

Linear Approximating for the Cipher Salsa 20 (II)

Li An-Ping

apli0001@sina.com

Abstract

This is a modification for our previous paper “linear approximating for the cipher Salsa20”. In this paper we will provide new linear analyses for the cipher Salsa20.

1. Preliminary

No long ago, in our paper [2], we shown some basic results about the linear analysis, and as an application, we also give two linear approximating for the cipher Salsa20 [1], a candidate algorithm for *eECRYPT*, unfortunately, the traces of linear approximating is not correct for our carelessness that missed the rotations of the integers in the rowround (columnround) functions. Here we will provide some new linear analyses for the cipher Salsa20. For the convenience, in the beginning we repeat the basic results in the section 2 of our paper [2].

Suppose z is a binary segment of length h , denoted by $z[i]$ the i -th bit, and let

$$s_1(z) = \sum_i z[i], \quad s_0(z) = h - s_1(z), \quad \text{and} \quad d(z) = s_0(z) - s_1(z),$$

that is, $s_0(z)$ and $s_1(z)$ are

the numbers of the bit “0” and bit “1” in z respectively, and $d(z)$ is the bias of them. Let x

and y be two integers of length h bits, denoted by $L(x, y) = (x + y) \oplus (x \oplus y)$, and define

$$D_i = (2^{2h} - 2 \sum_{x,y} L(x, y)[i]) / 2^{2h},$$

$$D = \sum_{x,y} d(L(x, y)) / h \cdot 2^{2h}.$$
(1.1)

We have the following result

Proposition 1

$$D_i = 1/2^i, \quad 0 \leq i < h.$$
(1.2)

$$D = \frac{2}{h} \cdot \left(1 - \frac{1}{2^h} \right).$$
(1.3)

Suppose that z is a integer variable over the domain Ω , denoted by $\delta(z) = \bigoplus_i z[i]$ and define

$$\Delta_z = |\Omega| - 2 \sum_{z \in \Omega} \delta(z) / |\Omega|.$$
(1.4)

We have

Proposition 2

$$\Delta_{L(x,y)} = 1/2^{16}.$$
(1.5)

Suppose that $\{w_i\}_1^s$ are a set of Boolean variables over the domain Ω , let $w = \bigoplus_i w_i$. If the

variables of the set $\{w_i\}_1^s$ are independent of each other, then it has that

$$\Delta_w = \prod_i \Delta_{w_i} \quad (1.6)$$

For generic case that the variables w_i are not independent there are no similar results known, nevertheless, we will give some considerations in statistic means.

Let us consider a special case that each w_i is a Boolean function on the domain Ω , that is,

$w_i = w_i(t), t \in \Omega$. We view the set $\{w_1(t), w_2(t), \dots, w_s(t) \mid t \in \Omega\}$ as the point set randomly taken over the domain Ω^s and let $w^{(1)}(t) = \bigoplus_i w_i(t)$, then for the variable $w^{(1)}(t)$, the bias

$\Delta_{w^{(1)}}$ will be near the expectation, i.e.

$$\Delta_{w^{(1)}} \approx (\Delta_w)^{1/s}, \quad (1.7)$$

in other words, $\Delta_{w^{(1)}}$ will be distributed centralized at $(\Delta_w)^{1/s}$.

Similarly, if each w_i is a Boolean function on the domain Ω^k , $w_i = w_i(t), t \in \Omega^k$, then the bias of the variable $w^{(k)}(t) = \bigoplus_i w_i(t)$ will be distributed centralized at $(\Delta_w)^{k/s}$, simply,

$$\Delta_{w^{(k)}} \approx (\Delta_w)^{k/s}. \quad (1.8)$$

2. Linear approximating for Salsa20

For an integer z , as usually denoted by $R(z, t)$ the operation of rotating z by t bits. In Salsa20, there are defined a function $z = \text{quarterround}(y)$, where $y = (y_0, y_1, y_2, y_3)$ and $z = (z_0, z_1, z_2, z_3)$ are two arrays of four 32-bits integers.

$$\begin{aligned} z_1 &= y_1 \oplus R(y_0 + y_3, 7) \\ z_2 &= y_2 \oplus R(z_1 + y_0, 9) \\ z_3 &= y_3 \oplus R(z_2 + z_1, 13) \\ z_0 &= y_0 \oplus R(z_3 + z_2, 18) \end{aligned} \quad (2.1)$$

Then, it is easy to obtain that for an integer $i, 0 \leq i < 32$,

$$\begin{aligned} z_1[7+i] &= y_1[7+i] \oplus (y_0 \oplus y_3)[i] \oplus L(y_0, y_3)[i], \\ z_2[9+i] &= y_2[9+i] \oplus (z_1 \oplus y_0)[i] \oplus L(z_1, y_0)[i], \\ z_3[13+i] &= y_3[13+i] \oplus (z_2 \oplus z_1)[i] \oplus L(z_2, z_1)[i], \\ z_0[18+i] &= y_0[18+i] \oplus (z_3 \oplus z_2)[i] \oplus L(z_3, z_2)[i]. \end{aligned} \quad (2.2)$$

where the indices in brackets will be modulo 32.

