A TMDTO Attack Against Lizard

Subhamoy Maitra[1], Nishant Sinha[2], Akhilesh Siddhanti[3], Ravi Anand[4], Sugata Gangopadhyay[2]

[1]Indian Statistical Institute, Kolkata, subho@isical.ac.in
[2]Indian Institute of Technology, Roorkee, {nishantsinha.iitr, gsugata}@gmail.com
[3]BITS Pilani, Goa Campus, akhileshsiddhanti@gmail.com
[4]Indian Institute of Technology Kharagpur, ravianandsps@gmail.com

Abstract. Lizard is a very recently proposed lightweight stream cipher that claims 60 bit security against distinguishing (related to state recovery) and 80 bit security against key recovery attack. This cipher has 121 bit state size. In this paper, we first note that using ψ key stream bits one can recover ψ unknown bits of the state when τ state bits are fixed to a specific pattern. This is made possible by guessing the remaining state bits. This helps us in mounting a TMDTO attack with preprocessing complexity 2^{67}, and the maximum of Data, Time and Memory complexity during the online phase as 2^{54}. The parameters in the online phase are significantly less than 2^{60}.

Keywords: Cryptanalysis, Lizard, Lightweight Stream Cipher, Time Memory Data Trade-Off (TMDTO) Attack.

1 Introduction

Time-Memory Trade-off attack on block ciphers has been first proposed in [13]. Later a generic TMDTO attack was introduced against stream ciphers in [7] modeling a similar situation as that of block ciphers. Given a stream cipher with search space $N = 2^n$, the classical TMDTO theory says that internal state of cipher can be recovered if $TM^2D^2 = N^2$, where T is the time complexity, M is the memory required and D is the total amount of data available to the attacker during the online phase. The pre-computation time is calculated as $P = \frac{N}{D}$ (this complexity is generally more than the exhaustive key search). The attack involves computing a chain of m pseudo-random states of size $\frac{N}{D}$ by repeatedly finding its image t times using the non-invertible function f that we like to attack. Here, a random state ζ is considered as the input to the function, and the output $f(\zeta)$ is the key stream output corresponding to that state equal to the same length as the state. This leads to formulation of $\frac{t}{D}$ tables, introducing a natural constraint of $t \geq D$. By choosing parameters of $T = M = 2^{\frac{n}{2}}$ and $D = 2^{\frac{n}{4}}$ for a search space of $N = 2^n$ (i.e., state size of n-bits), one can deduce the internal state. If the state size is less than twice the security-level mentioned for the cipher, then this is a generic attack, when the secret key can be directly recovered from the state. Hence, a basic rule of thumb in designing stream ciphers is to keep the state size atleast twice more than the secret key size. Added to this, one may also note that if the sum of key and IV size is less than the state size, then the attack is guided by the sum [14]. However, there are some ciphers, where discovering the secret state does not automatically guarantee obtaining the secret key directly (as in Lizard [11]) and in such a situation this attack does not immediately follow for secret key recovery.

Presently, new lightweight stream ciphers are being designed where we have state size lesser than twice the key size. The initial design, Sprout [1] has been attacked soon after the proposal (see [17,9,18]). Plantlet [16], an improvement over Sprout is also presented very recently and a generic TMDTO analysis has been presented in [12] very recently (earlier only differential fault attack [15] had been reported). The idea of Plantlet involves use of secret key bits not only during the Key Scheduling Algorithm (KSA) but also during the Pseudo Random Generation Algorithm (PRGA). The design of Lizard [11] is from a different viewpoint, and is similar to classical designs in which the secret key bits are not used during PRGA. Very recently certain cryptanalytic observations have been made [5] in connection with related keys of Lizard. This cipher has a state size of 121 bits and a generic TMDTO attack will obtain the state of this cipher in 2^{60} online complexity. Thus the challenge is to obtain a reduced complexity than this.

