New Weaknesses in the Keystream Generation Algorithms of the Stream Ciphers TPy and Py*

Gautham Sekar Souradyuti Paul Bart Preneel

Katholieke Universiteit Leuven, Dept. ESAT/COSIC,
Kasteelpark Arenberg 10,
B–3001, Leuven-Heverlee, Belgium
{gautham.sekar, souradyuti.paul, bart.preneel}@esat.kuleuven.be

Abstract

The stream ciphers Py, Py6 designed by Biham and Seberry were promising candidates in the ECRYPT-eSTREAM project because of their impressive speed. Since their publication in April 2005, a number of cryptanalytic weaknesses of the ciphers have been discovered. As a result, a strengthened version Pypy was developed to repair these weaknesses; it was included in the category of 'Focus ciphers' of the Phase II of the eSTREAM competition. However, even the new cipher Pypy was not free from flaws, resulting in a second redesign. This led to the generation of three new ciphers TPypy, TPy and TPy6. The designers claimed that TPy would be secure with a key size up to 256 bytes, i.e., 2048 bits. In February 2007, Sekar *et al.* published an attack on TPy with 2^{281} data and comparable time. This paper shows how to build a distinguisher with $2^{268.6}$ key/IVs and one outputword for each key (i.e., the distinguisher can be constructed within the design specifications); it uses a different set of weak states of the TPy. Our results show that distinguishing attacks with complexity lower than the brute force exist if the key size of TPy is longer than 268 bits. Therefore, for longer keys, our attack constitutes an academic break of the cipher. Furthermore, we discover a large number of similar bias-producing states of TPy and provide a general framework to compute them. The attacks on TPy are also shown to be effective on Py.

1 Introduction

Timeline: the Py-family of Ciphers

- **April 2005.** The ciphers Py and Py6, designed by Biham and Seberry, were submitted to the ECRYPT project for analysis and evaluation in the category of software based stream ciphers [2]. The impressive speed of the cipher Py in software (about 2.5 times faster than the RC4) made it one of the fastest and most attractive contestants. The cipher is designed to be used with a key of size 32 bytes (the key size may vary between 1 byte and 256 bytes) and an IV of size 16 bytes (the IV size can vary between 1 and 64 bytes).
- **March 2006 (at FSE 2006).** Paul, Preneel and Sekar reported distinguishing attacks with $2^{89.2}$ data and comparable time against the cipher Py [7]. Crowley [4] later reduced the complexity to 2^{72} by employing a Hidden Markov Model.

*The first author is supported by an IWT SoBeNeT project. The second author is funded by the IBBT (Interdisciplinary Institute for BroadBand Technology), a research institute founded by the Flemish Government in 2004. This is a revised version of the paper published in the proceedings of ISC 2007.

- **March 2006 (at the Rump session of FSE 2006).** A new cipher, namely Pypy, was proposed by the designers to rule out the aforementioned distinguishing attacks on Py [3].

- **May 2006 (presented at Asiacrypt 2006).** Distinguishing attacks were reported against Py6 with $2^{68.6}$ data and comparable time by Paul and Preneel [8].

- **October 2006 (to be presented at Eurocrypt 2007).** Wu and Preneel showed key recovery attacks against the ciphers Py, Pypy, Py6 with chosen IVs [10]. This attack was subsequently improved by Isobe *et al.* [5].

- **January 2007.** Three new ciphers TPypy, TPy, TPy6 were proposed by the designers [1]. These three ciphers can very well be viewed as the strengthened versions of the previous ciphers Py, Pypy and Py6 where the above attacks do not apply. The ciphers are designed to be secure for any key size between 1 byte and 256 bytes.

- **February 2007.** Sekar *et al.* published an attack on TPy which requires 2^{281} data and comparable time [9].

In this paper, we show distinguishing attacks on the ciphers TPy and Py with data complexity $2^{268.6}$ each. These results outperform the most recent attack on TPy which requires 2^{281} data [9]. However, it is worth noting that the attacks described in [7] can also be applied to TPy. In the design specifications, the TPy and the Py are claimed to be compatible with key size ranging from 8 bits to 2048 bits. If the ciphers are used with key size longer than 268 bits then our attacks are better than exhaustive search. It is also worth noting that the distinguisher can be built within the design specifications of the ciphers. To derive the distinguisher, $2^{268.6}$ randomly chosen key/IVs are used and for each of them one outputword is collected. Note that, according to the design specification, TPy can run for 2^{61} rounds (note that each round generates 8 bytes as output) per key where our distinguisher requires only 8 rounds per key.

In addition to the above distinguisher, we detect biases in a large number of outputs at rounds r, $r+2$, t and u where $r > 0$; $t, u \geq 5$; $t \notin \{r, r+2, u\}$; $u \notin \{r, r+2, t\}$. We provide a general framework to compute the biases due to the presence of arbitrarily many weak states. However, we were unable to combine those biases into a more efficient attack. Combining multiple distinguishers into a single and more efficient one is still an alluring open problem.

2 The Round Function of TPy

The round functions of the TPy and the Py are identical. Here, we analyze only the round function of TPy and hence do not describe the key setup and IV setup. Algorithm 1 describes a single round of the TPy. Array P (which is a permutation of $[0, 1, ..., 255]$), array Y (which contains 260 32-bit elements) and the 32-bit variable s are the inputs to the algorithm. Here, 'rotate(A)' denotes a cyclic rotation of the elements of array A by one position. The '$ROTL32(s, k)$' operation means that the 32-bit variable s is rotated to the left by k positions. The output generated in line 5 of the algorithm is labeled 'first output-word' and the output-word of line 6 is labeled 'second output-word'.

3 Notation and Convention

- $O_{a(b)}$ denotes the bth bit ($b = 0$ denotes the least significant bit or lsb) of the first output-word generated at round a. We do not use the second output-word anywhere in our analysis.

- P_a, Y_{a+1} and s_a are the inputs to the algorithm at round a. It is easy to see that when this convention is followed, $O_a = (ROTL32(s_a, 25) \oplus Y_a[256]) + Y_a[P_a[26]]$- the index '$a$' is maintained throughout the expression.

Algorithm 1 A Step of TPy

Require: $Y[-3, ..., 256]$, $P[0, ..., 255]$, a 32-bit variable s

Ensure: 64-bit random output

/*Update and rotate P*/
swap ($P[0]$, $P[Y[185]\&255]$);
rotate (P);
/* Update s*/
$s+ = Y[P[72]] - Y[P[239]]$;
$s = ROTL32(s, ((P[116] + 18)\&31))$;
/* Output 8 bytes (least significant byte first)*/
output $((ROTL32(s, 25) \oplus Y[256]) + Y[P[26]])$;
output $((\quad s \quad \oplus Y[-1]) + Y[P[208]])$;
/* Update and rotate Y*/
$Y[-3] = (ROTL32(s, 14) \oplus Y[-3]) + Y[P[153]]$;
rotate(Y);

- $Y_a[b]$, $P_a[b]$ denote the bth elements of array Y_a and P_a respectively.
- $Y_a[b]_i$, $P_a[b]_i$ denote the ith bit ($i = 0$ denotes the lsb) of $Y_a[b]$, $P_a[b]$ respectively.
- The operators '+' and '−' denote *addition modulo* 2^{32} and *subtraction modulo* 2^{32} respectively, except when used with expressions which relate two elements of array P. In this case they denote *addition and subtraction over* $\mathbb{Z}$.
- The symbol '$\oplus$' denotes bitwise *exclusive-or* and $\bigcap$ denotes set intersection.
- In $O_{a(i)}$, $s_{a(i)}$ and $Y_a[P_b[X]]_i$, the index representing bit position, i.e., i denotes $i \bmod 32$.
- $Y_a^c[P_b[X]]_i$ denotes the complement of $Y_a[P_b[X]]_i$.
- The pseudorandom bit generation algorithm of a stream cipher is denoted by PRBG.

