Noname manuscript No.
(will be inserted by the editor)

Improved Differential Fault Attack on MICKEY 2.0

Subhadeep Banik · Subhamoy Maitra · Santanu Sarkar

Received: date / Accepted: date

Abstract In this paper we describe several ideas related to Differential Fault Attack (DFA) on MICKEY 2.0, a stream cipher from eStream hardware profile. Using the standard assumptions for fault attacks, we first show that if the adversary can induce random single bit faults in the internal state of the cipher, then by injecting around $2^{16.7}$ faults and performing $2^{32.5}$ computations on an average, it is possible to recover the entire internal state of MICKEY at the beginning of the key-stream generation phase. We further consider the scenario where the fault may affect more than one (at most three) neighbouring bits and in that case we require around $2^{18.4}$ faults on an average to mount the DFA. We further show that if the attacker can solve multivariate equations (say, using SAT solvers) then the attack can be carried out using around $2^{14.7}$ faults in the single-bit fault model and $2^{16.06}$ faults for the multiple-bit scenario.

Keywords eStream · Fault attacks · MICKEY 2.0 · Stream Cipher

1 Introduction

The stream cipher MICKEY 2.0 [1] was designed by Steve Babbage and Matthew Dodd as a submission to the eStream project. The cipher has been selected as a part of eStream's final hardware portfolio. MICKEY is a synchronous, bit-oriented stream cipher designed for low hardware complexity and high speed. After a TMD tradeoff attack [17] against the initial version of MICKEY (version 1), the designers responded with a tweak to the design by increasing the state size from 160 to 200 bits and altering the values of some control bit tap locations. These changes were incorporated in MICKEY 2.0 and these are the only differences between MICKEY version 1 and MICKEY 2.0. While MICKEY 2.0 uses an 80-bit key and a variable length IV, a modified version of the cipher, MICKEY-128 2.0 that uses a 128-bit key [2] was also proposed by the designers.

The name MICKEY is derived from "Mutual Irregular Clocking KEY-stream generator" which describes the behavior of the cipher. The state consists of two 100-bit shift registers named R and S, each of which is irregularly clocked and controlled by the other. The cipher specification underlines that each key can be used with up to 2^{40} different IVs of the same length, and that 2^{40} key-stream bits can be generated from each Key-IV pair. Very little cryptanalysis of MICKEY 2.0 is available in literature. In fact it has been noted in [10, Section 3.2] that other than the observation related to time or power analysis attacks [13] on straightforward implementations of the MICKEY family, there have been no known cryptanalytic advances on these

This is a substantially revised and extended version of the paper "A Differential Fault Attack on MICKEY 2.0" by S. Banik and S. Maitra that has been presented in Workshop on Cryptographic Hardware and Embedded Systems 2013 (CHES 2013), Santa Barbara, California, USA during August 20–23, 2013 and published in LNCS, Vol. 8086, pp. 215–232. Section 5 is an addition over the conference version showing further improvement.

Subhadeep Banik
Indian Statistical Institute, 203, B.T. Road, Kolkata-108, India. Tel.: +91-33-25752821, E-mail: s.banik_r@isical.ac.in

Subhamoy Maitra
Indian Statistical Institute, 203, B.T. Road, Kolkata-108, India. E-mail: subho@isical.ac.in

Santanu Sarkar
Chennai Mathematical Institute, Chennai 603103, India. E-mail: sarkar.santanu.bir@gmail.com

ciphers. The work in this paper presents cryptanalytic result of MICKEY 2.0 in terms of differential fault attack.

Since the work of [8, 9], fault attacks have been considered as important cryptographic tools to analyse the strengths of cryptographic primitives. Such attacks on stream ciphers were first described by Hoch and Shamir [14]. A typical fault attack [14] involves random injection of faults (using laser shots/clock glitches [21, 22]) in a device (typically initialized by a secret key) which changes one or more bits of its internal state. The adversary then attempts to deduce information about the internal state (and if possible, the secret key too) using the output stream from this faulty device. In order to perform the attack, certain privileges are required like the ability to re-key the device, control the timing of the fault etc. The attack becomes impractical and unrealistic if the adversary is granted too many privileges. In this work we assume the following privileges from the adversarial point of view that are generally acceptable in cryptanalytic literature.

1. We assume that the adversary can re-key the cipher with the original Key-IV and restart cipher operations multiple times.
2. She has precise control over the timing of the fault injection.
3. Initially we assume that she can inject a fault that alters the bit value of one random register location in either the R or the S register. Later, in Section 4, we explore the situation when she can inject a fault that may affect more than one value in contiguous register locations. We present explicit results considering the events when upto three contiguous register locations may be affected in R or S.
4. She is, however, unable to fix the exact location of the R or S register where she wants to inject the fault. Obtaining the fault location by comparison of the fault-free and the faulty key-streams is one of the challenges while mounting the fault attack.

There are published works where the assumptions made are quite strong and requires the adversary to have more control over fault injections, e.g., the works [4, 7, 18] consider that the attacker can reproduce multiple faults in the same (but unknown) locations. A detailed physical implementation using such fault model is presented in [7, Section IIIB]. In this work we use a more relaxed fault model (as in [5] for Grain family) in which the adversary is not required to fault an unknown register location multiple number of times.

Differential fault attack is a special class of fault attack in which the attacker uses the difference between fault-free and faulty key-streams to deduce the internal state or the secret key of the cipher. In case of MICKEY 2.0, the differential attack is possible due to the rather simplistic nature of the output function (r_0+s_0) used to produce key-stream bits. Additionally, there are some interesting combinatorial properties of the state update functions in MICKEY that help facilitate the attack that we shall describe.

The organization of the paper is as follows. In Section 2, we present a description of the cipher which is suitable for our analysis, where we also present some notations that will be henceforth used in the paper. The complete attack, assuming that the adversary is able to induce single bit faults in random register locations, is described in Section 3. In Section 4 we explore the case when the adversary is able to induce a fault that affects the bit values of (random) consecutive (upto 3) register locations. In Section 5 we propose improvements of the attack using SAT Solvers. Section 6 concludes the paper.

2 Our description of MICKEY 2.0 PRGA and some notations

A detailed description of MICKEY 2.0 is available in [1]. It uses an 80-bit key and a variable length IV, the length of which may vary between 0 and 80 bits. The physical structure of the cipher consists of two 100 bit registers R and S. Both the registers are initialized to the all-zero state, and the three stages of register update (i) IV loading, (ii) Key Loading, and (iii) Pre-Clock are executed sequentially before the production of the first key-stream bit. Thereafter, during the PRGA (Pseudo Random bitstream Generation Algorithm), key-stream bits are produced.

We will now provide an alternate description of this stage of operation (PRGA) in MICKEY 2.0. Consider the binary variables a_0, a_1, a_2, a_3. Let a_0 be defined as

$$a_0 = \begin{cases} a_2, \text{ if } a_1 = 0 \\ a_3, \text{ if } a_1 = 1. \end{cases}$$

Then it is straightforward to see that a_0 can be expressed as a multivariate polynomial over GF(2), i.e.,

$$a_0 = (1 + a_1) \cdot a_2 + a_1 \cdot a_3.$$

The state registers R and S, during the PRGA, are updated by a call to the $CLOCK\_KG$ routine, which in turn calls the $CLOCK\_R$ and the $CLOCK\_S$ routine. In both these routines, the state is updated via a number of If-Else constructs. As a result of this, the state update may be equivalently expressed as a series of multi-variate polynomials over GF(2).

Let $r_0, r_1, \ldots, r_{99}, s_0, s_1, \ldots, s_{99}$ denote the internal state at a certain round during the MICKEY PRGA and let $r'_0, r'_1, \ldots, r'_{99}, s'_0, s'_1, \ldots, s'_{99}$ denote the internal state at the next round. Then it is possible to write

$$r'_i = \rho_i(r_0, r_1, \ldots, r_{99}, s_0, s_1, \ldots, s_{99}),$$

$$s'_i = \beta_i(r_0, r_1, \ldots, r_{99}, s_0, s_1, \ldots, s_{99}),$$

$\forall i \in [0, 99]$, where ρ_i, β_i are polynomial functions over GF(2). The exact forms of ρ_i, β_i are described in Appendix C.

Before describing the attack, let us fix certain notations that will be used henceforth.

1. $R_t = [r_0^t, r_1^t, \ldots, r_{99}^t], S_t = [s_0^t, s_1^t, \ldots, s_{99}^t]$ is used to denote the internal states of the R, S registers at the beginning of the round t of the PRGA. That is, r_i^t, s_i^t respectively denotes the i^{th} bit of the registers R, S at the beginning of round t of the PRGA. Note that $r_i^{t+1} = \rho_i(R_t, S_t)$ and $s_i^{t+1} = \beta_i(R_t, S_t)$.
2. The value of the variables $CONTROL\_BIT\_R$ and $CONTROL\_BIT\_S$, at the PRGA round t, are denoted by the variables CR_t, CS_t respectively. These bits are used by the R, S registers to exercise mutual self control over each other. Note that $CR_t = r_{67}^t + s_{34}^t$ and $CS_t = r_{33}^t + s_{67}^t$.
3. $R_{t,\Delta r_\phi}(t_0), S_{t,\Delta r_\phi}(t_0)$ (resp. $R_{t,\Delta s_\phi}(t_0), S_{t,\Delta s_\phi}(t_0)$) are used to denote the internal states of the cipher at the beginning of round t of the PRGA, when a fault has been injected in location ϕ of R(resp. S) at the beginning of round t_0 of the PRGA.
4. $z_{i,\Delta r_\phi}(t_0)$ or $z_{i,\Delta s_\phi}(t_0)$ denotes the key-stream bit produced in the i^{th} PRGA round, after a fault has been injected in location ϕ of R or S at the beginning of round t_0 of the PRGA. By z_i, we refer to the fault-free key-stream bit produced in the i^{th} PRGA round.

3 Complete description of the Attack

We start with some technical results that will be used later.

Lemma 1 *Consider the first* 100 *internal states of the MICKEY 2.0 PRGA. If* r_{99}^t *and* CR_t *are known* $\forall t \in [0, 99]$*, then the initial state* R_0 *can be calculated efficiently.*

Proof Let the values of r_{99}^t and CR_t be known $\forall t \in [0, 99]$. We notice that the functions ρ_i for all values of $i \in [1, 99]$ are of the form $\rho_i(\cdot) = r_{i-1} + (s_{34} + r_{67}) \cdot r_i + \alpha_i \cdot r_{99}$, where $s_{34} + r_{67}$ is the value of $CONTROL\_BIT\_R$. Also, $\alpha_i = 1$, if $i \in RTAPS$ (this is a set of tap locations related to the design of MICKEY 2.0, see [1]) and is 0 otherwise. Now consider the following equation governing r_{99}^{99} :

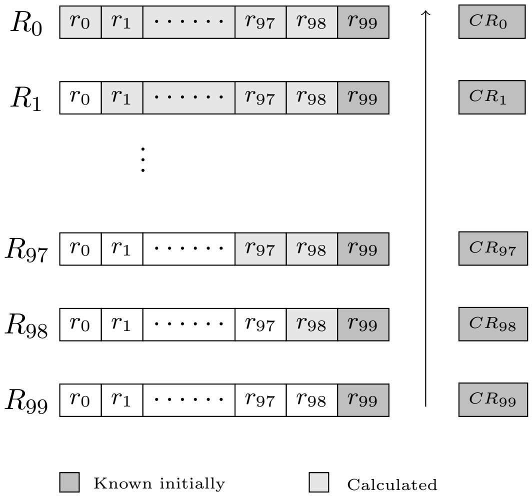


Fig. 1: Constructing the state R_0. Starting from PRGA round 99, any bit calculated at PRGA round i is used to determine state bits of round $i - 1$.

$$r_{99}^{99} = \rho_{99}(R_{98}, S_{98}) = r_{98}^{98} + CR_{98} \cdot r_{99}^{98} + \alpha_{99} \cdot r_{99}^{98}.$$

In the above equation, r_{98}^{98} is the only unknown and it appears as a linear term, and so its value can be calculated immediately. We therefore know the values of 2 state bits of R_{98}: r_{99}^{98}, r_{98}^{98}. Similarly look at the equations governing r_{99}^{98}, r_{98}^{98}:

$$r_{99}^{98} = r_{98}^{97} + CR_{97} \cdot r_{99}^{97} + \alpha_{99} \cdot r_{99}^{97},$$