Towards that, we try to obtain some information related to the state from the key stream bits. After carefully studying the PRGA of the cipher, it is found that 13 bits of key stream provide 13 unknown bits of the state. The constraint is that, we need to fix 28 bits of the state to a specific pattern and then need to guess 80 bits. Thus, the search space becomes 2^{80} and our parameters are such that we require 2^{67} complexity during the preprocessing. During the online phase, the parameters are $T = M = D = 2^{54}$. Note that each of the parameters during the online phase is less than 2^{60} and thus this is the first TMDTO analysis against Lizard other than the designers themselves. The designers specified that Lizard can be used for generating 2^{18} key stream bits only for a specific key and IV. However, we like to point out that obtaining a total amount of 2^{54} key stream bits (corresponding to D) from different sources will work in such a scenario. We will be able to obtain one state corresponding to a specific Key/IV for which the attack finally succeeds. Further, from theoretical point of view, our findings show certain weaknesses against this instance of lightweight stream cipher. The designers of Lizard [11, Appendix B1] indeed acknowledged the applicability of BSW-sampling [6,7] describing this as highly cipher specific. We should also like to mention at this point that our observation is not same as the BG attack [3,10]. Indeed we consider the situation $TP = DP = MP = N$, which is similar to the condition of BG trade-off. However, we do not have $P = T$ here as in the BG scenario. Our condition $MD = MT = DT < N$ follows from the idea of sampling resistance as given in [7] and further, we try to minimize the maximum of T, M, D by making $T = M = D$. Thus, this cannot be achieved directly by BG kind of attack on stream ciphers. In fact, we show that if ψ bits of the secret state can be recovered from key stream bits (either fixing some state bits or not), then we can achieve the parameters $P = 2^{\frac{\nu+\psi}{2}}$ and $T = M = D = 2^{\frac{\nu-\psi}{2}}$, where the state size is of ν bits. Needless to mention that how a higher preprocessing complexity will be accepted is a matter of interpretation. At the same time, we note that identifying the equations for guessing some state bits from key stream bits is important as well, since this reduces the online complexity significantly.

This motivates us to study the exact design of Lizard, and identify parameters for which is referred as conditional BSW-sampling. The term 'conditional' comes as we need to fix a few state bits with a specific pattern. In [11, Section 4.2], the ideas of TMDTO attack has been discussed clearly, but the applicability of conditional BSW sampling has not been studied in detail. Thus we apply conditional BSW sampling in this paper to analyze Lizard [11]. We should also like to mention that the computational power has developed a lot and we should look at the computations related to Bitcoin for estimating the achievable time complexity in real life. The current hash rate for Bitcoin is 6 Million Tera-Hash/Sec, which is more than 2^{62} and in a day it is more than 2^{78} operations. This is achieved in a distributed environment over the Bitcoin network [8]. Further, we have ASIC hardwares that claims around 13.5 Tera-Hash/Sec at USD 3000 [2], which provides more than 2^{60} operations in a day. Each such operation is almost competitive with the unit of computation in the TMDTO attack on a lightweight cipher like Lizard [11].

Before proceeding further, let us describe the organization of this paper. A brief description of Lizard is mentioned in Section 2. Section 3 illustrates the procedure of fixing state bits for a successful attack in obtaining some unknown state bits by looking at the key stream. Section 4 explains the preparation of offline tables and the description of the attack in detail. Section 5 concludes the paper.

2 Description of Lizard

The 121-bit inner state of Lizard is divided into two NFSRs namely NFSR1 and NFSR2. At time t, the first NFSR, NFSR1 is denoted by $(S_{(0+i)}, \ldots, S_{(30+i)})$ and the second NFSR, NFSR2 by $(B_{(0+i)}, \ldots, B_{(89+i)})$. The clocking of the cipher can be divided into following three functions:

