Distinguishing Attack on the ABC v.1 and v.2

Shahram Khazaei Mohammad Kiaei

{khazaaei, mohammadkiaei}@yahoo.com
Sharif University of Technology,
Electrical Engineering Department
Tehran, Iran
Sept 2005

Abstract

ABC is a synchronous stream proposed as a candidate to ECRYPT Project which has been withdrawn because of the attacks proposed in [4, 7]. The attacks benefit the non-bijectivity property of one of the ABC components called C which is a parametric function from $GF(2)^{32}$ into $GF(2)^{32}$. The designers of ABC updated it to a new (longer) version called ABC v.2. The focus of those cryptanalyses on ABC, which is called ABC v.1 hereafter, was to recover the initial state of the cipher. Although these attacks are not any longer applicable on ABC v.2, we use the same weakness of C function to mount a distinguishing attack on both versions of ABC with data, time and memory complexities of $O(2^{32})$.
Keywords. ECRYPT Stream Cipher Project, ABC stream cipher, Cryptanalysis, Distinguishing Attack, Security Evaluation.

1 Introduction

ABC v.1[1] [1] is a synchronous stream cipher proposed as a candidate to ECRYPT Stream Cipher Project -a multi-year effort to identify new stream ciphers that might become suitable for widespread adoption [5]. ABC v.1 was optimized for software applications and work with a 128-bit key and a 128-bit IV. It consists of 38, 32-bit registers; three of them denoted by $\mathbf{z}_0$, $\mathbf{z}_1$ and $\mathbf{x}$ are considered as the state of the cipher and the rest 35 registers, denoted by $\mathbf{d}_0$, $\mathbf{d}_1$, $\mathbf{e}$, $\mathbf{e}_0$, $\mathbf{e}_1$, … and $\mathbf{e}_{31}$, are considered as constant parameters fed to the cipher. The values of these 38 registers are determined during a key expansion routine. Although each register is a 32-bit word, the total number of initial bits are 1195 bits and not 1216 bits because of restrictions $\mathbf{z}_0 \equiv 2$ or 3 (mod 4), $\mathbf{e}_{31} \equiv 2^{16} \pmod{2^{17}}$, $\mathbf{d}_0 \equiv 1 \pmod 2$ and $\mathbf{d}_1 \equiv 0 \pmod 4$ which are made during initialization. The main components of ABC are three functions denoted by A, B and C. In [4] and [7] it has been shown that C is far from a random bijective function and more likely behaves as a randomly chosen function. Using this weakness, a correlation-based divide and conquer attack with time complexity of $O(2^{95})$ using $O(2^{32})$ keystream words has been applied by targeting the 2^{63}

[1]The original name is ABC, however it is called ABC v.1 in this paper to avoid uncertainty.

possible initial states of $\mathbf{z}_0$ and $\mathbf{z}_1$ registers. The recovery of the end of the internal state is possible with time complexity of $O(2^{77})$ using the same $O(2^{32})$ keystream words [4]. Some flaws in the IV setup have also been mentioned in [4].

The designers of ABC v.1 proposed a new longer version of it, called ABC v.2 [2, 3]. The updated version uses three more 32-bit registers $\mathbf{d}_2$, $\mathbf{z}_2$ and $\mathbf{z}_3$. The new restrictions on the initial value of the registers are $\mathbf{z}_0 \equiv 2$ or 3 (mod 4), $\mathbf{e}_{31} \equiv 2^{16} \pmod{2^{17}}$, $\mathbf{d}_0 \equiv 1$ (mod 4), $\mathbf{d}_1 \equiv 0$ (mod 4) and $\mathbf{d}_2 \equiv 0$ (mod 4) which increases the initial state size to 1288 bits.

Since the total number of possible initial states of $\mathbf{z}_0$, $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z}_3$ is 2^{127}, the time complexity of the attacks in [4] and [7] is $O(2^{32}2^{127})$ which is more than that of exhaustive key search. However, in this paper we show that a distinguishing attack with data, time and memory complexities of $O(2^{32})$ is applicable to both versions of ABC using the same non-bijectivity weakness of the C function. The goal of distinguishing attack is to distinguish the keystream of the cipher from a truly random sequence with small error probability.

The outline of this paper is as follows. In Sect. 2, a brief description of the ABC v.1 and v.2 is given. The proposed distinguishing attack is described in Sect. 3 and the paper is concluded in Sect. 4.