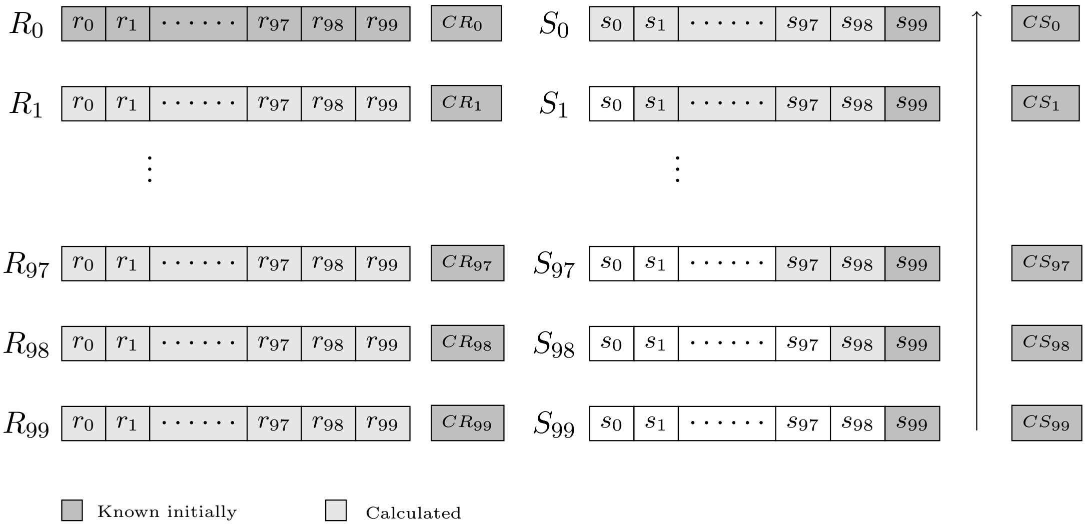
$$r_{98}^{98} = r_{97}^{97} + CR_{97} \cdot r_{98}^{97} + \alpha_{98} \cdot r_{99}^{97}.$$

As before, r_{98}^{97} is the lone unknown term in the first equation whose value is determined immediately. After this, r_{97}^{97} becomes the only unknown linear term in the next equation whose value too is determined easily. Thus we know 3 bits of R_{97}: r_{97+i}^{97}, $i = 0, 1, 2$. Continuing in such a bottom-up manner we can successively determine 4 bits of R_{96}, 5 bits of R_{95} and eventually all the 100 bits of R_0. (The process is explained pictorially in Figure 1.) □

Lemma 2 *Consider the first* 100 *internal states of the MICKEY 2.0 PRGA. If* R_0 *is known and* s_{99}^t, CS_t, CR_t *are known* $\forall t \in [0, 99]$*, then the initial state* S_0 *of the register* S *can be determined efficiently.*

Proof Since R_0 is known and so is CR_t for each $t \in [0, 99]$, we can construct all the bits of R_1 by calculating

$$r_i^1 = r_{i-1}^0 + CR_0 \cdot r_i^0 + \alpha_i \cdot r_{99}^0,\ \forall i \in [1, 99],$$



R_0 r_0 r_1 ······ r_{97} r_{98} r_{99} CR_0
S_0 s_0 s_1 ······ s_{97} s_{98} s_{99} CS_0
R_1 r_0 r_1 ······ r_{97} r_{98} r_{99} CR_1
S_1 s_0 s_1 ······ s_{97} s_{98} s_{99} CS_1
R_{97} r_0 r_1 ······ r_{97} r_{98} r_{99} CR_{97}
S_{97} s_0 s_1 ······ s_{97} s_{98} s_{99} CS_{97}
R_{98} r_0 r_1 ······ r_{97} r_{98} r_{99} CR_{98}
S_{98} s_0 s_1 ······ s_{97} s_{98} s_{99} CS_{98}
R_{99} r_0 r_1 ······ r_{97} r_{98} r_{99} CR_{99}
S_{99} s_0 s_1 ······ s_{97} s_{98} s_{99} CS_{99}
Known initially
Calculated

Fig. 2: Constructing the state S_0. Starting from PRGA round 99, any bit calculated at PRGA round i is used to determine state bits of round $i-1$.

and r_0^1 is given by $r_0^0 \cdot CR_0 + r_{99}^0$. Once all the bits of R_1 are known, all the bits of R_2 can be determined by calculating

$$r_i^2 = r_{i-1}^1 + CR_1 \cdot r_i^1 + \alpha_i \cdot r_{99}^1,\ \forall i \in [1,99],$$

and $r_0^2 = r_0^1 \cdot CR_1 + r_{99}^1$. Similarly all the bits of the states $R_3, R_4, \ldots, R_{99}$ can be calculated successively. As before, we begin by observing that the functions β_i for all values of $i \in [1,99]$ are of the form

$$\beta_i(\cdot) = s_{i-1} + \lambda_i \cdot (s_{67} + r_{33}) \cdot s_{99} + \hat{\beta}_i(s_i, s_{i+1}, \ldots, s_{99}),$$

where $s_{67} + r_{33}$ is the value of $CONTROL\_BIT\_S$ and $\hat{\beta}_i$ is a function that depends on $s_i, s_{i+1}, \ldots, s_{99}$ but not any of $s_0, s_1, \ldots, s_{i-1}$. $\lambda_i = 1$ if $FB0_i \neq FB1_i$ (these are bit-sequences related to the design of MICKEY 2.0, see [1]) and is 0 otherwise.

Now consider the following equation governing s_{99}^{99}:

$$s_{99}^{99} = \beta_{99}(R_{98}, S_{98}) = s_{98}^{98} + \lambda_{99} \cdot CS_{98} \cdot s_{99}^{98} + \hat{\beta}_{99}(s_{99}^{98}).$$

In the above equation s_{98}^{98} is the only unknown and it appears as a linear term, and so its value can be calculated immediately. We therefore know the values of the 2 state bits of S_{98}: s_{99}^{98}, s_{98}^{98}. Similarly consider the equations involving s_{99}^{98}, s_{98}^{98} :

$$s_{99}^{98} = s_{98}^{97} + \lambda_{99} \cdot CS_{97} \cdot s_{99}^{97} + \hat{\beta}_{99}(s_{99}^{97}),$$

$$s_{98}^{98} = s_{97}^{97} + \lambda_{98} \cdot CS_{97} \cdot s_{99}^{97} + \hat{\beta}_{98}(s_{98}^{97}, s_{99}^{97}).$$

As before, s_{98}^{97} is the lone unknown term in the first equation whose value can be determined immediately. After this, s_{97}^{97} becomes the only unknown linear term in the next equation whose value can also be obtained easily. Thus we know 3 bits of S_{97}: $s_{97+i}^{97},\ i = 0, 1, 2$. Continuing in such a bottom-up manner, we can successively determine 4 bits of S_{96}, 5 bits of S_{95} and eventually all the 100 bits of S_0. (The process is explained pictorially in Figure 2.) □

Table 1: The functions θ_i

i	$\theta_i(\cdot)$
0	$r_0 + s_0$
1	$r_0 \cdot r_{67} + r_0 \cdot s_{34} + r_{99} + s_{99}$
2	$r_0 \cdot r_{66} \cdot r_{67} + r_0 \cdot r_{66} \cdot s_{34} + r_0 \cdot r_{67} \cdot r_{99} +$ $r_0 \cdot r_{67} \cdot s_{33} + r_0 \cdot r_{67} \cdot s_{34} \cdot s_{35} + r_0 \cdot r_{67} \cdot s_{34} +$ $r_0 \cdot r_{67} + r_0 \cdot r_{99} \cdot s_{34} + r_0 \cdot s_{33} \cdot s_{34} + r_0 \cdot s_{34} \cdot s_{35} +$ $r_{33} \cdot s_{99} + r_{66} \cdot r_{99} + r_{67} \cdot r_{99} \cdot s_{34} + r_{98} + r_{99} \cdot s_{33} +$ $r_{99} \cdot s_{34} \cdot s_{35} + r_{99} \cdot s_{34} + r_{99} + s_{67} \cdot s_{99} + s_{98}$

3.1 Faulting specific bits of R, S

The output key-stream bits $z_t, z_{t+1}, \ldots$ can also be expressed as polynomial functions over R_t, S_t. We have

$$\begin{aligned} z_t &= r_0^t + s_0^t = \theta_0(R_t, S_t), \\ z_{t+1} &= r_0^{t+1} + s_0^{t+1} \\ &= \rho_0(R_t, S_t) + \beta_0(R_t, S_t) = \theta_1(R_t, S_t), \\ z_{t+2} &= r_0^{t+2} + s_0^{t+2} \\ &= \rho_0(R_{t+1}, S_{t+1}) + \beta_0(R_{t+1}, S_{t+1}) = \theta_2(R_t, S_t). \end{aligned}$$

The exact forms of $\theta_0, \theta_1, \theta_2$ are given in Table 1.

In the rest of this section we will assume that the adversary can (a) re-key the device containing the cipher with the original Key-IV, (b) apply faults to specific bit locations in the R, S registers and (c) exercise control over the timing of fault injection. Note that (b) is a stronger assumption, but we do not need it in our attack. We are using this assumption here to build a subroutine. In the next sub-section we shall demonstrate how the adversary can partially identify the location of any fault injected at a random position by comparing the faulty and fault-free key-streams.

We observe the following differential properties of the functions $\theta_0, \theta_1, \theta_2$.

- $\theta_1(\ldots, r_{67}, \ldots) + \theta_1(\ldots, 1 + r_{67}, \ldots) = r_0$,
- $\theta_1(r_0, \ldots) + \theta_1(1 + r_0, \ldots) = s_{34} + r_{67}$,
- $\theta_2(\ldots, s_{99}) + \theta_2(\ldots, 1 + s_{99}) = s_{67} + r_{33}$.

These differential properties have the following immediate implications.

$$z_{t+1} + z_{t+1,\Delta r_{67}}(t) = r_0^t \tag{1}$$

$$z_{t+1} + z_{t+1,\Delta r_0}(t) = CR_t \tag{2}$$

$$z_{t+2} + z_{t+2,\Delta s_{99}}(t) = CS_t \tag{3}$$

The above equations hold for all the values of $t = 0, 1, 2, \ldots$. This implies that if the adversary is able to re-key the device with the original Key-IV pair multiple times and apply faults at the PRGA rounds $t = 0, 1, 2, 3, \ldots, 100$ at precisely[1] the R register locations $0, 67$ and the S register location 99, then by observing the difference between the fault-free and faulty key-stream bits, she would be able to recover the values of r_0^t, CR_t, CS_t for all values of $t = 0, 1, 2, \ldots, 100$. The fault at each register location must be preceded by rekeying.

3.1.1 Determining the other bits

Hereafter, the values s_0^t for all $t = 0, 1, 2, \ldots, 100$ may be found by solving: $s_0^t = z_t + r_0^t$. Since $\beta_0(\cdot) = s_{99}$, this implies that $s_0^{t+1} = s_{99}^t,\ \forall t = 0, 1, 2, \ldots$. Therefore, calculating the values of $s_0^t,\ \forall t \in [1, 100]$ is the same as calculating $s_{99}^t,\ \forall t \in [0, 99]$. The values of $r_{99}^t,\ \forall t \in [0, 99]$ are obtained as follows. Consider the equation for z_{t+1}:

$$\begin{aligned} z_{t+1} = \theta_1(R_t, S_t) &= r_0^t \cdot r_{67}^t + r_0^t \cdot s_{34}^t + r_{99}^t + s_{99}^t \\ &= CR_t \cdot r_0^t + r_{99}^t + s_{99}^t,\ \forall t \in [0, 99]. \end{aligned}$$

Here, r_{99}^t is the only unknown linear term in these equations and hence its value too can be determined immediately. At this point, we have the following state bits with us:

$$[r_0^t,\ r_{99}^t,\ CR_t,\ s_0^t,\ s_{99}^t,\ CS_t],\quad \forall t \in [0, 99].$$

Now by using the techniques presented in Lemma 1, we can determine all the bits of the state R_0. Thereafter using Lemma 2, one can determine all the bits of S_0. Thus we have recovered the entire internal state at the beginning of the PRGA.

3.2 How to identify the random locations where faults are injected

In this subsection we will show how the adversary can identify the locations of randomly applied faults to the registers R and S. Although it will not be possible to conclusively determine the location of faults applied to each and every location of R and the S registers, we will show that the adversary can, with some probability, identify faulty streams corresponding to locations $0, 67$ of R and 99 of S. The adversary will then use the techniques described in Subsection 3.1 to complete the attack.

To help with the process of fault location identification, we define the first and second Signature Vectors for the location ϕ of R as

$$\Psi_{r_\phi}^1[i] = \begin{cases} 1, \text{ if } z_{t+i} = z_{t+i,\Delta r_\phi}(t) \text{ for all } R_t, S_t, \\ 0, \text{ otherwise.} \end{cases}$$

$$\Psi_{r_\phi}^2[i] = \begin{cases} 1, \text{ if } z_{t+i} \neq z_{t+i,\Delta r_\phi}(t) \text{ for all } R_t, S_t, \\ 0, \text{ otherwise.} \end{cases}$$

for $i = 0, 1, 2, \ldots, l - 1$. Here $l \approx 40$ is a suitably chosen constant.

Remark 1 The value of l should be large enough so that one can differentiate, with probability almost 1, 100 randomly generated bit sequences over GF(2) by comparing the first l bits of each sequence. This requires the value of l to be at least $2 \cdot \log_2 100 \approx 14$. We take $l = 40$, as computer simulations show that this value of l is sufficient to make a successful distinction with high probability.

