2^{-b}, then the random variable $X_1 \oplus X_2$ has bias 2^{-a-b+1}. Except special mention, "random" stands for "uniform random", in the appropriate sample space.

1.2 Description of Rabbit

Rabbit is a synchronous stream cipher, and both encryption and decryption are performed by xoring the argument (plaintext or ciphertext) with the keystream derived from the 128-bit secret key and the 64-bit IV (public). The internal state at epoch $t \geq 0$ is composed of the sequences of 32-bit values $\{x_{j,t}\}_{0 \leq j \leq 7}$ and $\{c_{j,t}\}_{0 \leq j \leq 7}$, plus the bit $\phi_{7,t}$. The state at $t = 0$ is the initial state, obtained after key & IV setup, and the state at epoch $t \geq 1$ is recursively defined by the equations given below.

$$c_{0,t} = c_{0,t-1} + a_0 + \phi_{7,t-1} \mod 2^{32},$$
$$c_{j,t} = c_{j,t-1} + a_j + \phi_{j-1,t} \mod 2^{32}, \ 0 < j < 8,$$

where

$$\phi_{j,t+1} = \begin{cases} 1 - \mathbf{1}_{\mathbb{Z}/2^{32}\mathbb{Z}}(c_{0,t-1} + a_0 + \phi_{7,t-1}) \text{ if } j = 0 \\ 1 - \mathbf{1}_{\mathbb{Z}/2^{32}\mathbb{Z}}(c_{j,t-1} + a_j + \phi_{j-1,t}) \text{ if } j > 0 \end{cases},$$

and the a_j's are constants: $a_0 = a_3 = a_6 = \mathsf{4D34D34D}$, $a_1 = a_4 = a_7 = \mathsf{D34D34D3}$, $a_2 = a_5 = \mathsf{34D34D34D}$. (and $\mathbf{1}_{\mathbb{Z}/2^{32}\mathbb{Z}} : \mathbb{N} \to \{0,1\}$ is the indicator function of the set $\mathbb{Z}/2^{32}\mathbb{Z}$).

$$x_{0,t+1} = g_{0,t} + (g_{7,t} \lll 16) + (g_{6,t} \lll 16)$$
$$x_{1,t+1} = g_{1,t} + (g_{0,t} \lll 8) + g_{7,t}$$
$$x_{2,t+1} = g_{2,t} + (g_{1,t} \lll 16) + (g_{0,t} \lll 16)$$
$$x_{3,t+1} = g_{3,t} + (g_{2,t} \lll 8) + g_{1,t}$$
$$x_{4,t+1} = g_{4,t} + (g_{3,t} \lll 16) + (g_{2,t} \lll 16)$$
$$x_{5,t+1} = g_{5,t} + (g_{4,t} \lll 8) + g_{3,t}$$
$$x_{6,t+1} = g_{6,t} + (g_{5,t} \lll 16) + (g_{4,t} \lll 16)$$
$$x_{7,t+1} = g_{7,t} + (g_{6,t} \lll 8) + g_{5,t},$$

with

$$g_{j,t} = (x_{j,t} + c_{j,t+1})^2 \oplus [(x_{j,t} + c_{j,t+1})^2 \gg 32] \mod 2^{32}.$$

The function g is thus essential for the security of Rabbit. (In the above equation, additions are computed modulo 2^{32}, and squarings modulo 2^{64}, otherwise the second operand of the $\oplus$ would be null.)

At round $t \geq 0$, a 128-bit block of keystream s_t is extracted as follows.

$$s_t^{[15...0]} = x_{0,t}^{[15...0]} \oplus x_{5,t}^{[31...16]} \qquad s_t^{[31...16]} = x_{0,t}^{[31...16]} \oplus x_{3,t}^{[15...0]}$$
$$s_t^{[47...32]} = x_{2,t}^{[15...0]} \oplus x_{7,t}^{[31...16]} \qquad s_t^{[63...48]} = x_{2,t}^{[31...16]} \oplus x_{5,t}^{[15...0]}$$
$$s_t^{[79...64]} = x_{4,t}^{[15...0]} \oplus x_{1,t}^{[31...16]} \qquad s_t^{[95...80]} = x_{4,t}^{[31...16]} \oplus x_{7,t}^{[15...0]}$$
$$s_t^{[111...96]} = x_{6,t}^{[15...0]} \oplus x_{3,t}^{[31...16]} \qquad s_t^{[127...112]} = x_{6,t}^{[31...16]} \oplus x_{1,t}^{[15...0]}$$

There, $s_t^{[a...b]}$, $a \geq b$, denotes the $(a - b + 1)$-bit subblock going from the b-th to the a-th bit (right-to-left).

