Fast Correlation Attack on Stream Cipher ABC v3

Haina Zhang * Lin Li † Xiaoyun Wang ‡

Abstract

ABC v3 is a stream cipher proposed as a candidate to ECRYPT Estream Project which enters the second evaluation phase. Its key length is 128 bits. In this paper, We show that, there are at least $2^{103.71}$ weak keys among 2^{128} random primary keys, and for each weak key, the expanded key can be recovered with about $2^{33.6}$ keystream words and $2^{50.56}$ operations. The attack can be applied to ABC v1 and v2 to retrieve the expanded key generated by a weak key with the same complexity, but the number of weak keys is about $2^{97} + 2^{95.29}$. It reveals that ABC v3 incurs more weak keys than that of ABC v1 and v2.

1 Introduction

ABC v1 [6] is a synchronous stream cipher submitted to the ECRYPT call for Stream Cipher Primitives. In the first evaluation phase, it has been withdrawn because of the attacks proposed in [7, 8, 9], then the designers of ABC updated it to a new longer version called ABC v2 [11]. The designers claimed that the ABC v2 offers a security level of 2^{128} and no hidden weaknesses have been incorporated in the design of ABC.

Berbain and Gilbert [7] presented an attack on ABC v1 which needs 2^{32} keystream words and 2^{95} operations to retrieves the complete internal state (also called expanded key). But the attack isn't applicable to ABC v2. Khazaei [8] proposed a Divide and Conquer attack on ABC v1 which requires 10×2^{32} keystream words and 2^{95} operations. In [9], Khazaei and Kiaei mounted a distinguishing attack on ABC v1 and v2 which can distinguish the keystream of the cipher from a truly

*Shandong University, Jinan 250100. Email:hnzhang@math.sdu.edu.cn.
†Shandong University, Jinan 250100. Email:llin@mail.sdu.edu.cn.
‡Shandong University and Tsinghua University, Beijing 100087. Email:xywang@sdu.edu.cn.

random sequence with the data, time and memory complexity of $O(2^{32})$. But the designers [10] showed that the attack is wrong.

The recent attack on ABC v1 and v2 was proposed by Wu and Preneel [13]. They found a linear expression of carry bits which has a substantial probability bias and results in a fast correlation to break ABC v2 with 2^{96} weak keys. In [15], we presented the strict theoretical proof for the probability advantage of some linear expressions .

In order to avoid of Wu-Preneel attack, the designers improved the key expansion algorithm to eliminate the probability bias. Our cryptanalysis reveals that, the improved key expansion algorithm results in another type of weak keys, and the number of weak keys is up to $2^{103.71}$. For any weak key, we can apply a fast correlation attack to recover all the expanded subkeys of ABC v3 with about $2^{33.6}$ keystream words and $2^{50.56}$ operations.

This paper is organized as follows: In section 2, a brief description of ABC v3 is given. In section 3, we discuss how the probability bias occurs by searching a shrunken ABC v3 cipher. In section 4, we prove that there are a large amount of weak keys, and present the their occurrence probability. We employ a fast correlation attack to break ABC v3 in section 5, 6. Finally, the paper is concluded in section 7.

2 A Brief Description of ABC v3

ABC v3 is a synchronous stream cipher optimized for software applications. It uses a 128-bit key and a 128-bit IV. Throughout this paper, the symbols $\oplus$, $\gg$, $\ll$, $\ggg$, + are respectively used for 32-bit XOR, right shift, left shift, right rotation and addition modulo 2^{32}.

2.1 Key Stream Generator

ABC v3 consists of three components, the key stream generator of the cipher is given in Fig. 1