4 Motivational Observations

Our major observation is the detection of a relation between the elements of the internal state and the outputs of the TPy which can, eventually, be used to build a distinguishing attack on the cipher. The relation is outlined in the following theorem.

Theorem 1 $O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)} = 0$ *if the following* 17 *conditions are simultaneously satisfied.*

1. $P_1[116] \equiv -18 \bmod 32$ *(event* E_1*),*
2. $P_2[116] \equiv 7 \bmod 32$ *(event* E_2*),*
3. $P_3[116] \equiv -4 \bmod 32$ *(event* E_3*),*
4. $P_7[116] \equiv 3 \bmod 32$ *(event* E_4*),*
5. $P_8[116] \equiv 3 \bmod 32$ *(event* E_5*),*
6. $P_1[72] = P_2[239] + 1$ *(event* E_6*),*
7. $P_1[239] = P_2[72] + 1$ *(event* E_7*),*

8. $P_7[72] = P_8[72] + 1$ *(event E_8)*,

9. $P_7[239] = P_8[239] + 1$ *(event E_9)*,

10. $P_3[72] = 254$ *(event E_{10})*,

11. $P_1[26] = P_3[239] + 2$ *(event E_{11})*,

12. $P_1[72] = 3$ *(event E_{12})*,

13. $P_3[26] = 0$ *(event E_{13})*,

14. $P_1[239] = P_7[26] + 6$ *(event E_{14})*,

15. $P_7[153] = 252$ *(event E_{15})*,

16. $P_6[153] = P_8[26] + 2$ *(event E_{16})*,

17. $d_{7(i-7)} \oplus d_{8(i-7)} \oplus c_{1(i)} \oplus d_{3(i)} \oplus d_{1(i+7)} \oplus c_{3(i+7)} \oplus c_{7(i+7)} \oplus e_{7(i+7)} \oplus c_{8(i+7)} \oplus e_{8(i+7)} = 0$ *(event E_{17})*.[1]

Proof. First, we state and prove two lemmata which will be used to establish the theorem.

Lemma 1 *If*

1. $P_1[116] \equiv -18 \bmod 32$,

2. $P_3[116] \equiv -4 \bmod 32$,

3. $P_7[116] \equiv 3 \bmod 32$,

4. $P_8[116] \equiv 3 \bmod 32$

then the following equations are satisfied:

1. $O_{1(i)} = s_{0(i+7)} \oplus Y_1[P_1[72]]_{i+7} \oplus Y_1^c[P_1[239]]_{i+7} \oplus Y_1[256]_i \oplus Y_1[P_1[26]]_i \oplus c_{1(i)} \oplus d_{1(i+7)}$,

2. $O_{3(i+7)} = s_{2(i)} \oplus Y_3[P_3[72]]_i \oplus Y_3^c[P_3[239]]_i \oplus Y_3[256]_{i+7} \oplus Y_3[P_3[26]]_{i+7} \oplus c_{3(i+7)} \oplus d_{3(i)}$,

3. $O_{7(i+7)} = Y_7[P_7[72]]_{i-7} \oplus Y_7^c[P_7[239]]_{i-7} \oplus Y_6[-3]_{i+7} \oplus Y_7[P_7[26]]_{i+7} \oplus Y_6[P_6[153]]_{i+7} \oplus c_{7(i+7)} \oplus d_{7(i-7)} \oplus e_{7(i+7)}$,

4. $O_{8(i+7)} = Y_8[P_8[72]]_{i-7} \oplus Y_8^c[P_8[239]]_{i-7} \oplus Y_7[-3]_{i+7} \oplus Y_8[P_8[26]]_{i+7} \oplus Y_7[P_7[153]]_{i+7} \oplus c_{8(i+7)} \oplus d_{8(i-7)} \oplus e_{8(i+7)}$.

Proof. From Figure 1, we get

$$Y_n[i] = Y_{n+1}[i-1] \tag{1}$$

when $-2 \le i \le 256$. When $i = -3$,

$$Y_{n+1}[256] = (ROTL32(s_i, 14) \oplus Y_n[-3]) + Y_n[P_n[153]].$$

Generalizing (1), we have

$$Y_n[i] = Y_{n+k}[i-k] \tag{2}$$

when $-3 \le i - k \le 255$. Line 5 of Algorithm 1 gives

[1] The terms c, d, e are the carries generated in certain expressions, the descriptions of which can be found in the proof of Theorem 1.

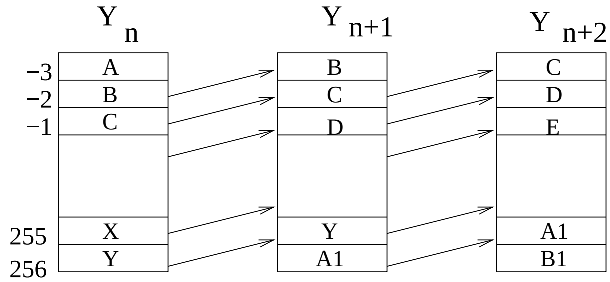

Figure 1: The figure shows the update of the S-box Y. $Y_n[i] = Y_{n+1}[i-1]$ when $-2 \le i \le 256$. $Y_{n+1}[256] = A1$ when $i = -3$ and $A1 = (ROTL32(s_n, 14) \oplus A) + Y_n[P_n[153]]$. Generalizing the above, we can write $Y_n[i] = Y_{n+k}[i-k]$ when $-3 \le i-k \le 255$.

$$O_7 = (ROTL32(s_7, 25) \oplus Y_7[256]) + Y_7[P_7[26]]. \tag{3}$$

Let the c_7 denote the carry in the above equation. Since $ROTL32(s_7, 25)_i = s_{7(i-25 \bmod 32)}$,

$$O_{7(i)} = s_{7(i-25 \bmod 32)} \oplus Y_7[256]_i \oplus Y_7[P_7[26]]_i \oplus c_{7(i)}. \tag{4}$$

Lines 3 and 4 of Algorithm 1 give us

$$s_7 = ROTL32(s_6 + Y_7[P_7[72]] - Y_7[P_7[239]], P_7[116] + 18 \bmod 32) \tag{5}$$
$$\Rightarrow s_{7(j)} = s_{6(j-k \bmod 32)} \oplus Y_7[P_7[72]]_{j-k \bmod 32} \oplus Y_7^c[P_7[239]]_{j-k \bmod 32} \oplus d_{7(j-k \bmod 32)} \tag{6}$$

where $k = P_7[116] + 18 \bmod 32$, $d_{7(i)} = f_{7(i)} \oplus g_{7(i)}$ and $d_{7(0)} = 1$ (f_7 and g_7 are the carry terms in (5) which are explained in Sect. 5.2). For simplicity, henceforth we denote $X_{(i \bmod 32)}$ by $X_{(i)}$. Thus (6) becomes,

$$s_{7(j)} = s_{6(j-k)} \oplus Y_7[P_7[72]]_{j-k} \oplus Y_7^c[P_7[239]]_{j-k} \oplus d_{7(j-k)}. \tag{7}$$