2 A Brief Description of ABC v.1 and v.2

ABC works with 32-bit integer values. A 32-bit vector (a[31], a[30], ..., a[1], a[0]) is denoted by integer a where $a = \sum_{i=0}^{31} a[i]2^i$. Throughout this paper the symbols $\oplus$, >>, << and >>> are respectively used for 32-bit XOR, right shift, left shift and right rotation, and the symbols + and $\cdot$ are respectively used for addition and multiplication module 2^{32}.

Both versions of ABC consist of three components A, B and C. In the first version, the component A is a linear transformation over $GF(2)^{64}$ defined by

$$\mathrm{A}(z_1, z_0) = (z_1 \oplus (z_1 << 31) \oplus (z_0 >> 1), z_1) \quad (1)$$

and in the second version it is a linear transformation over $GF(2)^{128}$ and defined by

$$\mathrm{A}(z_3, z_2, z_1, z_0) = (z_2 \oplus (z_1 << 31) \oplus (z_0 >> 1), z_3, z_2, z_1) \quad (2)$$

The component B is a single cycle T-function over $GF(2)^{32}$ defined by

$$\mathrm{B}(x) = d_0 + 5 \cdot (x \oplus d_0) \quad (3)$$

in v.1, and by

$$\mathrm{B}(x) = ((x \oplus d_0) + d_1) \oplus d_2 \quad (4)$$

in v.2, where d_0, d_1 and d_2 represent 32-bit key and IV dependent constants.

The component C is a mapping over $GF(2)^{32}$ which is the same for both versions of the ABC and defined by

$$\mathrm{C}(x) = (e + \sum_{i=0}^{31} x[i]e_i) >>> 16 \quad (5)$$

where e and e_i's are key and IV dependent constants determined during the key schedule and $x[i]$ is the i^{th} bit of the word x.

The key stream generation routine of both version of ABC involves the described components and consists ofthe following 3 steps.

ABC v.1 KEY STREAM GENERATOR

INPUT: $(\mathbf{z}_1, \mathbf{z}_0) \in GF(2)^{64}$, $\mathbf{x} \in GF(2)^{32}$

$$(\mathbf{z}_1, \mathbf{z}_0) \leftarrow A(\mathbf{z}_1, \mathbf{z}_0) \quad (6)$$

$$\mathbf{x} \leftarrow \mathbf{z}_1 + B(\mathbf{x}) \quad (7)$$

$$\mathbf{y} \leftarrow \mathbf{z}_0 + C(\mathbf{x}) \quad (8)$$

OUTPUT: $(\mathbf{z}_1, \mathbf{z}_0) \in GF(2)^{64}$, $\mathbf{x} \in GF(2)^{32}$, $\mathbf{y} \in GF(2)^{32}$

ABC v.2 KEY STREAM GENERATOR

INPUT: $(\mathbf{z}_3, \mathbf{z}_2, \mathbf{z}_1, \mathbf{z}_0) \in GF(2)^{128}$, $\mathbf{x} \in GF(2)^{32}$

$$(\mathbf{z}_3, \mathbf{z}_2, \mathbf{z}_1, \mathbf{z}_0) \leftarrow A(\mathbf{z}_3, \mathbf{z}_2, \mathbf{z}_1, \mathbf{z}_0) \quad (9)$$

$$\mathbf{x} \leftarrow \mathbf{z}_3 + B(\mathbf{x}) \quad (10)$$

$$\mathbf{y} \leftarrow \mathbf{z}_0 + C(\mathbf{x}) \quad (11)$$

OUTPUT: $(\mathbf{z}_3, \mathbf{z}_2, \mathbf{z}_1, \mathbf{z}_0) \in GF(2)^{128}$, $\mathbf{x} \in GF(2)^{32}$, $\mathbf{y} \in GF(2)^{32}$

3 Description of the Attack

Since the result of applying our attack to both versions of ABC is exactly the same, it is only considered for ABC v.1. Denote the state sequence of the registers $\mathbf{z}_0$, $\mathbf{x}$ and $\mathbf{y}$ respectively by $\{z_n\}_{n=0}^{\infty}$, $\{x_n\}_{n=0}^{\infty}$ and $\{y_n\}_{n=1}^{\infty}$. Then the key stream generation routine, relations (6)-(8), can be expressed by the following recursive equations

ABC v.1 KEY STREAM GENERATOR

$$z_{n+1} = z_n \oplus (z_n << 31) \oplus (z_{n-1} >> 1) \quad (12)$$

$$x_n = z_{n+1} + B(x_{n-1}) \quad (13)$$

$$y_n = z_n + C(x_n) \quad (14)$$

for $n \geq 1$, where z_0, z_1 and x_0 are respectively the initial values of $\mathbf{z}_0$, $\mathbf{z}_1$ and $\mathbf{x}$ registers in ABC v.1 which are determined by its key schedule.