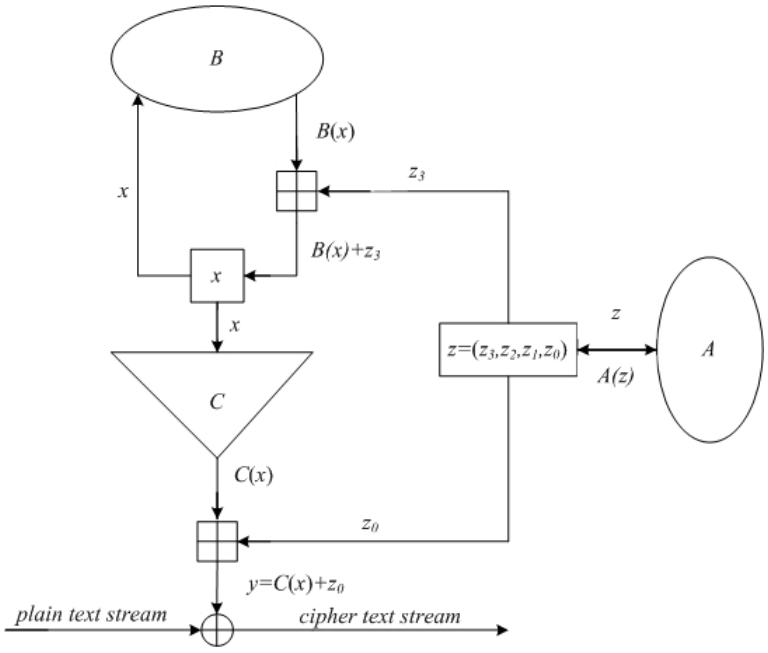

Fig.1. Keystream generator of ABC v3

1. Component A is a linear transformation of the 128−bit LFSR, and its characteristic polynomial is $f(x) = x(x^{127}+x^{63}+1)$. The internal state is represented by $Z = (z_3, z_2, z_1, z_0)$, $z_i \in GF(2^{32})$, $0 \le i < 4$. At each clock, the LFSR is updated as follows:

$$Temp \leftarrow z_2 \oplus (z_1 << 31) \oplus (z_0 >> 1)$$

$$z_0 \leftarrow z_1,\ \ z_1 \leftarrow z_2,\ \ z_2 \leftarrow z_3,\ \ z_3 \leftarrow Temp.$$

2. Component B is a single cycle T-function. Where d_0, d_1, $d_2 \in GF(2^{32})$ represent 32-bit keys and IV dependent words.

$$B(x) = ((x \oplus d_0) + d_1) \oplus d_2 (mod\ 2^{32}).$$

where $d_0 \equiv 0$, $d_2 \equiv 0$, $d_1 \equiv 1$ (mod 4).

3. Component C is a mapping from $GF(2^{32})$ to $GF(2^{32})$ which involves key dependent constant 32-bit words $e, e_i (0 \le i < 32)$. $\delta_i(x)$ is the i-th bit selection function that determines the i-th bit of a 32-bit integer number.

$$C(x) = ((e + \sum_{i=0}^{31} e_i \delta_i(x))\ mod\ 2^{32}) >>> 16,\ \ e_{31} \equiv 2^{16} (mod\ 2^{17}).$$

2.2 Key Expansion

The main idea behind the key expansion routine is to use the ABC keystream generator with feedback. A single call of ABC keystream generator will be denoted here as the function g:

$$g(z, x, d_0, d_1, d_2, e, \{e_i\}_{i=0}^{31}) \ \ i.e. \ \ g : Z/2^{1312}Z \longrightarrow Z/2^{32}Z.$$

Let K be the 128-bit primary key, $K = (k_3, k_2, k_1, k_0) \ \in (Z/2^{32}Z)^4$.

In all ABC keystream generator calls during the key expansion algorithm the fixed values e', $\{e'_i\}_{i=0}^{31}$ of the coefficients of C are used. During the key expansion the following temporary variables

$$z' = (z'_3, z'_2, z'_1, z'_0) \in (Z/2^{32}Z)^4, x'_0, \ d'_0, \ d'_1, \ d'_2, \zeta \in Z/2^{32}Z.$$

are set directly using the primary key K. Then after a number of warmups with feedback the ABC keystream generator is run to fill the ABC secret state. At each warmup iteration a 32-bit output of generator is fed back to some part of the state by means of bitwise addition modulo 2. After the warmup phase the ABC keystream generator is used to obtain the values for $(d_0, d_1, d_2, x, z, \{e_i\}_{i=0}^{31}, e)$, composing the expanded key of ABC v3.

The values for $\{e_i\}_{i=0}^{31}$ are obtained in a special way to avoid the combinations of values that are weak (i.e. the values which are vulnerable to the Wu-Preneel attack in [13]).

Definition 1. Denote a as a $k-$bit integer, we define $\delta_s^r(a)$ as the integer

$$\sum_{i=s}^{r} \delta_i(a) 2^{i-s}, 0 \leq s \leq r \leq k-1.$$

The Key Expansion Algorithm of e, $\{e_i\}_{i=0}^{31}$

$e \leftarrow g(z', x', d'_0, d'_1, d'_2, e', \{e'_i\}_{i=0}^{31})$;
for i from 0 to 31 do
  $e_i \leftarrow g(z', x', d'_0, d'_1, d'_2, e', \{e'_i\}_{i=0}^{31})$;
  $e_i \leftarrow e_i \ AND \ g(z', x', d'_0, d'_1, d'_2, e', \{e'_i\}_{i=0}^{31})$;
  $e_i \leftarrow e_i \ AND \ g(z', x', d'_0, d'_1, d'_2, e', \{e'_i\}_{i=0}^{31})$;
  $e_i \leftarrow e_i \ AND \ g(z', x', d'_0, d'_1, d'_2, e', \{e'_i\}_{i=0}^{31})$;
end for
for i from 0 to 31 do
  for j from 0 to 2 do

