Weaknesses in the Pseudorandom Bit Generation Algorithms of the Stream Ciphers TPypy and TPy[⋆]

Gautham Sekar, Souradyuti Paul, and 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 were designed by Biham and Seberry for the ECRYPT-eSTREAM project in 2005. However, due to several recent cryptanalytic attacks on them, a strengthened version Pypy was proposed to rule out those attacks. The ciphers have been promoted to the 'Focus' ciphers of the Phase II of the eSTREAM project. The impressive speed of the ciphers make them the forerunners in the competition. Unfortunately, even the new cipher Pypy was found to retain weaknesses, forcing the designers to again go for modifications. As a result, three new ciphers TPypy, TPy and TPy6 were built. Among all the members of the Py-family of ciphers, the TPypy is conjectured to be the strongest. So far, there is no known attack on the TPypy. This paper shows that the security of TPypy does not grow exponentially with the key-size. The main achievement of the paper is the detection of input-output correlations of TPypy that allow us to build a distinguisher with 2^{281} output words and comparable time. The cipher TPypy was claimed by the designers to be secure with keysize up to 256 bytes, i.e., 2048 bits. Our results establish that the TPypy fails to provide adequate security if the keysize is longer than 35 bytes, i.e., 280 bits. Because of remarkable similarities between the TPypy and the TPy, our attacks are shown to be effective for TPy also. The paper also points out how the other members of the Py-family (i.e., TPy6, Py6, Pypy and Py6) are also weak against the current and some existing attacks.

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

[⋆] This work was supported in part by the Concerted Research Action (GOA) Mefisto 2000/06 and Ambiorix 2005/11 of the Flemish Government and in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

category of software based stream ciphers [1]. 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.

- **March 2006 (at FSE 2006).** Paul, Preneel and Sekar reported distinguishing attacks with $2^{89.2}$ data and comparable time against the cipher Py [8]. Crowley [4] later reduced the complexity to 2^{72} by employing a Hidden Markov Model.
- **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 [2].
- **May 2006 (presented at Asiacrypt 2006).** Distinguishing attacks were reported against Py6 with 2^{68} data and comparable time by Paul and Preneel [9].
- **October 2006 (to be presented at Eurocrypt 2007).** Wu and Preneel showed key recovery attacks against the ciphers Py, Pypy, Py6 with chosen IVs. This attack was subsequently improved by Isobe *et al.* [5].
- **January 2007.** Three new ciphers TPypy, TPy, TPy6 were proposed by the designers [3]. 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. So far there exist no attacks on TPypy, TPy and TPy6.

Contribution of the paper. From the previous discussion, the list that orders the Py-family of ciphers in terms of increasing security is: Py6→Py→ Pypy → TPy6 → TPy → TPypy. In this paper we build a distinguishing attack on the strongest member of the Py-family of cipher TPypy with 2^{281} data. The TPypy is normally used with a 32-byte key and a 16-byte initial value (or IV). However, the key size may vary from 1 byte to 256 bytes and the IV can be of any size from 1 byte to 64 bytes. When TPypy is used with key of size longer than 35 bytes (or, 280 bits), our attack is better than the exhaustive key search and therefore, constitutes an academic break of the cipher. Ideally the security of a stream cipher should increase exponentially with the key-size. Our major contribution is that, for the TPypy (as well as TPy, TPy6, Pypy and Py) the security does not grow exponentially beyond 35 bytes. These weaknesses result in the *first attacks* on TPypy, TPy and TPy6.

For Pypy and Py, the attacks outlined in this paper are also valid with data and time complexities 2^{281}. It is now very important to make a distinction between the current attacks and previous attacks on Pypy and Py.

Our attack on Pypy (and also Py) and the attack on it by Wu and Preneel. Recently, Wu and Preneel [10] have reported a key recovery attack on Pypy based on the weaknesses of its IV setup [11]. There are three major limitations of their attack: (i) the attack does not work if the IV size is shorter than 10 bytes; (ii) the attack-model assumes a powerful adversary who can control the IVs, something which is difficult to implement in practice. On the other hand, the attack described in this paper does not depend on any of the above constraints. More precisely, the attacks described in the paper work (i) for IV of any length, even if (ii) the adversary has no control over the IVs.

In Sect. 4, we detect correlation between the inputs and the outputs of the TPypy at rounds 1, 3, 5 and 9. The correlation gives rise to a bias in the output distribution which is quantified in Sect. 5. In Sect. 6, we generalize the results of Sect. 4. The construction of a distinguisher is shown in Sect. 7. Finally, we conclude with a discussion on (1) the applicability of these attacks on *all* other Py-family of ciphers, (2) an implication of the attacks when combined with those of [8] and (3) the possibility of the existence of stronger distinguishers for all the Py-family of ciphers by combining the various biases.

2 The Round Function of TPypy and TPy

The TPypy and TPy use the same initialization, that is, their key setup and IV setup algorithms are identical [3]. Algorithm 1 describes one round of TPypy. The only difference is in the round function – TPy generates two outputwords (lines 5 and 6), each of 4 bytes, in every round where TPypy outputs only one of them (line 6). We assume that the key/IV setups of both TPypy and TPy generate perfectly random outputs. The round function takes as inputs the array P (which is a permutation of the elements of the set $\{0, ..., 255\}$), the array Y (which contains 260 elements (each element is a 32-bit integer) and a 32-bit variable s. The operation 'rotate(X)' implies a cyclic rotation of the elements of array X by one position. The '$ROTL32(s, n)$' function cyclically rotates the variable s to the left by n positions (see Fig. 1).

Algorithm 1 A Round of TPypy and TPy

```
Input: $Y[-3, ..., 256]$, $P[0, ..., 255]$, a 32-bit variable $s$
Output: 64-bit random output
    /*Update and rotate $P$*/
 1: swap ($P[0]$, $P[Y[185]\&255]$);
 2: rotate ($P$);
    /* Update s*/
 3: $s+ = Y[P[72]] - Y[P[239]]$;
 4: $s = ROTL32(s, ((P[116] + 18)\&31))$;
    /* Output 4 or 8 bytes (least significant byte first)*/
 5: output $((ROTL32(s, 25) \oplus Y[256]) + Y[P[26]])$;/* This step is skipped for TPypy*/
 6: output $((\quad s \quad \oplus Y[-1]) + Y[P[208]])$;
    /* Update and rotate $Y$*/
 7: $Y[-3] = (ROTL32(s, 14) \oplus Y[-3]) + Y[P[153]]$;
 8: rotate($Y$);
```

3 Notation and Convention

We follow the same convention as described in [8]. The $O_{i(j)}$ denotes the jth bit (where $j = 0$ denotes the lsb) of the outputword generated in round i. We denote the arrays obtained after the key/IV setup by P_0, Y_1 and s_0. This ensures uniform indices in the formula for the output generated at round 1. For example, when the above convention is followed, $O_1 = (s_1 \oplus Y_1[-3]) + Y_1[P_1[153]]$. Hence at the beginning of step m, we have P_m, Y_{m+1} and s_m. Next, the $Y_m[k]$ and the $P_m[k]$ denote the kth elements of the arrays Y_m and P_m respectively. The $Y_m[k]_j$, the $P_m[k]_j$ denote the jth bit (where $j = 0$ denotes the lsb) of $Y_m[k]$, $P_m[k]$ respectively. The operators '+' and '−' denote *addition modulo* 2^{32} and *subtraction modulo* 2^{32} respectively, except when used in expressions of the form $P_m[k] = P_n[l] \pm x$, where they denote *addition and subtraction over* $\mathbb{Z}$. The symbol '$\oplus$' denotes bitwise *exclusive-or.*

4 Motivational Observation

Our principal observation is the detection of a relation between inputs and outputs of TPypy (and hence TPy) which is formulated in the theorem below. Such types of weaknesses are sometimes difficult to eliminate from the stream ciphers based on arrays and modular addition as analyzed by Paul in his Ph.D. thesis [7].

