Discrete Applied Mathematics 208 (2016) 88–97

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Chosen IV cryptanalysis on reduced round ChaCha and Salsa

Subhamoy Maitra

Applied Statistics Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India

ARTICLE INFO

Article history:
Received 13 July 2015
Received in revised form 18 February 2016
Accepted 22 February 2016
Available online 20 April 2016

Keywords:
Stream cipher
ChaCha
Salsa
Non-randomness
Probabilistic Neutral Bit (PNB)
ARX cipher

ABSTRACT

Recently, ChaCha20 (the stream cipher ChaCha with 20 rounds) is in the process of being a standardized and thus it attracts serious interest in cryptanalysis. The most significant effort to analyse Salsa and ChaCha was explained by Aumasson et al. long back (FSE 2008) and further, only minor improvements could be achieved. In this paper, first we revisit the work of Aumasson et al. to provide a clearer insight of the existing attack (2^{248} complexity for ChaCha7, i.e., 7 rounds) and show certain improvements (complexity around 2^{243}) by exploiting additional Probabilistic Neutral Bits. More importantly, we describe a novel idea that explores proper choice of IVs corresponding to the keys, for which the complexity can be improved further (2^{239}). The choice of IVs corresponding to the keys is the prime observation of this work. We systematically show how a single difference propagates after one round and how the differences can be reduced with proper choices of IVs. For Salsa too (Salsa20/8, i.e., 8 rounds), we get improvement in complexity, reducing it to $2^{245.5}$ from $2^{247.2}$ reported by Aumasson et al.

1. Introduction

The Salsa20 [2] stream cipher was designed by Bernstein in 2005 as a candidate for eStream [9] and Salsa20/12 was accepted in the eStream software portfolio. There are several works that studied the cryptanalysis of Salsa [4,5,13,1,6,12,8,7] and these works show weaknesses of this cipher in reduced rounds. The central idea in this field of cryptanalysis is as follows.

- Apply some input difference at the initial state and then investigate for biases at some output.
- Once one can proceed a few rounds forward as above, it may be possible to get back a few rounds from a final state to obtain further non-randomness.

The ChaCha [3] stream cipher was proposed in early 2008 to conjecturally provide better diffusion and cryptanalytic resistance than Salsa. Though ChaCha was designed long back, the cipher got renewed attention in recent time due to its deployment in several applications of Google as evident from the following news report [11]: "Given recent attacks against older, commonly-used encryption modes RC4 and CBC, the Google team began implementing new algorithms — ChaCha 20 for symmetric encryption and Poly1305 for authentication in OpenSSL and NSS in March 2013". The only significant cryptanalysis of reduced round Salsa and ChaCha was presented long back in [1] that introduced Probabilistic Neutral Bits (PNBs) and the works [12,7] achieved only some incremental advancements over [1]. The idea of Column Chaining Distinguisher (CCD) [12] could only provide minor advantage over the complexities described in [1] for both Salsa and ChaCha. In [7], an interesting observation regarding round reversal of Salsa was studied, but no significant cryptanalytic improvement could be obtained using this method. However, the work of [7] revisits the attack of [1] in detail, providing some corrections to the values of the parameters that constitute the attack complexity.

E-mail address: subho@isical.ac.in.

http://dx.doi.org/10.1016/j.dam.2016.02.020
0166-218X/

1.1. Contribution

In this paper, we attempt to obtain significant improvement in the cryptanalysis of ChaCha over the work of [1]. First, in Section 1.3, we put a disciplined effort to study the same method of [1] in greater details. This is in line of [7], where the work of [1] was revisited too in the context of Salsa. We perform detailed experiments and exploit more PNBs than in [1] to obtain better results. In [1], the time complexity of the attack against ChaCha7 (7 round ChaCha) was estimated as 2^{248}, though a careful analysis shows that it is actually slightly better (less), which is $2^{246.71}$. The effort of [12] with the idea of CCD required $2^{246.5}$ complexity and thus one may note that the improvement is indeed minor. In Section 2, with additional PNBs and following the similar method as in [1], we show that the complexity can be reduced to $2^{242.82}$.