We do not recall the key & IV setup procedure, since our further developments does not relate to this initialisation process, but instead considers a random initial state.

2 The function g

The function g aims at destroying linear relations in the cipher, *"following initial ideas from chaos theory"* [10]. Indeed the modulo square is known to be a strongly non-linear operation, and the authors of Rabbit were particularly attentive to its security properties: they studied its algebraic degree, differential properties, second degree approximations [7], diffusion properties [3], and the correlation between Hamming weights of the argument and its image [5], and found no notable weakness. One clear drawback of this function is the slowness in hardware of the 32-bit integer squaring (8 squarings are necessary for 128 bits of keystream), however Rabbit is fast in software environments (see benchmarks [9]).

2.1 Basic properties

Consider the function $G_n : \{0,1\}^n \to \{0,1\}^n$,

$$G_n : x \to x^2 \oplus (x^2 \gg n) \mod 2^n,$$

for some integer $n \geq 2$. It is clear that properties holding for a random argument of G_{32} would also hold for random arguments of g and a random internal state. Below several properties of G_n are presented, where each one is followed by a brief justification.

Property 1 *For all even $n \geq 2$, $G_n(1) = G_n(2^{n/2}) = 1$, so G_n is not bijective for even n's.*

Indeed we have $(2^{n/2})^2 = 2^n \equiv 0 \mod 2^n$, thus $G_n(2^{n/2}) = 2^n \gg n$, which is equal to one since the only non-null bit in position $n+1$ is shifted back to the least significant offset. To illustrate this, let $n = 4$ (so $2^{n/2} = 4 = 0100$):

$$
\begin{aligned}
G_4(4) = 0100^2 \oplus (0100^2 \gg 4) &= 10000 \oplus (10000 \gg 4) \\
&= 10000 \oplus 1 \\
&\equiv 1 \mod 2^4.
\end{aligned}
$$

Property 2 *For all even $n \geq 2$, if $x < 2^{n/2}$, then $G_n(x) = x^2$.*

Indeed, $x < 2^{n/2} \Rightarrow x^2 < 2^n$, and so the non-null bits are all deleted by the rotation.

Property 3 *For all $n \geq 2$, if the $2n$-bit words $x = x_1 || x_2$ and $y = y_1 || y_2$ are perfect squares, such that $|x_i| = |y_i| = n$ and $x_1 \oplus x_2 = y_1 \oplus y_2$, then $G_n(\sqrt{x}) = G_n(\sqrt{y})$.*

This simply comes from the fact that $G_n(\sqrt{x_1 || x_2}) = x_1 \oplus x_2$.

Property 4 *For all $n \geq 2$, $G_n(0) = 0$ and $G_n(2^n - 1) = 2^n - 1$.*

The first statement is trivial, and the second comes from the following equality (proven in Appendix B):

$$(2^n - 1)^2 = \left(\sum_{k=0}^{n-1} 2^k \right)^2 = 1 + \sum_{k=n+1}^{2n-1} 2^k, \text{ for } n \geq 2.$$

Indeed, the property is easily proved using this result:

$$(2^n - 1)^2 \oplus ((2^n - 1)^2 \gg n) = \left(1 + \sum_{k=n+1}^{2n-1} 2^k\right) \oplus \sum_{k=1}^{n-1} 2^k$$

$$\equiv \sum_{k=0}^{n-1} 2^k \mod 2^n$$

$$= 2^n - 1.$$

Informally, the number $2^n - 1$ only has non-null bits in the n first offsets, and the squaring shifts all those bits but the least significant of n positions, replacing them by zeros. For instance, consider $n = 4$ (so $2^n - 1 = 15 = \textit{1111}$):

$$G_4(15) = \textit{1111}^2 \oplus (\textit{1111}^2 \gg 4) = \textit{11100001} \oplus (\textit{11100001} \gg 4)$$

$$= \textit{11100001} \oplus \textit{1110}$$

$$\equiv \textit{1111} \mod 2^4.$$

Property 5 *For all even* $n \geq 4$*, if* $x = \sum_{k=n/2-1}^{n-2} 2^k$*, then* $G_n(x) = x$.