Suppose that z is an integer, and I is a subset of \mathbb{Z}_{32} , the ring of the integers mod 32, denoted by $z[I] = \bigoplus_{i \in I} z[i]$. For an integer k , $0 \leq k < 32$, denote $I^{(k)} = \{i + k \bmod 32 \mid i \in I\}$.

Clearly, the set I is one-to-one correspondent to an integer such that I is the set of bit “1” positions of the integer. So, we also view I as an integer if no confusion. In this case, then it is clear that $I^{(k)} = R(I, k)$. Hence, the equations in (2.2) can be extended to that

$$\begin{aligned} z_1[I^{(7)}] &= y_1[I^{(7)}] \oplus (y_0 \oplus y_3)[I] \oplus L(y_0, y_3)[I], \\ z_2[J^{(9)}] &= y_2[J^{(9)}] \oplus (z_1 \oplus y_0)[J] \oplus L(z_1, y_0)[J], \\ z_3[K^{(13)}] &= y_3[K^{(13)}] \oplus (z_2 \oplus z_1)[K] \oplus L(z_2, z_1)[K], \\ z_0[P^{(18)}] &= y_0[P^{(18)}] \oplus (z_3 \oplus z_2)[P] \oplus L(z_3, z_2)[P]. \end{aligned} \quad (2.3)$$

We have the following

Claim: For Salsa20, a linear approximating can be implemented from any position of the output.

To prove this claim, it suffice to prove that for any four subsets X_1, X_2, X_3, X_0 of \mathbb{Z}_{32} , there are the subsets A_1, A_2, A_3, A_0 , and C_1, C_2, C_3, C_0 such that

$$\begin{aligned} z_1[X_1] \oplus z_2[X_2] \oplus z_3[X_3] \oplus z_0[X_0] &= y_1[A_1] \oplus y_2[A_2] \oplus y_3[A_3] \oplus y_0[A_0] \oplus \\ &L(y_0, y_3)[C_1] \oplus L(z_1, y_0)[C_2] \oplus L(z_2, z_1)[C_3] \oplus L(z_3, z_2)[C_0]. \end{aligned} \quad (2.4)$$

In order to get a solution of the equation (2.4), we XOR the four equations of (2.3), it follows that

$$\begin{aligned} & z_1[K \oplus J \oplus I^{(7)}] \oplus z_2[P \oplus K \oplus J^{(9)}] \oplus z_3[P \oplus K^{(13)}] \oplus z_0[P^{(18)}] \\ &= y_1[I^{(7)}] \oplus y_2[J^{(9)}] \oplus y_3[I \oplus K^{(13)}] \oplus y_0[I \oplus J \oplus P^{(18)}] \oplus \\ &L(y_0, y_3)[I] \oplus L(z_1, y_0)[J] \oplus L(z_2, z_1)[K] \oplus L(z_3, z_2)[P]. \end{aligned} \quad (2.5)$$

From (2.5) we can see that we have merely to solve the following equations system

$$\begin{cases} P^{(18)} = X_0, \\ P \oplus K^{(13)} = X_3, \\ P \oplus K \oplus J^{(9)} = X_2, \\ K \oplus J \oplus I^{(7)} = X_1. \end{cases} \quad (2.6)$$

The solution of the system above is that,

$$\begin{aligned} P &= X_0^{(14)}, \\ K &= X_0^{(1)} \oplus X_3^{(19)}, \\ J &= X_0^{(5)} \oplus X_0^{(24)} \oplus X_3^{(10)} \oplus X_2^{(23)}, \\ I &= X_1^{(25)} \oplus X_0^{(30)} \oplus X_0^{(26)} \oplus X_0^{(17)} \oplus X_2^{(16)} \oplus X_3^{(3)} \oplus X_3^{(12)}, \end{aligned} \quad (2.7)$$

By (2.5), we obtain the solution of (2.4)

$$\begin{aligned}
A_1 &= X_1 \oplus X_0^{(5)} \oplus X_0^{(1)} \oplus X_0^{(24)} \oplus X_2^{(23)} \oplus X_3^{(10)} \oplus X_3^{(19)}, \\
A_2 &= X_0^{(14)} \oplus X_0^{(1)} \oplus X_3^{(19)} \oplus X_2, \\
A_3 &= X_1^{(25)} \oplus X_0^{(30)} \oplus X_0^{(26)} \oplus X_0^{(17)} \oplus X_0^{(14)} \oplus X_3 \oplus X_3^{(3)} \oplus X_3^{(12)} \oplus X_2^{(16)}, \\
A_0 &= X_1^{(25)} \oplus X_0^{(30)} \oplus X_0^{(26)} \oplus X_0^{(17)} \oplus X_0^{(5)} \oplus X_0^{(24)} \oplus X_0 \\
&\quad \oplus X_3^{(3)} \oplus X_3^{(12)} \oplus X_3^{(10)} \oplus X_2^{(16)} \oplus X_2^{(23)}, \\
C_1 &= X_1^{(25)} \oplus X_0^{(30)} \oplus X_0^{(26)} \oplus X_0^{(17)} \oplus X_2^{(16)} \oplus X_3^{(3)} \oplus X_3^{(12)}, \\
C_2 &= X_0^{(5)} \oplus X_0^{(24)} \oplus X_3^{(10)} \oplus X_2^{(23)}, \\
C_3 &= X_0^{(1)} \oplus X_3^{(19)}, \\
C_0 &= X_0^{(14)}
\end{aligned} \tag{2.8}$$