Theorem 1. *$O_{1(0)} \oplus O_{3(0)} \oplus O_{5(0)} \oplus O_{9(0)} = 0$ if the following 14 conditions on the elements of P and Y are simultaneously satisfied.*

1. $P_2[116] \equiv -18 (\bmod 32)$ *(event E_1),*
2. $P_3[116] \equiv 0 (\bmod 32)$ *(event E_2) or* $P_3[116] \equiv -18 (\bmod 32)$ *(event E'_2),*
3. $P_4[116] \equiv -18 (\bmod 32)$ *(event E_3),*
4. $P_5[116] \equiv 0 (\bmod 32)$ *when* $P_3[116] \equiv 0 (\bmod 32)$ *occurs (event E_4) or* $P_5[116] \equiv -18 (\bmod 32)$ *when* $P_3[116] \equiv -18 (\bmod 32)$ *occurs (event E'_4),*
5. $P_6[116] \equiv -18 (\bmod 32)$ *(event E_5),*
6. $P_7[116] \equiv -18 (\bmod 32)$ *(event E_6),*
7. $P_8[116] \equiv -18 (\bmod 32)$ *(event E_7),*
8. $P_9[116] \equiv -18 (\bmod 32)$ *(event E_8),*
9. $P_2[72] = P_3[239] + 1$ *(event E_9),*
10. $P_2[239] = P_3[72] + 1$ *(event E_{10}),*
11. $P_4[72] = P_5[239] + 1$ *(event E_{11}),*
12. $P_4[239] = P_5[72] + 1$ *(event E_{12}),*
13. $\sum_{i=6}^{9} (Y_i[P_i[72]] - Y_i[P_i[239]]) = 0$ *(event E_{13}),*
14. $Y_9[P_9[208]]_0 \oplus Y_5[P_5[153]]_0 \oplus Y_5[P_5[208]]_0 \oplus Y_3[P_3[153]]_0 \oplus Y_3[P_3[208]]_0 \oplus Y_1[P_1[208]]_0 \oplus Y_6[256]_0 \oplus Y_4[256]_0 \oplus Y_3[5]_0 \oplus Y_1[3]_0 = 0$ *(event E_{14}).*

Proof. From line 6 of Algorithm 1, it is found that

$$O_1 = (s_1 \oplus Y_1[-1]) + Y_1[P_1[208]] \tag{1}$$

$$\Rightarrow O_{1(0)} = s_{1(0)} \oplus Y_1[-1]_0 \oplus Y_1[P_1[208]]_0 \,. \tag{2}$$

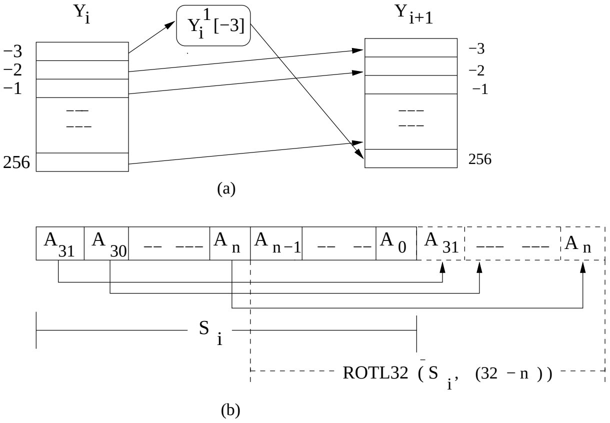

Fig. 1. (a) $Y_i[k] = Y_{i+1}[k-1]$ for $-2 \leq k \leq 256$; $Y_{i+1}[256] = Y_i^1[-3]$ when $k = -3$ $(Y_i^1[-3] = (ROTL32(s_i, 14) \oplus Y_i[-3]) + Y_i[P_i[153]])$; (b) s_i and $ROTL32(s_i, (32-n))$ where $0 \leq n \leq 31$

From line 7 of Algorithm 1 and Fig. 1, it is obtained that

$$Y_4[256] = (ROTL32(s_3, 14) \oplus Y_3[-3]) + Y_3[P_3[153]] \tag{3}$$

$$\Rightarrow Y_4[256]_0 = ROTL32(s_3, 14)_0 \oplus Y_3[-3]_0 \oplus Y_3[P_3[153]_0\,. \tag{4}$$

But, $ROTL32(s_3, 14)_0 = s_{3(18)}$ and $Y_3[-3] = Y_1[-1] \Rightarrow Y_3[-3]_0 = Y_1[-1]_0$. Substituting these two results in (4) and rearranging the terms we get

$$Y_1[-1]_0 = s_{3(18)} \oplus Y_4[256]_0 \oplus Y_3[P_3[153]]_0\,. \tag{5}$$

Putting (5) in (2) gives

$$O_{1(0)} = s_{1(0)} \oplus s_{3(18)} \oplus Y_4[256]_0 \oplus Y_3[P_3[153]]_0 \oplus Y_1[P_1[208]]_0. \tag{6}$$

Similarly,

$$O_{3(0)} = s_{3(0)} \oplus s_{5(18)} \oplus Y_6[256]_0 \oplus Y_5[P_5[153]]_0 \oplus Y_3[P_3[208]]_0\,. \tag{7}$$

Looking at (2), one can write the formulas for $O_{5(0)}$ and $O_{9(0)}$ as

$$O_{5(0)} = s_{5(0)} \oplus Y_5[-1]_0 \oplus Y_5[P_5[208]]_0,$$
$$O_{9(0)} = s_{9(0)} \oplus Y_9[-1]_0 \oplus Y_9[P_9[208]]_0.$$

Again from Fig. 1, we have

$$Y_5[-1] = Y_1[3] \Rightarrow Y_5[-1]_0 = Y_1[3]_0,$$
$$Y_9[-1] = Y_3[5] \Rightarrow Y_9[-1]_0 = Y_3[5]_0.$$

Therefore,

$$O_{5(0)} = s_{5(0)} \oplus Y_1[3]_0 \oplus Y_5[P_5[208]]_0, \tag{8}$$
$$O_{9(0)} = s_{9(0)} \oplus Y_3[5]_0 \oplus Y_9[P_9[208]]_0. \tag{9}$$