This follows from the equality (proven in Appendix B)

$$\left(\sum_{k=\frac{n}{2}-1}^{n-2} 2^k\right)^2 = 2^{n-2} + \sum_{k=\frac{3}{2}n-1}^{2n-3} 2^k, \text{ for } 4 \leq n \text{ even.}$$

To see Property 5, set $x = \sum_{k=n/2-1}^{n-2} 2^k$, and compute

$$x^2 \oplus (x^2 \gg n) = \left(2^{n-2} + \sum_{k=\frac{3}{2}n-1}^{2n-3} 2^k\right) \oplus \left(\sum_{\frac{n}{2}-1}^{n-3} 2^k\right)$$

$$\equiv 2^{n-2} \oplus \left(\sum_{\frac{n}{2}-1}^{n-3} 2^k\right) \mod 2^n$$

$$\equiv \sum_{k=n/2-1}^{n-2} 2^k \mod 2^n$$

$$= x.$$

For instance, for $n = 8$, $\sum_{k=3}^{6} 2^k = 120 = \textit{1111000}$, and so

$$G_8(120) = \textit{1111000}^2 \oplus (\textit{1111000}^2 \gg 8) = \textit{11100001000000} \oplus \textit{111000}$$

$$\equiv \textit{1111000} \mod 2^8.$$

Property 6 *For all* $n \geq 2$*, the number* $2^n + 1$ *is square-free if and only if any non-null* x *verifies* $G_n(x) \neq 0$.

(Recall that a number is square-free if and only if can be decomposed as a product of distinct prime numbers.) To see this property, let's prove separately the two implications. First, we demonstrate the contraposed of the proposition "if any non-null x verifies $G_n(x) \neq 0$, then $2^n + 1$ is square-free", which is "if $2^n + 1$ is not square-free, then there exists a non-null x such that $G_n(x) = 0$": let x be an argument of G_n, and denote $x^2 = y$. If $G_n(x) = 0$, y can be written $y = y_1 || y_1$, with $|y_1| = n$ (n is not necessarily even here), so we have $y = y_1 + 2^n y_1 = y_1 \cdot (2^n + 1)$. Consider the equation

$$y_1 \cdot (2^n + 1) = x^2,$$

with unknowns x and y. We assumed that $2^n + 1$ was not square-free, so there exists an integer $d \geq 2$ such that d^2 divides $2^n + 1$, thus a valid non-null solution is $y_1 = (2^n + 1)/d^2 < 2^n$. For instance, the perfect square $2^3 + 1$ leads to the solution $(x, y_1) = (3, 1)$, and so

$$\begin{aligned} G_3(3) &= 011^2 \oplus (011^2 \gg 3) \\ &= 001001 \oplus (001001 \gg 3) \\ &\equiv 0 \mod 2^3. \end{aligned}$$

(Another solution here is $(6, 4)$). So we proved the existence of a non-null x mapping to zero. To prove the converse implication, assume that $2^n + 1$ is square-free: refering to our previous equation, y_1 must be of the form $d^2 \cdot (2^n + 1) > 2^n$, but we need $y_1 < 2^n$, thus the only solution is 0 in that case.

It may be interesting to search a criterion of n for $2^n + 1$ to have square factors, but this comes out of the scope of this paper. In the specific case of interest for Rabbit, we have $2^{32} + 1 = 641 \times 6700417$, both those divisors being primes, thus $G_{32}(x) = 0$ only if $x = 0$.

2.2 Distribution of single bits

Table 1 and Table 2 give the distribution of the bits in a random image of the reduced versions G_8 and G_{16}. Our notations deserve a few explanations: we present the binary logarithm of the bias instead of the bias itself (biases are all strictly positive here, so their logarithm is well-defined); the integer value Δ is the difference between the quantity of zeros of a uniform distribution (2^{n-1} here) and the quantity observed in the distribution of the function G_n.