If $j = i - 25 \bmod 32$, then (7) becomes

$$s_{7(i-25)} = s_{6(i-k-25)} \oplus Y_7[P_7[72]]_{i-k-25} \oplus Y_7^c[P_7[239]]_{i-k-25} \oplus d_{7(i-k-25)}. \tag{8}$$

Substituting (8) in (4), we get,

$$O_{7(i)} = s_{6(i-k-25)} \oplus Y_7[P_7[72]]_{i-k-25} \oplus Y_7^c[P_7[239]]_{i-k-25} \oplus Y_7[256]_i \oplus Y_7[P_7[26]]_i \oplus c_{7(i)} \oplus d_{7(i-k-25)}. \tag{9}$$

Next, we have

$$Y_7[256] = (ROTL32(s_6, 14) \oplus Y_6[-3]) + Y_6[P_6[153]], \tag{10}$$
$$Y_7[256]_i = s_{6(i-14)} \oplus Y_6[-3]_i \oplus Y_6[P_6[153]]_i \oplus e_{7(i)} \tag{11}$$

where e_7 is the carry term in (10). Substituting (11) in (9), we get,

$$O_{7(i)} = s_{6(i-k-25)} \oplus s_{6(i-14)} \oplus Y_7[P_7[72]]_{i-k-25} \oplus Y_7^c[P_7[239]]_{i-k-25} \oplus Y_6[-3]_i \oplus Y_7[P_7[26]]_i \oplus Y_6[P_6[153]]_i \oplus c_{7(i)} \oplus d_{7(i-k-25)} \oplus e_{7(i)}. \tag{12}$$

Now, if $k = -11$ (i.e., $k \equiv -11 \bmod 32 \Rightarrow P_7[116] + 18 \equiv -11 \bmod 32 \Rightarrow P_7[116] \equiv 3 \bmod 32$) then $s_{6(i-k-25)} \oplus s_{6(i-14)} = 0$. Hence, when $P_7[116] \equiv 3 \bmod 32$, (12) becomes

$$O_{7(i)} = Y_7[P_7[72]]_{i-14} \oplus Y_7^c[P_7[239]]_{i-14} \oplus Y_6[-3]_i \oplus Y_7[P_7[26]]_i \oplus Y_6[P_6[153]]_i \oplus c_{7(i)} \oplus d_{7(i-14)} \oplus e_{7(i)}. \tag{13}$$

By similar arguments, when $P_8[116] \equiv 3 \bmod 32$,

$$\begin{aligned} O_{8(i)} &= Y_8[P_8[72]]_{i-14} \oplus Y_8^c[P_8[239]]_{i-14} \oplus Y_7[-3]_i \oplus Y_8[P_8[26]]_i \\ &\quad \oplus Y_7[P_7[153]]_i \oplus c_{8(i)} \oplus d_{8(i-14)} \oplus e_{8(i)}. \end{aligned} \tag{14}$$

From (9), we get

$$\begin{aligned} O_{1(i)} &= s_{0(i-k-25)} \oplus Y_1[P_1[72]]_{i-k-25} \oplus Y_1^c[P_1[239]]_{i-k-25} \oplus Y_1[256]_i \\ &\quad \oplus Y_1[P_1[26]]_i \oplus c_{1(i)} \oplus d_{1(i-k-25)}. \end{aligned} \tag{15}$$

When $k = 0$ (i.e., $P_1[116] \equiv -18 \bmod 32$), the above equation reduces to

$$O_{1(i)} = s_{0(i+7)} \oplus Y_1[P_1[72]]_{i+7} \oplus Y_1^c[P_1[239]]_{i+7} \oplus Y_1[256]_i \oplus Y_1[P_1[26]]_i \oplus c_{1(i)} \oplus d_{1(i+7)}. \tag{16}$$

Similarly, when $P_3[116] \equiv -4 \bmod 32$, we have

$$O_{3(i+7)} = s_{2(i)} \oplus Y_3[P_3[72]]_i \oplus Y_3^c[P_3[239]]_i \oplus Y_3[256]_{i+7} \oplus Y_3[P_3[26]]_{i+7} \oplus c_{3(i+7)} \oplus d_{3(i)}. \tag{17}$$

From (13) and (14), we derive the following results:

$$\begin{aligned} O_{7(i+7)} &= Y_7[P_7[72]]_{i-7} \oplus Y_7^c[P_7[239]]_{i-7} \oplus Y_6[-3]_{i+7} \oplus Y_7[P_7[26]]_{i+7} \oplus Y_6[P_6[153]]_{i+7} \oplus c_{7(i+7)} \\ &\quad \oplus d_{7(i-7)} \oplus e_{7(i+7)}, \end{aligned} \tag{18}$$

$$\begin{aligned} O_{8(i+7)} &= Y_8[P_8[72]]_{i-7} \oplus Y_8^c[P_8[239]]_{i-7} \oplus Y_7[-3]_{i+7} \oplus Y_8[P_8[26]]_{i+7} \oplus Y_7[P_7[153]]_{i+7} \oplus c_{8(i+7)} \\ &\quad \oplus d_{8(i-7)} \oplus e_{8(i+7)}. \end{aligned} \tag{19}$$

This completes the proof. □

Now we state the second lemma.

Lemma 2 $s_{0(i+7)} = s_{2(i)}$ *if the following conditions are simultaneously satisfied,*

1. $P_1[116] \equiv -18 \bmod 32$,
2. $P_2[116] \equiv 7 \bmod 32$,
3. $P_1[72] = P_2[239] + 1$,
4. $P_1[239] = P_2[72] + 1$.

Proof. Equation (5) gives us:

$$s_1 = ROTL32(s_0 + Y_1[P_1[72]] - Y_1[P_1[239]], P_1[116] + 18 \bmod 32).$$

The first condition ($P_1[116] \equiv -18 \bmod 32$) reduces this to

$$s_1 = s_0 + Y_1[P_1[72]] - Y_1[P_1[239]].$$

Therefore,

$$s_2 = ROTL32(s_0 + Y_2[P_2[72]] - Y_2[P_2[239]] + Y_1[P_1[72]] - Y_1[P_1[239]], P_2[116] + 18 \bmod 32).$$

Conditions 3 and 4 reduce the above equation to

$$s_2 = ROTL32(s_0, P_2[116] + 18 \bmod 32).$$

Table 1: Terms generated in $O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)}$, when events E_1 to E_7 simultaneously occur, grouped by their bit positions