From (6), (7), (8) and (9),

$$\begin{aligned} O_{1(0)} \oplus O_{3(0)} \oplus O_{5(0)} \oplus O_{9(0)} = {} & s_{1(0)} \oplus s_{3(18)} \oplus s_{3(0)} \oplus s_{5(18)} \oplus s_{5(0)} \oplus s_{9(0)} \\ & \oplus Y_1[P_1[208]]_0 \oplus Y_3[P_3[153]]_0 \oplus Y_3[P_3[208]]_0 \\ & \oplus Y_5[P_5[153]]_0 \oplus Y_5[P_5[208]]_0 \oplus Y_9[P_9[208]]_0 \\ & \oplus Y_1[3]_0 \oplus Y_3[5]_0 \oplus Y_4[256]_0 \oplus Y_6[256]_0. \end{aligned}$$

The formulas for s_2 and s_3 are given below (see Algorithm 1):

$$s_2 = ROTL32(s_1 + Y_2[P_2[72]] - Y_2[P_2[239]], P_2[116] + 18 \bmod 32), \tag{10}$$
$$s_3 = ROTL32(s_2 + Y_3[P_3[72]] - Y_3[P_3[239]], P_3[116] + 18 (\bmod 32). \tag{11}$$

Condition 1 (i.e., $P_2[116] \equiv -18 \bmod 32$) reduces (10) to

$$s_2 = s_1 + Y_2[P_2[72]] - Y_2[P_2[239]].$$

Therefore, (11) becomes

$$s_3 = ROTL32(s_1 + \sum_{i=2}^{3}(Y_i[P_i[72]] - Y_i[P_i[239]]), P_3[116] + 18 \bmod 32). \tag{12}$$

Now, condition 9 (i.e., $P_2[72] = P_3[239] + 1$) and condition 10 (i.e., $P_2[239] = P_3[72] + 1$) together imply $\sum_{i=2}^{3}(Y_i[P_i[72]] - Y_i[P_i[239]]) = 0$ and hence reduce (12) to

$$s_3 = ROTL32(s_1, P_3[116] + 18 \bmod 32). \tag{13}$$

Now, when

1. event E_2 (that is, $P_3[116] \equiv 0 \bmod 32$) occurs, (13) becomes

$$s_3 = ROTL32(s_1, 18) \Rightarrow s_{3(18)} = ROTL32(s_1, 18)_{18} = s_{1(0)};$$

2. event E_2' (i.e., $P_3[116] \equiv -18 \bmod 32$) occurs, (13) becomes

$$s_3 = ROTL32(s_1, 0) = s_1 \Rightarrow s_{3(0)} = s_{1(0)}.$$

Hence, when the event

1. $E_1 \cap E_2 \cap E_9 \cap E_{10}$ occurs then $s_{3(18)} = s_{1(0)}$,
2. $E_1 \cap E_2' \cap E_9 \cap E_{10}$ occurs then $s_{3(0)} = s_{1(0)}$.

Similarly, when the event

1. $E_3 \cap E_4 \cap E_{11} \cap E_{12}$ occurs then $s_{5(18)} = s_{3(0)}$,
2. $E_3 \cap E'_4 \cap E_{11} \cap E_{12}$ occurs then $s_{5(0)} = s_{3(0)}$.

By similar arguments, it is seen that $s_{9(0)} = s_{5(0)}$ when condition 5 (that is, $P_6[116] \equiv -18 \bmod 32$), condition 6 (i.e., $P_7[116] \equiv -18 \bmod 32$), condition 7 (i.e., $P_8[116] \equiv -18 \bmod 32$), condition 8 (i.e., $P_9[116] \equiv -18 \bmod 32$) and condition 13 (i.e., $\sum_{i=6}^{9}(Y_i[P_i[72]] - Y_i[P_i[239]]) = 0$) are simultaneously satisfied. Condition 13 generates many sets of conditions on the elements of the S-box P which are provided in Table 1. Each row in the table lists a set of 4 conditions on the elements of array P. For example, the 4 conditions listed in the first row are:

1. $P_6[72] = P_7[239] + 1$,
2. $P_7[72] = P_6[239] - 1$,
3. $P_8[72] = P_9[239] + 1$,
4. $P_9[72] = P_8[239] - 1$.

The intersection of these 4 events is denoted by G_1. The occurrence of G_j $(1 \leq j \leq 9)$ implies the occurrence of the event E_{13}. Likewise, we have 16 sets of conditions, denoted by H_k $(1 \leq k \leq 16)$, on the elements of P where each set comprises 5 conditions. When those 5 conditions are simultaneously satisfied, condition 14 of Theorem 1 is satisfied. One such set of 5 conditions is shown below.

Table 1. When G_j $(1 \leq j \leq 9)$ occurs, condition 13 is satisfied

Event	$P_6[72]$	$P_7[72]$	$P_8[72]$	$P_9[72]$
G_1	$P_7[239] + 1$	$P_6[239] - 1$	$P_9[239] + 1$	$P_8[239] - 1$
G_2	$P_7[239] + 1$	$P_8[239] + 1$	$P_9[239] + 1$	$P_6[239] - 3$
G_3	$P_7[239] + 1$	$P_9[239] + 2$	$P_6[239] - 2$	$P_8[239] - 1$
G_4	$P_8[239] + 2$	$P_6[239] - 1$	$P_9[239] + 1$	$P_7[239] - 2$
G_5	$P_8[239] + 2$	$P_9[239] + 2$	$P_6[239] - 2$	$P_7[239] - 2$
G_6	$P_8[239] + 2$	$P_9[239] + 2$	$P_7[239] - 1$	$P_6[239] - 3$
G_7	$P_9[239] + 3$	$P_6[239] - 1$	$P_7[239] - 1$	$P_8[239] - 1$
G_8	$P_9[239] + 3$	$P_8[239] + 1$	$P_6[239] - 2$	$P_7[239] - 2$
G_9	$P_9[239] + 3$	$P_8[239] + 1$	$P_7[239] - 1$	$P_6[239] - 3$

1. $P_5[153] = 3 \Rightarrow Y_5[P_5[153]] = Y_5[3] = Y_3[5] \Rightarrow Y_5[P_5[153]]_0 = Y_3[5]_0$,
2. $P_3[153] = 1 \Rightarrow Y_3[P_3[153]] = Y_3[1] = Y_1[3] \Rightarrow Y_3[P_3[153]]_0 = Y_1[3]_0$,
3. $P_9[208] = 253 \Rightarrow Y_9[P_9[208]] = Y_9[253] = Y_6[256] \Rightarrow Y_9[P_9[208]]_0 = Y_6[256]_0$,
4. $P_5[208] = 255 \Rightarrow Y_5[P_5[208]] = Y_5[255] = Y_4[256] \Rightarrow Y_5[P_5[208]]_0 = Y_4[256]_0$,
5. $P_1[208] = P_3[208] + 2 \Rightarrow Y_3[P_3[208]] = Y_1[P_1[208]]$
   $\Rightarrow Y_3[P_3[208]]_0 = Y_1[P_1[208]]_0$.