Table 1. Distribution of the bits in $G_8(x)$.

offset	0	1	2	3	4	5	6	7
$\log_2$ bias	-5.00	-5.00	-5.19	-5.00	-4.68	-5.19	-5.00	-5.00
Δ	8	8	7	8	10	7	8	8

It appears that G_8 and G_{16} are unbalanced; more zeros than ones appear at each offset. This property also holds for the full version G_{32}; experimental results, presented in Table 3, show that

$$-17.35 < \log_2 \text{bias} < -16.40,$$

with the highest bias reached in position 4, and the smallest in position 13.

Table 2. Distribution of the bits in $G_{16}(x)$.

offset	0	1	2	3	4	5	6	7
$\log_2$ bias	-9.00	-9.00	-8.80	-8.89	-9.00	-8.46	-9.45	-8.92
Δ	128	128	147	138	128	186	94	135

offset	8	9	10	11	12	13	14	15
$\log_2$ bias	-8.74	-8.89	-8.98	-8.81	-8.82	-9.61	-8.88	-9.01
Δ	153	138	130	146	145	84	139	127

Table 3. Distribution of the bits in $G_{32}(x)$.

offset	0	1	2	3	4	5	6	7
$\log_2$ bias	-17.00	-17.00	-16.92	-17.12	-16.40	-17.03	-16.76	-17.11

offset	8	9	10	11	12	13	14	15
$\log_2$ bias	-16.49	-17.33	-16.89	-16.99	-16.95	-17.34	-16.81	-17.03

offset	16	17	18	19	20	21	22	23
$\log_2$ bias	-16.42	-17.06	-16.73	-16.87	-16.71	-17.02	-16.83	-17.04

offset	24	25	26	27	28	29	30	31
$\log_2$ bias	-16.77	-17.21	-16.88	-17.10	-16.55	-17.26	-16.67	-17.03

2.3 Distribution of n-bit patterns

This section aims at giving more precise results on the distribution of g, which cannot be obtained by the previous distributions of the bits, because of the non-independence of the variables. As expected from the previous results, the computation of n-bit patterns for $n = 2, \ldots, 4$ revealed strong deviation from the uniform distribution.

Here we just consider the case $n = 2$: Table 4 below presents the distribution of digrams in the least significant bits (LSB's). We give the logarithm of the *absolute value* of the biases, and add the superscript $\star$ for negative biases; Δ is again the difference between the quantity of a uniform distribution and the quantity observed, for each pattern.

Table 4. Distribution of the digrams in the LSB's of $G_{32}(x)$.

pattern	*00*	*01*	*10*	*11*
$\log_2$ |bias|	-17.00*	0	0	-17.00
Δ	-32 767	0	0	32 769

By computing the distribution of digrams in the 16-th and 17-th, and in the 24-th and 25th bits (see tables 7 and 8 in Appendix A), we can deduce the distribution in the $x_{k,t}$'s, presented in Table 5. Since the spread of the carry bit dramatically scrambles the patterns, we get no significant bias in the digram distribution of $x_{5,t}^{[31\ldots16]}$, but if the keystream subblocks were computed as $s_t^{[15\ldots0]} = x_{0,t}^{[15\ldots0]} \oplus x_{5,t}^{[15\ldots0]}$, then the 2-bit patterns in the LSB's would have biases respectively $2^{-40.99}$, $-2^{-40.99}$, $2^{-43.89}$, and $-2^{-43.93}$.

As predicted, the bits in the images by G_{32} do not exactly behave like independent random variables: indeed, the pattern *000* occurs with bias $2^{-17.62}$, whereas it should be $2^{-17.39}$ if the variables were independent. Table 9 in Appendix A gives the distribution of the 3-bit patterns, which is also statistically close the the distribution obtained under the independance hypothesis.