Bit position: $i-7$	**Bit position:** i	**Bit position:** $i+7$
$Y_7[P_7[72]]$	$Y_1[256]$	$Y_1[P_1[72]]$
$Y_7[P_7[239]]$	$Y_1[P_1[26]]$	$Y_1[P_1[239]]$
$Y_8[P_8[72]]$	$Y_3[P_3[72]]$	$Y_3[256]$
$Y_8[P_8[239]]$	$Y_3[P_3[239]]$	$Y_3[P_3[26]]$
Carries	Carries	$Y_6[P_6[153]]$
		$Y_6[-3]$
		$Y_7[P_7[26]]$
		$Y_7[P_7[153]]$
		$Y_7[-3]$
		$Y_8[P_8[26]]$
		Carries

Finally, with condition 2 (i.e., $P_2[116] \equiv 7 \bmod 32$), the previous equation becomes

$$\begin{aligned} s_2 &= ROTL32(s_0, 25) \\ \Rightarrow s_{2(i)} &= ROTL32(s_0, 25)_i = s_{0(i-25)} \\ &= s_{0(i+7)}. \end{aligned} \tag{20}$$

This completes the proof. □

Now we observe that, when the conditions listed under (i) Lemma 1 (i.e., events E_1, E_3, E_4 and E_5) and (ii) Lemma 2 (i.e., events E_1, E_2, E_6 and E_7) are simultaneously satisfied, then the expression $O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)}$ is the XOR of the terms which are listed in Table 1 (grouped according to the bit positions).[2] Similarly, the 'carries' in Table 1 are elaborated in Table 2.

Table 2: Carry terms generated in $O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)}$ grouped by their bit positions

Bit position: $i-7$	**Bit position:** i	**Bit position:** $i+7$
d_7	c_1	d_1
d_8	d_3	c_3
		c_7
		e_7
		c_8
		e_8

If the Y-terms in Table 1 are pairwise equated (this is achieved when the events E_8 through to E_{16} occur) then we get

$$\begin{aligned} O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)} &= d_{7(i-7)} \oplus d_{8(i-7)} \oplus c_{1(i)} \oplus d_{3(i)} \oplus d_{1(i+7)} \oplus c_{3(i+7)} \oplus c_{7(i+7)} \oplus e_{7(i+7)} \\ &\oplus c_{8(i+7)} \oplus e_{8(i+7)}. \end{aligned} \tag{21}$$

[2]Note that none of the terms listed in Table 1 is of the form A^c because we used the fact that $A^c \oplus B^c = A \oplus B$ in (16), (17), (18) and (19).

Now, when the RHS of (21) equals zero (i.e., E_{17} occurs) we get

$$O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)} = 0.$$

This completes the proof. □

5 Computation of the Bias

In this section, we quantify the bias in the outputs of TPy induced by the fortuitous events similar to the one described in Sect. 4. Now it is important to note that there may be *more than one set of 17 conditions* possible, where each of them results in $O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)} = 0$ (let us assume that there are n such sets). In Theorem 1, we listed one such set. Our experiments suggest that these n sets are mutually independent, however, a formal proof of that is nontrivial.

Each of the events E_1 to E_5 occurs with approximate probability $\frac{1}{32}$ and each of the events E_6 to E_{16} occurs with probability which is approximately $\frac{1}{256}$. Let p denote the probability that condition 17 is satisfied. Let F denote the event $\bigcap_{j=1}^{16} E_j$. Therefore,

$$P[F] = (\frac{1}{32})^5 \cdot (\frac{1}{256})^{11}.$$

We see that there are n F-like events (i.e., the intersection of 16 conditions). Let F_n denote the union of these n events. Since, each event occurs with approximately the same probability,

$$\begin{aligned} P[F_n] &\approx n \cdot P[F] \\ &\approx n \cdot (\frac{1}{32})^5 \cdot (\frac{1}{256})^{11} \\ &= n \cdot \frac{1}{2^{113}}. \end{aligned}$$

From Table 1, we get the maximum number of ways that terms of a particular column can be pairwise equated and hence the upper bound on n can be calculated to be $2 \cdot 2 \cdot 945 = 3780$, that is, $n < 3780$.

5.1 Formulating the Bias

Now, we establish a formula to compute $P[O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)} = 0]$, under the assumption of a perfectly random key/IV setup and the uniformity of bits when F_n does not occur.Our experiments suggest that it is infeasible to find a set of conditions such that the overall bias (computed on the basis of the aforementioned assumption of randomness in the event that F_n does not occur) is canceled or reduced in magnitude. Therefore, this assumption is reasonable. Let T denote $O_{1(i)} \oplus O_{3(i+7)} \oplus O_{7(i+7)} \oplus O_{8(i+7)}$. Then using Bayes' rule we get

$$\begin{aligned} P[T=0] &= P[T=0|F_n \cap E_{17}] \cdot P[F_n \cap E_{17}] + P[T=0|F_n^c \cup E_{17}^c] \cdot P[F_n^c \cup E_{17}^c] \\ &= P[T=0|F_n \cap E_{17}] \cdot P[F_n \cap E_{17}] + P[T=0|F_n^c \cap E_{17}] \cdot P[F_n^c \cap E_{17}] \\ &+ P[T=0|F_n \cap E_{17}^c] \cdot P[F_n \cap E_{17}^c] + P[T=0|F_n^c \cap E_{17}^c] \cdot P[F_n^c \cap E_{17}^c] \\ &= 1 \cdot (n \cdot p \cdot \frac{1}{2^{113}}) + \frac{1}{2} \cdot (1 - n \cdot \frac{1}{2^{113}}) \cdot p + 0 \cdot P[F_n \cap E_{17}^c] + \frac{1}{2} \cdot (1 - n \cdot \frac{1}{2^{113}}) \cdot (1-p) \\ &= \frac{1}{2} + n \cdot (2p-1) \cdot \frac{1}{2^{114}}. \end{aligned} \tag{22}$$

Hence, we see that the distribution of the outputs $(O_{1(i)}, O_{3(i+7)}, O_{7(i+7)}, O_{8(i+7)})$ is biased. The bias is equal to $n \cdot (2p-1) \cdot \frac{1}{2^{114}}$. In the following section, we provide formulas to compute p, i.e., the probability that E_{17} occurs; or more generally, the probability that the 17th condition of each of the n F-like events occurs, i.e., $P[d_{7(i-7)} \oplus d_{8(i-7)} \oplus c_{1(i)} \oplus d_{3(i)} \oplus d_{1(i+7)} \oplus c_{3(i+7)} \oplus c_{7(i+7)} \oplus e_{7(i+7)} \oplus c_{8(i+7)} \oplus e_{8(i+7)}] = 0$.

5.2 Biases in the Carry Terms

In this section, we provide formulas to calculate the bias in the carry terms. The carry terms c and e are generated in expressions of the form $(S \oplus X) + Z$. We now proceed to calculate $P[c_{l(i)} = 0]$ assuming that S, X and Z are uniformly distributed and independent. Under this assumption, $P[S_i = 0] = P[X_i = 0] = P[Z_i = 0] = \frac{1}{2}$, that is, the probability that the carry bit at position i equals zero depends only on i. Stated otherwise, $P[c_{(i)} = 0] = P[e_{(i)} = 0]$. Let $P[c_{(i)} = 0]$ be denoted by p_i. Since there is no carry on the lsb, $p_0 = 1$. We now have Table 3.