Similarly one can define Signature Vectors for any location ϕ the register S.

$$\Psi_{s_\phi}^1[i] = \begin{cases} 1, \text{ if } z_{t+i} = z_{t+i,\Delta s_\phi}(t) \text{ for all } R_t, S_t, \\ 0, \text{ otherwise.} \end{cases}$$

$$\Psi_{s_\phi}^2[i] = \begin{cases} 1, \text{ if } z_{t+i} \neq z_{t+i,\Delta s_\phi}(t) \text{ for all } R_t, S_t, \\ 0, \text{ otherwise.} \end{cases}$$

The task for the fault location identification routine is to determine the fault location ϕ of R (or S) by analyzing the difference between the sequences $z_t, z_{t+1}, \ldots$ and $z_{t,\Delta r_\phi}(t), z_{t+1,\Delta r_\phi}(t), \ldots$ (or $z_{t,\Delta s_\phi}(t), \ldots$) by using the Signature Vectors $\Psi_{r_\phi}^1, \Psi_{r_\phi}^2$ (or $\Psi_{s_\phi}^1, \Psi_{s_\phi}^2$). Note that the i^{th} bit of $\Psi_{r_\phi}^1$ is 1 if and only if the $(t + i)^{th}$ key-stream bits produced by R_t, S_t and $R_{t,\Delta r_\phi}(t), S_{t,\Delta r_\phi}(t)$

[1] We would again like to point out that our actual attack does not need precise fault injection at all locations of R, S. This will be explained in the next sub-section.

are the same for all choices of the internal state R_t, S_t and that i^{th} bit of $\Psi^2_{r_\phi}$ is 1 if the above key-stream bits are different for all choices of the internal state.

The concept of Signature Vectors to deduce the location of a randomly applied fault was introduced in [4]. However the analysis of [4] cannot be reproduced for MICKEY 2.0, since a lot of different register locations have the same Signature Vector. However one can observe the following which are important to mount the attack.

Theorem 1 *The following statements hold for the Signature Vectors $\Psi^1_{r_\phi}, \Psi^2_{r_\phi}$, $\Psi^1_{s_\phi}, \Psi^2_{s_\phi}$ of MICKEY 2.0.*

A. $\Psi^1_{r_\phi}[0] = 1, \forall\phi \in [1, 99]$ *and* $\Psi^2_{r_0}[0] = 1$.
B. $\Psi^1_{r_\phi}[0] = \Psi^1_{r_\phi}[1] = 1, \forall\phi \in [1, 99] \setminus \{67, 99\}$.
C. $\Psi^2_{r_{99}}[1] = 1,$ *and* $\Psi^2_{r_{67}}[1] = 0$.
D. $\Psi^1_{s_\phi}[0] = 1, \forall\phi \in [1, 99]$ *and* $\Psi^2_{s_0}[0] = 1$.
E. $\Psi^1_{s_\phi}[0] = \Psi^1_{s_\phi}[1] = 1, \forall\phi \in [1, 99] \setminus \{34, 99\}$.
F. $\Psi^2_{s_{99}}[1] = 1,$ *and* $\Psi^2_{s_{34}}[1] = 0$.

Proof We present the proof for Case **A**. The proofs for the remaining cases are similar and those are available in Appendix A.

A. We have

$$\begin{aligned} z_t + z_{t,\Delta r_0}(t) &= \theta_0(R_t, S_t) + \theta_0(R_{t,\Delta r_0}(t), S_{t,\Delta r_0}(t)) \\ &= (r^t_0 + s^t_0) + (1 + r^t_0 + s^t_0) \\ &= 1,\ \forall R_t, S_t \in \{0, 1\}^{100}. \end{aligned}$$

So, $\Psi^2_{r_0}[0] = 1$. Also θ_0 is not a function of any r_i, s_i for $i \in [1, 99]$ and so

$$\theta_0(R_{t,\Delta r_\phi}(t), S_{t,\Delta r_\phi}(t)) = \theta_0(R_t, S_t)\ \forall\phi \in [1, 99]$$

and so we have

$$\begin{aligned} z_t + z_{t,\Delta r_\phi}(t) &= \theta_0(R_t, S_t) + \theta_0(R_{t,\Delta r_\phi}(t), S_{t,\Delta r_\phi}(t)) \\ &= 0,\ \forall\phi \in [1, 99],\ \forall R_t, S_t \in \{0, 1\}^{100}. \end{aligned}$$

So, $\Psi^1_{r_\phi}[0] = 1$ for all $\phi \in [1, 99]$.

Thus the proof. □

Now, consider the attack scenario in which the adversary is able to re-key the device with the same Key-IV multiple number of times and inject a single fault at a random location of register R at the beginning of any particular PRGA round $t \in [0, 100]$ and obtain faulty key-streams. She continues the process until she obtains 100 different faulty key-streams corresponding to 100 different fault locations in R and for each $t \in [0, 100]$ (as mentioned earlier this is done by comparing the first l bits of each faulty key-stream sequence). Assuming that every location has equal probability of getting injected by fault, the above process on an average takes around $100 \cdot \sum_{i=1}^{100} \frac{1}{i} \approx 2^{9.02}$ faults [12] and hence re-keyings for each value of $t \in [0, 100]$ and hence a total of $101 \cdot 2^{9.02} \approx 2^{15.68}$ faults. The process has to be repeated for the S register, and so the expected number of faults is $2 \cdot 2^{15.68} = 2^{16.68}$.

If we define the vectors $Z_t = [z_t, z_{t+1}, \ldots, z_{t+l-1}]$ and $\Delta_{r_\phi} Z_t = [z_{t,\Delta r_\phi}(t), z_{t+1,\Delta r_\phi}(t), \ldots, z_{t+l-1,\Delta r_\phi}(t)]$, then the adversary at this point has knowledge of the 100 differential key-streams $\eta_{t,r_\phi} = Z_t + \Delta_{r_\phi} Z_t$ for each value of $t \in [0, 100]$. The adversary, however, does not know the exact fault location corresponding to any differential stream, i.e., she has been unable to assign fault location labels to any of the differential streams. With this information in hand, we shall study the implications of the observations **A** to **F**.

Implication of A: For any $t \in [0, 100]$, $\Psi^2_{r_0}[0] = 1$ guarantees that there is at least one differential stream with $\eta_{t,r_\phi}[0] = 1$ whereas $\Psi^1_{r_\phi}[0] = 1, \forall\phi \in [1, 99]$ guarantees that that there is exactly one differential stream with this property. This implies that out of the 100 differential streams for any PRGA round t the one and only differential stream with this property must have been produced due to a fault on the 0^{th} location in R. Labelling of this stream helps us determine the values of CR_t for all $t \in [0, 100]$ from Eqn. (2).

Implication of B, C: Once the differential stream corresponding to the 0^{th} location has been labelled we now turn our attention to the remaining 99 streams. Statement **B** guarantees that of the remaining 99 streams at least 97 have the property:

(P1) $\eta_{t,r_\phi}[0] = \eta_{t,r_\phi}[1] = 0$.

Statement **C** guarantees that the number of streams with the property:

(P2) $\eta_{t,r_\phi}[0] = 0, \eta_{t,r_\phi}[1] = 1$,

is at most 2 and at least 1. If the number of streams that satisfy (P1) is 98 and (P2) is 1, then the lone stream satisfying (P2) must have been produced due to fault on location 99 of R. This immediately implies that $\eta_{t,r_{67}}[1] = 0$ which by Eqn. (1) in turn implies that $r^t_0 = 0$. Else if the number of streams satisfying (P1) is 97 and (P2) is 2 then it implies that the streams satisfying (P2) were produced due to faults in location $67, 99$ of R. This implies $\eta_{t,r_{67}}[1] = r^t_0 = 1$.

Repeating the entire process on Register S, one can similarly obtain the vectors $\Delta_{s_\phi} Z_t$ and the differential streams $\eta_{t,s_\phi} = Z_t + \Delta_{s_\phi} Z_t$ for all values of $t \in [0, 100]$. As before the streams η_{t,s_ϕ} are unlabeled. Let us now study the implications of **D, E, F**.

Implication of D: For any $t \in [0,100]$, $\Psi^2_{s_0}[0] = 1$ guarantees that there is at least one differential stream with $\eta_{t,s_\phi}[0] = 1$ whereas $\Psi^1_{s_\phi}[0] = 1, \forall \phi \in [1,99]$ guarantees that that there is exactly one differential stream with this property. This implies that out of the 100 differential streams for any PRGA round t the one and only differential stream with this property must have been produced due to a fault on the 0^{th} location in S.

Implication of E, F: Once the differential stream corresponding to the 0^{th} location has been labelled we now turn our attention to the remaining 99 streams. The statement **E** guarantees that of the remaining 99 streams at least 97 have the property

(P3) $\eta_{t,s_\phi}[0] = \eta_{t,s_\phi}[1] = 0$.

Statement **F** guarantees that the number of streams with the property

(P4) $\eta_{t,s_\phi}[0] = 0, \eta_{t,s_\phi}[1] = 1$,

is at most 2 and at least 1.

Case 1. If the number of streams that satisfy (P3) is 98 and (P4) is 1 then the lone stream satisfying (P4) must have been produced due to fault at location 99 of S. Once the stream corresponding to location 99 of S had been labelled, we can use Eqn (3) to determine $CS_t = \eta_{t,s_{99}}[2]$.

Case 2. If the number of streams satisfying (P3) is 97 and (P4) is 2 then it implies that the streams satisfying (P4) had been produced due to faults in location $34, 99$ of S.

(i) Now if the bit indexed 2 of both these vectors are equal then we can deduce $CS_t = \eta_{t,s_{99}}[2] = \eta_{t,s_{34}}[2]$.

(ii) A confusion occurs when $\eta_{t,s_{99}}[2] \neq \eta_{t,s_{34}}[2]$. In such a situation we would be unable to conclusively determine the value of CS_t.

Assuming independence, we assume that **Cases 1, 2** have equal probability of occurrence. Given that **Case 2** occurs, we can also assume that one of **2(i), 2(ii)** occurs with equal probability. Therefore, the probability of confusion, i.e., the probability that we are unable to determine the value of CS_t for any t can be estimated as $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. Let γ denote the number of $t \in [0,100]$ such that CS_t cannot be conclusively determined then γ is distributed according to $\gamma \sim Binomial(101, \frac{1}{4})$. Therefore the expected value of γ is $E(\gamma) = 101 \cdot \frac{1}{4} = 25.25$. Also the probability that

$$P(\gamma > 35) = \sum_{k=36}^{101} \binom{101}{k} \left(\frac{1}{4}\right)^k \left(\frac{3}{4}\right)^{101-k} \approx 0.01.$$

In such a situation, the adversary must guess the γ number of bit values of CS_t to perform the attack, which implies that the adversary must perform the calculations in Section 3.1 and Lemma 1, Lemma 2 a total of 2^γ times to complete the attack. For the correct value of the guesses, the calculated state R_0, S_0 will produce the given fault-free key-stream sequence. We present a complete description of the attack in Algorithm 1.

```
Generate and record the fault-free key-stream $z_0, z_1, z_2, \ldots$
for some Key-IV $K, IV$
$t \leftarrow 0$;
while $t \leq 100$ do
    while 100 different faulty key-stream sequences $\Delta_{r_\phi} Z_t$
    have not been obtained do
        Re-key the cipher with Key-IV $K, IV$;
        Inject a fault at a random unknown location
        $\phi \in [0,99]$ in $R$ at PRGA round $t$;
        Record the faulty key-stream sequence $\Delta_{r_\phi} Z_t$;
    end
    $t \leftarrow t + 1$;
end
Calculate $r_0^t, CR_t,\ \forall t \in [0,100]$ using A, B, C;
$t \leftarrow 0$;
while $t \leq 100$ do
    while 100 different faulty key-stream sequences $\Delta_{s_\phi} Z_t$
    have not been obtained do
        Re-key the cipher with Key-IV $K, IV$;
        Inject a fault at a random unknown location
        $\phi \in [0,99]$ in $S$ at PRGA round $t$;
        Record the faulty key-stream sequence $\Delta_{s_\phi} Z_t$;
    end
    $t \leftarrow t + 1$;
end
Using D, E, F calculate $CS_t$,  for all such $t \in [0,100]$ for
which there is no confusion;
Let the number of undecided $CS_t$ bits be $\gamma$;
for Each of the $2^\gamma$ guesses of the undecided $CS_t$'s do
    Use techniques of Subsection 3.1 to compute
    $r_0^t,\ r_{99}^t,\ CR_t,\ s_0^t,\ s_{99}^t,\ CS_t,\ \ \forall t \in [0,99]$;
    Use Lemma 1, Lemma 2 to compute $R_0, S_0$;
    if $R_0, S_0$ produce the sequence $z_0, z_1, z_2, \ldots$ then
        Output the required state $R_0, S_0$;
    end
end
```

Algorithm 1: Fault Attack against MICKEY 2.0

3.3 Issues related to the length of the IV

It is known that MICKEY 2.0 employs a variable length IV of length at most 80. So if v is the length of the IV then the cipher will run for $v + 80$ (Key loading) + 100 (Preclock) clock rounds before entering the PRGA phase. Our attack requires that the first faults are to be injected at the beginning of the PRGA. In order to do that the adversary must know the value of v. This not a strong assumption as IVs are assumed to be known. However even if the adversary does not know the IV or its length the attack can be performed. Since $0 \leq v \leq 80$ must be satisfied, the strategy of the adversary who does not know the value of v will be as follows. She will inject the first set of faults at clock round 260 which corresponds to the PRGA round $p = 260 - 180 - v = 80 - v$. After performing the attack, the adversary will end up constructing the internal state R_p, S_p instead of R_0, S_0. Finding the value of p by looking at the fault-free key-stream sequence is straightforward. However, finding R_0, S_0 is a slightly stronger result because, as reported in [17], there is a finite entropy loss for each state update operation in the MICKEY PRGA.