We restate condition 14 of Theorem 1 here:

$$Y_9[P_9[208]]_0 \oplus Y_5[P_5[153]]_0 \oplus Y_5[P_5[208]]_0 \oplus Y_3[P_3[153]]_0 \oplus Y_3[P_3[208]]_0$$
$$\oplus Y_1[P_1[208]]_0 \oplus Y_6[256]_0 \oplus Y_4[256]_0 \oplus Y_3[5]_0 \oplus Y_1[3]_0 = 0.$$

We observe that the above equation holds when the previous 5 conditions are satisfied simultaneously. The other 15 sets of conditions can be found in Table 2. The occurrence of H_k $(1 \leq k \leq 16)$ implies the occurrence of the event E_{14}. Let $D_{j,k}$ denote the event $E_1 \cap E_2 \cap E_3 \cap E_4 \cap E_5 \cdots \cap G_j \cap H_k$ and $F_{j,k}$ denote the event $E_1 \cap E_2' \cap E_3 \cap E_4' \cap E_5 \cdots \cap G_j \cap H_k$ where $1 \leq j \leq 9$, $1 \leq k \leq 16$. Hence, it follows that, if $E_{j,k}$ or $F_{j,k}$ occurs then

$$O_{1(0)} \oplus O_{3(0)} \oplus O_{5(0)} \oplus O_{9(0)} = 0.$$

This completes the proof. □

5 Estimation of the Bias in the Outputs

Under the assumption of a perfect key/IV setup (see Section 2), we now proceed to estimate $Prob[O_{1(0)} \oplus O_{3(0)} \oplus O_{5(0)} \oplus O_{9(0)} = 0]$. The conditions listed under Theorem 1 fall into one of the following categories:

1. $P_n[116] \equiv -18$ or $0 \bmod 32$, $2 \leq n \leq 9$,
2. an element of P is related to another element of P, and
3. an element of P is equated to a constant term.

The probability of occurrence of an event falling under category 2 or 3, is approximately $\frac{1}{256} = \frac{1}{2^8}$. It may be recalled from Sect. 4 that the event G_j $(1 \leq j \leq 9)$ is the intersection of 4 independent events, all of which fall under category 2 described above. Hence,

$$P[G_j] \approx (\frac{1}{2^8})^4 = \frac{1}{2^{32}}, 1 \leq j \leq 9. \tag{14}$$

Again, event H_k $(1 \leq k \leq 16)$ is the intersection of 5 independent events. The first four events come under category 2 and the fifth event falls under category 3. Therefore,

$$P[H_k] \approx (\frac{1}{2^8})^5 = \frac{1}{2^{40}}, 1 \leq k \leq 16. \tag{15}$$

Hence,

$$P[\cup_{j=1}^{9} G_j] = \sum_{j=1}^{9} P[G_j] \approx \frac{9}{2^{32}}. \tag{16}$$

Similarly,

$$P[\cup_{k=1}^{16} H_k] \approx \frac{16}{2^{40}} = \frac{1}{2^{36}}. \tag{17}$$

The independent events E_9, E_{10}, E_{11} and E_{12} also come under category 2. Therefore,

$$\begin{aligned} P[E_9] = P[E_{10}] = P[E_{11}] = P[E_{12}] &\approx \frac{1}{2^8} \\ \Rightarrow P[E_9 \cap E_{10} \cap E_{11} \cap E_{12}] &\approx (\frac{1}{2^8})^4 = \frac{1}{2^{32}}. \end{aligned} \tag{18}$$

Now we calculate the probabilities, using Bayes' rule, for events coming under category 1. We know that $P_2[116] \equiv -18 \bmod 32 \Rightarrow P_2[116] \in \{14, 46, 78, 110, 142, 174, 206, 238\}$. Therefore,

$$P[E_1] = \frac{8}{256}.$$

Now, $P_3[116] \equiv 0 \bmod 32$ (i.e., event E_2) implies $P_3[116] \in \{0, 32, 64, 96, 128, 160, 192, 224\}$. Since $P_3[116] \neq P_2[116]$,

$$\begin{aligned} P[E_2|E_1] &= \frac{8}{255} \\ \Rightarrow P[E_2 \cap E_1] = P[E_2|E_1] \cdot P[E_1] &= \frac{8}{256} \cdot \frac{8}{255}. \end{aligned}$$

On the other hand, if $P_3[116] \equiv -18 \bmod 32$ (i.e., event E_2') then $P_3[116] \in \{14, 46, 78, 110, 142, 174, 206, 238\}$ and hence

$$P[E_2' \cap E_1] = \frac{8}{256} \cdot \frac{7}{255}.$$

By similar arguments we have

$$\begin{aligned} P[E_3 \cap E_2 \cap E_1] = P[E_3|E_2 \cap E_1] \cdot P[E_2 \cap E_1] &= \frac{8}{256} \cdot \frac{8}{255} \cdot \frac{7}{254}, \\ P[E_4 \cap E_3 \cap E_2 \cap E_1] &= \frac{8}{256} \cdot \frac{8}{255} \cdot \frac{7}{254} \cdot \frac{7}{253}, \\ P[E_4' \cap E_3 \cap E_2 \cap E_1] &= \frac{8}{256} \cdot \frac{7}{255} \cdot \frac{6}{254} \cdot \frac{5}{253}. \end{aligned}$$

Proceeding similarly, we see that

$$P[\cap_{i=1}^{8} E_i] = \frac{8}{256} \cdot \frac{8}{255} \cdot \frac{7}{254} \cdot \frac{7}{253} \cdot \frac{6}{252} \cdot \frac{5}{251} \cdot \frac{4}{250} \cdot \frac{3}{249} \approx \frac{1}{2^{43.7}}, \tag{19}$$

and

$$P[E'] = \frac{8}{256} \cdot \frac{7}{255} \cdot \frac{6}{254} \cdot \frac{5}{253} \cdot \frac{4}{252} \cdot \frac{3}{251} \cdot \frac{2}{250} \cdot \frac{1}{249} \approx \frac{1}{2^{48.5}} \tag{20}$$

where E' denotes $E_1 \cap E_2' \cap E_3 \cap E_4' \cap E_5 \cap E_6 \cap E_7 \cap E_8$. Since $\cap_{i=1}^{8} E_i$ and E' are mutually exclusive,

$$P[(\cap_{i=1}^{8} E_i) \cup E'] = \frac{1}{2^{43.7}} + \frac{1}{2^{48.5}} \approx \frac{1}{2^{43.7}}. \tag{21}$$