Table 3: Truth table for computing p_i (NR=Not Required)

$c_{(i-1)}$	$S_{(i-1)}$	$X_{(i-1)}$	$Z_{(i-1)}$	$c_{(i)}$	Probability
0	0	0	0	0	$\frac{p_{i-1}}{8}$
0	0	0	1	0	$\frac{p_{i-1}}{8}$
0	0	1	0	0	$\frac{p_{i-1}}{8}$
0	0	1	1	0	$\frac{p_{i-1}}{8}$
0	1	0	0	0	$\frac{p_{i-1}}{8}$
0	1	0	1	1	NR
0	1	1	0	1	NR
0	1	1	1	0	$\frac{p_{i-1}}{8}$
1	0	0	0	0	$\frac{1-p_{i-1}}{8}$
1	0	0	1	1	NR
1	0	1	0	1	NR
1	0	1	1	0	$\frac{1-p_{i-1}}{8}$
1	1	0	0	1	NR
1	1	0	1	1	NR
1	1	1	0	1	NR
1	1	1	1	1	NR

From Table 3, using Bayes' rule we get

$$p_i = \frac{p_{i-1}}{2} + \frac{1}{4}.$$

Solving this recursion, given $p_0 = 1$, we get

$$p_i = \frac{1}{2} + \frac{1}{2^{i+1}}. \tag{23}$$

Now, the carry terms f and g are generated in expressions of the form $S + X - Z$. This can be rewritten as $S + X + Z^c + 1$ since the additions in these two expressions are modulo 2^{32}. The presence of two carries in $S + X + Z$ is demonstrated using the Figure 2. The carries generated in $S + X + Z^c + 1$ can be thought of as carries generated in $S + X + A$ where $A = Z^c$ and the carries on the lsb $f_{(0)} = 1$, $g_{(0)} = 0$. Let q_i denote $P[f_{(i)} = 0]$ and r_i denote $P[g_{(i)} = 0]$. Hence, $q_0 = 0$, $r_0 = 1$ and $r_1 = 1$. Now we have Table 4.

From Table 4, using Bayes' rule we get

$$q_i = \frac{1}{2} + \frac{4 \cdot q_{i-1} \cdot r_{i-1}}{8} - \frac{q_{i-1}}{4} - \frac{r_{i-1}}{4}, \tag{24}$$

$$r_{i+1} = \frac{1}{2} - \frac{q_{i-1} \cdot r_{i-1}}{4} + \frac{3 \cdot q_{i-1}}{8} + \frac{3 \cdot r_{i-1}}{8}. \tag{25}$$

Using the initial conditions, $q_0 = 0$, $r_0 = 1$ and $r_1 = 1$, q_i and r_i are computed recursively. Since $d_{m(i)}$ denotes $f_{m(i)} \oplus g_{m(i)}$ for any $m > 0$,

Table 4: Truth table for computing q_i and r_{i+1} using q_{i-1} and r_{i-1} (NR=Not Required)

$f_{(i-1)}$	$g_{(i-1)}$	$S_{(i-1)}$	$X_{(i-1)}$	$Z_{(i-1)}$	$f_{(i)}$	$g_{(i+1)}$	Probability
0	0	0	0	0	0	0	$\frac{q_{i-1} \cdot r_{i-1}}{8}$
0	0	0	0	1	0	0	$\frac{q_{i-1} \cdot r_{i-1}}{8}$
0	0	0	1	0	0	0	$\frac{q_{i-1} \cdot r_{i-1}}{8}$
0	0	0	1	1	1	0	NR
0	0	1	0	0	0	0	$\frac{q_{i-1} \cdot r_{i-1}}{8}$
0	0	1	0	1	1	0	NR
0	0	1	1	0	1	0	NR
0	0	1	1	1	1	0	NR
0	1	0	0	0	0	0	$\frac{q_{i-1} \cdot (1-r_{i-1})}{8}$
0	1	0	0	1	1	0	NR
0	1	0	1	0	1	0	NR
0	1	0	1	1	1	0	NR
0	1	1	0	0	1	0	NR
0	1	1	0	1	1	0	NR
0	1	1	1	0	1	0	NR
0	1	1	1	1	0	1	$\frac{q_{i-1} \cdot (1-r_{i-1})}{8}$
1	0	0	0	0	0	0	$\frac{(1-q_{i-1}) \cdot r_{i-1}}{8}$
1	0	0	0	1	1	0	NR
1	0	0	1	0	1	0	NR
1	0	0	1	1	1	0	NR
1	0	1	0	0	1	0	NR
1	0	1	0	1	1	0	NR
1	0	1	1	0	1	0	NR
1	0	1	1	1	0	1	$\frac{(1-q_{i-1}) \cdot r_{i-1}}{8}$
1	1	0	0	0	1	0	NR
1	1	0	0	1	1	0	NR
1	1	0	1	0	1	0	NR
1	1	0	1	1	0	1	$\frac{(1-q_{i-1}) \cdot (1-r_{i-1})}{8}$
1	1	1	0	0	1	0	NR
1	1	1	0	1	0	1	$\frac{(1-q_{i-1}) \cdot (1-r_{i-1})}{8}$
1	1	1	1	0	0	1	$\frac{(1-q_{i-1}) \cdot (1-r_{i-1})}{8}$
1	1	1	1	1	0	1	$\frac{(1-q_{i-1}) \cdot (1-r_{i-1})}{8}$

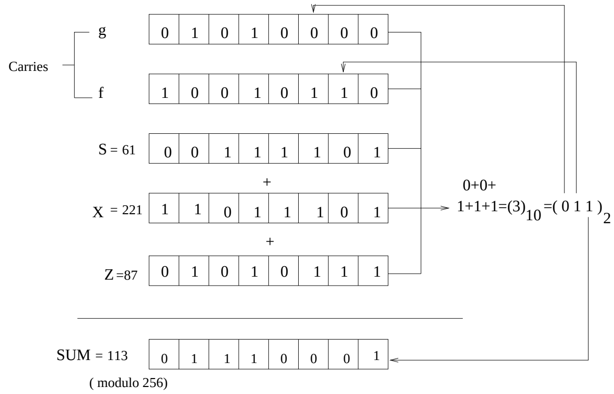

Figure 2: An example showing how the carries are generated when three 8-bit variables $S = 61$, $X = 221$ and $Z = 87$ are added

1. $P[d_{7(i-7)} = 0] = P[d_{8(i-7)} = 0] = q_{i-7 \bmod 32} \cdot r_{i-7 \bmod 32}$ $+ (1 - q_{i-7 \bmod 32}) \cdot (1 - r_{i-7 \bmod 32})$,
2. $P[c_{1(i)} = 0] = \frac{1}{2} + \frac{1}{2^{i+1}}$,
3. $P[d_{3(i)} = 0] = q_i \cdot r_i + (1 - q_i) \cdot (1 - r_i)$,
4. $P[d_{1(i+7)} = 0] = q_{i+7 \bmod 32} \cdot r_{i+7 \bmod 32}$ $+ (1 - q_{i+7 \bmod 32}) \cdot (1 - r_{i+7 \bmod 32})$,
5. $P[c_{3(i+7)} = 0] = P[c_{7(i+7)} = 0] = P[e_{7(i+7)} = 0] = P[c_{8(i+7)} = 0]$ $= P[e_{8(i+7)} = 0] = \frac{1}{2} + \frac{1}{2^{(i+7 \bmod 32)+1}}$.