3.4 Complexity of the Attack

As mentioned in Section 3.2, the attack requires the adversary to obtain 100 different faulty key-streams corresponding to all the fault locations in R for PRGA rounds $t \in [0, 100]$. This requires $101 \cdot 100 \cdot \sum_{i=1}^{100} \frac{1}{k} \approx 2^{15.68}$ faults on an average. The same process must be repeated for the register S and hence the expected number of total faults is $2^{16.68}$. The computational overload comes from guessing the γ bits of CS_t which cannot be found by observing the differential key-streams. This requires a computational effort proportional to 2^γ. Since γ is distributed according to $Binomial(101, \frac{1}{4})$, the expected value of γ is 25.25. The expected value of the computation complexity is therefore given by

$$E(2^\gamma) = \sum_{k=0}^{101} \binom{101}{k} \left(\frac{1}{4}\right)^k \left(\frac{3}{4}\right)^{101-k} 2^k \approx 2^{32.5}.$$

4 Case of Multiple bit faults

In this section we explore the situation in which the adversary is unable to induce a single bit flip of the internal state every time she injects a fault. We assume that the injection of fault may affect the bit values of at most three consecutive locations of the state (indeed this can be extended further, but the analysis will become very tedious). This gives rise to three situations (a) the attacker flips exactly one register bit (100 possibilities), (b) she flips 2 consecutive locations $i, i+1$ of R or S (99 possibilities), (c) she flips 3 consecutive locations $i, i+1, i+2$ of R or S (98 possibilities). Studying such a model makes sense if we attack an implementation of MICKEY where the register cells of the R and S registers are physically positioned linearly one after the other. Now, this attack scenario gives rise to $100 + 99 + 98 = 297$ different instances of faults due to any single fault injection, and we will assume that all these instances are equally likely to occur. As before we will assume that the adversary can re-key the device with the original Key-IV and obtain all the 297 faulty streams for any PRGA round $t \in [0, 100]$ by randomly injecting faults in either the R or S register. For each PRGA round, the attacker thus needs around $297 \cdot \sum_{i=1}^{297} \frac{1}{i} \approx 2^{10.7}$ faults. Thus the fault requirement for the register R is $101 \cdot 2^{10.7} = 2^{17.4}$. The process has to be repeated for the S register and so the total fault requirement is $2 \cdot 2^{17.4} = 2^{18.4}$.

Let $\mathbf{\Phi} = \{\phi_1, \phi_2, \ldots, \phi_k\}$ denotes the set of indices of k $(k \leq 3)$ continuous locations in the R (or S) register. The notations $R_{t,\Delta r_{\mathbf{\Phi}}}(t_0), S_{t,\Delta r_{\mathbf{\Phi}}}(t_0)$, $R_{t,\Delta s_{\mathbf{\Phi}}}(t_0)$, $S_{t,\Delta s_{\mathbf{\Phi}}}(t_0)$, $z_{i,\Delta r_{\mathbf{\Phi}}}(t_0), \Delta_{r_{\mathbf{\Phi}}} Z_t, \eta_{t,r_{\mathbf{\Phi}}}$, $\Psi^1_{r_{\mathbf{\Phi}}}[i], \Psi^2_{r_{\mathbf{\Phi}}}[i]$, and $\Psi^1_{s_{\mathbf{\Phi}}}[i]$, $\Psi^2_{s_{\mathbf{\Phi}}}[i]$ will be used in their usual meanings in the context of multiple faults at all locations in $\mathbf{\Phi}$.

To begin with, in the single bit fault case, the attack depends on the successful identification of the faulty streams produced due to faults in locations $0, 67$ of R and 99 of S. In the multiple bit fault case too, the success of the attack depends on the identification of faulty streams that have been produced due to faults in these locations. We will deal each of these situations separately.

4.1 The bit r_0 is affected.

This could happen in 3 ways: a) r_0 alone is toggled, b) r_0, r_1 are toggled, c) r_0, r_1, r_2 are toggled. Let us state the following technical result.

Proposition 1 *$\Psi^1_{r_{\mathbf{\Phi}}}[0] = 1, \forall \mathbf{\Phi}$ such that $0 \notin \mathbf{\Phi}$, but $\Psi^2_{r_{\mathbf{\Phi}}}[0] = 1, \forall \mathbf{\Phi}$ that contain 0.*

Proof Since θ_0 is a function of r_0, s_0 only we will have

$$\begin{aligned} z_t + z_{t,\Delta r_{\mathbf{\Phi}}}(t) &= \theta_0(R_t, S_t) + \theta_0(R_{t,\Delta r_{\mathbf{\Phi}}}(t), S_{t,\Delta r_{\mathbf{\Phi}}}(t)) \\ &= \begin{cases} 0, \text{if } 0 \notin \mathbf{\Phi}, \\ 1, \text{if } 0 \in \mathbf{\Phi} \end{cases} \end{aligned}$$

Hence the result. □

This implies that any faulty stream with its first bit different from the fault-free first bit must have been produced due to a fault that has affected r_0 and vice versa. Thus 3 out of the 297 faulty streams have this property and they can be identified easily. Furthermore since $\theta_1(R_t, S_t) + \theta_1(R_{t,\Delta r_{\mathbf{\Phi}}}(t), S_{t,\Delta r_{\mathbf{\Phi}}}(t)) = s^t_{34} + r^t_{67} = CR_t$ $\forall \mathbf{\Phi}$ containing 0, the second bit in the all these faulty streams are equal and the difference of this bit with the second fault-free bit gives us the value of CR_t.

4.2 The bits r_{67} and r_{99} are affected.

r_{67} could be affected in 6 ways : a) r_{67} alone is toggled, b) r_{66}, r_{67} are toggled, c) r_{67}, r_{68} are toggled, d) r_{65}, r_{66}, r_{67} are toggled, e) r_{66}, r_{67}, r_{68} are toggled and f) r_{67}, r_{68}, r_{69} are toggled. Also note that r_{99} could be affected in 3 ways: a) r_{99} is toggled, b) r_{98}, r_{99} are toggled and c) r_{97}, r_{98}, r_{99} are all toggled. Again we state the following propositions.

Proposition 2 *$\Psi^1_{r_\Phi}[0] = \Psi^1_{r_\Phi}[1] = 1, \forall \mathbf{\Phi}$ such that the indices $0, 67, 99 \notin \mathbf{\Phi}$.*

Proposition 3 *If $99 \in \mathbf{\Phi}$ then $\Psi^2_{r_\Phi}[1] = 1$. If $67 \in \mathbf{\Phi}$ then $\Psi^2_{r_\Phi}[1] = 0$.*

Proof Note that θ_0 is a function of only r_0, s_0 and θ_1 is a function of $r_0, r_{67}, r_{99}, s_{34}, s_{99}$ only.

$$z_{t+1} + z_{t+1,\Delta r_\Phi}(t) = \begin{cases} 0, & \text{if } 0, 67, 99 \notin \mathbf{\Phi}, & \text{(G)} \\ CR_t, & \text{if } 0 \in \mathbf{\Phi}, & \text{(H)} \\ r_0^t, & \text{if } 67 \in \mathbf{\Phi}, & \text{(K)} \\ 1, & \text{if } 99 \in \mathbf{\Phi}. & \text{(L)} \end{cases}$$

Hence the result. □

In the above, (G) implies that out of the remaining 294 differential streams at least $294 - 6 - 3 = 285$ satisfy

(P5) $\eta_{t,r_\Phi}[0] = \eta_{t,r_\Phi}[1] = 0$

and (L) implies that the number of differential streams with the property

(P6) $\eta_{t,r_\Phi}[0] = 0, \eta_{t,r_\Phi}[1] = 1$

is at least 3. A direct implication of (K) is that if the number of differential streams satisfying (P5) is 285 and (P6) is 9 then $r_0^t = 1$ and on the other hand if, the number of streams satisfying (P5) is 291 and (P6) is 3 then $r_0^t = 0$. These are exclusive cases, i.e., the number of streams satisfying (P5) can be either 285 or 291. Since the values of r_0^t, CR_t for all $t \in [0, 100]$ are now known, the attacker can now use the techniques of Section 3.1 and Lemma 1 to calculate the entire initial state R_0.

4.3 The bits s_0, s_{34} and s_{99} are affected.

Following previous descriptions, we know that there are respectively 3, 6, 3 possibilities of faults affecting s_0, s_{34}, s_{99}. Again, we present the following technical results before describing the attack.

Proposition 4 *$\Psi^1_{s_\Phi}[0] = 1, \forall \mathbf{\Phi}$ such that $0 \notin \mathbf{\Phi}$, but $\Psi^2_{s_\Phi}[0] = 1, \forall \mathbf{\Phi}$ that contain 0.*

Proposition 5 *$\Psi^1_{s_\Phi}[0] = \Psi^1_{s_\Phi}[1] = 1, \forall \mathbf{\Phi}$ such that the indices $0, 34, 99 \notin \mathbf{\Phi}$.*

Proposition 6 *If $99 \in \mathbf{\Phi}$ then $\Psi^2_{s_\Phi}[1] = 1$. If $34 \in \mathbf{\Phi}$ then $\Psi^2_{s_\Phi}[1] = 0$.*

Proof The proof is similar to those of previous propositions. Since θ_0 is a function of only r_0, s_0 and θ_1 is a function of $r_0, r_{67}, r_{99}, s_{34}, s_{99}$ only, we have

$$\begin{aligned} z_t + z_{t,\Delta s_\Phi}(t) &= \theta_0(R_t, S_t) + \theta_0(R_{t,\Delta s_\Phi}(t), S_{t,\Delta s_\Phi}(t)) \\ &= \begin{cases} 0, \text{ if } 0 \notin \mathbf{\Phi}, \\ 1, \text{ if } 0 \in \mathbf{\Phi} \end{cases} \end{aligned}$$

$$z_{t+1} + z_{t+1,\Delta s_\Phi}(t) = \begin{cases} 0, & \text{if } 34, 99 \notin \mathbf{\Phi}, & \text{(M)} \\ r_0^t, & \text{if } 34 \in \mathbf{\Phi}, & \text{(N)} \\ 1, & \text{if } 99 \in \mathbf{\Phi}. & \text{(O)} \end{cases}$$

□

Proposition 4 proves that there are exactly 3 differential streams out of 297 which have $\eta_{s_\Phi}[0] = 1$. Further, (M) implies that of the remaining 294 streams, at least $294 - 3 - 6 = 285$ satisfy

(P7) $\eta_{t,s_\Phi}[0] = \eta_{t,s_\Phi}[1] = 0$

and (O) implies that the number of streams that satisfy

(P8) $\eta_{t,s_\Phi}[0] = 0, \eta_{t,s_\Phi}[1] = 1$

is at least 3.

4.3.1 CASE I.

If the number of streams that satisfy (P7) is 291 and (P8) is 3 then the streams satisfying (P8) must have been produced due to the faults affecting s_{99}. For these streams $\eta_{s_\Phi}[2]$ is given by:

$$z_{t+2} + z_{t+2,\Delta s_\Phi}(t) = \begin{cases} CS_t, & \text{if } \mathbf{\Phi} = \{99\}, \\ 1 + CS_t, & \text{if } \mathbf{\Phi} = \{98, 99\} \\ 1 + CS_t. & \text{if } \mathbf{\Phi} = \{97, 98, 99\} \end{cases}$$

So, for 2 of these 3 streams we have $\eta_{s_\Phi}[2] = 1 + CS_t$. Hence, our strategy will be to look at the bit indexed 2 of these 3 streams. Two of them will be equal and we designate that value as $1 + CS_t$.