1. NFSR1 Update Function: NFSR1 is of 31 bit size and the update rule of this NFSR is:

$$\begin{aligned} S_{(31+i)} = {} & S_{(0+i)} \oplus S_{(2+i)} \oplus S_{(5+i)} \oplus S_{(6+i)} \oplus S_{(15+i)} \oplus S_{(17+i)} \oplus S_{(18+i)} \oplus S_{(20+i)} \oplus S_{(25+i)} \\ & \oplus S_{(8+i)}S_{(18+i)} \oplus S_{(8+i)}S_{(20+i)} \oplus S_{(12+i)}S_{(21+i)} \oplus S_{(14+i)}S_{(19+i)} \oplus S_{(17+i)}S_{(21+i)} \\ & \oplus S_{(20+i)}S_{(22+i)} \oplus S_{(4+i)}S_{(12+i)}S_{(22+i)} \oplus S_{(4+i)}S_{(19+i)}S_{(22+i)} \oplus S_{(7+i)}S_{(20+i)}S_{(21+i)} \\ & \oplus S_{(8+i)}S_{(18+i)}S_{(22+i)} \oplus S_{(8+i)}S_{(20+i)}S_{(22+i)} \oplus S_{(12+i)}S_{(19+i)}S_{(22+i)} \\ & \oplus S_{(20+i)}S_{(21+i)}S_{(22+i)} \oplus S_{(4+i)}S_{(7+i)}S_{(12+i)}S_{(21+i)} \oplus S_{(4+i)}S_{(7+i)}S_{(19+i)}S_{(21+i)} \\ & \oplus S_{(4+i)}S_{(12+i)}S_{(21+i)}S_{(22+i)} \oplus S_{(4+i)}S_{(19+i)}S_{(21+i)}S_{(22+i)} \oplus S_{(7+i)}S_{(8+i)}S_{(18+i)}S_{(21+i)} \\ & \oplus S_{(7+i)}S_{(8+i)}S_{(20+i)}S_{(21+i)} \oplus S_{(7+i)}S_{(12+i)}S_{(19+i)}S_{(21+i)} \\ & \oplus S_{(8+i)}S_{(18+i)}S_{(21+i)}S_{(22+i)} \oplus S_{(8+i)}S_{(20+i)}S_{(21+i)}S_{(22+i)} \oplus S_{(12+i)}S_{(19+i)}S_{(21+i)}S_{(22+i)} \end{aligned} \tag{1}$$

2. NFSR2 Update Function: The second register NFSR2 is of 90 bit and the update rule is:

$$\begin{aligned} B_{(89+i)} = {} & S_{(0+i)} \oplus B_{(0+i)} \oplus B_{(24+i)} \oplus B_{(49+i)} \oplus B_{(79+i)} \oplus B_{(84+i)} \oplus B_{(3+i)}B_{(59+i)} \\ & \oplus B_{(10+i)}B_{(12+i)} \oplus B_{(15+i)}B_{(16+i)} \oplus B_{(25+i)}B_{(53+i)} \oplus B_{(35+i)}B_{(42+i)} \\ & \oplus B_{(55+i)}B_{(58+i)} \oplus B_{(60+i)}B_{(74+i)} \oplus B_{(20+i)}B_{(22+i)}B_{(23+i)} \\ & \oplus B_{(62+i)}B_{(68+i)}B_{(72+i)} \oplus B_{(77+i)}B_{(80+i)}B_{(81+i)}B_{(83+i)} \end{aligned} \tag{2}$$

3. The Output Function: The output bit z_i is a function from $\{0,1\}^{53}$ to $\{0,1\}$. For round i,

$$z_i = \mathcal{L}_i \oplus \mathcal{Q}_i \oplus \mathcal{T}_i \oplus \overline{\mathcal{T}}_i \tag{3}$$

where:

$$\mathcal{L}_i = B_{(7+i)} \oplus B_{(11+i)} \oplus B_{(30+i)} \oplus B_{(40+i)} \oplus B_{(45+i)} \oplus B_{(54+i)} \oplus B_{(71+i)} \tag{4}$$

$$\mathcal{Q}_i = B_{(4+i)}B_{(21+i)} \oplus B_{(9+i)}B_{(52+i)} \oplus B_{(18+i)}B_{(37+i)} \oplus B_{(44+i)}B_{(76+i)} \tag{5}$$

$$\begin{aligned} \mathcal{T}_i = {} & B_{(5+i)} \oplus B_{(8+i)}B_{(82+i)} \oplus B_{(34+i)}B_{(67+i)}B_{(73+i)} \oplus B_{(2+i)}B_{(28+i)}B_{(41+i)}B_{(65+i)} \\ & \oplus B_{(13+i)}B_{(29+i)}B_{(50+i)}B_{(64+i)}B_{(75+i)} \oplus B_{(6+i)}B_{(14+i)}B_{(26+i)}B_{(32+i)}B_{(47+i)}B_{(61+i)} \end{aligned}$$

$$\oplus B_{(1+i)}B_{(19+i)}B_{(27+i)}B_{(43+i)}B_{(57+i)}B_{(66+i)}B_{(78+i)} \tag{6}$$

$$\overline{\mathcal{T}}_i = S_{(23+i)} \oplus S_{(3+i)}S_{(16+i)} \oplus S_{(9+i)}S_{(13+i)}B_{(48+i)} \oplus S_{(1+i)}S_{(24+i)}B_{(38+i)}B_{(63+i)} \tag{7}$$