$\zeta \leftarrow g(z', x', d'_0, d'_1, d'_2, e', \{e'_i\}_{i=0}^{31})$;
$\delta_i(e_{\delta_0^4(\zeta)}) \leftarrow 1; \quad \delta_i(e_{\delta_5^9(\zeta)}) \leftarrow 1; \quad \delta_i(e_{\delta_{10}^{14}(\zeta)}) \leftarrow 1;$
$\delta_i(e_{\delta_{15}^{19}(\zeta)}) \leftarrow 1; \quad \delta_i(e_{\delta_{20}^{24}(\zeta)}) \leftarrow 1; \quad \delta_i(e_{\delta_{25}^{29}(\zeta)}) \leftarrow 1;$
end for
$\zeta \leftarrow g(z', x', d'_0, d'_1, d'_2, e', \{e'_i\}_{i=0}^{31})$;
$\delta_i(e_{\delta_0^4(\zeta)}) \leftarrow 1; \quad \delta_i(e_{\delta_5^9(\zeta)}) \leftarrow 1;$
end for
$e_{30} \leftarrow e_{30}\ OR\ (e_{31}\ AND\ \ 0\text{x}0001\text{ffff});$
$e_{31} \leftarrow (e_{31}\ AND\ \ 0\text{xffff}0000)\ OR\ \ 0\text{x}0001\text{ffff}.$

The main purpose of this paper is to show that, the expanded subkeys e and $\{e_i\}_{i=0}^{31}$ are weak with a high probability, which can result in an efficient attack on the full cipher.

3 Some Notations and Conclusions

Component C is the kernel of ABC v3. We will observe some possible $0-1$ sequences generated by $C(x)$ which have probability bias of '0' occurrence in this section.

To describe our attack conveniently, we denote some notations as follows.

Let $x, e, e_0, e_1, \cdots, e_{k-1}$ be $k-$bit integers, $e_{k-1} \equiv 2^{\frac{k}{2}} (mod\ 2^{\frac{k}{2}+1})$, where $k = 2, 4, 6, 8, \cdots$.

The vector E_k is composed of $k+1$ integers, denoted as $(e, e_0, e_1, \cdots, e_{k-1})$.

The vector $\alpha(E_k, m)$ is composed of the m-th least significant bits of $k+1$ integers in E_k, denoted as $(\delta_m(e), \delta_m(e_0), \cdots, \delta_m(e_2), \delta_m(e_{k-1}))$, $0 \le m \le k-1$.

The function $S(E_k, x)$ is defined as

$$S(E_k, x) = (e + \sum_{i=0}^{k-1} e_i \delta_i(x))\ \ mod\ 2^k.$$

When x traverses all the integers in the interval $[0, 2^k - 1]$, we rank all the m-th bits of $S(E_k, x)$ as a sequence S_m:

$$S_m = (\delta_m(S(E_k, 0)),\ \delta_m(S(E_k, 1)), ...,\ \delta_m(S(E_k, 2^k - 1)))$$

We focus on the probability of '0' occurrence in S_m in our paper.

Definition 2. The probability of '0' occurrence in the sequence S_m is defined as

$$p(E_k, m) = 1 - \frac{\sum_{x=0}^{2^k-1} \delta_m(S(E_k, x))}{2^k}, 0 \le m \le k-1.$$

Let A is the set $\{E_k | p(E_k, m) \in X\}$ where $X \subseteq [0, 1]$. $N(E_k, m, X)$ is the element number of set A.

We only consider the case that $p(E_k, m) > 1/2$, and define $p(E_k, m) - 1/2$ as the probability bias of '0' in sequence S_m.

The following experiment data in Table 1 reflects that, some E_k can cause the probability bias of '0' in S_m. The question is that, under what conditions, the expanded subkey E_k can results in some S_m with available probability bias which can be applied to break ABC v3 cipher. The key K which generates such E_k is the weak key we are eager to explore.

Table 1: The distribution of $N(E_4, m, X)$, and the 1-st row denotes the intervals X

m	0.5	(0.5,0.6]	(0.6,0.7]	(0.7,0.8]	(0.8,0.9]	(0.9,1.0)
0	114688	0	0	0	0	0
1	90112	0	0	12288	0	0
2	131072	0	0	0	0	0
3	55904	12336	16320	6464	1600	112
m	0.0,1.0	[0.4,0.5)	[0.3,0.4)	[0.2,0.3)	[0.1,0.2)	(0.0,0.1)
0	16384	0	0	0	0	0
1	8192	0	0	20480	0	0
2	0	0	0	0	0	0
3	416	12496	17600	6336	1344	144

Let's recall the Wu-Preneel attack, the weak keys are those to produce the expanded subkey E_{32} under which $\alpha(E_{32}, 0)$ is 0-sequence, i.e., $\alpha(E_{32}, 0) = (0, 0, 0, \cdots, 0)$, so S_0 is a 0-sequence, and the probability bias of $p(E_k, m)$ is 1/2.