It has been well discussed in [4] and [7] that the C function, with probability $1 - e^{-1}2^{-32}$, is non-bijective and behaves as a randomly chosen function over $GF(2)^{32}$. The idea of our distinguishing attack is to use the non-uniform distribution of the C function and approximate the sum of two words module 2^{32} with their bitwise XOR. The idea of approximating + with $\oplus$ has already been used in [6] to perform distinguishing attack on

SOBER family stream ciphers. Applying this approximation for (14), introduce an error denoted by w_n. Using the linear recurrence (12) it then follows that

$$\begin{aligned} y_{n+2} \oplus y_{n+1} \oplus (y_{n+1} << 31) \oplus (y_n >> 1) = C_{n+2} \oplus w_{n+2} \oplus C_{n+1} \oplus w_{n+1} \\ \oplus (C_{n+1} << 31) \oplus (w_{n+1} << 31) \\ \oplus (C_n >> 1) \oplus (w_n >> 1) \end{aligned} \quad (15)$$

where $C(x_n)$ is denoted by C_n for simplicity.

The distribution of w_n is quite biased and identical for all n. Although not being necessary, it could be computed in time complexity of $O(2^{32})$ and saved in $O(2^{32})$ memory [8]. The distribution of C_n is also highly biased for a randomly chosen value of e and e_i's [4, 7] and is the same for all n.

In order to distinguish between the following two hypotheses

H_0: $\{y_n\}_{n=1}^N$ is a purely random sequence over $GF(2)^{32}$.

H_1: $\{y_n\}_{n=1}^N$ is the output sequence of ABC v.1.

we construct the sequence

$$u_n \overset{\Delta}{=} y_{n+2} \oplus y_{n+1} \oplus (y_{n+1} << 31) \oplus (y_n >> 1) \quad (16)$$

for $1 \le n \le N-2$.

Under hypothesis H_0, u_n is uniformly distributed while under hypothesis H_1 it is biased (non-uniformly distributed). Since the distribution of u_n depends on the value of e and e_i's, a LRT[2] is not applicable here. Instead, for a given sequence $\{y_n\}_{n=1}^N$, we build the table $\hat{F}$ with entries $\hat{F}[x]$ $(0 \le x \le 2^{32}-1)$, defined as the number of occurrences of word x in the sequence $\{u_n\}_{n=1}^N$. Then we can compute the following ad-hoc statistic

$$\chi^2 = \sum_{x=0}^{2^{32}-1} (\hat{F}[x]-1)^2 \quad (17)$$

The statistic χ^2 is expected to be much larger under hypothesis H_1 corresponding to ABC v.1 generator than under H_0 corresponding to a truly random generator. The decision rule for distinguishing between H_0 and H_1 then can be expressed as

$$\text{the sequence } \{y_n\} \text{ is from} \begin{cases} \text{the ABC v.1 Generator} & \text{if } \chi^2 \ge th \\ \text{a Truely Random Generator} & \text{if } \chi^2 < th \end{cases} \quad (18)$$

In [7] the distribution of $\Delta = 2^{-32} \sum_{x=0}^{2^{32}-1} |\hat{F}[x]-1|$ has been approximated by Normal distribution for $\lambda >> 1$ where $N = \lambda 2^{32}$, and the average and variance of Δ have been theoretically computed using some features of the distribution of u_n. Similarly, it is possible to compute the threshold th which minimizes the error probability of our distinguishing method which is equal to

[2] Likelihood Ratio Test

$$P_e = 1/2\,\Pr\{H_1 \mid H_0\} + 1/2\,\Pr\{H_0 \mid H_1\} \qquad (19)$$

assuming $\Pr\{H_1\} = \Pr\{H_0\} = 1/2$.