Next we show how our idea of exploiting specific IVs corresponding to the secret key helps in improving the attack of [1]. While the cryptanalysis of reduced round Salsa and ChaCha was presented long back [1], this observation of choosing selected IVs could not be discovered earlier. The cryptanalytic technique presented in [1] depends on some biases related to certain output differences in the forward direction given the input difference(s). Our idea improves such biases significantly. Using this, in Section 3, we show that for proper choices of IVs, the time complexity of ChaCha7 cryptanalysis reduces to $2^{238.94}$.

To explain further, we consider the scenario when the number of differences after the quarterround is the same as in the case when the modulo addition $+$ (nonlinear) is replaced by the linear operation $\oplus$. The differential thus passes with probability one. This happens for proper choices of IV words given the key words. One may also refer to these differences as conditional differences.

We apply the similar technique for Salsa in Section 4. For Salsa20/8, the complexity described in [1] was 2^{251}, but a detailed study in [7] shows that it is actually better, i.e., $2^{247.2}$. The idea of [12] using CCD required 2^{250}, and naturally the improvement was not significant. Our technique, considering the properly chosen IVs, improves the complexity to $2^{245.5}$.

Before proceeding further, let us explain the structure of ChaCha stream cipher as provided in the draft towards standardization [10]. We will come back to the description related to Salsa in Section 4.

1.2. Description of ChaCha

The cipher state is of 16 words, each 32-bit and it can be written in 4×4 matrix format as follows:

$$X = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & v_0 & v_1 & v_2 \end{pmatrix}.$$

The rightmost matrix shows the initial state, that takes four predefined constants $c_0, \ldots, c_3$ as $c_0 = \texttt{0x61707865}$, $c_1 = \texttt{0x3320646e}$, $c_2 = \texttt{0x79622d32}$, $c_3 = \texttt{0x6b206574}$, 256-bit key $k_0, \ldots, k_7$, 32-bit block counter t_0 and 96-bit nonce v_0, v_1, v_2.

Primitive nonlinear operation here is the quarterround function. Each quarterround (a, b, c, d) consists of four ARX rounds, each of which comprises of addition (A), cyclic left rotation (R) and XOR (X) operation (one each) as given below.

$$\left.\begin{array}{lll} a = a + b; & d = d \oplus a; & d = d \lll 16; \\ c = c + d; & b = b \oplus c; & b = b \lll 12; \\ a = a + b; & d = d \oplus a; & d = d \lll 8; \\ c = c + d; & b = b \oplus c; & b = b \lll 7; \end{array}\right\}. \tag{1}$$

Each columnround works as four quarterrounds on each of the four columns of the state matrix and each diagonalround consists of four quarterrounds on each of the four diagonals. In ChaCha20, ten times each the rowround and ten times each the diagonalround is applied alternatively on the initial state (total 20 times).

In each of the odd rounds, we first apply quarterround on all the four columns in the following order. This is a complete columnround.

$$\text{quarterround}(x_0, x_4, x_8, x_{12}), \quad \text{quarterround}(x_1, x_5, x_9, x_{13}),$$
$$\text{quarterround}(x_2, x_6, x_{10}, x_{14}), \quad \text{and} \quad \text{quarterround}(x_3, x_7, x_{11}, x_{15}).$$

In each of the even rounds, we consider the order

$$\text{quarterround}(x_0, x_5, x_{10}, x_{15}), \quad \text{quarterround}(x_1, x_6, x_{11}, x_{12}),$$
$$\text{quarterround}(x_2, x_7, x_8, x_{13}), \quad \text{and} \quad \text{quarterround}(x_3, x_4, x_9, x_{14}).$$

This describes a complete diagonalround.

By $X^{(r)}$, we mean that r such rounds are applied (in total, alternatively the columnround in odd rounds and diagonalround in even rounds, the initial round applied is considered as the round 1) on the initial state X. Here $X^{(0)}$ is the initial state X. Finally, after R rounds we have $X^{(R)}$. Then a keystream block of 16 words or 512 bits is obtained as

$$Z = X + X^{(R)}.$$

For ChaCha20, there are 20 rounds, i.e., $R = 20$. It is quite natural that less rounds achieve higher speed and conjecturally, more rounds provide higher security.