In order to prevent Wu-preneel attack, the ABC designers modified key expansion algorithm (section 2.2) to guarantee the numbers of '1' and '0' are close to balance. Especially, for both S_0 and S_{16}, the numbers of '1' and '0' are absolutely balanced.

The following conclusions show that S_0 and S_{16} are balanced.

Conclusion 1. For $m = 16$, S_{16} is balanced, i.e., $p(E_{32}, 16) = \frac{1}{2}$.
Proof. Denote x as a 32-bit integer. Suppose that

$$\sum_{x=0}^{2^{31}-1} \delta_{16}(S(E_{32}, x)) = N_1,\ N_0 = 2^{31} - N_1.$$

So,

$$\begin{aligned}
\sum_{x=0}^{2^{32}-1}\delta_{16}(S(E_{32},x)) &= \sum_{x=0}^{2^{31}-1}\delta_{16}(S(E_{32},x)) + \sum_{x=2^{31}}^{2^{32}-1}\delta_{16}(S(E_{32},x)) \\
&= \sum_{x=0}^{2^{31}-1}\delta_{16}(S(E_{32},x)) + \sum_{x=0}^{2^{31}-1}\delta_{16}(S(E_{32},(x+2^{31}))) \\
&= \sum_{x=0}^{2^{31}-1}\delta_{16}(S(E_{32},x)) + \sum_{x=0}^{2^{31}-1}\delta_{16}(e+\sum_{x=0}^{30}\delta_i(x)e_i+e_{31})) \\
&= \sum_{x=0}^{2^{31}-1}\delta_{16}(S(E_{32},x)) + \sum_{x=0}^{2^{31}-1}\delta_{16}(e+\sum_{x=0}^{31}\delta_i(x)e_i+e_{31}) \quad 1) \\
&= \sum_{x=0}^{2^{31}-1}\delta_{16}(S(E_{32},x)) + \sum_{x=0}^{2^{31}-1}\delta_{16}(S(E_{32},x))\oplus 1) \qquad 2)
\end{aligned}$$

Because $\delta_i(e_{31}) = 0, i = 0, 1, ..15$, and $\delta_{16}(e_{31}) = 1$, it is easy to see that 1) results in 2).

It is obvious that

$$\sum_{x=0}^{2^{32}-1}\delta_{16}(S(E_{32},x)) = N_0 + N_1 = 2^{31}.$$

i.e.

$$p(E_{32},16) = 1 - \frac{2^{31}}{2^{32}} = \frac{1}{2}.$$

Conclusion 2. For $m = 0$, if the hamming weight of $\alpha(E_{32}, 0)$ is greater than 1, S_0 is balanced, i.e., $p(E_{32}, 0) = \frac{1}{2}$.

Proof. Select any e_i which satisfies $\delta_0(e_i) = 1$ as e_{31}, the proof is similar to that of conclusion 1.

From conclusion 1, ABC v3 resists on the Wu-Preneel attack. In the following of this paper, we are interested in other sequences S_m ($m \neq 0, 16$) which have a high probability $p(E_{32}, m)$.

4 Weak Keys and Their Occurrence Probability

In this section, we mainly explore those E_{32} which generate some sequence S_m with $p(E_{32}, m) \geq \frac{1}{2} + \varepsilon$, and we call such a E_k as a ε-weak E_k.

Component C is based on the knapsack function of high non-linear. Because of both the memory and computation limits, it is hard to decide the exact probability only by computer exhaustive search for a higher ε. Therefore, we search a series of shrunken ciphers of ABC v3, and estimate the probability bias of '0' occurrence in S_m by the asymptotic approximation.

Definition 3. We define the set $M(k, m, \varepsilon)$ as

$$\{E_k | p(E_k, m) \geq \frac{1}{2} + \varepsilon\}$$

Here, $\varepsilon > 0,\ 0 \leq m \leq k-1,\ k = 2, 4, 6, 8....$ All the sets $M(k, m, \varepsilon)$ consist of all the ε-weak E_k.

By searching the shrunken ciphers with $k = 8, 10, 12, 14, 16,$ and selecting $r = m$, $s = m + 2$ $(m \geq 2)$, $\varepsilon = 0.03$, we find that, if E_k lies in $M(k, m, \varepsilon)$, then $\delta_s^r(e)$ and $\delta_s^r(e_i)$ $(0 \leq i \leq k-2)$ drop into $N = \{0, 1, 2, 5, 6, 7\}$ with probability 0.99, and the probability does not depend on k. This implies a solution to search for the ε-weak E_k by searching a small space $\{\delta_s^r(e), \delta_s^r(e_i) \in N,\ 0 \leq i \leq k-2\}$ instead of the full space $\{\delta_s^r(e), \delta_s^r(e_i) \in \{0, 1, 2, 3, 4, 5, 6, 7\},\ 0 \leq i \leq k-2\}$, and the search space is cut down $(\frac{8}{6})^{32} \approx 2^{13.28}$ times which coincides with our experiment data. Utilizing this property , we can present an efficient algorithm to search for many ε-weak E_k.