If the events coming under category 1 are assumed to be independent then the values of the probabilities in (19) and (20) are identical. In such case each of them is equal to

$$(\frac{8}{256})^8 = 2^{-40}$$

instead of $2^{-43.7}$ and $2^{-48.5}$ respectively. The difference is notable and attributed to the fact that an event falling under category 1 has a larger number of outcomes than an event coming under category 2 or 3. Finally, we have 4 independent events: $(\cap_{i=1}^{8} E_i) \cup E'$, $E_9 \cap E_{10} \cap E_{11} \cap E_{12}$, G_j and H_k. We find that the intersection of these 4 events is $D_{j,k} \cup F_{j,k}$ (see Sect. 4). Hence from (14), (15), (18) and (21), we get

$$\begin{aligned} P[D_{j,k} \cup F_{j,k}] &= P[(\cap_{i=1}^{8} E_i) \cup E'] \cdot P[E_9 \cap E_{10} \cap E_{11} \cap E_{12}] \cdot P[G_j] \cdot P[H_k] \\ &= \frac{1}{2^{43.7}} \cdot \frac{1}{2^{32}} \cdot \frac{1}{2^{32}} \cdot \frac{1}{2^{40}}. \end{aligned}$$

Let A denote the event $\bigcup_{j=1,k=1}^{9,16} D_{j,k} \cup F_{j,k}$. Then, from (16) and (17)

$$P[A] = \frac{1}{2^{43.7}} \cdot \frac{1}{2^{32}} \cdot \frac{9}{2^{32}} \cdot \frac{1}{2^{36}} \approx \frac{1}{2^{140.5}}. \tag{22}$$

Now we are ready to calculate the probability $P[O_{1(0)} \oplus O_{3(0)} \oplus O_{5(0)} \oplus O_{9(0)} = 0]$. Note that the outputs are uniformly distributed if event A does not occur.[1] Let R_0 denote $O_{1(0)} \oplus O_{3(0)} \oplus O_{5(0)} \oplus O_{9(0)}$. Now, using Bayes' rule we have

$$\begin{aligned} P[R_0 = 0] &= P[R_0 = 0|A] \cdot P[A] \\ &\quad + P[R_0 = 0|A^c] \cdot P[A^c] \\ &= 1 \cdot 2^{-140.5} + \frac{1}{2} \cdot (1 - 2^{-140.5}) \\ &= \frac{1}{2} \cdot (1 + 2^{-140.5}). \end{aligned} \tag{23}$$

Note that the above probability would have been exactly 1/2 if the TPypy had been an ideal stream cipher.

6 Generalizations of the Attack

In this section, we show that the outputs ($O_{1(i)}$, $O_{3(i)}$, $O_{5(i)}$, $O_{9(i)}$, $1 \leq i \leq 31$) are also biased. Next we calculate the probability $P[O_{1(i)} \oplus O_{3(i)} \oplus O_{5(i)} \oplus O_{9(i)} = 0]$ where $0 \leq i \leq 31$. From (1) and (3) we get,

$$O_{1(i)} = s_{1(i)} \oplus Y_1[-1]_i \oplus Y_1[P_1[208]]_i \oplus c_{1(i)}, \tag{24}$$

$$Y_4[256]_i = ROTL32(s_3, 14)_i \oplus Y_3[-3]_i \oplus Y_3[P_3[153]]_i \oplus d_{1(i)}, \tag{25}$$

[1] This fact is established from the assumption that the key/IV setups are perfect and produce uniformly distributed initial state.

where $0 \leq i \leq 31$ and c_1, d_1 are the carry terms in (1) and (3) respectively. Similarly,

$$O_{3(i)} = s_{3(i)} \oplus Y_3[-1]_i \oplus Y_3[P_3[208]]_i \oplus c_{3(i)} \tag{26}$$
$$O_{5(i)} = s_{5(i)} \oplus Y_5[-1]_i \oplus Y_5[P_5[208]]_i \oplus c_{5(i)}, \tag{27}$$
$$O_{9(i)} = s_{9(i)} \oplus Y_9[-1]_i \oplus Y_9[P_9[208]]_i \oplus c_{9(i)} \tag{28}$$

where $0 \leq i \leq 31$. Now, $ROTL32(s_3, 14)_i = s_{3(i+18 \bmod 32)}$ and $Y_3[-3]_i = Y_1[-1]_i$. Substituting these two results in (25) and rearranging the terms we get,

$$Y_1[-1]_i = s_{3((i+18) \bmod 32)} \oplus Y_4[256]_i \oplus Y_3[P_3[153]]_i \oplus d_{1(i)}. \tag{29}$$

Putting (29) in (24) we get,

$$O_{1(i)} = s_{1(i)} \oplus s_{3((i+18) \bmod 32)} \oplus Y_4[256]_i \oplus Y_3[P_3[153]]_i \oplus Y_1[P_1[208]]_i \oplus c_{1(i)} \oplus d_{1(i)}. \tag{30}$$

Similarly, if d_3 denotes the carry term in

$$Y_6[256] = (ROTL32(s_5, 14) \oplus Y_5[-3]) + Y_5[P_5[153]],$$

we have,

$$O_{3(i)} = s_{3(i)} \oplus s_{5((i+18) \bmod 32)} \oplus Y_6[256]_i \oplus Y_5[P_5[153]]_i \oplus Y_3[P_3[208]]_i \oplus c_{3(i)} \oplus d_{3(i)}. \tag{31}$$

Since $Y_5[-1]_i = Y_1[3]_i$ and $Y_9[-1]_i = Y_3[5]_i$, $\forall i \in \{0, ..., 31\}$, (27 and (28) can be written as:

$$O_{5(i)} = s_{5(i)} \oplus Y_1[3]_i \oplus Y_5[P_5[208]]_i \oplus c_{5(i)}, \tag{32}$$
$$O_{9(i)} = s_{9(i)} \oplus Y_3[5]_i \oplus Y_9[P_9[208]]_i \oplus c_{9(i)}. \tag{33}$$

From (30), (31), (32) and (33),

$$\begin{aligned} O_{1(i)} \oplus O_{3(i)} \oplus O_{5(i)} \oplus O_{9(i)} = {} & s_{1(i)} \oplus s_{3((i+18) \bmod 32)} \oplus s_{3(i)} \oplus s_{5((i+18) \bmod 32)} \\ & \oplus s_{5(i)} \oplus Y_1[P_1[208]]_i \oplus Y_3[P_3[153]]_i \oplus Y_3[P_3[208]]_i \\ & \oplus s_{9(i)} \oplus Y_5[P_5[153]]_i \oplus Y_5[P_5[208]]_i \oplus Y_9[P_9[208]]_i \\ & \oplus Y_1[3]_i \oplus Y_3[5]_i \oplus Y_4[256]_i \oplus Y_6[256]_i \\ & \oplus c_{1(i)} \oplus c_{3(i)} \oplus c_{5(i)} \oplus c_{9(i)} \oplus d_{1(i)} \oplus d_{3(i)}. \end{aligned} \tag{34}$$

Let the event E_{14} under Theorem 1 be redefined as follows.

$$Y_9[P_9[208]]_i \oplus Y_5[P_5[153]]_i \oplus Y_5[P_5[208]]_i \oplus Y_3[P_3[153]]_i \oplus Y_3[P_3[208]]_i \oplus Y_1[P_1[208]]_i \oplus Y_6[256]_i \oplus Y_4[256]_i \oplus Y_3[5]_i \oplus Y_1[3]_i = 0. \tag{35}$$