Suppose that $z = \text{salsa20}(x) = x + y$, $y = \text{doubleround}^{10}(x)$, x , y and z are the binary segments of length 512 bits. Then for any a 512-bits segment a , we can find a 512-bits segment b such that

$$a \& y = (b \& x) \oplus \left(\bigoplus_{(u,v)} L(u,v)[C_{u,v}] \right) \tag{2.9}$$

where $C_{u,v}$ are constants determined by a and (u,v) are the pairs involved the additions in the function $\text{doubleround}(x)$. So, we have

$$a \& z = (a \& x) \oplus (b \& x) \oplus L(x,y)[a] \oplus \left(\bigoplus_{(u,v)} L(u,v)[C_{u,v}] \right) \tag{2.10}$$

Denoted by $c = a \oplus b$, and $z_a = a \& z$, $x_c = c \& x$, $\varepsilon = L(x,y)[a] \oplus \left(\bigoplus_{(u,v)} L(u,v)[C_{u,v}] \right)$,

then (2.10) becomes that

$$z_a \oplus x_c = \varepsilon. \tag{2.11}$$

Consequently,

$$\Delta_{z_a \oplus x_c} = \Delta_\varepsilon. \tag{2.12}$$

The right-hand of (2.12) Δ_ε can be estimated with (1.2), (1.7) and (1.8), and so obtain an estimation for the bias $\Delta_{z_a \oplus x_c}$. We have computed the values of bias $(\Delta_w)^{1/s}$ for a number of positions, including every bit of the output of Salsa20, the results all are

$$(\Delta_w)^{1/s} < 1/2^{17}. \tag{2.13}$$

The test results suggest the conjecture that the inequality (2.13) is true for all the positions.

Some of computed results are listed in the following table, where s is the number of non-zero bits in the constants a and $C_{u,v}$, that is, the number of the bits activated in the linear

approximating, and μ is the sum of the positions of these s bits in the 32-bits integers.

a	μ	s	μ/s
2^0	67991	4098	16.59
2^{32}	70279	4186	16.79
2^{64}	70343	4185	16.81
2^{96}	69166	4141	16.70
2^{128}	69166	4141	16.70
2^{160}	67991	4098	16.59
2^{192}	70279	4186	16.79
2^{224}	70343	4185	16.81

Table 1

Now we take a example $a = 1$, through the calculation by formula (2.8), it has

$$\begin{aligned}
 b = & (0x47651b66,0xa4333500,0x89fc980,0xaf47a340, \\
 & 0xc3f7c5e0,0xbab84666,0xbcd93640,0xa8819cc0, \\
 & 0xcab7de40,0x1dc3b140,0x5e110b66,0x861f3640 \\
 & 0x523b7640,0xec1bbec0,0x510da340,0x14c11b66)
 \end{aligned} \tag{2.14}$$

and,

$$\mu = 67991, \quad s=4098, \quad \mu/s=16.6 \tag{2.15}$$

For this example, we have made some tests for $\Delta_{z_a \oplus x_c}$ with 2^{30} keystream in the case $key = 1, 2, \dots, 16$, the most of them near $1/2^{16}$.

According to the analysis above, we have the following conclusions that

Conclusion 1: For a discrete secret key and $a = 1$, the bias $\Delta_{z_a \oplus x_c}$ will be distributed centered at $1/2^{33.2}$.

Conclusion 2: For a discrete secret key, and $a = 2^{k*32}, 0 \leq k < 16$, the bias $\Delta_{z_a \oplus x_c}$ will be distributed centered about $1/2^{34}$.

So, we have

Conclusion 3: Salsa20 is distinguishable from truly random one in a discrete key of keystreams

in most of time.

It maybe should be mentioned that the term x_c in a approximating shown above, it is likely that contains the part of key bits, but they are constant in a discrete secret key and so will not make effect for the size of $\Delta_{z_a \oplus x_c}$.

References

- [1] Daniel Bernstein, Synchronous Stream Cipher Salsa20, Available at <http://www.ecrypt.eu.org/stream/salsa20.html>
- [2] Li An-Ping, Linear approximating for the cipher Salsa20, available at <http://www.ecrypt.eu.org/stream/papersdir/056.pdf>