Searching Algorithm for Weak E_k

1. Given $\varepsilon > 0$, select k to be $4, 6, 8, \cdots$, m to be $2, 3, \cdots, k-1$.
2. Select E_k randomly, $k = 4, 6, 8, \cdots$
3. Compute $\delta_{m-2}^m(e)$. If $\delta_{m-2}^m(e) = 3$, set the $m-$th least significant bit of e as '1'. If $\delta_{m-2}^m(e) = 4$, set the $m-$th least significant bit of e as '0'. Similar operation to e_i $(0 \leq i \leq k-2)$.
4. Compute $p(E_k, m)$. If E_k is in $M(k, m, \varepsilon)$, output E_k.

Step 3 performs a transformation as:

$$7 \longleftarrow 3 \quad i.e. \quad (111)_2 \longleftarrow (011)_2$$

$$0 \longleftarrow 4 \quad i.e. \quad (000)_2 \longleftarrow (100)_2$$

After correcting e and e_i is step 3, $\delta_{m-2}^m(e)$ and $\delta_{m-2}^m(e_i)$ fall into the set $\{0, 1, 2, 5, 6, 7\}$ instead of $\{0, 1, 2, 3, 4, 5, 6, 7\}$. There are two advantages for the performance.

First, it does not change the hamming weight of $\alpha(E_k, m)$. Second, it improves the efficiency to search for weak E_k in $M(k, m, \varepsilon)$ among the corrected E_k.

It's clear that all the corrected E_k after step 3 abide by the following probability distribution.

1. $Pr(m, j, i) = Pr(m, j) = 0, j = 3, 4$
2. $Pr(m, j, i) = Pr(m, j) = 1/4, j = 0, 7$
3. $Pr(m, j, i) = Pr(m, j) = 1/8, j = 1, 2, 5, 6$

such that

$$Pr(m, j, i) = Pr(\delta_s^r(e_i) = j),\ Pr(m, j) = Pr(\delta_s^r(e) = j)$$

and $i = 0, 1, ..., k - 2$

Let B is the set of all the corrected E_{32} such that $B = \{E_{32}|$ for some m, E_{32} satisfying the above probability distributions 1-3$\}$

Experiment 2. We run the searching algorithm to find weak E_{32}. For every available m, $m \neq 1$, we search for 2^{20} E_{32} which fall into the set B, and all the numbers of the searched weak E_{32} corresponding to every m are listed in Table 2. Let $C_{weak}(E_{32}, m, \epsilon)$ are the weak E_{32} in B searched with $p(E_{32}, m) \geq 1/2 + \varepsilon$. It is remarked that, there are some common weak E_{32} among the sets $C_{weak}(E_{32}, m, \epsilon)$. We neglect to count these common weak E_{32} because of their little ratio.

Summing up the data of Table 2, we know that a E_{32} in set B is weak with average probability

$$Pr_a = Pr(E_{32} \in \bigcup_{m \neq 1} C_{weak}(E_{32}, m, \epsilon)) = 2^{-6.53}.$$

Remark. From the experiment data in table 2, we know that $C_{weak}(E_{32}, 0, \epsilon)$ and $C_{weak}(E_{32}, 16, \epsilon)$ are empty which coincides with the 0-1 balance in S_0 and S_{16}. When $m = 1$, the searching algorithm is not available, we label '$\star$' for it.

After deciding the probability that a E_{32} in B is weak, the remaining is to decide such probability that, for any primary key K of ABC v3, the expanded key E_{32} falls into the set B.

We suppose that any expanded key by the key expansion algorithm has an average hamming weight 16, and the following probability holds.

$$Pr(\delta_m^{m+2}(e) = \beta) \approx \frac{1}{8}, Pr(\delta_m^{m+2}(e_i) = \beta) \approx \frac{1}{8}, 0 \leq \beta \leq 7.$$

Table 2: The 0.03-weak E_{32} searched from 2^{20} $E_{32} \in B$

m	0	1	2	3	4	5	6	7
	0	$\star$	232	435	520	616	623	623
m	8	9	10	11	12	13	14	15
	656	627	640	618	622	604	612	660
m	16	17	18	19	20	21	22	23
	0	168	226	187	218	228	219	211
m	24	25	26	27	28	29	30	31
	234	194	232	235	216	236	237	219

The above assumption should be right, and the fact is claimed by the ABC designers, and our experiment data also is consistent with the above distributions.