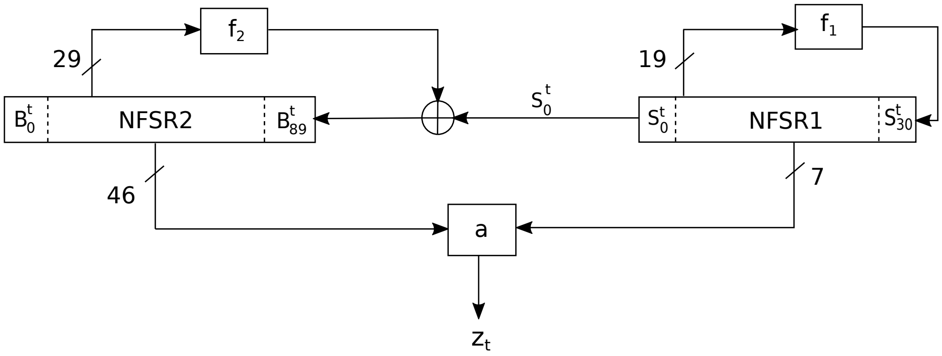

Fig. 1: Lizard: PRGA mode.

The state initialization process is divided into 4 phases.

Phase 1: Key and IV Loading: Let $K = (K_0, \ldots, K_{119})$ be the 120-bit key and $IV = (IV_0, \ldots, IV_{63})$ the 64-bit public IV. The state is initialized as follows:

$$B_j^0 = \begin{cases} K_j \oplus IV_j, & \text{for } 0 \le j \le 63 \\ K_j, & \text{for } 64 \le j \le 89 \end{cases}$$

$$S_j^0 = \begin{cases} K_{(j+90)}, & \text{for } 0 \le j \le 28 \\ K_{119}, & \text{for } j = 29 \\ 1 & \text{for } j = 30 \end{cases}$$

Phase 2: Grain-like Mixing: In this phase the output bit z_i is fed back into both NFSRs for $0 \le t \le 127$. This type of approach is used in Grain family.

Phase 3: Second Key Addition: In this phase, the 120-bit key is XORed to both NFSRs as follows:

$$B_j^{129} = B_j^{128} \oplus K_j, \quad \text{for } 0 \le j \le 89$$

$$S_j^{129} = \begin{cases} S_j^{128} \oplus K_{(j+90)}, & \text{for } 0 \le j \le 29 \\ 1, & \text{for } j = 30 \end{cases}$$

Phase 4: Final Diffusion This phase is exactly similar to phase 2 except z_t is not fed back into the NFSRs. In this phase, one has to run both NFSRs 128 rounds. So after this phase, registers are $(S_0^{257}, \ldots, S_{30}^{257})$ and $(B_0^{257}, \ldots, B_{89}^{257})$. Now Lizard is ready to produce output key stream bits.

3 Computation of Conditional Sampling Resistance of Lizard

In this section, we will be discussing the technique of recovering some state bits by cleverly exploiting Lizard's output function. We fix certain bits of the state to get rid of nonlinear terms. Some possible counts of state bits recovered corresponding to the number of state bits fixed in the output equation are shown in Table 1.

Instances (j)	Bits fixed ($\lvert\mathscr{F}_j\rvert$)	Keystream bits used = Bits recovered ($\lvert\mathscr{R}_j\rvert$)	Bits Guessed ($121 - \lvert\mathscr{F}_j\rvert - \lvert\mathscr{R}_j\rvert$)
1	12	9	100
2	16	10	95
3	20	11	90
4	24	12	85
5	27	13	81
6	30	14	77

Table 1: Possible combinations of bits fixed to bits recovered.