Now, using the techniques described in Sect. 4, it can be easily verified that when the first 13 conditions listed under Theorem 1 and (35) are simultaneously satisfied, (34) reduces to

$$O_{1(i)} \oplus O_{3(i)} \oplus O_{5(i)} \oplus O_{9(i)} = c_{1(i)} \oplus c_{3(i)} \oplus c_{5(i)} \oplus c_{9(i)} \oplus d_{1(i)} \oplus d_{3(i)}. \tag{36}$$

Let,

$$\begin{aligned}R_i &= O_{1(i)} \oplus O_{3(i)} \oplus O_{5(i)} \oplus O_{9(i)},\\ T_i &= c_{1(i)} \oplus c_{3(i)} \oplus c_{5(i)} \oplus c_{9(i)} \oplus d_{1(i)} \oplus d_{3(i)}. \end{aligned} \tag{37}$$

Now,

$$\begin{aligned}P[R_i = 0] &= P[R_i = 0|A] \cdot P[A]\\ &\quad + P[R_i = 0|A^c] \cdot P[A^c]\\ &= P[T_i = 0] \cdot P[A]\\ &\quad + P[T_i = 0|A^c] \cdot P[A^c]\\ &= P[T_i = 0] \cdot 2^{-140.5} + \frac{1}{2} \cdot (1 - 2^{-140.5}). \end{aligned} \tag{38}$$

We observe that the carry terms c, d are generated in expressions of the form $(S \oplus A) + B$, where S, A and B are uniformly distributed and independent 32-bit variables. Hence, we can use the following result from [8].

$$P[c_{1(i)} = 0] = \frac{1}{2} + \frac{1}{2^{i+1}}.$$

Let $p = \frac{1}{2} + \frac{1}{2^{i+1}}$. Also,

$$P[c_{3(i)}] = P[c_{5(i)}] = P[c_{9(i)}] = P[d_{1(i)}] = P[d_{3(i)}] = p.$$

We see that $T_i = 0$ when an even number of terms on the RHS of (37) equate to zero. Since the probability of any of the terms equating to zero is p,

$$\begin{aligned}P[T_i = 0] &= p^6 + \binom{6}{2} \cdot p^4 \cdot (1-p)^2 + \binom{6}{4} \cdot p^2 \cdot (1-p)^4 + (1-p)^6\\ &= p^6 + 15 \cdot p^4 \cdot (1-p)^2 + 15 \cdot p^2 \cdot (1-p)^4 + (1-p)^6. \end{aligned} \tag{39}$$

Substituting $p = \frac{1}{2} + \frac{1}{2^{i+1}}$ in (39) and simplifying the resultant expression we get,

$$P[T_i = 0] = \frac{1}{2} + 15 \cdot 2^{-(5+2i)} + 2^{-(5+6i)}. \tag{40}$$

When $i = 0$, we get $P[T_i = 0] = 1$ which is the expected value. Finally, putting (40) in (38) we get,

$$\begin{aligned}P[R_i = 0] &= \frac{1}{2} + (15 \cdot 2^{-(5+2i)} + 2^{-(5+6i)}) \cdot 2^{-140.5}\\ &= \frac{1}{2} + 15 \cdot 2^{-(145.5+2i)} + 2^{-(145.5+6i)}. \end{aligned} \tag{41}$$

Using the techniques described above and in Sect. 4 and 5, one can show that (41) applies to outputs generated at rounds t, $t+2$, $t+4$ and $t+8$, i.e.,

$$\begin{aligned}P[O_{t(i)} \oplus O_{t+2(i)} \oplus O_{t+4(i)} \oplus O_{t+8(i)} = 0] &= \frac{1}{2} + 15 \cdot 2^{-(145.5+2i)}\\ &\quad + 2^{-(145.5+6i)}. \end{aligned} \tag{42}$$

And when $i = 0$ (i.e., for the lsb), (42) becomes:

$$P[O_{t(0)} \oplus O_{t+2(0)} \oplus O_{t+4(0)} \oplus O_{t+8(0)} = 0] = \frac{1}{2} \cdot (1 + 2^{-140.5}). \qquad (43)$$

7 The Distinguisher

A distinguisher is an algorithm which distinguishes a stream of bits from a stream of bits that follow the uniform distribution. The distinguisher we construct, using the observations described in the previous sections, collects the lsbs of sufficiently many outputs $(O_t,\ O_{t+2},\ O_{t+4},\ O_{t+8})$. 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 $O_{t(0)} \oplus O_{t+2(0)} \oplus O_{t+4(0)} \oplus O_{t+8(0)} = 0$, X is the distribution of the outputs O_t, O_{t+2}, O_{t+4} and O_{t+8} produced by a perfectly random keystream generator and Y is the distribution of the outputs produced by TPypy. From (43), $p = \frac{1}{2}$, $q = \frac{1}{2^{140.5}}$. Hence $O(\frac{1}{(2^{-140.5})^2}) = O(2^{281})$ output samples are needed to construct the distinguisher with non-negligible probability of success. It should be noted that distinguishers can be constructed by considering higher order bits of $(O_t,\ O_{t+2},\ O_{t+4},\ O_{t+8})$. However, as the maximum bias is found in the lsb, the best distinguisher requires $O(2^{281})$ samples to distinguish TPypy from random.

8 Adapting the Attacks to TPy6, Pypy, Py and Py6

The attacks described in the previous sections can be adjusted for TPy6 also, in a similar way the attack on Py in [8] was modified to work for Py6 in [9]. Since the round functions of TPypy, TPy and TPy6 are identical with those of Pypy, Py and Py6 respectively, it is easy to see that the aforementioned attacks on TPypy, TPy and TPy6 are also applicable to Pypy, Py and Py6.

Moreover, it is important to note that the existing distinguishing attacks on the Py and Py6 as described in [4] and [9] are also effective on TPy and TPy6. Therefore, the best distinguishers on TPy and TPy6 are with data 2^{72} and 2^{68}.

9 Conclusions and Future Work

This paper for the first time finds weaknesses in the stream cipher TPypy which is conjectured to the most secure candidate of the Py-family of ciphers. Precisely, the paper shows that the security of the stream cipher TPypy does not grow exponentially with the key-size. This is established by constructing a distinguisher which works with 2^{281} outputwords and comparable time. Note that, when TPypy is used with key-size longer than 35 bytes, i.e., 280 bits, our attack constitutes an academic break of the cipher.

Given the striking similarities between TPy and TPypy (see Algorithm 1), the current attacks can also be applied to TPy. In fact, we have further noted that some of the existing attacks also work for TPy and TPy6. Moreover, in Appendix B, we present additional weaknesses in TPy by combining the results of this paper with the results described in [8]. Thus we have many weaknesses in the round functions of TPy and TPypy. It seems quite possible to combine these weaknesses to construct stronger distinguishers for both the ciphers. Crowley, in [4], describes a method which uses a Hidden Markov Model to combine the distinguishers on Py. This method reduced the data complexity of an attack on Py by a factor of 2^{17}. The same method can certainly be applied here too, but a complete description is beyond the scope of this paper.