It is easy to show that, for any expanded key E_{32}, the probability of E_{32} falling into set B is about:

$$Pr_b = Pr(E_{32} \in B) = \frac{C_{32}^8 C_{24}^8 C_{16}^4 C_{12}^4 C_8^4}{(C_{32}^{16})^3} = 2^{-18.76}.$$

So, the probability that any expanded subkey E_{32} is 0.03-weak is about

$$Pr_a.Pr_b = 2^{-25.29},$$

and the total 0.03-weak E_{32} is about $2^{102.71}$.

Similarly, by consideration of probability bias with '1' in S_m. we can get another $2^{102.71}$ 0.03-weak E_{32}.

We define a primary key K as a weak key if it's corresponding expanded subkey E_{32} is 0.03-weak in our attack. So, the total 0.03-weak E_{32} is up to $2^{103.71}$.

The following is two examples of weak E_{32} for ABC v3.

The next section is to discuss how to break ABC v3 when the initial K is weak.

5 Distinguishing the Weak Keys

5.1 The Linear Expressions with Probability Advantages

Before we give the efficient attack on ABC v3 with weak keys. We first recall two important linear expressions with probability advantages introduced in [13] and [15].

Table 3: Two examples of 0.03-weak E_{32} searched by key expansion algorithm

	363c321b	21a9208d	b311b840	4f4b7fe0	06dd900a
	a2c0df59	58625794	c5dee834	37ea47e7	2e114eef
$p(E_{32}, 10)$	5c0cbf1f	07ceb784	7671c0ba	83801813	4b1bff7c
$\Downarrow$	2ac3775c	d10446ff	62cf49bd	e6dff846	092f0006
0.5739	dfbab85e	74a95fc9	d7b5f788	293787b7	b41cfff1
	c7347820	ec16a0bc	dd0ef743	62180f97	fbdee8ca
	90a18fb5	75b3d9d2	26650000		
	3418fe95	6bb79d53	f0c81a32	80784c16	a838b9cb
	b8ffb09d	5fc9b4c4	7657a469	8da04924	064f4366
$p(E_{32}, 18)$	5c287218	45206110	6ecf3e01	5b980794	587f852c
$\Downarrow$	b7385fbb	85dfc44d	be1f7df6	42b7f8f7	fd804ab1
0.7067	2587434f	1f77a5ee	87d79dfc	7d20d3a4	8730d6ca
	c9e7fdfa	b8407d90	14a76110	294ec363	83284f00
	249f953f	8797f7df	66390000		

Lemma 1. Denote a and b as two random and independent $n-$bit integers. Let $c_n = \delta_n(a+b)$, where c_n denotes the carry bit at the n-th least significant bit position. Denote the most significant bit of a as a_{n-1}.Then $Pr(c_n \oplus a_{n-1} = 0) = \frac{3}{4}$.

See Lemma 1 in section 3.1 in [13].

Lemma 2. Denote $a_i, b_i, c_i (1 \leq i \leq 3)$ as three random n-bit integers. If $c_{i,n} = \delta_n(a_i + b_i)$, $a_1 \oplus a_2 = a_3$, then

$$Pr(c_{1,n} \oplus c_{2,n} \oplus c_{3,n} = 0) = \frac{4}{7} + \frac{3}{7} \times \frac{1}{8^n}.$$

See Theorem 2 in section 4.1 in [15].

5.2 Distinguishing Weak Keys

We notice that the output word is $y = z_0 + C(x)$ and the component A is a linear feedback shift register with primitive polynomial $g(x) = x^{127} + x^{63} + 1$. Take the $2^5 th$ power of $g(x)$, we obtain

$$g(x)^{2^5} = x^{127\times32} + x^{63\times32} + 1. \tag{1}$$

Denote y, z_0, x at the i-th step as y_i, z_0^i, x_i. Since each time 32 bits are updated, the distance between $\delta_m(z_0^i)$ and $\delta_m(z_0^{i+j})$ is $32j$. According to (1), we obtain the following linear recurrence

$$\delta_m(z_0^i) \oplus \delta_m(z_0^{i+63}) \oplus \delta_m(z_0^{i+127}) = 0. \tag{2}$$

Notice that $C(x) = S(E_{32}, x) \ggg 16$, we know

$$\delta_m(y_i) = \delta_u(S(E_{32}, x_i)) \oplus \delta_m(z_0^i) \oplus c_{i,m}, \;\; u \geq 0, \; u = m - 16 \;\; mod \; 32. \tag{3}$$

where $c_{i,m} = \delta_m(z_{0,m}^i + C_m(x))$, and $z_{0,m}^i = z_0^i \;\; mod \; 2^m$, $C_m(x) = C(x) \;\; mod \; 2^m$
From (2) and (3), we know

$$\delta_m(y_i) \oplus \delta_m(y_{i+63}) \oplus \delta_m(y_{i+127}) = \varphi \oplus c_{i,m} \oplus c_{i+63,m} \oplus c_{i+127,m}.$$

Here, $\varphi = \delta_u(S(E_{32}, x_i)) \oplus \delta_u(S(E_{32}, x_{i+63})) \oplus \delta_u(S(E_{32}, x_{i+127}))$.