4.3.2 CASE II.

If the number of streams that satisfy (P7) is 285 and (P8) is 9 then the streams have been produced due to faults that have affected s_{34} and s_{99}. We have the identity

$$\sum_{\mathbf{\Phi}:\ 34 \in \mathbf{\Phi}} \eta_{t,s_\Phi}[2] = r_0^t \cdot r_{67}^t \cdot s_{34}^t + r_{99}^t \cdot s_{34}^t.$$

Therefore, the sum of the bits indexed 2 of all the differential streams that satisfy (P8) is

$$\sum_{\mathbf{\Phi}:\ 34 \text{ or } 99 \in \mathbf{\Phi}} \eta_{t,s_\Phi}[2] = CS_t + r_0^t \cdot r_{67}^t \cdot s_{34}^t + r_{99}^t \cdot s_{34}^t.$$

At this time the entire initial state of the register R and all the values of CR_t for $t \in [0, 100]$ are known to us. Hence, by Lemma 2, all values of r_i^t for all $t > 0$ can be calculated by clocking the register R forward. Also, since $CR_t = r_{67}^t + s_{34}^t$ is known, $s_{34}^t = CR_t + r_{67}^t$ can be calculated easily. Therefore, in the previous equation, CS_t becomes the only unknown and thus its value can be calculated immediately.

At this point of time we have r_0^t, CR_t, CS_t for all values of $t = 0, 1, 2, \ldots, 100$. Now using the techniques of Section 3.1 and Lemmata 1, 2, we will be able to determine the entire initial state R_0, S_0. Note that using this fault model although the fault requirement increases, the adversary does not have to bear the additional computational burden of guessing γ values of CS_t.

5 Improvement Using SAT Solver

The main idea of algebraic cryptanalysis is to solve multivariate polynomial systems that describe a cipher and this has been successfully exploited in DFA also. For a very brief introduction in this, one may refer [19, Section 5]. The DFA on Trivium [19] requires only 2 faults. Our very recent work on DFA against Grain family [20] also shows that the number of faults can be reduced significantly (not more than 10). With this motivation, we tried to exploit similar ideas for fault attacks against MICKEY 2.0. Our analysis shows improvements over our result in Section 3.4; however, not as significant as what could be achieved for Trivium or Grain family. Nevertheless, we identify several other combinatorial patterns towards the improved DFA against MICKEY 2.0 in this section. We will start with the following simple technical result.

Lemma 3 *Suppose $r_0^t = 0$ for some $t \in [0, 99]$. Then the location of a random fault can be identified deterministically when it injects the 99^{th} location of R.*

Proof This follows from Theorem 1B, 1C. We have already seen in Section 3.2, that for any t, if $r_0^t = 0$, then the number of differential streams satisfying (P2) is exactly 1. It follows from Theorem 1B, 1C, that this differential stream must have been produced due to fault on location 99 of R. □

Now we will prove another result when $r_0^t = 1$.

Lemma 4 *Suppose $r_0^t = 1$ for some $t \in [0, 99]$. Then to decide that r_0^t is indeed 1 and furthermore to find the value of CR_t, one needs to inject around 183.33 faults on average.*

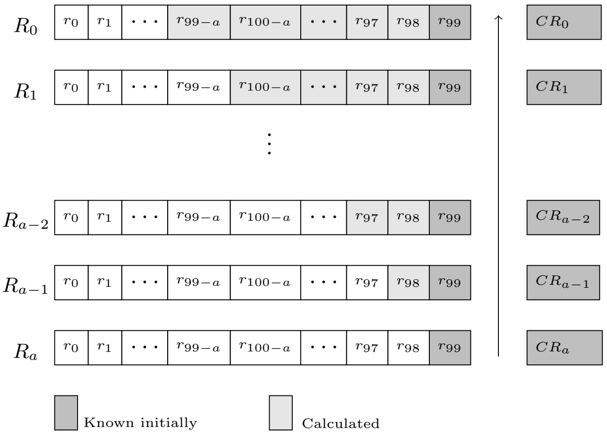


Fig. 3: Constructing the last a bits of the state R_0.

Proof From Theorem 1A, 1B, 1C and their implications, it is clear that if $r_0^t = 1$, then the number of differential streams satisfying (P2) is 2 (produced due to faults on locations 67, 99 of R) and if $r_0^t = 0$, then the number of differential streams satisfying (P2) is 1. Hence for any t, in the process of applying random faults, as soon as the attacker obtains 2 streams satisfying (P2), she can conclude that $r_0^t = 1$.

Also from the implication of Theorem 1A, we know that finding CR_t requires the faulty key-stream from location 0 of R. So, for any fixed t, if $r_0^t = 1$, then deducing r_0^t and CR_t requires faulty key-streams from locations 0, 67, 99 of R only. By injecting random faults, the attacker can expect to inject these 3 locations by applying $100 + \frac{100}{2} + \frac{100}{3} = 183.33$ random faults. Hence the result. □

Note that, this is much less than the $2^{9.02}$ faults required to obtain the 100 distinct faulty key-streams corresponding to each fault location in R as discussed in Section 3.2.

Hence when $r_0^t = 1$, we do not need to inject fault at every location of R_t to find the value of r_0^t and CR_t. On the other hand when $r_0^t = 0$, using Lemma 3, we can identify the faulty key-stream resulting from fault on location 99 of R. We will use these faulty key-streams as location of fault is known in our attack.

We will now state a more general form of Lemma 1.

Lemma 5 *Let $a \in [0, 99]$ be an integer. If we assume that r_{99}^t and CR_t are known $\forall t \in [0, a]$, then the state bits $r_{99-a}^0, r_{100-a}^0, \ldots, r_{99}^0$ of the initial state R_0 may be calculated efficiently.*

Proof The proof is exactly similar to that of Lemma 1. In Lemma 1, we started with 1 bit of R_{99}, i.e., r_{99}^{99}, and then worked backwards to calculate the last 2 bits of

R_{98}, 3 bits of R_{97} and in this manner the entire of R_0. In this case we will start with 1 bit of R_a, i.e., r_{99}^a and backtrack to calculate the last 2 bits of R_{a-1}, 3 bits of R_{a-2} and in this manner the bits $r_{99-a}^0, r_{100-a}^0, \dots, r_{99}^0$ of R_0. The process is explained pictorially in Figure 3. □

We will now investigate the situation when attacker injects faults at each round $t \in [0, a]$. Using Theorem 1 and its implications, the attacker can deduce the values of r_0^t and CR_t ∀ $t \in [0, a]$. She can then find the values of r_{99}^t, ∀ $t \in [0, a]$, using the arguments of Section 3.1.1. Then using Lemma 5, she can compute $r_{99-a}^0, r_{100-a}^0, \dots, r_{99}^0$. Now, let us write the state R_0 as

$$[r_0^0, x_1, \dots, x_{98-a}, r_{99-a}^0, r_{100-a}^0, \dots, r_{99}^0],$$

where x_i's are unknown for $1 \le i \le 98 - a$ and r_i^0 are known for $i = 0$ and $99 - a \le i \le 99$. We can write the state S_0 as

$$[y_0, y_1, \dots, y_{99}],$$

where y_i's are unknown for $0 \le i \le 99$.

We will now describe the technique to formulate multivariate equations in x_i, y_i over GF(2) which we will solve using a SAT Solver. We will formulate equations for the fault-free key-stream bits first. We have already seen that the state bits of $R_1, S_1, R_2, S_2, \dots, R_k, S_k, \dots$ can be expressed as polynomials over the state bits of R_0, S_0. However, the algebraic degree and complexity of these polynomials increase exponentially with increasing k. So much so that we could not compute the form of these polynomials for $k > 4$ on a normal Desktop PC. To circumvent this situation, we take resort to introducing new variables at every PRGA round of the cipher.

In the first round of PRGA, we introduce 200 new variables u_i^1 and v_i^1 for $0 \le i \le 99$, where u_i^1 corresponds to the state R_1 and v_i^1 corresponds to S_1. Hence we formulate 201 new equations which are

1. $z_0 = r_0^0 + y_0$
2. $u_i^1 = \rho_i(r_0^0, \dots, x_{99-a}, r_{100-a}, \dots, r_{99}, y_0, \dots, y_{99})$
3. $v_i^1 = \beta_i(r_0^0, \dots, x_{99-a}, r_{100-a}, \dots, r_{99}, y_0, \dots, y_{99})$.

Hence, the states R_1 and S_1, obtained after running one round of PRGA, becomes

$$[u_0^1, \dots, u_{99}^1] \text{ and } [v_0^1, \dots, v_{99}^1]$$

respectively. This technique is repeated in each successive round accompanied by the introduction of 200 new variables. As MICKEY's state update function is highly non linear, this approach enables us to compute the symbolic forms (via a series of equations) of any PRGA state R_T, S_T. Instead, if at each round $k > 0$, the variables u_i^k, v_i^k were replaced by their equivalent algebraic expressions in x_i, y_i, this would never have been possible efficiently. By introducing new variables, after T rounds, we have a total of $201T$ equations.

We will now formulate equations generated due to faulty key-stream bits. The attacker can determine any faulty key-stream conclusively when it has been produced due to fault at location 0 of R. So after T rounds, we have a total of T faulty key-stream sequences generated due to fault on 0^{th} location of R. To use these faulty key-streams, we proceed as follows. Consider the case when an injected fault has toggled the location 0 of R at $t = 0$. We denote this faulty state by the vector

$$[1 + r_0^0, x_1, \dots, x_{98-a}, r_{99-a}^0, r_{100-a}^0, \dots, r_{99}^0] \text{ and}$$

$$[y_0, y_1, \dots, y_{99}].$$

As before we use 200 new variables $\overline{u}_i^1, \overline{v}_i^1$ to the next faulty state. So we again get 201 new equations

1. $z_{0,\Delta r_0}(0) = 1 + r_0^0 + y_0$
2. $\overline{u}_i^1 = \rho_i(1 + r_0^0, \dots, x_{99-a}, r_{100-a}, \dots, r_{99}, \dots, y_{99})$
3. $\overline{v}_i^1 = \beta_i(1 + r_0^0, \dots, x_{99-a}, r_{100-a}, \dots, r_{99}, \dots, y_{99})$.

As before, we repeat the above for T' rounds with 200 new variables in each round. Again, this results in a total of $201T'$ equations. The process can be repeated for fault at any round $t \in [0, T]$. New equations and variables are formulated accordingly in each case.

Again from Lemma 3, we know that ∀ t, we can identify any faulty key-stream sequence produced due to fault on location 99 of R, when $r_0^t = 0$. So whenever $r_0^t = 0$, we can formulate more equations. For example if $r_0^0 = 0$, we start with the state

$$[r_0^0, x_1, \dots, x_{98-a}, r_{99-a}^0, r_{100-a}^0, \dots, 1 + r_{99}^0] \text{ and}$$

$$[y_0, y_1, \dots, y_{99}],$$

and thereafter form equations by the introduction of new variables in each round.

5.1 Experiments

We assume all except the first 25 bits bits of R_0 have been found out by injecting faults and thereafter using Lemma 5, i.e., we take $a = 75$. We need to find S_0 which contains 100 unknown bits. To restrict the total number of equations, we use only first 38 key-stream bits, i.e., we take $T = 38$. We also use faulty key-stream bits for only first $T' = 12$ rounds when the location of a faulty key-stream can be conclusively identified. We feed the equation so formed into the SAT Solver Cryptominisat-2.9.5 [23] available with SAGE 5.7 [24]. The solver is able to find the remaining 125 unknown bits in 1345.80 seconds on an average (averaged over 100 trials) on a PC powered by an Intel Dual Core Processor, with a CPU speed of 1.83 GHz and 2 GB RAM.

Fault Requirement: Since $a = 75$, we need to apply faults in the first 75 PRGA rounds. Among the 75 rounds, we can assume value of r_0^t will be 1 at expected $\frac{75}{2}$ times. Whenever $r_0^t = 1$, by Lemma 4, only 183.33 faults are sufficient. So expected number of faults required in our attack is

$$\frac{75}{2} \cdot 100 \sum_{i=1}^{100} \frac{1}{i} + \frac{75}{2}\left(100 + \frac{100}{2} + \frac{100}{3}\right) \approx 2^{14.68}.$$

Thus we have a four-fold improvement in the number of faults compared to Section 3.4, where expected number of fault was $2^{16.68}$. This is the improvement achieved when we solve non-linear equations using a SAT solver.

5.2 Multiple bit faults

From the discussion in Section 4, it is clear that the attacker can not conclusively determine whether a given faulty key-stream has been produced due to a single bit or a multiple bit fault. Hence the attacker cannot use faulty key-streams to formulate equations. The best she can do is as follows. Find a bits of R_0 by applying faults and then find the remaining bits of R_0, S_0 by formulating equations for the fault-free key-stream bits. By extensive experimentation, we have found that to obtain a solution in reasonable time, the value of a has to be 100, i.e., we need to find out the entire state of R_0 before using the SAT solver. Using the technique of formulating equations using extra variables, which was described in the previous subsection, we were able to find the entire S_0 using the SAT Solver, within 1206.18 seconds on an average (averaged over 100 trials).

Fault Requirement: The number of different $\mathbf{\Phi}$ for which $0 \in \mathbf{\Phi}$ is 3. Assuming $r_0^t = 1$, among the remaining $297 - 3 = 294$ different faulty key-streams, 9 would satisfy (P6). Of these 9, three are due to fault on location 99 and six are due to location 67. However when $r_0^t = 0$, the number of streams satisfying (P6) is only 3. Hence for any t, as soon as the attacker can obtain four different key-streams satisfying (P6), she can conclude $r_0^t = 1$. Hence when $r_0^t = 1$, it can be proved that the attacker requires 187.25 faults on average (See Appendix B for a theoretical justification of this figure) to deduce the value of r_0^t and CR_t.