Since the results in [7] is valid for $\lambda >> 1$, i.e. $\lambda \geq 10$, we did not follow the proposed method. Instead, we have estimated P_e experimentally over 1000 samples under each one of the hypothesis H_0 and H_1 on reduced versions[3] of the ABC, where 32-bit words of the actual ABC were replaced by m-bit words for $\lambda = 1, 2, 4, 8$ and $m = 8, 10, 12, 14, 16, 18, 20$. The results are given in Tables 1 and 2 respectively for ABC v.1 and v.2. We also estimated P_e for the statistic Δ; the resulting error probability is slightly more that that of the statistic χ^2.

Table 1. Estimated P_e for an m-bit word version of ABC v.1 for $N = \lambda 2^m$ and the statistic χ^2

λ \ m	8	10	12	14	16	18	20
1	0.4605	0.4770	0.4760	0.4935	0.4565	0.4190	0.3180
2	0.3815	0.4125	0.4175	0.4295	0.4140	0.2740	0.0155
4	0.2590	0.2885	0.3185	0.3355	0.3180	0.0370	0.0000
8	0.1305	0.1405	0.1565	0.1750	0.1160	0.0000	0.0000

Table 2. Estimated P_e for an m-bit word version of ABC v.2 for $N = \lambda 2^m$ and the statistic χ^2

λ \ m	8	10	12	14	16	18	20
1	0.4805	0.4745	0.4735	0.4990	0.4465	0.3325	0.0225
2	0.4125	0.4105	0.4245	0.4510	0.3870	0.0330	0.0000
4	0.2875	0.2930	0.2910	0.3445	0.1075	0.0000	0.0000
8	0.1420	0.1515	0.1425	0.1985	0.0075	0.0000	0.0000

Theoretic analysis of error probability is interesting for small values of λ, but it seems to be difficult. Due to accessing to an ordinary processor, we had to confine ourselves to reduced versions of ABC. One with higher computational power can estimate the error probability for the real versions of ABC. However, at least the results of Tables 1 and 2 convince us that the proposed distinguishing attack is applicable on both actual versions of ABC with negligible error probability using about 2^{32} output words.

4 Conclusion

In this paper, using the linear approximation of module some and non-bijectivity weakness of C function in ABC family ciphers, we mount a distinguishing attack on them with data, time and memory complexities of $O(2^m)$ where m is the word size of the cipher (m = 32 for ABC v.1 and v.2). It seems hard to achieve a strong cipher using a randomly chosen s-box.

[3] The recurrence equation of $\{z_n\}$ was considered as $z_{n+1} = z_n \oplus (z_n << (m-1)) \oplus (z_{n-1} >> 1)$ for v.1 and as $z_{n+3} = z_{n+1} \oplus (z_n << (m-1)) \oplus (z_{n-1} >> 1)$ for v.2.

References

1- V. Anashin, A. Bogdanov, I. Kizhvatov and S. Kumar “ABC: A New Fast Flexible Stream Cipher”, ECRYPT Stream Cipher Project Report 2005/001, 2005, available at http://www.ecrypt.eu.org/stream/

2- V. Anashin, A. Bogdanov, I. Kizhvatov, “Increasing the ABC Stream Cipher Period”, ECRYPT Stream Cipher Project Report 2005/050, 2005, available at http://www.ecrypt.eu.org/stream/

3- V. Anashin, A. Bogdanov, I. Kizhvatov and S. Kumar “ABC: A New Fast Flexible Stream Cipher, Specification Version 2”, available at http://crypto.rsuh.ru/papers/

4- C. Berbain, H. Gilbert, “Cryptanalysis of ABC”, ECRYPT Stream Cipher Project Report 2005/048, 2005, available at http://www.ecrypt.eu.org/stream/

5- eSTREAM, the ECRYPT Stream Cipher Project. http://www.ecrypt.eu.org/stream/

6- P. Ekdahl and T. Johansson, “Distinguishing Attacks on Sober-t16 and t32”, Fast Software Encryption 2002, LNCS 2365, J. Daemen, V. Rijmen, Eds., Springer-Verlag, pp. 210-224, 2002.

7- S. Khazaei, “Divide and Conquer Attack on ABC Stream Cipher”, ECRYPT Stream Cipher Project Report 2005/052, 2005, available at http://www.ecrypt.eu.org/stream/

8- A. Maximov, “On Linear Approximation of Modulo Sum”, Fast Software Encryption (FSE) 2004, India, February 2004, pp: 483-484.