We illustrate the technique with the help of an example. We denote the set of fixed state bits by $\mathscr{F}_j$. The state bits recovered will be denoted by $\mathscr{R}_j$, which is done by observing the key stream bits z_t, fixing state bits $\mathscr{F}_j$ and guessing of remaining state bits appearing in the output equation. Let us fix $|\mathscr{F}_1| = 12$ bits. We aim to recover $\mathscr{R}_1 = \{S_{(3+i)} : i = 0, \dots, 8\}$, which means $|\mathscr{R}_1| = 9$ bits.

Let $\mathscr{F}_1 = \mathscr{A}_1 \cup \mathscr{A}_2$, where $\mathscr{A}_1 = \{S_{(16+i)} : i = 0, \dots, 8\}$ and $\mathscr{A}_2 = \{B_{(48+k)} : k = 0, \dots, 2\}$. We fix the elements of set $\mathscr{A}_1$ with 1's and $\mathscr{A}_2$ with 0's. The equation (7) :

$$\overline{\mathcal{T}}_i = S_{(23+i)} \oplus S_{(3+i)} S_{(16+i)} \oplus S_{(9+i)} S_{(13+i)} B_{(48+i)} \oplus S_{(1+i)} S_{(24+i)} B_{(38+i)} B_{(63+i)}$$

can now be written as:

$$\overline{\mathcal{T}}_i = S_{(23+i)} \oplus S_{(3+i)} \oplus S_{(1+i)} S_{(24+i)} B_{(38+i)} B_{(63+i)}. \tag{8}$$

for $i = 0, 1, 2$. From equation (3),

$$z_i = \mathcal{L}_i \oplus \mathcal{Q}_i \oplus \mathcal{T}_i \oplus \overline{\mathcal{T}}_i$$

and using equation (8) we have:

$$S_{(3+i)} = z_i \oplus L_i \oplus Q_i \oplus T_i \oplus S_{(23+i)} \oplus S_{(1+i)} S_{(24+i)} B_{(38+i)} B_{(63+i)}. \tag{9}$$

Now, we recover bits S_3, S_4 and S_5 by substituting $i = 0, 1, 2$ (following the order) in equation (9). This is made possible by guessing the state bits appearing on the right hand side of equation (9), except S_3, S_{24}, B_{48}, B_{49} and B_{50}, and using the key stream bits z_0, z_1 and z_2 (since S_3 is recovered and rest are already fixed). One may observe that the number of guessed bits required decreases for every additional bit recovered. For example, when we substitute $i = 0$, most of the state bits appearing on the right hand side of equation (9) are guessed which are listed in Table 2. When we substitute $i = 1$, we have to guess lesser number of internal state bits as most of them are already guessed in the previous step.

From $\mathscr{A}_1$, we rewrite equation (7) as:

$$\overline{\mathcal{T}}_i = S_{(23+i)} \oplus S_{(3+i)} \oplus S_{(19+i)} B_{(48+i)} \oplus S_{(1+i)} S_{(24+i)} B_{(38+i)} B_{(63+i)} \tag{10}$$

for $i = 3, \ldots, 8$. Hence, S_6, S_7, S_8, S_9, S_{10} and S_{11} can be recovered by using equation (10) with the help of equation (3). Further simplification gives the following equation:

$$S_{(3+i)} = z_i \oplus L_i \oplus Q_i \oplus T_i \oplus S_{(23+i)} \oplus S_{(19+i)} B_{(48+i)} \oplus S_{(1+i)} S_{(24+i)} B_{(38+i)} B_{(63+i)} \tag{11}$$

For $i = 3$ in equation 11, we need to guess very few bits as most bits have been already fixed or guessed earlier. For the case of $i = 7$ and $i = 8$ we need to know the values of S_{31}, S_{32} and B_{90}, which are nothing but the feedback bits. All the tap positions necessary to generate the feedback bits are already known at this stage, except B_0 and S_0 which need to be guessed as well. Thus, we have successfully recovered the entire set $\mathscr{R}_1$, i.e. by fixing $|\mathscr{F}_1|$ internal state bits with particular values, using 12 key stream bits and by guessing the remaining 100 internal state bits. The bits guessed for every bit recovered has been shown in Table 2.