For a weak key, $Pr(\delta_m(S(E_{32}, x_i)) = 0) \geq 0.53$, $2 \leq m \leq 31, m \neq 16$, using the lemma 2 and Piling-up Lemma, the bias occurs.

$$Pr(\delta_m(y_i) \oplus \delta_m(y_{i+63}) \oplus \delta_m(y_{i+127}) = 0) \geq 0.5000123.$$

From normal distributed formula, the maximal number of the keystream outputs which are required to distinguish a weak key with success rate 99.8% is $N = 2 \times (0.500014 - \frac{1}{2})^{-2} = 2^{33.6}$. The complexity is $2 \times (32 - 3) \times 2^{33.6} = 2^{39.46}$.

6 Recovering the Expanded keys of ABC v3

The initial value of the LFSR is recovered by exploiting the strong correlation between the LFSR and the keystream. From Lemma 1 in section 5.1, we obtain

$$Pr(\delta_{n-1}(z_0^i) \oplus c_{i,n} = 0) = \frac{3}{4}. \tag{4}$$

From (3) and (4),we get

$$Pr(\delta_n(y_i) \oplus \delta_{n-1}(z_0^i) \oplus \delta_n(z_0^i) = 0) = \frac{1}{4} + \frac{1}{2} Pr(\delta_u(S(E_{32}, x_i)) = 0). \tag{5}$$

Here, $0 < n \leq 31, n \neq 16, 17, u \geq 0, u = n - 16 \;\; mod \; 32$,

For a weak key, $Pr(\delta_u(S(E_{32}, x_i)) = 0) \geq 0.53$, according to (5), we get the following correlation:

$$Pr(\delta_n(y_i) \oplus \delta_{n-1}(z_0^i) \oplus \delta_n(z_0^i) = 0) \geq 0.515.$$

The fast correlation attack Algorithm B of Meier and Staffelbach [1] can be applied to recover the LFSR. Fixing n, then u will be fixed correspondingly. Let $p = \frac{1}{4} + \frac{1}{2}Pr(\delta_u(S(E_{32}, x_i)) = 0)$, $v_i = \delta_{n-1}(z_0^i) \oplus \delta_n(z_0^i)$, and $w_i = \delta_n(y_i)$. By squaring the polynomial (1) iteratively, we obtain a number of linear relations for every v_i:

$$v_i \oplus v_{i+63\cdot 2^j} \oplus v_{i+127\cdot 2^j} = 0 \quad (j \geq 0). \tag{6}$$

From (5) and (6), we obtain

$$s = Pr(w_i \oplus w_{i+63\cdot 2^j} \oplus v_{i+127\cdot 2^j} = 0 | v_i = w_i) = p^2 + (1-p)^2.$$

where each value of j indicates one relation of w_i, similar to $w_{i+63\cdot 2^j}$ and $w_{i+127\cdot 2^j}$. In the average there are m relations for w_i as

$$m = m(N, k, t) = log_2(\frac{N}{2k})(t+1), \tag{7}$$

where N is the number of outputs, $k = 127$(the length of the LFSR), $t = 2$(taps) for ABC v3. The probability that w_i satisfies at most h of m relations

$$U(p, m, h) = \sum_{r=0}^{h} C_m^r (ps^r(1-s)^{r-l} + (1-p)(1-s)^r s^{m-r}).$$

The probability that $w_i = v_i$ and that at most h of m relations are satisfied

$$V(p, m, h) = \sum_{r=0}^{h} C_m^r ps^r(1-s)^{m-r}.$$

The probability that $w_i \neq v_i$ and that at most h of m relations are satisfied

$$W(p, m, h) = \sum_{r=0}^{h} C_m^r (1-p)(1-s)^r s^{m-r}.$$

With regard to the described method to correct digits if they satisfy at most h relations, the total number of digits of w_i changed by $U(p, m, h) \cdot N$, moreover the number of erroneously changed digits is $V(p, m, h) \cdot N$, and the number of correctly changed digits is $W(p, m, h) \cdot N$. So, the relative increase is

$$I(p, m, h) = W(p, m, h) - V(p, m, h)$$

To measure the correction effect, Meier and Staffelbach introduced a function $F(p, t, N/k) = I(p, m, h_{max}) \cdot (N/k)$ (here, h_{max} is the value h such that $I(p, m, h)$ is

maximum for given p and m). Of course, if $F(p, t, N/k) \leq 0$, algorithm B definitely fails.

If algorithm B is valid, its computational complexity is of order $O(k)$, i.e. linear in the length k of the LFSR. But the approximate supreme bound of its complexity is unknown. In order to determine its approximate value and the number of the output words which can be utilized to recover the internal state of ABC v3 for weak keys, we present a variational algorithm of algorithm B.