Using the above formulas, the value of p can be computed for any given i. Running simulation, we find that the maximum bias in the chosen outputs occurs when $i = 25$ which corresponds to $p = 0.5 - 2^{-34.2}$. Hence, (22) gives us

$$P[T = 0] = \frac{1}{2} - \frac{n}{2^{147.2}}$$
$$\Rightarrow P[T = 1] = \frac{1}{2} + \frac{n}{2^{147.2}},$$

when $i = 25$. Substituting $n = 3780$ in the above equation, we get:

$$P[T = 1] = \frac{1}{2} + \frac{1}{2^{135.3}}. \tag{26}$$

This is an upper bound on the probability that the outputs $(O_{1(i)}, O_{3(i+7)}, O_{7(i+7)}, O_{8(i+7)})$ of TPy are biased. From Sect. 4, we found that $n \geq 1$. From the previous discussion, we see that $n < 3780$. Hence, $1 \leq n < 3780$. If $n = 1$, then $P[T = 1] = \frac{1}{2} + \frac{1}{2^{147.2}}$. Thus,

$$\frac{1}{2}(1 + \frac{1}{2^{146.2}}) \leq P[T = 1] \leq \frac{1}{2}(1 + \frac{1}{2^{134.3}}). \tag{27}$$

6 The Distinguisher

A distinguisher is an algorithm which distinguishes a given stream of bits from a stream of bits generated by a perfect PRBG. The distinguisher is constructed by collecting sufficiently many outputs $(O_{1(25)}, O_{3(0)}, O_{7(0)}, O_{8(0)})$ generated by as many key/IVs. To compute the minimum number of samples required to establish the distinguisher, we use the following corollary of a theorem from [6].

Corollary 1 *If an event e occurs in a distribution X with probability p and in Y with probability* $p(1+q)$ *then, if* $p = \frac{1}{2}$, $O(\frac{1}{q^2})$ *samples are required to distinguish X from Y with non-negligible probability of success.*

In the present case, e is the event $O_{1(25)} \oplus O_{3(0)} \oplus O_{7(0)} \oplus O_{8(0)} = 0$, X is the distribution of the outputs O_1, O_3, O_7 and O_8 produced by a perfectly random keystream generator and Y is the distribution of the outputs produced by TPy. From (27), $p = \frac{1}{2}$ and the highest value of $q = \frac{1}{2^{134.3}}$. Hence $O(\frac{1}{(2^{-134.3})^2}) = O(2^{268.6})$ output samples are needed to construct the best distinguisher with a non-negligible probability of success. Note that this is an improvement by a factor of $2^{12.4}$ over the data complexity of 2^{281} obtained in [9].

7 A Family of Distinguishers

In Sect. 4 we found that the outputs at rounds 1, 3, 7 and 8 are biased allowing us to build a distinguisher. It is found that there exist plenty of 4-tuples of biased outputs. The generalization is presented in the following theorem.

Theorem 2 *The distribution of the outputs* $(O_{r(i)}, O_{r+2(i+7)}, O_{t(i+7)}, O_{u(i+7)})$ *of the TPy are biased for many suitably chosen (r, t, u)'s where* $r > 0$; $t, u \geq 5$; $t \notin \{r, r+2, u\}$; $u \notin \{r, r+2, t\}$.

The proof is similar to the proof furnished for Theorem 1, however, a detailed proof has been provided in the Appendix A. This allows us to construct a family of distinguishers for the cipher TPy. It seems possible to combine these huge number of distinguishers in order to construct one single efficient distinguisher; however, any concrete mathematical model to combine them is still an interesting open problem. Another major implication of the above generalization theorem is the fact that the TPy outputs will remain always biased no matter how many initial outputwords are discarded from the keystream.

8 Attacks on Py

The PRBG of the cipher Py is identical with that of TPy. The attacks described in the previous sections exploit the weaknesses in the PRBG of TPy only. Therefore, all the attacks are applicable to Py also.

9 Conclusions and Open Problems

The paper develops a family of distinguishers from the outputs $(O_{r(i)}, O_{r+2(i+7)}, O_{t(i+7)}, O_{u(i+7)})$ of TPy (and Py), where $r > 0$; $t, u \geq 5$; $t \notin \{r, r+2, u\}$; $u \notin \{r, r+2, t\}$. Note that the TPy is one of the strongest members of the Py-family of ciphers. The best distinguisher works with data complexity $2^{268.6}$ which records an improvement of a factor of 5404 over the previous attack. In addition, we detect a large number of bias-producing states of TPy and compute them in a general framework. It is reasonable to assume that these weak states can be combined to mount a more efficient attack on TPy; however, methods to combine many distinguishers into a single yet more efficient one is still an open problem. We were unable to find the exact value of the bias in the distribution of the outputs $(O_{1(25)}, O_{3(0)}, O_{7(0)}, O_{8(0)})$. We leave this as an open problem.

References

[1] E. Biham, J. Seberry, "Tweaking the IV Setup of the Py Family of Ciphers – The Ciphers Tpy, TPypy, and TPy6," Published on the author's webpage at `http://www.cs.technion.ac.il/ biham/`, January 25, 2007.

[2] E. Biham, J. Seberry, "Py (Roo): A Fast and Secure Stream Cipher using Rolling Arrays," *ecrypt submission*, 2005.

[3] E. Biham, J. Seberry, "Pypy (Roopy): Another Version of Py," *ecrypt submission*, 2006.

[4] P. Crowley, "Improved Cryptanalysis of Py," *Workshop Record of SASC 2006 - Stream Ciphers Revisited*, ECRYPT Network of Excellence in Cryptology, February 2006, Leuven (Belgium), pp. 52-60.

[5] T. Isobe, T. Ohigashi, H. Kuwakado M. Morii, "How to Break Py and Pypy by a Chosen-IV Attack," eSTREAM, ECRYPT Stream Cipher Project, Report 2006/060.

[6] I. Mantin, A. Shamir, "A Practical Attack on Broadcast RC4," *Fast Software Encryption 2001* (M. Matsui, ed.), vol. 2355 of *LNCS*, pp. 152-164, Springer-Verlag, 2001.

[7] S. Paul, B. Preneel, G. Sekar, "Distinguishing Attacks on the Stream Cipher Py," *Fast Software Encryption 2006* (M. Robshaw, ed.), vol. 4047 of *LNCS*, pp. 405-421, Springer-Verlag, 2006.

[8] S. Paul, B. Preneel "On the (In)security of Stream Ciphers Based on Arrays and Modular Addition," *Asiacrypt 2006* (X. Lai and K. Chen, eds.), vol. 4047 of *LNCS*, pp. 69-83, Springer-Verlag, 2006.

[9] G. Sekar, S. Paul, B. Preneel, "Weaknesses in the Pseudorandom Bit Generation Algorithms of the Stream Ciphers TPypy and TPy," available at `http://eprint.iacr.org/2007/075.pdf`.

[10] H. Wu, B. Preneel, "Differential Cryptanalysis of the Stream Ciphers Py, Py6 and Pypy," Eurocrypt 2007 (to appear).