State bit recovered	State bits guessed for recovering $\mathscr{R}_1$	State bits guessed for recovering $\mathscr{R}_2$	State bits guessed for recovering $\mathscr{R}_6$
S_3	$B_1, B_2, B_4, B_5, B_6, B_7, B_8, B_9, B_{11}, B_{13}, B_{14}, B_{18}, B_{19}, B_{21}, B_{26}, B_{27}, B_{28}, B_{29}, B_{30}, B_{32}, B_{34}, B_{37}, B_{38}, B_{40}, B_{41}, B_{43}, B_{44}, B_{45}, B_{47}, B_{52}, B_{54}, B_{57}, B_{61}, B_{63}, B_{64}, B_{65}, B_{66}, B_{67}, B_{71}, B_{73}, B_{75}, B_{76}, B_{78}, B_{82}, S_1$	$B_1, B_2, B_4, B_5, B_6, B_7, B_9, B_{13}, B_{14}, B_{18}, B_{19}, B_{21}, B_{26}, B_{27}, B_{28}, B_{29}, B_{30}, B_{32}, B_{34}, B_{37}, B_{38}, B_{40}, B_{41}, B_{43}, B_{44}, B_{45}, B_{47}, B_{52}, B_{54}, B_{57}, B_{61}, B_{63}, B_{64}, B_{65}, B_{66}, B_{67}, B_{71}, B_{73}, B_{75}, B_{76}, B_{78}, B_{82}, S_1$	$B_1, B_2, B_4, B_5, B_6, B_7, B_{26}, B_{27}, B_{28}, B_{29}, B_{30}, B_{32}, B_{34}, B_{37}, B_{38}, B_{40}, B_{41}, B_{43}, B_{44}, B_{45}, B_{47}, B_{52}, B_{61}, B_{63}, B_{64}, B_{65}, B_{66}, B_{67}, B_{71}, B_{73}, B_{75}, B_{76}, B_{78}, B_{82}, S_1$
S_4	$B_3, B_{10}, B_{12}, B_{15}, B_{20}, B_{22}, B_{31}, B_{33}, B_{35}, B_{39}, B_{42}, B_{46}, B_{51}, B_{53}, B_{55}, B_{58}, B_{62}, B_{68}, B_{72}, B_{74}, B_{77}, B_{79}, B_{83}, S_2, S_{25}$	$B_3, B_{10}, B_{12}, B_{15}, B_{20}, B_{22}, B_{31}, B_{33}, B_{35}, B_{39}, B_{42}, B_{46}, B_{51}, B_{55}, B_{58}, B_{62}, B_{68}, B_{72}, B_{74}, B_{77}, B_{79}, B_{83}, S_2, S_{25}$	$B_3, B_{22}, B_{31}, B_{33}, B_{35}, B_{39}, B_{42}, B_{46}, B_{51}, B_{58}, B_{62}, B_{68}, B_{72}, B_{74}, B_{77}, B_{79}, B_{83}, S_2, S_{25}$
S_5	$B_{16}, B_{23}, B_{36}, B_{56}, B_{59}, B_{69}, B_{80}, B_{84}, S_{26}$	$B_{16}, B_{23}, B_{36}, B_{56}, B_{59}, B_{69}, B_{80}, B_{84}, S_{26}$	$B_{16}, B_{23}, B_{36}, B_{59}, B_{69}, B_{80}, B_{84}, S_{26}$
S_6	$B_{17}, B_{24}, B_{60}, B_{70}, B_{81}, B_{85}, S_{12}, S_{27}$	$B_{24}, B_{60}, B_{70}, B_{81}, S_{12}, S_{27}$	$B_{24}, B_{60}, B_{70}, B_{81}, S_{12}, S_{27}$
S_7	$B_{25}, B_{86}, S_{13}, S_{28}$	$B_{25}, B_{86}, S_{13}, S_{28}$	B_{25}, S_{13}, S_{28}
S_8	B_{87}, S_{14}, S_{29}	B_{87}, S_{14}, S_{29}	S_{14}, S_{29}
S_9	B_{88}, S_{15}, S_{30}	B_{88}, S_{15}, S_{30}	S_{15}, S_{30}
S_{10}	B_{89}, S_0	B_{89}, S_0	S_0
S_{11}	B_0	B_0	B_0
B_{85}	N/A	–	–
B_{86}	N/A	N/A	–
B_{87}	N/A	N/A	–
B_{88}	N/A	N/A	–
B_{89}	N/A	N/A	–

Table 2: Recovery of state bits for $\mathscr{R}_1$, $\mathscr{R}_2$ and $\mathscr{R}_6$.