Table 5. Distribution of the digrams in $x_{k,t}$, $0 \leq k < 8$.

pattern	*00*	*01*	*10*	*11*
k even	-22.28*	-19.46	-20.45*	-20.96*
k odd	-21.72	-21.72*	-31.42*	-31.42*

3 Bias in the keystream

In this section we give a method to compute the bias of each bit in a keystream block s_t, and illustrate it with the computation of the bias of $s_t^{[0]}$, the first bit of s_t. Recall that

$$s_t^{[0]} = x_{0,t}^{[0]} \oplus x_{5,t}^{[16]},$$

so we first compute the bias of $x_{0,t}^{[0]}$ and $x_{5,t}^{[16]}$, under the assumption that the arguments of g are random. We denote hereafter p_i the probability that the i-th bit of $G_{32}(x)$ is 1, and set $q_i = 1 - p_i$, $0 \leq i < 32$ (refer to Table 3 for the values of the p_i's).

It is clear that $x_{0,t}$ is odd (first bit non-null) if and only if either $g_{0,t}$ and $(g_{7,t} \lll 16)$ and $(g_{6,t} \lll 16)$ are odd, or if exactly one of them is odd. Those three random variables $g_{0,t}$, $g_{6,t}$, and $g_{7,t}$, are assumed independent, therefore $x_{0,t}^{[0]}$ is non-null with probability

$$\varrho_0(x_{0,t}) = p_0 \cdot p_{16}^2 + p_0 \cdot q_{16}^2 + 2 \cdot q_0 \cdot p_{16} \cdot q_{16} \leq \frac{1}{2} - 2^{-47.85}.$$

Since we have to take into account the carries of the sum, computing the bias of the 17-th bit of $x_{5,t}$ is less simple. Let p' and q' be the distribution vectors (for ones and zeros) of the 32-bit word $(g_{3,t} + g_{5,t})$. Then the i-th bit of $(g_{4,t} \lll 8) + (g_{3,t} + g_{5,t}) = x_{5,t}$ is non-null with probability

$$\begin{aligned}
\varrho_i(x_{5,t}) &= R_{i-1}(p_j \cdot p_i' + q_j \cdot q_i') + (1 - R_{i-1})(p_j \cdot q_i' + q_j \cdot p_i') \\
&= (2 \cdot p_j \cdot p_i' - p_j - p_i')(2 \cdot R_{i-1} - 1) + R_{i-1},
\end{aligned}$$

where $j = i - 8 \mod 32$, and R_i is the probability that the i-th offset returns a non-null carry bit, expressed by the recurrence

$$\begin{aligned}
R_0 &= p_0 \cdot p_0' \\
R_i &= R_{i-1}\left(p_j \cdot q_i' + q_j \cdot p_i' + p_j \cdot p_i'\right) + (1 - R_{i-1})\left(p_j \cdot p'i\right) \\
&= R_{i-1}(p_j + p_i' - 2 \cdot p_j \cdot p_i') + p_j \cdot p_i'.
\end{aligned}$$

Here we suppose implicitly that the probability to return a carry bit at a given offset is independent from the bit distribution after this position, whereas it is not; we will assume that this approximation does not significantly weaken the results.

The bias of $s_t^{[0]}$ can thus be expressed using the above equations: first compute the distribution of the first 17 bits of $(g_{3,t} + g_{5,t})$, then combine it with the distribution of $(g_{4,t} \lll 8)$ to get the carries distribution, and finally compute from this the bias of the 17-th bit of $x_{5,t}$.

Using the multi-precision library GMP [2], to compute the $\varrho_i(x_{5,t})$'s from the results in Table 3, we find that $x_{5,t}^{[16]}$ has bias $\geq 2^{-76.73}$, and so

$$P(s_t^{[0]} = 1) \leq \frac{1}{2} - 2^{-124.50}.$$

Table 6. Bias in $x_{2k+1,t}$, $0 \leq k < 4$.

offset	0	1	2	3	4	5	6	7
$\log_2$ bias	-48.77	-63.27	-52.72	-55.34	-59.92	-59.31	-60.17	-63.24

offset	8	9	10	11	12	13	14	15
$\log_2$ bias	-63.96	-67.61	-68.64	-70.96	-72.04	-75.17	-75.47	-77.50

offset	16	17	18	19	20	21	22	23
$\log_2$ bias	-76.73	-79.07	-78.72	-79.21	-79.03	-80.04	-79.45	-79.90

offset	24	25	26	27	28	29	30	31
$\log_2$ bias	-78.90	-80.00	-79.36	-79.82	-87.84	-81.24	-79.61	-79.99