On the other hand if $r_0^t = 0$, there is a total of 291 different faulty streams which satisfy (P5) and only 3 which satisfy (P6). Now in the process of applying random fault and resetting, as soon as we obtain 286 streams that satisfy (P5), we can conclude that $r_0^t = 0$. Hence in this case, the expected number of faults is approximately $297 \cdot \sum_{i=6}^{291} \frac{1}{i} = 1178.77$.

Thus, the expected number of faults required to find R_0 is

$$\frac{100}{2}\left(187.2 + 1178.77\right) \approx 2^{16.06}.$$

This is more than four-fold improvement over the $2^{18.4}$ faults reported in Section 4.

6 Conclusion

A differential fault attack against the stream cipher MICKEY 2.0 is presented. The work is one of the first cryptanalytic attempts against this cipher and requires reasonable computational effort. The attack works due to the simplicity of the output function and certain register update operations of MICKEY 2.0 and would have been thwarted had these been of a more complex nature. It would be interesting to study efficient countermeasures with minimum tweak in the design.

Given our work in this paper, differential fault attacks are now known against all of the three ciphers in the hardware portfolio of eStream. The attacks on all the 3 ciphers use exactly the same fault model that is similar to what described in this paper. Let us now summarize the fault requirements.

Cipher	State size	Average # Faults
Trivium [16]	288	3.2
Grain v1 [20]	160	≈ 10
MICKEY 2.0	200	$\approx 2^{14.7}$

To the best of our knowledge, there was no published fault attack on MICKEY 2.0. prior to our work. One of the reasons this remained open for such a long time could be that the cipher uses irregular clocking to update its state registers. Hence it becomes difficult to determine the location of a randomly applied fault injected in either the R or S register by simply comparing the faulty and fault-free key-streams. The idea explained in Theorem 1 and its implications are instrumental in mounting the attack. The total number of faults is indeed much higher when we compare it with the other two eStream hardware candidates. However, this seems natural as MICKEY 2.0 has more complex structure than Trivium or Grain v1. This is also important to point out that while Grain and Trivium are susceptible to DFA with very few faults when SAT solvers are exploited, such drastic results could not be attained for MICKEY 2.0.

References

1. S. Babbage and M. Dodd. The stream cipher MICKEY 2.0. ECRYPT Stream Cipher Project Report. Available at `http://www.ecrypt.eu.org/stream/p3ciphers/mickey/mickey_p3.pdf`.
2. S. Babbage and M. Dodd. The stream cipher MICKEY-128 2.0. ECRYPT Stream Cipher Project Report. Available at `http://www.ecrypt.eu.org/stream/p3ciphers/mickey/mickey128_p3.pdf`.
3. S. Banik and S. Maitra. A Differential Fault Attack on MICKEY 2.0. In CHES 2013, LNCS, Vol. 8086, pp. 215–232.
4. S. Banik, S. Maitra and S. Sarkar. A Differential Fault Attack on the Grain Family of Stream Ciphers. In CHES 2012, LNCS, Vol. 7428, pp. 122–139.
5. S. Banik, S. Maitra and S. Sarkar. A Differential Fault Attack on the Grain Family under Reasonable Assumptions. In INDOCRYPT 2012, LNCS, Vol. 7668, pp. 191–208.
6. S. Banik, S. Maitra and S. Sarkar. A Differential Fault Attack on Grain Family under Reasonable Assumptions. In INDOCRYPT 2012, LNCS, Vol. 7668, pp. 191–208.
7. A. Berzati, C. Canovas, G. Castagnos, B. Debraize, L. Goubin, A. Gouget, P. Paillier and S. Salgado. Fault Analysis of Grain-128. In IEEE International Workshop on Hardware-Oriented Security and Trust, 2009, pp. 7–14.
8. E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In CRYPTO 1997, LNCS, Vol. 1294, pp. 513–525.
9. D. Boneh, R. A. DeMillo and R. J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In EUROCRYPT 1997, LNCS, Vol. 1233, pp. 37–51.
10. C. Cid and M. Robshaw (Editors), S. Babbage, J. Borghoff and V. Velichkov (Contributors). The eSTREAM Portfolio in 2012, 16 January 2012, Version 1.0. Available at `http://www.ecrypt.eu.org/documents/D.SYM.10-v1.pdf`.
11. ECRYPT Stream Cipher Project. eSTREAM Portfolio of Stream Ciphers. Latest report published in January, 2012. `http://www.ecrypt.eu.org/stream/`
12. P. Erdős and A. Rényi. On a classical problem of probability theory. Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei 6: 215–220, MR 0150807, 1961. Available at `http://www.renyi.hu/~p_erdos/1961-09.pdf`.
13. B. Gierlichs, L. Batina, C. Clavier, T. Eisenbarth, A. Gouget, H. Handschuh, T. Kasper, K. Lemke-Rust, S. Mangard, A. Moradi and E. Oswald. Susceptibility of eSTREAM Candidates towards Side Channel Analysis. In Proceedings of SASC 2008, available via `http://www.ecrypt.eu.org/stvl/sasc2008/`.
14. J. J. Hoch and A. Shamir. Fault Analysis of Stream Ciphers. In CHES 2004, LNCS, Vol. 3156, pp. 1–20.
15. M. Hojsík and B. Rudolf. Differential Fault Analysis of Trivium. In FSE 2008, LNCS, Vol. 5086, pp. 158–172.
16. M. Hojsík and B. Rudolf. Floating Fault Analysis of Trivium. In INDOCRYPT 2008, LNCS, Vol. 5365, pp. 239–250.
17. J. Hong and W. Kim. TMD-Tradeoff and State Entropy Loss Considerations of stream cipher MICKEY. In INDOCRYPT 2005, LNCS, Vol. 3797, pp. 169–182.
18. S. Karmakar and D. Roy Chowdhury. Fault analysis of Grain-128 by targeting NFSR. In AFRICACRYPT 2011, LNCS, Vol. 6737, pp. 298–315.
19. M. S. E. Mohamed , S. Bulygin and J. Buchmann. Improved Differential Fault Analysis of Trivium. In COSADE 2011, Darmstadt, Germany, February 24–25, 2011.
20. S. Sarkar, S. Banik and S. Maitra. Differential Fault Attack against Grain family with very few faults and minimal assumptions. IACR eprint archive, 2013:494. Available at `http://eprint.iacr.org/2013/494.pdf`.
21. S. P. Skorobogatov. Optically Enhanced Position-Locked Power Analysis. In CHES 2006, LNCS, Vol. 4249, pp. 61–75.
22. S. P. Skorobogatov and R. J. Anderson. Optical Fault Induction Attacks. In CHES 2002, LNCS, Vol. 2523, pp. 2–12.
23. M. Soos. CryptoMiniSat-2.9.5. `http://www.msoos.org/cryptominisat2/`.
24. W. Stein. Sage Mathematics Software. Free Software Foundation, Inc., 2009. Available at `http://www.sagemath.org`. (Open source project initiated by W. Stein and contributed by many).

Appendix A: Proofs for Theorem 1B-F

B. Since θ_1 is a function of $r_0, r_{67}, s_{34}, r_{99}, s_{99}$ only, for any $\phi \in [1,99] \setminus \{67,99\}$ we have

$$\theta_1(R_{t,\Delta r_\phi}(t), S_{t,\Delta r_\phi}(t)) = \theta_1(R_t, S_t).$$

Therefore $z_{t+1} + z_{t+1,\Delta r_\phi}(t)$ equals

$$\begin{aligned}&\theta_1(R_t, S_t) + \theta_1(R_{t,\Delta r_\phi}(t), S_{t,\Delta r_\phi}(t))\\ &\quad = 0,\ \forall \phi \in [1,99] \setminus \{67,99\},\ \forall R_t, S_t \in \{0,1\}^{100}.\end{aligned}$$

So, $\Psi^1_{r_\phi}[1] = 1$ for all $\phi \in [1,99] \setminus \{67,99\}$.

C. We have $z_{t+1} + z_{t+1,\Delta r_{99}}(t)$ equals

$$\begin{aligned}&\theta_1(R_t, S_t) + \theta_1(R_{t,\Delta r_{99}}(t), S_{t,\Delta r_{99}}(t))\\ &\quad = (r^t_0 \cdot r^t_{67} + r^t_0 \cdot s^t_{34} + r^t_{99} + s^t_{99}) +\\ &\qquad (r^t_0 \cdot r^t_{67} + r^t_0 \cdot s^t_{34} + 1 + r^t_{99} + s^t_{99})\\ &\quad = 1,\ \forall R_t, S_t \in \{0,1\}^{100}.\end{aligned}$$

So, $\Psi^2_{r_{99}}[1] = 1$. Also $z_{t+1} + z_{t+1,\Delta r_{67}}(t)$ equals

$$\begin{aligned}&\theta_1(R_t, S_t) + \theta_1(R_{t,\Delta r_{67}}(t), S_{t,\Delta r_{67}}(t))\\ &\quad = (r^t_0 \cdot r^t_{67} + r^t_0 \cdot s^t_{34} + r^t_{99} + s^t_{99}) +\\ &\qquad (r^t_0 \cdot (1 + r^t_{67}) + r^t_0 \cdot s^t_{34} + r^t_{99} + s^t_{99})\\ &\quad = r^t_0 \neq 0 \text{ or } 1,\ \forall R_t, S_t \in \{0,1\}^{100}.\end{aligned}$$

So, $\Psi^2_{r_{67}}[1] = 0$.

D. We have

$$\begin{aligned}z_t + z_{t,\Delta s_0}(t) &= \theta_0(R_t, S_t) + \theta_0(R_{t,\Delta s_0}(t), S_{t,\Delta s_0}(t))\\ &= (r^t_0 + s^t_0) + (r^t_0 + 1 + s^t_0)\\ &= 1,\ \forall R_t, S_t \in \{0,1\}^{100}.\end{aligned}$$

So, $\Psi^2_{s_0}[0] = 1$. Also θ_0 is not a function of any r_i, s_i for $i \in [1,99]$ and so

$$\theta_0(R_{t,\Delta s_\phi}(t), S_{t,\Delta s_\phi}(t)) = \theta_0(R_t, S_t)$$

for all $\phi \in [1, 99]$ and so we have

$$\begin{aligned} z_t + z_{t,\Delta s_\phi}(t) &= \theta_0(R_t, S_t) + \theta_0(R_{t,\Delta s_\phi}(t), S_{t,\Delta s_\phi}(t)) \\ &= 0,\ \forall \phi \in [1, 99],\ \forall R_t, S_t \in \{0,1\}^{100}. \end{aligned}$$

So, $\Psi^1_{s_\phi}[0] = 1$ for all $\phi \in [1, 99]$.

E. Since θ_1 is a function of $r_0, r_{67}, s_{34}, r_{99}, s_{99}$ only, for any $\phi \in [1, 99] \setminus \{34, 99\}$ we have

$$\theta_1(R_{t,\Delta s_\phi}(t), S_{t,\Delta s_\phi}(t)) = \theta_1(R_t, S_t).$$

Therefore $z_{t+1} + z_{t+1,\Delta s_\phi}(t)$ equals

$$\begin{aligned} &\theta_1(R_t, S_t) + \theta_1(R_{t,\Delta s_\phi}(t), S_{t,\Delta s_\phi}(t)) \\ &= 0,\ \forall \phi \in [1, 99] \setminus \{34, 99\},\ \forall R_t, S_t \in \{0,1\}^{100}. \end{aligned}$$

So, $\Psi^1_{s_\phi}[1] = 1$ for all $\phi \in [1, 99] \setminus \{34, 99\}$.

F. We have $z_{t+1} + z_{t+1,\Delta s_{99}}(t)$ equals

$$\begin{aligned} &\theta_1(R_t, S_t) + \theta_1(R_{t,\Delta s_{99}}(t), S_{t,\Delta s_{99}}(t)) \\ &= (r^t_0 \cdot r^t_{67} + r^t_0 \cdot s^t_{34} + r^t_{99} + s^t_{99}) + \\ &\quad (r^t_0 \cdot r^t_{67} + r^t_0 \cdot s^t_{34} + r^t_{99} + 1 + s^t_{99}) \\ &= 1,\ \forall R_t, S_t \in \{0,1\}^{100}. \end{aligned}$$

So, $\Psi^2_{s_{99}}[1] = 1$. Also $z_{t+1} + z_{t+1,\Delta s_{34}}(t)$ equals

$$\begin{aligned} &\theta_1(R_t, S_t) + \theta_1(R_{t,\Delta s_{34}}(t), S_{t,\Delta s_{34}}(t)) \\ &= (r^t_0 \cdot r^t_{67} + r^t_0 \cdot s^t_{34} + r^t_{99} + s^t_{99}) + \\ &\quad (r^t_0 \cdot r^t_{67} + r^t_0 \cdot (1 + s^t_{34}) + r^t_{99} + s^t_{99}) \\ &= r^t_0 \neq 0 \text{ or } 1,\ \forall R_t, S_t \in \{0,1\}^{100}. \end{aligned}$$

So, $\Psi^2_{s_{34}}[1] = 0$.