Variational Algorithm B.

1. Select N, p and Num which is a counter.
2. Determine m according to formula (7), initialize the iteration counter $Num = 0$.
3. Find the value of $h = h_{max}$ such that $I(p, m, h)$ is maximum. If $I_{max} = I(p, m, h_{max}) \leq 0$, then return Step 1. If $I_{max} > 0$, increase Num with '1'.
4. If $I_{max} > 1 - p$, then terminate. Else, replace p with $p + I_{max}$, return Step 3.

It is obvious that p can become small with the increase of N in the condition that Algorithm B is still efficient, and the number of weak keys will increase corresponding. Perform the variational algorithm B with $N = 2^{32}$ and $p = 0.515$, we obtain that the final p is 0.98 with $Num = 0x3ec4$. Therefore, the maximal computational complexity is

$$2 \cdot log_2(\frac{N}{2k})(t + 1) \cdot N \cdot Num = 2^{50.56}.$$

After recovering the subkey of LFSR, we can find the subkeys of the function $B(x)$ and $C(x)$ by the similar method in [13] with about $2^{17.5}$ keystream words and $2^{32.84}$ operations. So the total complexity to recover all the subkeys of ACB v3 is about $2^{33.6}$ keystream words and $2^{50.56}$ operations.

A similar attack can be applied to break ABC v1 and v2. The numbers of weak keys for ABC v1 and v2 are both at least $2^{97} + 2^{95.29}$ with at most $2^{33.6}$ key stream words and $2^{50.56}$ operations. So, ABC v1, v2 and v3 are all insecure.

7 Conclusion

In the paper, we have showed that there is a large amount of weak keys up to $2^{103.7}$. For the full ABC v3 with a weak key, our attack can retrieve the all the expanded keys (all the internal states) with $2^{33.6}$ keystream words and $2^{50.56}$ operations. Our

cryptanalysis implies that ABC v3 is vulnerable to weak key attack than the early two versions.

References

[1] W. Meier, O. Staffelbach, Fast Correlation Attacks on Stream Ciphers. Journal of Cryptology 1(3), pp. 159-176, 1989.

[2] A.Klimov, A.Shamir. A new class of invertible mappings, in Cryptographic Hardware and Embedded Systems 2002 (B.S.Kaliski Jr.etal., eds.),Lect. Notes in Comp. Sci.,Vol. 2523, Springer-Verlag, 2003,pp.470(483. 22, 26).

[3] A.Klimov, A.Shamir.Cryptographic applications of T-functions,in: Selected Areas in Cryptography -2003

[4] A.Klimov, A.Shamir, New Cryptographic Primitives Based on Multiword T-functions, in: Fast Software Encryption -2004, 11th Int'Workshop. Lect. Notes Comp. Sci., Vol. 3017, Springer-Verlag, 2004£pp. 1-15

[5] E.Kasper. What do you think about T-functions? T-79.515 Cryptography: Special Topics. March 15, 2005. Cryptography: Special Topics.

[6] V.Anashin, A.Bogdanov, I.Kizhvatov, S.Kumar. ABC: A new fast exible stream cipher. ECRYPT Stream Cipher Project Report 2005/001, 2005.http://www.ecrypt.eu.org/stream.

[7] C.Berbain and H.Gilbert. Cryptanalysis of ABC. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/048, 2005.http://www.ecrypt.eu.org/stream.

[8] S.Khazaei.Divide and conquer attack on ABC stream cipher.eSTREAM, ECRYPT Stream Cipher Project, Report 2005/052, 2005.http://www.ecrypt.eu.org/stream.

[9] S.Khazaei and M.Kiaei. Distinguishing attack on theABC v.1 and v.2. eSTREAM, ECRYPT Stream Cipher Project, Report2005/061, 2005.

[10] V.Anashin, A.Bogdanov, I.Kizhvatov. Increasing the ABC stream cipher period. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/050, 2005. http://www.ecrypt.eu.org/stream.

[11] V.Anashin, A.Bogdanov, I.Kizhvatov, and S.Kumar. ABC: A new fast exible stream cipher. Version 2, 2005.http://crypto.rsuh.ru/papers/abc-spec-v2.pdf.

[12] V.Anashin,A.Bogdanov,I.Kizhvatov.ABC is safe and sound.

[13] H. J. Wu, B. Preneel, Cryptanalysis of ABC v2.http://www.ecrypt.eu.org/stream/abc.html,Feb of 2006.

[14] M.Daum, Narrow T-functions, CITS Research Group,Ruhr-University Bochum, daum@cits.rub.de

[15] H.N. Zhang, S.H. Wang, X.Y. Wang, Two linear expressions with probability advantages in symmetric ciphers. http://www.ecrypt.eu.org/stream/papersdir/2006/046.pdf.