By referring to the description of the cipher in Section 1.2, the same calculus can be done for all the other subblocks. One would get exactly the same distribution as for $s_t^{[15...0]}$ for the subblocks $s_t^{[47...32]}$, $s_t^{[79...64]}$, and $s_t^{[111...96]}$ (the distribution of the other subblocks is very close to this), that is,

$$P(s_t^{[k]} = 1) \leq \frac{1}{2} - 2^{-124.50}, \text{ for } k \equiv 0 \mod 16.$$

Indeed, $x_{i,t}$ and $x_{j,t}$ have the same distribution if and only if i and j have the same parity, for all $t \geq 0$. We give in Table 6 this distribution for odd subscripts (which is close to the distribution for even ones); from these results, it is clear that the bias in the LSB of a subblock is much greater than at the other offsets (the bias in the second bit is about 2^{-141}). Therefore, the distinguisher only based on those bits is close to the optimal one: it would require about 2^{247} keystream blocks (see Theorem 2 in [12]), produced by random keys with random IV's, which is much greater than the cost of exhaustive search.

Eventually, our results are clearly not a threat for Rabbit, but stress the non-uniformity of the bits' distribution given a random initial state, under certain assumptions.

Acknowledgements

References

1. eSTREAM, the ECRYPT Stream Cipher Project. Official site: http://www.ecrypt.eu.org/stream/.
2. GNU multiple precision arithmetic library 4.2.1, 2006. Available at http://www.swox.com/gmp/.
3. Cryptico A/S. Algebraic analysis of Rabbit, 2003. White paper.
4. Cryptico A/S. Analysis of the key setup function in Rabbit, 2003. White paper.
5. Cryptico A/S. Hamming weights of the g-function, 2003. White paper.
6. Cryptico A/S. Periodic properties of Rabbit, 2003. White paper.
7. Cryptico A/S. Second degree approximations of the g-function, 2003. White paper.
8. Cryptico A/S. Security analysis of the iv-setup for Rabbit, 2003. White paper.
9. Daniel J. Bernstein. Notes on the ECRYPT stream cipher project (eSTREAM). Timings available at http://cr.yp.to/streamciphers/#timings.
10. Martin Boesgaard, Mette Vesterager, Thomas Christensen, and Erik Zenner. The stream cipher Rabbit. eSTREAM [1] Report 2005/024, 2005. Previously published in [11].
11. Martin Boesgaard, Mette Vesterager, Thomas Pedersen, Jesper Christiansen, and Ove Scavenius. Rabbit: A new high-performance stream cipher. In Thomas Johansson, editor, *FSE'03*, volume 2887 of *Lecture Notes in Computer Science*, pages 307–329. Springer, 2003.
12. Itsik Mantin and Adi Shamir. A practical attack on broadcast RC4. In Mitsuru Matsui, editor, *FSE'01*, volume 2355 of *Lecture Notes in Computer Science*, pages 152–164. Springer, 2001.

A

Table 7. Digrams distribution in the 16-th and 17-th bits of $G_{32}(x)$.

pattern	*00*	*01*	*10*	*11*
$\log_2 |\text{bias}|$	-16.50*	-18.15	-20.72*	-16.95
Δ	-46 258	14 780	-2 480	33 960

Table 8. Digrams distribution in the 24-th and 25-th bits of $G_{32}(x)$.

pattern	*00*	*01*	*10*	*11*
$\log_2 |\text{bias}|$	-16.73*	-18.55	-21.74	-17.27
Δ	-39 604	11 185	1 223	27 198

Table 9. Trigrams distribution in the LSB's of $G_{32}(x)$.

pattern	*000*	*001*	*010*	*011*	*100*	*101*	*110*	*111*
$\log_2 |\text{bias}|$	-17.62*	-17.61*	-20.78	-19.50	-18.51*	-17.61	-20.78*	-17.28
Δ	-21 273	-21 481	2 386	5 810	-11 494	21 481	-2 386	26 959