A Proof of the Theorem 2

Claim 1 *The distribution of the outputs* ($O_{r(i)}$, $O_{r+2(i+7)}$, $O_{t(i+7)}$, $O_{u(i+7)}$) *of the TPy are biased for many suitably chosen* (r, t, u)*'s where* $r > 0$; $t, u \geq 5$; $t \notin \{r, r+2, u\}$; $u \notin \{r, r+2, t\}$.

Proof. First, we state and prove two lemmata which will be used to establish the theorem.

Lemma 3 *If*

1. $P_r[116] \equiv -18 \bmod 32$,
2. $P_{r+2}[116] \equiv -4 \bmod 32$,
3. $P_t[116] \equiv 3 \bmod 32$,
4. $P_u[116] \equiv 3 \bmod 32$

then the following equations are satisfied:

1. $O_{r(i)} = s_{r-1(i+7)} \oplus Y_r[P_r[72]]_{i+7} \oplus Y_r^c[P_r[239]]_{i+7} \oplus Y_r[256]_i \oplus Y_r[P_r[26]]_i \oplus c_{r(i)} \oplus d_{r(i+7)}$,
2. $O_{r+2(i+7)} = s_{r+1(i)} \oplus Y_{r+2}[P_{r+2}[72]]_i \oplus Y_{r+2}^c[P_{r+2}[239]]_i \oplus Y_{r+2}[256]_{i+7} \oplus Y_{r+2}[P_{r+2}[26]]_{i+7}$
$\oplus c_{r+2(i+7)} \oplus d_{r+2(i)}$,
3. $O_{t(i+7)} = Y_t[P_t[72]]_{i-7} \oplus Y_t^c[P_t[239]]_{i-7} \oplus Y_{t-1}[-3]_{i+7} \oplus Y_t[P_t[26]]_{i+7} \oplus Y_{t-1}[P_{t-1}[153]]_{i+7} \oplus c_{t(i+7)}$
$\oplus d_{t(i-7)} \oplus e_{t(i+7)}$,
4. $O_{u(i+7)} = Y_u[P_u[72]]_{i-7} \oplus Y_u^c[P_u[239]]_{i-7} \oplus Y_{u-1}[-3]_{i+7} \oplus Y_u[P_u[26]]_{i+7} \oplus Y_{u-1}[P_{u-1}[153]]_{i+7} \oplus$
$c_{u(i+7)}$
$\oplus d_{u(i-7)} \oplus e_{u(i+7)}$.

Proof. Line 5 of Algorithm 1 gives

$$O_t = (ROTL32(s_t, 25) \oplus Y_t[256]) + Y_t[P_t[26]], \tag{28}$$

Let c_t denote the carry in the above equation. Since $ROTL32(s_t, 25)_i = s_{t(i-25)}$,

$$O_{t(i)} = s_{t(i-25)} \oplus Y_t[256]_i \oplus Y_t[P_t[26]]_i \oplus c_{t(i)}. \tag{29}$$

Lines 3 and 4 of Algorithm 1 give us

$$s_t = ROTL32(s_{t-1} + Y_t[P_t[72]] - Y_t[P_t[239]], P_t[116] + 18 \bmod 32), \tag{30}$$

$$\Rightarrow s_{t(j)} = s_{t-1(j-k)} \oplus Y_t[P_t[72]]_{j-k} \oplus Y_t^c[P_t[239]]_{j-k} \oplus d_{t(j-k)} \tag{31}$$

where $k = P_t[116] + 18 \bmod 32$, $d_{t(i)} = f_{t(i)} \oplus g_{t(i)}$ and $d_{t(0)} = 1$ (f_t and g_t are the carry terms in (30). If $j = i - 25 \bmod 32$, then (31) becomes

$$s_{t(i-25)} = s_{t-1(i-k-25)} \oplus Y_t[P_t[72]]_{i-k-25} \oplus Y_t^c[P_t[239]]_{i-k-25} \oplus d_{t(i-k-25)}. \tag{32}$$

Substituting (32) in (29), we get,

$$O_{t(i)} = s_{t-1(i-k-25)} \oplus Y_t[P_t[72]]_{i-k-25} \oplus Y_t^c[P_t[239]]_{i-k-25} \oplus Y_t[256]_i \oplus Y_t[P_t[26]]_i \oplus c_{t(i)} \oplus d_{t(i-k-25)}. \tag{33}$$

Next, we have

$$\begin{aligned} Y_t[256] &= (ROTL32(s_{t-1}, 14) \oplus Y_{t-1}[-3]) + Y_{t-1}[P_{t-1}[153]], & (34)\\ Y_t[256]_i &= s_{t-1(i-14)} \oplus Y_{t-1}[-3]_i \oplus Y_{t-1}[P_{t-1}[153]]_i \oplus e_{t(i)} & (35) \end{aligned}$$

where e_t is the carry term in (34). Substituting (35) in (33), we get,

$$\begin{aligned} O_{t(i)} = {} & s_{t-1(i-k-25)} \oplus s_{t-1(i-14)} \oplus Y_t[P_t[72]]_{i-k-25} \oplus Y_t^c[P_t[239]]_{i-k-25} \oplus Y_{t-1}[-3]_i \\ & \oplus Y_t[P_t[26]]_i \oplus Y_{t-1}[P_{t-1}[153]]_i \oplus c_{t(i)} \oplus d_{t(i-k-25)} \oplus e_{t(i)}. \end{aligned} \tag{36}$$

Now, if $k = -11$ (i.e., $k \equiv -11 \bmod 32 \Rightarrow P_t[116] + 18 \equiv -11 \bmod 32 \Rightarrow P_t[116] \equiv 3 \bmod 32$) then $s_{t-1(i-k-25)} \oplus s_{t-1(i-14)} = 0$. Hence, when $P_t[116] \equiv 3 \bmod 32$, (36) becomes

$$\begin{aligned} O_{t(i)} = {} & Y_t[P_t[72]]_{i-14} \oplus Y_t^c[P_t[239]]_{i-14} \oplus Y_{t-1}[-3]_i \oplus Y_t[P_t[26]]_i \\ & \oplus Y_{t-1}[P_{t-1}[153]]_i \oplus c_{t(i)} \oplus d_{t(i-14)} \oplus e_{t(i)}. \end{aligned} \tag{37}$$

By similar arguments, when $P_u[116] \equiv 3 \bmod 32$,

$$\begin{aligned} O_{u(i)} = {} & Y_u[P_u[72]]_{i-14} \oplus Y_u^c[P_u[239]]_{i-14} \oplus Y_{u-1}[-3]_i \oplus Y_u[P_u[26]]_i \\ & \oplus Y_{u-1}[P_{u-1}[153]]_i \oplus c_{u(i)} \oplus d_{u(i-14)} \oplus e_{u(i)}. \end{aligned} \tag{38}$$

From (33), we get

$$\begin{aligned} O_{r(i)} = {} & s_{r-1(i-k-25)} \oplus Y_r[P_r[72]]_{i-k-25} \oplus Y_r^c[P_r[239]]_{i-k-25} \oplus Y_r[256]_i \\ & \oplus Y_r[P_r[26]]_i \oplus c_{r(i)} \oplus d_{r(i-k-25)}. \end{aligned} \tag{39}$$