□

Appendix B: 187.25 faults are sufficient to deduce $r^t_0 = 1$ and find CR_t

From Section 5.2, we know that we have a total of 297 different faulty streams. To deduce that $r^t_0 = 1$ and find CR_t, by injecting random faults, we want to obtain 4 different streams out of a set of 9 specific streams and 1 out of a set of 3 other streams. To find the expected number of faults to achieve this target, we will use the following proposition.

Proposition 7 *Consider five real numbers $a_1, \ldots, a_5$ in $(0, 1)$. Then, we have the following identities*

1. $$\sum_{r_1=0}^{\infty} \cdots \sum_{r_5=0}^{\infty} \left[\prod_{i=1}^{5} a_i^{r_i} \right] = \prod_{i=1}^{5} \frac{1}{(1 - a_i)}$$

2. $$\sum_{\substack{r_1=0 \\ \vdots \\ r_5=0}}^{\infty} \left[\sum_{i=1}^{5} r_i \cdot \prod_{i=1}^{5} a_i^{r_i} \right] = \sum_{i=1}^{5} \frac{a_i}{(1 - a_i)^2} \prod_{\substack{j=1 \\ j \neq i}}^{5} \frac{1}{(1 - a_j)}$$

Suppose, we first obtain the 4 streams of the set of 9 in $r_1 + 1, r_1 + r_2 + 2, r_1 + r_2 + r_3 + 3$ and $r_1 + r_2 + r_3 + r_4 + 4$ attempts respectively. Thereafter, we obtain the remaining streams from the set of 3 after another $r_5 + 1$ trials, i.e., we require $r_1 + r_2 + r_3 + r_4 + r_5 + 5$ faults in total. We call this event $\mathcal{E}_{r_1,\ldots,r_5}$. Then $\mathsf{Pr}(\mathcal{E}_{r_1,\ldots,r_5}) =$

$$\begin{aligned} &a_1^{r_1} \frac{9}{297} \cdot a_2^{r_2} \cdot \frac{8}{297} \cdot a_3^{r_3} \cdot \frac{7}{297} \cdot a_4^{r_4} \cdot \frac{6}{297} \cdot a_5^{r_5} \cdot \frac{3}{297} \\ &= a_1^{r_1} a_2^{r_2} a_3^{r_3} a_4^{r_4} a_5^{r_5} \cdot \frac{9072}{297^5}, \end{aligned}$$

where $a_1 = \frac{285}{297}, a_2 = \frac{286}{297}, a_3 = \frac{287}{297}, a_4 = \frac{288}{297}, a_5 = \frac{294}{297}$. Here a_i's denote the failure probabilities, i.e., a_i denotes the probability that, after obtaining $i-1$ required streams, a random fault produces no stream of interest.

We may also fulfill our target by some other "ordering" of events. For example, we first obtain 3 streams from the set of 9, then the single stream from the other set of 3 and finally the remaining stream from the first set. There are 5 orderings in total. Denote by b_i, c_i, d_i, e_i the failure probabilities, in each of the other orderings.

It is easy to see that, $b_1 = c_1 = d_1 = e_1 = a_1,\ b_2 = c_2 = d_2 = a_2, b_3 = c_3 = a_3,\ b_4 = a_4,\ b_5 = c_5 = d_5 = e_5 = \frac{291}{297},\ c_4 = d_4 = e_4 = \frac{290}{297},\ d_3 = e_3 = \frac{289}{297},\ e_2 = \frac{288}{297}$.

Considering all cases, the required expected value is

$$E = \sum_{\substack{r_1=0 \\ \vdots \\ r_5=0}}^{\infty} \left(5 + \sum_{i=0}^{5} r_i\right) \left(\prod_{i=1}^{5} a_i^{r_i} + \cdots + \prod_{i=1}^{5} e_i^{r_i}\right) \cdot \frac{9072}{297^5}$$

Now using Proposition 7, we get $E = 187.25$.

Appendix C: The functions $\rho_i\ \forall i \in [0, 99]$

i	ρ_i
0	$r_0 \cdot r_{67} + r_0 \cdot s_{34} + r_{99}$
1	$r_0 + r_1 \cdot r_{67} + r_1 \cdot s_{34} + r_{99}$
2	$r_1 + r_2 \cdot r_{67} + r_2 \cdot s_{34}$
3	$r_2 + r_3 \cdot r_{67} + r_3 \cdot s_{34} + r_{99}$
4	$r_3 + r_4 \cdot r_{67} + r_4 \cdot s_{34} + r_{99}$
5	$r_4 + r_5 \cdot r_{67} + r_5 \cdot s_{34} + r_{99}$
6	$r_5 + r_6 \cdot r_{67} + r_6 \cdot s_{34} + r_{99}$
7	$r_6 + r_7 \cdot r_{67} + r_7 \cdot s_{34}$
8	$r_7 + r_8 \cdot r_{67} + r_8 \cdot s_{34}$
9	$r_8 + r_9 \cdot r_{67} + r_9 \cdot s_{34} + r_{99}$
10	$r_9 + r_{10} \cdot r_{67} + r_{10} \cdot s_{34}$
11	$r_{10} + r_{11} \cdot r_{67} + r_{11} \cdot s_{34}$
12	$r_{11} + r_{12} \cdot r_{67} + r_{12} \cdot s_{34} + r_{99}$
13	$r_{12} + r_{13} \cdot r_{67} + r_{13} \cdot s_{34} + r_{99}$
14	$r_{13} + r_{14} \cdot r_{67} + r_{14} \cdot s_{34}$
15	$r_{14} + r_{15} \cdot r_{67} + r_{15} \cdot s_{34}$
16	$r_{15} + r_{16} \cdot r_{67} + r_{16} \cdot s_{34} + r_{99}$
17	$r_{16} + r_{17} \cdot r_{67} + r_{17} \cdot s_{34}$
18	$r_{17} + r_{18} \cdot r_{67} + r_{18} \cdot s_{34}$
19	$r_{18} + r_{19} \cdot r_{67} + r_{19} \cdot s_{34} + r_{99}$
20	$r_{19} + r_{20} \cdot r_{67} + r_{20} \cdot s_{34} + r_{99}$
21	$r_{20} + r_{21} \cdot r_{67} + r_{21} \cdot s_{34} + r_{99}$
22	$r_{21} + r_{22} \cdot r_{67} + r_{22} \cdot s_{34} + r_{99}$

i	ρ_i
23	$r_{22} + r_{23} \cdot r_{67} + r_{23} \cdot s_{34}$
24	$r_{23} + r_{24} \cdot r_{67} + r_{24} \cdot s_{34}$
25	$r_{24} + r_{25} \cdot r_{67} + r_{25} \cdot s_{34} + r_{99}$
26	$r_{25} + r_{26} \cdot r_{67} + r_{26} \cdot s_{34}$
27	$r_{26} + r_{27} \cdot r_{67} + r_{27} \cdot s_{34}$
28	$r_{27} + r_{28} \cdot r_{67} + r_{28} \cdot s_{34} + r_{99}$
29	$r_{28} + r_{29} \cdot r_{67} + r_{29} \cdot s_{34}$
30	$r_{29} + r_{30} \cdot r_{67} + r_{30} \cdot s_{34}$
31	$r_{30} + r_{31} \cdot r_{67} + r_{31} \cdot s_{34}$
32	$r_{31} + r_{32} \cdot r_{67} + r_{32} \cdot s_{34}$
33	$r_{32} + r_{33} \cdot r_{67} + r_{33} \cdot s_{34}$
34	$r_{33} + r_{34} \cdot r_{67} + r_{34} \cdot s_{34}$
35	$r_{34} + r_{35} \cdot r_{67} + r_{35} \cdot s_{34}$
36	$r_{35} + r_{36} \cdot r_{67} + r_{36} \cdot s_{34}$
37	$r_{36} + r_{37} \cdot r_{67} + r_{37} \cdot s_{34} + r_{99}$
38	$r_{37} + r_{38} \cdot r_{67} + r_{38} \cdot s_{34} + r_{99}$
39	$r_{38} + r_{39} \cdot r_{67} + r_{39} \cdot s_{34}$
40	$r_{39} + r_{40} \cdot r_{67} + r_{40} \cdot s_{34}$
41	$r_{40} + r_{41} \cdot r_{67} + r_{41} \cdot s_{34} + r_{99}$
42	$r_{41} + r_{42} \cdot r_{67} + r_{42} \cdot s_{34} + r_{99}$
43	$r_{42} + r_{43} \cdot r_{67} + r_{43} \cdot s_{34}$
44	$r_{43} + r_{44} \cdot r_{67} + r_{44} \cdot s_{34}$
45	$r_{44} + r_{45} \cdot r_{67} + r_{45} \cdot s_{34} + r_{99}$
46	$r_{45} + r_{46} \cdot r_{67} + r_{46} \cdot s_{34} + r_{99}$
47	$r_{46} + r_{47} \cdot r_{67} + r_{47} \cdot s_{34}$
48	$r_{47} + r_{48} \cdot r_{67} + r_{48} \cdot s_{34}$
49	$r_{48} + r_{49} \cdot r_{67} + r_{49} \cdot s_{34}$
50	$r_{49} + r_{50} \cdot r_{67} + r_{50} \cdot s_{34} + r_{99}$
51	$r_{50} + r_{51} \cdot r_{67} + r_{51} \cdot s_{34}$
52	$r_{51} + r_{52} \cdot r_{67} + r_{52} \cdot s_{34} + r_{99}$
53	$r_{52} + r_{53} \cdot r_{67} + r_{53} \cdot s_{34}$
54	$r_{53} + r_{54} \cdot r_{67} + r_{54} \cdot s_{34} + r_{99}$
55	$r_{54} + r_{55} \cdot r_{67} + r_{55} \cdot s_{34}$
56	$r_{55} + r_{56} \cdot r_{67} + r_{56} \cdot s_{34} + r_{99}$
57	$r_{56} + r_{57} \cdot r_{67} + r_{57} \cdot s_{34}$
58	$r_{57} + r_{58} \cdot r_{67} + r_{58} \cdot s_{34} + r_{99}$
59	$r_{58} + r_{59} \cdot r_{67} + r_{59} \cdot s_{34}$
60	$r_{59} + r_{60} \cdot r_{67} + r_{60} \cdot s_{34} + r_{99}$
61	$r_{60} + r_{61} \cdot r_{67} + r_{61} \cdot s_{34} + r_{99}$
62	$r_{61} + r_{62} \cdot r_{67} + r_{62} \cdot s_{34}$
63	$r_{62} + r_{63} \cdot r_{67} + r_{63} \cdot s_{34} + r_{99}$
64	$r_{63} + r_{64} \cdot r_{67} + r_{64} \cdot s_{34} + r_{99}$
65	$r_{64} + r_{65} \cdot r_{67} + r_{65} \cdot s_{34} + r_{99}$
66	$r_{65} + r_{66} \cdot r_{67} + r_{66} \cdot s_{34} + r_{99}$
67	$r_{66} + r_{67} \cdot s_{34} + r_{67} + r_{99}$
68	$r_{67} \cdot r_{68} + r_{67} + r_{68} \cdot s_{34}$
69	$r_{67} \cdot r_{69} + r_{68} + r_{69} \cdot s_{34}$
70	$r_{67} \cdot r_{70} + r_{69} + r_{70} \cdot s_{34}$
71	$r_{67} \cdot r_{71} + r_{70} + r_{71} \cdot s_{34} + r_{99}$
72	$r_{67} \cdot r_{72} + r_{71} + r_{72} \cdot s_{34} + r_{99}$
73	$r_{67} \cdot r_{73} + r_{72} + r_{73} \cdot s_{34}$
74	$r_{67} \cdot r_{74} + r_{73} + r_{74} \cdot s_{34}$
75	$r_{67} \cdot r_{75} + r_{74} + r_{75} \cdot s_{34}$
76	$r_{67} \cdot r_{76} + r_{75} + r_{76} \cdot s_{34}$
77	$r_{67} \cdot r_{77} + r_{76} + r_{77} \cdot s_{34}$
78	$r_{67} \cdot r_{78} + r_{77} + r_{78} \cdot s_{34}$
79	$r_{67} \cdot r_{79} + r_{78} + r_{79} \cdot s_{34} + r_{99}$
80	$r_{67} \cdot r_{80} + r_{79} + r_{80} \cdot s_{34} + r_{99}$
81	$r_{67} \cdot r_{81} + r_{80} + r_{81} \cdot s_{34} + r_{99}$
82	$r_{67} \cdot r_{82} + r_{81} + r_{82} \cdot s_{34} + r_{99}$
83	$r_{67} \cdot r_{83} + r_{82} + r_{83} \cdot s_{34}$
84	$r_{67} \cdot r_{84} + r_{83} + r_{84} \cdot s_{34}$
85	$r_{67} \cdot r_{85} + r_{84} + r_{85} \cdot s_{34}$
86	$r_{67} \cdot r_{86} + r_{85} + r_{86} \cdot s_{34}$
87	$r_{67} \cdot r_{87} + r_{86} + r_{87} \cdot s_{34} + r_{99}$
88	$r_{67} \cdot r_{88} + r_{87} + r_{88} \cdot s_{34} + r_{99}$
89	$r_{67} \cdot r_{89} + r_{88} + r_{89} \cdot s_{34} + r_{99}$
90	$r_{67} \cdot r_{90} + r_{89} + r_{90} \cdot s_{34} + r_{99}$
91	$r_{67} \cdot r_{91} + r_{90} + r_{91} \cdot s_{34} + r_{99}$
92	$r_{67} \cdot r_{92} + r_{91} + r_{92} \cdot s_{34} + r_{99}$
93	$r_{67} \cdot r_{93} + r_{92} + r_{93} \cdot s_{34}$
94	$r_{67} \cdot r_{94} + r_{93} + r_{94} \cdot s_{34} + r_{99}$
95	$r_{67} \cdot r_{95} + r_{94} + r_{95} \cdot s_{34} + r_{99}$
96	$r_{67} \cdot r_{96} + r_{95} + r_{96} \cdot s_{34} + r_{99}$
97	$r_{67} \cdot r_{97} + r_{96} + r_{97} \cdot s_{34} + r_{99}$
98	$r_{67} \cdot r_{98} + r_{97} + r_{98} \cdot s_{34}$
99	$r_{67} \cdot r_{99} + r_{98} + r_{99} \cdot s_{34}$