B

Proposition 1. *For all integer $n \geq 2$,*

$$\left(\sum_{k=0}^{n-1} 2^k\right)^2 = 1 + \sum_{k=n+1}^{2n-1} 2^k.$$

Proof. By recurrence; for $n = 2$, we have $(2^0 + 2^1)^2 = 9 = 1 + 2^3$. Now assume that the equality holds for n, and let

$$\Omega = \left(\sum_{k=0}^{n-1} 2^k\right)^2,$$

then the $(n+1)$-th term gives

$$\left(\sum_{k=0}^{n} 2^k\right)^2 = \sum_{k=0}^{n} \left(2^k \sum_{j=0}^{n} 2^j\right)$$

$$= \Omega + \sum_{k=0}^{n-1} 2^{k+n} + \sum_{k=0}^{n} 2^{k+n}$$

$$= \Omega + \sum_{k=n+1}^{2n-1} 2^k + 2^{2n+1}$$

$$= 1 + \sum_{k=n+2}^{2n+1} 2^k.$$

We conclude that the equality holds for all $n \geq 2$. □

Proposition 2. *For all even integer $n \geq 4$,*

$$\left(\sum_{k=\frac{n}{2}-1}^{n-2} 2^k \right)^2 = 2^{n-2} + \sum_{k=\frac{3}{2}n-1}^{2n-3} 2^k.$$

Proof. By recurrence; for $n = 4$, $(2+2^2)^2 = 36 = 2^2 + 2^5$. Now assume that the equality holds for n, and let

$$\Omega = \left(\sum_{k=\frac{n}{2}-1}^{n-2} 2^k \right)^2,$$

then the $(n+2)$-th term gives

$$\left(\sum_{k=\frac{n+2}{2}-1}^{n+2-2} 2^k \right)^2 = \left(\sum_{k=\frac{n}{2}}^{n} 2^k \right)^2$$

$$= \sum_{k=\frac{n}{2}}^{n} \left(2^k \sum_{j=\frac{n}{2}}^{n} 2^j \right)$$

$$= \Omega + (2^{n-1} + 2^n) \left(\sum_{k=\frac{n}{2}}^{n} 2^k + \sum_{k=\frac{n}{2}}^{n-2} 2^k \right) - 2^{\frac{n}{2}-1} \left(\sum_{k=\frac{n}{2}}^{n-2} 2^k + \sum_{k=\frac{n}{2}-1}^{n-2} 2^k \right)$$

$$= \Omega + (2^{n-1} + 2^n) \left(\sum_{k=\frac{n}{2}+1}^{n-2} 2^k + 2^{n+1} \right) - 2^{\frac{n}{2}-1} \left(\sum_{k=\frac{n}{2}+1}^{n-1} 2^k + 2^{\frac{n}{2}-1} \right)$$

$$= \Omega + \sum_{k=\frac{3n}{2}}^{2n-3} 2^k + \sum_{k=\frac{3n}{2}+1}^{2n-2} 2^k + 2^{2n} + 2^{2n+1} - \sum_{k=n}^{\frac{3n}{2}-2} 2^k - 2^{n-2}$$

$$= 2^{n-2} + \left(\sum_{k=\frac{3n}{2}-1}^{2n-3} 2^k \right) + \sum_{k=\frac{3n}{2}}^{2n-3} 2^k + \sum_{k=\frac{3n}{2}+1}^{2n-2} 2^k + 2^{2n} + 2^{2n+1} - \sum_{k=n}^{\frac{3n}{2}-2} 2^k - 2^{n-2}.$$

At this point the negative sum added with the $2^{\frac{3n}{2}-1}$ of the first sum gives 2^n, and the remaining negative term cancels the first term 2^{n-2}, so we get

$$2^n + \sum_{k=\frac{3n}{2}}^{2n-3} 2^k + \sum_{k=\frac{3n}{2}}^{2n-3} 2^k + \sum_{k=\frac{3n}{2}+1}^{2n-2} 2^k + 2^{2n} + 2^{2n+1} = 2^n + \sum_{k=\frac{3n}{2}+2}^{2n+1} 2^k.$$

We conclude that the equality holds for all even $n \geq 4$. □