When $k = 0$ (i.e., $P_r[116] \equiv -18 \bmod 32$), the above equation reduces to

$$\begin{aligned} O_{r(i)} = {} & s_{r-1(i+7)} \oplus Y_r[P_r[72]]_{i+7} \oplus Y_r^c[P_r[239]]_{i+7} \oplus Y_r[256]_i \oplus Y_r[P_r[26]]_i \\ & \oplus c_{r(i)} \oplus d_{r(i+7)}. \end{aligned} \tag{40}$$

Similarly, when $P_{r+2}[116] \equiv -4 \bmod 32$, we have

$$\begin{aligned} O_{r+2(i+7)} &= s_{r+1(i)} \oplus Y_{r+2}[P_{r+2}[72]]_i \oplus Y^c_{r+2}[P_{r+2}[239]]_i \oplus Y_{r+2}[256]_{i+7} \\ &\oplus Y_{r+2}[P_{r+2}[26]]_{i+7} \oplus c_{r+2(i+7)} \oplus d_{r+2(i)}. \end{aligned} \tag{41}$$

From (37) and (38), we derive the following results:

$$\begin{aligned} O_{t(i+7)} &= Y_t[P_t[72]]_{i-7} \oplus Y^c_t[P_t[239]]_{i-7} \oplus Y_{t-1}[-3]_{i+7} \oplus Y_t[P_t[26]]_{i+7} \\ &\oplus Y_{t-1}[P_{t-1}[153]]_{i+7} \oplus c_{t(i+7)} \oplus d_{t(i-7)} \oplus e_{t(i+7)}, \end{aligned} \tag{42}$$

$$\begin{aligned} O_{u(i+7)} &= Y_u[P_u[72]]_{i-7} \oplus Y^c_u[P_u[239]]_{i-7} \oplus Y_{u-1}[-3]_{i+7} \oplus Y_u[P_u[26]]_{i+7} \\ &\oplus Y_{u-1}[P_{u-1}[153]]_{i+7} \oplus c_{u(i+7)} \oplus d_{u(i-7)} \oplus e_{u(i+7)}. \end{aligned} \tag{43}$$

This completes the proof. □

Now we state the second lemma.

Lemma 4 *For $r > 0$, $s_{r-1(i+7)} = s_{r+1(i)}$ if the following conditions are simultaneously satisfied,*

1. $P_r[116] \equiv -18 \bmod 32$,
2. $P_{r+1}[116] \equiv 7 \bmod 32$,
3. $P_r[72] = P_{r+1}[239] + 1$,
4. $P_r[239] = P_{r+1}[72] + 1$.

Table 5: Terms generated in $O_{r(i)} \oplus O_{r+2(i+7)} \oplus O_{t(i+7)} \oplus O_{u(i+7)}$, when the conditions listed under Lemma 3 and Lemma 4 are simultaneously satisfied, grouped by their bit positions

Bit position: $i-7$	**Bit position:** i	**Bit position:** $i+7$
$Y_t[P_t[72]]$	$Y_r[256]$	$Y_r[P_r[72]]$
$Y_t[P_t[239]]$	$Y_r[P_r[26]]$	$Y_r[P_r[239]]$
$Y_u[P_u[72]]$	$Y_{r+2}[P_{r+2}[72]]$	$Y_{r+2}[256]$
$Y_u[P_u[239]]$	$Y_{r+2}[P_{r+2}[239]]$	$Y_{r+2}[P_{r+2}[26]]$
Carries	Carries	$Y_{t-1}[P_{t-1}[153]]$
		$Y_{t-1}[-3]$
		$Y_t[P_t[26]]$
		$Y_{u-1}[P_{u-1}[153]]$
		$Y_{u-1}[-3]$
		$Y_u[P_u[26]]$
		Carries

Proof. Equation (30) gives us:

$$s_r = ROTL32(s_{r-1} + Y_r[P_r[72]] - Y_r[P_r[239]], P_r[116] + 18 \bmod 32).$$

The first condition ($P_r[116] \equiv -18 \bmod 32$) reduces this to

$$s_r = s_{r-1} + Y_r[P_r[72]] - Y_r[P_r[239]].$$

Therefore,

$$s_{r+1} = ROTL32(s_{r-1} + Y_{r+1}[P_{r+1}[72]] - Y_{r+1}[P_{r+1}[239]] + Y_r[P_r[72]] - Y_r[P_r[239]], P_{r+1}[116] + 18 \bmod 32).$$

Conditions 3 and 4 reduce the above equation to

$$s_{r+1} \quad = \quad ROTL32(s_{r-1}, P_{r+1}[116] + 18 \bmod 32).$$

Finally, with condition 2 (i.e., $P_{r+1}[116] \equiv 7 \bmod 32$), the previous equation becomes

$$\begin{aligned} s_{r+1} &= ROTL32(s_{r-1}, 25) \\ \Rightarrow s_{r+1(i)} &= ROTL32(s_{r-1}, 25)_i = s_{r-1(i-25)} = s_{r-1(i+7)}. \end{aligned} \tag{44}$$

This completes the proof. □

Now we observe that, when the conditions listed under Lemma 3 and Lemma 4 are simultaneously satisfied, then the expression $O_{r(i)} \oplus O_{r+2(i+7)} \oplus O_{t(i+7)} \oplus O_{u(i+7)}$ is the XOR of the terms which are listed in Table 5 (grouped according to the bit positions).[3] Similarly, the 'carries' in Table 5 are elaborated in Table 6. If the Y-terms in Table 5 are pairwise equated, we get

Table 6: Carry terms generated in $O_{r(i)} \oplus O_{r+2(i+7)} \oplus O_{t(i+7)} \oplus O_{u(i+7)}$ grouped by their bit positions

Bit position: $i-7$	**Bit position:** i	**Bit position:** $i+7$
d_t	c_r	d_r
d_u	d_{r+2}	c_{r+2}
		c_t
		e_t
		c_u
		e_u

$$\begin{aligned} O_{r(i)} \oplus O_{r+2(i+7)} \oplus O_{t(i+7)} \oplus O_{u(i+7)} \quad = \quad & d_{t(i-7)} \oplus d_{u(i-7)} \oplus c_{r(i)} \oplus d_{r+2(i)} \oplus d_{r(i+7)} \\ & \oplus c_{r+2(i+7)} \oplus c_{t(i+7)} \oplus e_{t(i+7)} \oplus c_{u(i+7)} \oplus e_{u(i+7)}. \end{aligned} \tag{45}$$

Now, when the RHS of (45) equals zero, we get

$$O_{r(i)} \oplus O_{r+2(i+7)} \oplus O_{t(i+7)} \oplus O_{u(i+7)} = 0.$$

For a particular set of (r, t, u), we can have a set of 17 conditions similar to the set of conditions listed under Theorem 1, where $r = 1$, $t = 7$ and $u = 8$. In this way we can generate arbitrarily many (r, t, u)'s such that the outputs at rounds r, $r+2$, t and u are biased. This completes the proof. □

[3] Note that none of the terms listed in Table 5 is of the form A^c because we used the fact that $A^c \oplus B^c = A \oplus B$ in (40), (41), (42) and (43).