References

1. E. Biham, J. Seberry, "Py (Roo): A Fast and Secure Stream Cipher using Rolling Arrays," *ecrypt submission*, 2005.
2. E. Biham, J. Seberry, "Pypy (Roopy): Another Version of Py," *ecrypt submission*, 2006.
3. 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.
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, "Cryptanalysis of Stream Ciphers Based on Arrays and Modular Addition," *PhD thesis, Katholieke Universiteit Leuven, 2006.*
8. 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.
9. 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.
10. H. Wu and B. Preneel, "Differential Cryptanalysis of the Stream Ciphers Py, Py6 and Pypy," Eurocrypt 2007, LNCS, Springer-Verlag, 2007.
11. H. Wu, B. Preneel, "Attacking the IV Setup of Py and Pypy," available at `http://www.ecrypt.eu.org/stream/papersdir/2006/050.pdf`.

A Disjoint Events

Here, we prove that for event E_{13} to occur with probability 1, the 9 conditions G_j, $1 \leq j \leq 9$ can be considered exhaustive. That is, E_{13} occurs with random probability $\frac{1}{32}$ when none of the events G_j occur. From the discussion in Sect. 4, we know that $s_{5(0)} = s_{9(0)}$ when the following conditions are simultaneously satisfied:

1. $P_6[116] \equiv -18 (\bmod 32)$ (event E_5),
2. $P_7[116] \equiv -18 (\bmod 32)$ (event E_6),
3. $P_8[116] \equiv -18 (\bmod 32)$ (event E_7),
4. $P_9[116] \equiv -18 (\bmod 32)$ (event E_8),
5. $\sum_{i=6}^{9} (Y_i[P_i[72]] - Y_i[P_i[239]]) = 0$ (event E_{13}).

It may be recalled from Sect. 4 that Table 1 shows nine events (G_1, ..., G_9), where the occurrence of any G_i, $1 \leq i \leq 9$, implies that event E_{13} occurs. It is to be noted that the nine events are not pairwise disjoint. This can be demonstrated as follows. Let us consider the events G_1 and G_2 and let us assume that $P_6[239] = P_8[239] + 2$. Hence, $P_6[239] - 1 = P_8[239] + 1$ and $P_8[239] - 1 = P_6[239] - 3$, with the result that rows 1 and 2 become identical. Therefore, $G_1 = G_2$ when $P_6[239] = P_8[239] + 2$. Similarly, any condition which equates two events, say G_i and G_j (i, $j \in \{1, ..., 9\}$ and $i \neq j$), will relate $P_m[239]$ and $P_n[239]$ (n, $m \in \{6, 7, 8, 9\}$, $n \neq m$). The relation is of the form $P_m[239] = P_n[239] + (n - m)$, and the probability of its occurrence is approximately $\frac{1}{256}$. We restate event E_{13} here,

$$Y_6[P_6[72]] - Y_6[P_6[239]] + Y_7[P_7[72]] - Y_7[P_7[239]] + Y_8[P_8[72]] - Y_8[P_8[239]] + Y_9[P_9[72]] - Y_9[P_9[239]] = 0.$$

We see that any relation of the above form will only equate two Y terms preceded by the same '$-$' sign ($Y_m[P_m[239]]$ and $Y_n[P_n[239]]$) and thus the terms do not get cancelled. Hence it is not necessary that this relation must hold for E_{13} to occur. Now, the probability that

- the 4 conditions forming G_i (where $1 \leq i \leq 9$) are simultaneously satsified and
- a relation of the form $P_m[239] = P_n[239] + (n - m)$ (where n, $m \in \{6, 7, 8, 9\}$, $n \neq m$) holds,

will be approximately equal to $(\frac{1}{256})^5 = \frac{P[G_i]}{256}$. Such a relation of the above form, which is an extra condition on the elements of P and which amounts to no significant increase in the probability that E_{13} occurs, is therefore redundant. There is still one important point that we have overlooked - it is not necessary that relations are to be drawn between elements of P in order that condition 13 under Theorem 1 be satisfied. One could try to relate the elements of array Y directly. However, any such a relation would occur with a probability of approximately $\frac{1}{2^{32}}$- which is much lesser than $P[P_m[239] = P_n[239] + (n - m)] \approx \frac{1}{2^8}$.

In summary, the satisfiability of the four conditions listed in each row of Table 1, is sufficient for our analysis and any extra condition is discarded.

Table 2 below shows 16 events (H_1,..., H_{16}), where the occurrence of any H_i, $1 \leq i \leq 16$, implies that event E_{14} occurs. Under each event H_i we have 5 conditions on the elements of array P. From the table, one can infer that barring the 5th condition, the other 4 conditions equate an element of P to a constant term. Following this observation, it is easy to show that $H_i \neq H_j$ for any i, $j \in \{1, ..., 16\}$ where $i \neq j$. Now, by similar arguments as above, one can prove that for event E_{14} to occur with probability 1, the 16 conditions H_k, $1 \leq k \leq 16$ are exhaustive. That is, E_{14} occurs with random probability $\frac{1}{2}$ when none of the events H_k occur.

B Another Statistical Weakness in TPy

In this section, we combine the results of this paper with the results of [8] to find a new weakness in TPy. The algorithm for TPy includes line 5 of Algorithm 1. We use the convention followed in [8]. We label the outputs generated in line 5 and line 6 as the '1st output-word' and '2'nd output-word respectively. Let $O_{m,n}$ (where $m \in \{1, 2\}$) denote the mth output-word generated in the nth round of TPy. $O_{m,n(j)}$ denotes the j the bit of $O_{m,n}$. The following is a corollary of Theorem 2 of [8].

Corollary 2. $O_{1,8(0)} = O_{2,10(0)}$ *if the following six conditions on the elements of the S-box P are simultaneously satisfied.*

1. $P_9[116] \equiv -18 (\bmod 32)$ *(event L_1),*
2. $P_{10}[116] \equiv 7 (\bmod 32)$ *(event L_2),*
3. $P_9[72] = P_{10}[239] + 1$ *(event L_3),*
4. $P_9[239] = P_{10}[72] + 1$ *(event L_4),*
5. $P_9[26] = 1$ *(event L_5),*
6. $P_{10}[208] = 254$ *(event L_6).*

We recall the following from Sect. 4.

1. Event $L_1 = E_8$,
2. $E = E_1 \cap E_2 \cap E_3 \cap E_4 \cap E_5 \cdots \cap E_{14}$ and
3. $F = E_1 \cap E_2' \cap E_3 \cap E_4' \cap E_5 \cdots \cap E_{14}$.