The functions β_i $\forall i \in [0, 99]$

i	β_i
0	s_{99}
1	$s_0 + s_1 \cdot s_2 + s_1 + s_{99}$
2	$s_1 + s_2 \cdot s_3 + s_{99}$
3	$r_{33} \cdot s_{99} + s_2 + s_3 \cdot s_4 + s_3 + s_{67} \cdot s_{99} + s_{99}$
4	$r_{33} \cdot s_{99} + s_3 + s_4 \cdot s_5 + s_4 + s_5 + s_{67} \cdot s_{99} + 1$
5	$s_4 + s_5 \cdot s_6 + s_6 + s_{99}$
6	$r_{33} \cdot s_{99} + s_5 + s_6 \cdot s_7 + s_{67} \cdot s_{99}$
7	$r_{33} \cdot s_{99} + s_6 + s_7 \cdot s_8 + s_7 + s_{67} \cdot s_{99} + s_{99}$
8	$r_{33} \cdot s_{99} + s_7 + s_8 \cdot s_9 + s_{67} \cdot s_{99} + s_{99}$
9	$r_{33} \cdot s_{99} + s_8 + s_9 \cdot s_{10} + s_9 + s_{10} + s_{67} \cdot s_{99} + s_{99} + 1$
10	$r_{33} \cdot s_{99} + s_9 + s_{10} \cdot s_{11} + s_{10} + s_{67} \cdot s_{99} + s_{99}$
11	$s_{10} + s_{11} \cdot s_{12} + s_{11} + s_{12} + s_{99} + 1$
12	$s_{11} + s_{12} \cdot s_{13} + s_{12} + s_{13} + s_{99} + 1$
13	$s_{12} + s_{13} \cdot s_{14} + s_{14} + s_{99}$
14	$r_{33} \cdot s_{99} + s_{13} + s_{14} \cdot s_{15} + s_{15} + s_{67} \cdot s_{99} + s_{99}$
15	$r_{33} \cdot s_{99} + s_{14} + s_{15} \cdot s_{16} + s_{15} + s_{67} \cdot s_{99}$
16	$s_{15} + s_{16} \cdot s_{17} + s_{17}$
17	$r_{33} \cdot s_{99} + s_{16} + s_{17} \cdot s_{18} + s_{17} + s_{67} \cdot s_{99} + s_{99}$
18	$r_{33} \cdot s_{99} + s_{17} + s_{18} \cdot s_{19} + s_{67} \cdot s_{99}$
19	$s_{18} + s_{19} \cdot s_{20} + s_{20} + s_{99}$
20	$r_{33} \cdot s_{99} + s_{19} + s_{20} \cdot s_{21} + s_{67} \cdot s_{99} + s_{99}$
21	$r_{33} \cdot s_{99} + s_{20} + s_{21} \cdot s_{22} + s_{21} + s_{22} + s_{67} \cdot s_{99} + s_{99} + 1$
22	$r_{33} \cdot s_{99} + s_{21} + s_{22} \cdot s_{23} + s_{22} + s_{67} \cdot s_{99} + s_{99}$
23	$s_{22} + s_{23} \cdot s_{24} + s_{24} + s_{99}$
24	$r_{33} \cdot s_{99} + s_{23} + s_{24} \cdot s_{25} + s_{24} + s_{67} \cdot s_{99} + s_{99}$
25	$r_{33} \cdot s_{99} + s_{24} + s_{25} \cdot s_{26} + s_{26} + s_{67} \cdot s_{99} + s_{99}$
26	$s_{25} + s_{26} \cdot s_{27} + s_{26} + s_{99}$
27	$s_{26} + s_{27} \cdot s_{28} + s_{27} + s_{28} + s_{99} + 1$
28	$r_{33} \cdot s_{99} + s_{27} + s_{28} \cdot s_{29} + s_{28} + s_{67} \cdot s_{99} + s_{99}$
29	$s_{28} + s_{29} \cdot s_{30} + s_{30}$
30	$r_{33} \cdot s_{99} + s_{29} + s_{30} \cdot s_{31} + s_{30} + s_{31} + s_{67} \cdot s_{99} + 1$
31	$r_{33} \cdot s_{99} + s_{30} + s_{31} \cdot s_{32} + s_{31} + s_{67} \cdot s_{99} + s_{99}$
32	$s_{31} + s_{32} \cdot s_{33} + s_{32} + s_{33} + s_{99} + 1$
33	$r_{33} \cdot s_{99} + s_{32} + s_{33} \cdot s_{34} + s_{33} + s_{67} \cdot s_{99}$
34	$s_{33} + s_{34} \cdot s_{35}$
35	$s_{34} + s_{35} \cdot s_{36} + s_{36}$
36	$s_{35} + s_{36} \cdot s_{37}$
37	$r_{33} \cdot s_{99} + s_{36} + s_{37} \cdot s_{38} + s_{37} + s_{67} \cdot s_{99}$
38	$r_{33} \cdot s_{99} + s_{37} + s_{38} \cdot s_{39} + s_{38} + s_{67} \cdot s_{99}$
39	$r_{33} \cdot s_{99} + s_{38} + s_{39} \cdot s_{40} + s_{67} \cdot s_{99} + s_{99}$
40	$r_{33} \cdot s_{99} + s_{39} + s_{40} \cdot s_{41} + s_{40} + s_{67} \cdot s_{99} + s_{99}$
41	$r_{33} \cdot s_{99} + s_{40} + s_{41} \cdot s_{42} + s_{67} \cdot s_{99} + s_{99}$
42	$s_{41} + s_{42} \cdot s_{43} + s_{42}$
43	$s_{42} + s_{43} \cdot s_{44} + s_{43} + s_{44} + 1$
44	$s_{43} + s_{44} \cdot s_{45} + s_{44} + s_{99}$
45	$r_{33} \cdot s_{99} + s_{44} + s_{45} \cdot s_{46} + s_{46} + s_{67} \cdot s_{99}$
46	$s_{45} + s_{46} \cdot s_{47}$
47	$s_{46} + s_{47} \cdot s_{48} + s_{48} + s_{99}$
48	$r_{33} \cdot s_{99} + s_{47} + s_{48} \cdot s_{49} + s_{67} \cdot s_{99}$
49	$r_{33} \cdot s_{99} + s_{48} + s_{49} \cdot s_{50} + s_{49} + s_{50} + s_{67} \cdot s_{99} + s_{99} + 1$
50	$s_{49} + s_{50} \cdot s_{51}$
51	$r_{33} \cdot s_{99} + s_{50} + s_{51} \cdot s_{52} + s_{67} \cdot s_{99} + s_{99}$
52	$r_{33} \cdot s_{99} + s_{51} + s_{52} \cdot s_{53} + s_{67} \cdot s_{99}$
53	$s_{52} + s_{53} \cdot s_{54} + s_{53}$
54	$r_{33} \cdot s_{99} + s_{53} + s_{54} \cdot s_{55} + s_{55} + s_{67} \cdot s_{99} + s_{99}$
55	$s_{54} + s_{55} \cdot s_{56} + s_{55}$
56	$s_{55} + s_{56} \cdot s_{57} + s_{56} + s_{57} + s_{99} + 1$
57	$r_{33} \cdot s_{99} + s_{56} + s_{57} \cdot s_{58} + s_{57} + s_{67} \cdot s_{99} + s_{99}$
58	$r_{33} \cdot s_{99} + s_{57} + s_{58} \cdot s_{59} + s_{67} \cdot s_{99} + s_{99}$
59	$s_{58} + s_{59} \cdot s_{60} + s_{60} + s_{99}$
60	$s_{59} + s_{60} \cdot s_{61} + s_{61}$
61	$r_{33} \cdot s_{99} + s_{60} + s_{61} \cdot s_{62} + s_{61} + s_{62} + s_{67} \cdot s_{99} + s_{99} + 1$
62	$r_{33} \cdot s_{99} + s_{61} + s_{62} \cdot s_{63} + s_{62} + s_{63} + s_{67} \cdot s_{99} + 1$
63	$r_{33} \cdot s_{99} + s_{62} + s_{63} \cdot s_{64} + s_{63} + s_{67} \cdot s_{99} + s_{99}$
64	$r_{33} \cdot s_{99} + s_{63} + s_{64} \cdot s_{65} + s_{64} + s_{67} \cdot s_{99}$
65	$s_{64} + s_{65} \cdot s_{66} + s_{65} + s_{66} + s_{99} + 1$
66	$s_{65} + s_{66} \cdot s_{67} + s_{66}$
67	$r_{33} \cdot s_{99} + s_{66} + s_{67} \cdot s_{68} + s_{67} \cdot s_{99} + s_{68}$
68	$s_{67} + s_{68} \cdot s_{69} + s_{68}$
69	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{68} + s_{69} \cdot s_{70} + s_{70}$
70	$s_{69} + s_{70} \cdot s_{71} + s_{70} + s_{71} + 1$
71	$s_{70} + s_{71} \cdot s_{72} + s_{71} + s_{72} + 1$
72	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{71} + s_{72} \cdot s_{73} + s_{72} + s_{73} + 1$

i	β_i
73	$s_{72} + s_{73} \cdot s_{74} + s_{74}$
74	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{73} + s_{74} \cdot s_{75} + s_{74} + s_{75} + 1$
75	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{74} + s_{75} \cdot s_{76} + s_{75} + s_{76} + s_{99} + 1$
76	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{75} + s_{76} \cdot s_{77} + s_{76} + s_{77} + s_{99} + 1$
77	$s_{76} + s_{77} \cdot s_{78} + s_{77} + s_{78} + 1$
78	$s_{77} + s_{78} \cdot s_{79} + s_{99}$
79	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{78} + s_{79} \cdot s_{80} + s_{80}$
80	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{79} + s_{80} \cdot s_{81}$
81	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{80} + s_{81} \cdot s_{82} + s_{81} + s_{82} + 1$
82	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{81} + s_{82} \cdot s_{83} + s_{83} + s_{99}$
83	$s_{82} + s_{83} \cdot s_{84} + s_{84} + s_{99}$
84	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{83} + s_{84} \cdot s_{85} + s_{85}$
85	$s_{84} + s_{85} \cdot s_{86} + s_{86} + s_{99}$
86	$s_{85} + s_{86} \cdot s_{87} + s_{86} + s_{87} + s_{99} + 1$
87	$s_{86} + s_{87} \cdot s_{88} + s_{87} + s_{99}$
88	$s_{87} + s_{88} \cdot s_{89} + s_{88} + s_{89} + 1$
89	$s_{88} + s_{89} \cdot s_{90}$
90	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{89} + s_{90} \cdot s_{91} + s_{91} + s_{99}$
91	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{90} + s_{91} \cdot s_{92} + s_{99}$
92	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{91} + s_{92} \cdot s_{93} + s_{92} + s_{99}$
93	$s_{92} + s_{93} \cdot s_{94}$
94	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{93} + s_{94} \cdot s_{95}$
95	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{94} + s_{95} \cdot s_{96} + s_{95} + s_{99}$
96	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{95} + s_{96} \cdot s_{97} + s_{96} + s_{99}$
97	$s_{96} + s_{97} \cdot s_{98} + s_{98}$
98	$s_{97} + s_{98} \cdot s_{99} + s_{99}$
99	$r_{33} \cdot s_{99} + s_{67} \cdot s_{99} + s_{98}$