Now, we have the following theorem:

Theorem 2. $O_{2,1(0)} \oplus O_{2,3(0)} \oplus O_{2,5(0)} \oplus O_{2,9(0)} \oplus O_{2,10(0)} \oplus O_{1,8(0)} = 0$ *if*

1. *event E or F occurs, and*
2. *the event $L_2 \cap L_3 \cap L_4 \cap L_5 \cap L_6$ occurs.*

The proof is straightforward and follows from Sect. 4 and the proof of Theorem 1 in [8]. It is to be noted that for event E_{14} to occur with probability 1, the 16 conditions H_k, $1 \leq k \leq 16$ are not exhaustive. A careful look at Table 2 would reveal the reason. All the events except H_6, H_8, H_{10} and H_{12}, have one of the conditions of the form $P_3[X] = 1$, where $X \in \{153, 208\}$. Now, $P_3[X] = 1 \Rightarrow P_9[26] \neq 1$. Therefore, we have E_{14} to occur with probability 1 when $H_6 \cup H_8 \cup H_{10} \cup H_{12}$ occurs.

Table 2. When H_k $(1 \leq k \leq 16)$ occurs, condition 14 is satisfied.

Event	$Y_5[3]$ $(= Y_3[5] = Y_1[7])$	$Y_1[3](= Y_3[1])$	$Y_6[256](= Y_9[253])$	$Y_4[256](= Y_5[255])$	**5th Condition**
H_1	$Y_5[P_5[153]]$ $(P_5[153] = 3)$	$Y_3[P_3[153]]$ $(P_3[153] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[208]]$ $(P_5[208] = 255)$	$Y_3[P_3[208]] = Y_1[P_1[208]]$ $(P_1[208] = P_3[208] + 2)$
H_2	$Y_5[P_5[153]]$ $(P_5[153] = 3)$	$Y_3[P_3[208]]$ $(P_3[208] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[208]]$ $(P_5[208] = 255)$	$Y_3[P_3[153]] = Y_1[P_1[208]]$ $(P_1[208] = P_3[153] + 2)$
H_3	$Y_5[P_5[208]]$ $(P_5[208] = 3)$	$Y_3[P_3[153]]$ $(P_3[153] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[153]]$ $(P_5[153] = 255)$	$Y_3[P_3[208]] = Y_1[P_1[208]]$ $(P_1[208] = P_3[208] + 2)$
H_4	$Y_5[P_5[208]]$ $(P_5[208] = 3)$	$Y_3[P_3[208]]$ $(P_3[208] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[153]]$ $(P_5[153] = 255)$	$Y_3[P_3[153]] = Y_1[P_1[208]]$ $(P_1[208] = P_3[153] + 2)$
H_5	$Y_3[P_3[153]]$ $(P_3[153] = 5)$	$Y_3[P_3[208]]$ $(P_3[208] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[208]]$ $(P_5[208] = 255)$	$Y_5[P_5[153]] = Y_1[P_1[208]]$ $(P_1[208] = P_5[153] + 4)$
H_6	$Y_3[P_3[153]]$ $(P_3[153] = 5)$	$Y_1[P_1[208]]$ $(P_1[208] = 3)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[208]]$ $(P_5[208] = 255)$	$Y_5[P_5[153]] = Y_3[P_3[208]]$ $(P_3[208] = P_5[153] + 2)$
H_7	$Y_3[P_3[153]]$ $(P_3[153] = 5)$	$Y_3[P_3[208]]$ $(P_3[208] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[153]]$ $(P_5[153] = 255)$	$Y_5[P_5[208]] = Y_1[P_1[208]]$ $(P_1[208] = P_5[208] + 4)$
H_8	$Y_3[P_3[153]]$ $(P_3[153] = 5)$	$Y_1[P_1[208]]$ $(P_1[208] = 3)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[153]]$ $(P_5[153] = 255)$	$Y_5[P_5[208]] = Y_3[P_3[208]]$ $(P_3[208] = P_5[208] + 2)$
H_9	$Y_3[P_3[208]]$ $(P_3[208] = 5)$	$Y_3[P_3[153]]$ $(P_3[153] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[208]]$ $(P_5[208] = 255)$	$Y_5[P_5[153]] = Y_1[P_1[208]]$ $(P_1[208] = P_5[153] + 4)$
H_{10}	$Y_3[P_3[208]]$ $(P_3[208] = 5)$	$Y_1[P_1[208]]$ $(P_1[208] = 3)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[208]]$ $(P_5[208] = 255)$	$Y_5[P_5[153]] = Y_3[P_3[153]]$ $(P_3[153] = P_5[153] + 2)$
H_{11}	$Y_3[P_3[208]]$ $(P_3[208] = 5)$	$Y_3[P_3[153]]$ $(P_3[153] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[153]]$ $(P_5[153] = 255)$	$Y_5[P_5[208]] = Y_1[P_1[208]]$ $(P_1[208] = P_5[208] + 4)$
H_{12}	$Y_3[P_3[208]]$ $(P_3[208] = 5)$	$Y_1[P_1[208]]$ $(P_1[208] = 3)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[153]]$ $(P_5[153] = 255)$	$Y_5[P_5[208]] = Y_3[P_3[153]]$ $(P_3[153] = P_5[208] + 2)$
H_{13}	$Y_1[P_1[208]]$ $(P_1[153] = 7)$	$Y_3[P_3[153]]$ $(P_3[153] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[208]]$ $(P_5[208] = 255)$	$Y_5[P_5[153]] = Y_3[P_3[208]]$ $(P_3[208] = P_5[153] + 2)$
H_{14}	$Y_1[P_1[208]]$ $(P_1[153] = 7)$	$Y_3[P_3[208]]$ $(P_3[208] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[208]]$ $(P_5[208] = 255)$	$Y_5[P_5[153]] = Y_3[P_3[153]]$ $(P_3[153] = P_5[153] + 2)$
H_{15}	$Y_1[P_1[208]]$ $(P_1[153] = 7)$	$Y_3[P_3[153]]$ $(P_3[153] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[153]]$ $(P_5[153] = 255)$	$Y_5[P_5[208]] = Y_3[P_3[208]]$ $(P_3[208] = P_5[208] + 2)$
H_{16}	$Y_1[P_1[208]]$ $(P_1[153] = 7)$	$Y_3[P_3[208]]$ $(P_3[208] = 1)$	$Y_9[P_9[208]]$ $(P_9[208] = 253)$	$Y_5[P_5[153]]$ $(P_5[153] = 255)$	$Y_5[P_5[208]] = Y_3[P_3[153]]$ $(P_3[153] = P_5[208] + 2)$

As before, one can try to generalize the above theorem for round t and bit i. It is also possible to draw other corollaries from Theorem 2 of [8] and use them in tandem with generalizations of Theorem 1 to find many more relationships among the outputs generated by TPy. The probability that the conditions listed under Theorem 2 are simultaneously satisfied, will be lesser than $P[E \cup F] = 2^{-140.5}$. Hence the distinguisher thus constructed will require more than 2^{281} samples. Nevertheless, the existence of a large number of biases in the outputs of TPy and TPypy can be exploited to construct stronger distinguishers for both the ciphers.