Now, we would like to illustrate the recovery of $\mathscr{R}_2 = \mathscr{R}_1 \cup \{B_{85}\}$, i.e. recovering 10 bits of internal state by fixing 16 bits of state variables. For recovering $\mathscr{R}_2$, we first recover $\mathscr{R}_1$ by fixing $\mathscr{F}_1$ (following the same method as above) along with fixing 4 additional bits, i.e. $\mathscr{F}_2 = \mathscr{F}_1 \cup \{B_8, B_{11}, B_{17}, B_{53}\}$.

From the constraint $B_{(i+44)} = 1$ and $B_{(i+8)} = 0$, equation (5) and equation (6) can be written as follows:

$$Q_i = B_{(i+4)}B_{(i+21)} \oplus B_{(i+9)}B_{(i+52)} \oplus B_{(i+18)}B_{(i+37)} \oplus B_{(i+76)}. \tag{12}$$

$$\begin{aligned} T_i = {} & B_{(i+5)} \oplus B_{(i+34)}B_{(i+67)}B_{(i+73)} \oplus B_{(i+2)}B_{(i+28)}B_{(i+41)}B_{(i+65)} \\ & \oplus B_{(i+13)}B_{(i+29)}B_{(i+50)}B_{(i+64)}B_{(i+75)} \oplus B_{(i+6)}B_{(i+14)}B_{(i+26)}B_{(i+32)}B_{(i+47)}B_{(i+61)} \\ & \oplus B_{(i+1)}B_{(i+19)}B_{(i+27)}B_{(i+43)}B_{(i+57)}B_{(i+66)}B_{(i+78)}. \end{aligned} \tag{13}$$

Thus, equation (3) can be written as:

$$\begin{aligned} B_{(i+76)} = {} & z_i \oplus L_i \oplus B_{(i+4)}B_{(i+21)} \oplus B_{(i+9)}B_{(i+52)} \oplus B_{(i+18)}B_{(i+37)} \oplus B_{(i+5)} \\ & \oplus B_{(i+34)}B_{(i+67)}B_{(i+73)} \oplus B_{(i+2)}B_{(i+28)}B_{(i+41)}B_{(i+65)} \\ & \oplus B_{(i+13)}B_{(i+29)}B_{(i+50)}B_{(i+64)}B_{(i+75)} \oplus B_{(i+6)}B_{(i+14)}B_{(i+26)}B_{(i+32)}B_{(i+47)}B_{(i+61)} \\ & \oplus B_{(i+1)}B_{(i+19)}B_{(i+27)}B_{(i+43)}B_{(i+57)}B_{(i+66)}B_{(i+78)} \oplus \overline{\mathcal{T}}_i. \end{aligned} \tag{14}$$

By putting $i = 9$ in equation (14), we get state bit B_{85} by guessing all state bits on the right hand side of the equation except S_{32} and S_{33}, which are feedback bits, that can be simply calculated from the already guessed bits. Note that fixing $B_{53} = 1$ would have been alone enough for recovering B_{85}, and hence $\mathscr{R}_2$. However, when substituting $i = 3$ in equation (11) (to recover S_6), we encounter $B_{85} \cdot B_{11}$ as a product term on the RHS of the equation. If only we fix B_{11} to 0, we can skip the guessing of B_{85} for recovering S_6 (which is during recovery of $\mathscr{R}_1$ itself). In other words, we either guess B_{85} and recover S_6 (and attain $\mathscr{R}_1$) or fix B_{11} to 0 and recover S_6. Choosing the second option enables us to recover B_{85} (and hence $\mathscr{R}_2$). A similar reason holds true for fixing B_8 and B_{17} to 0's. The only rule is that we do not guess what we need to recover, which can be done by carefully fixing some bits to 0's.

Now, we describe the last instance of our result, in which 14 internal state bits are recovered by fixing 30 bits i.e. $\mathscr{R}_6 = \mathscr{R}_2 \cup \{B_{86}, B_{87}, B_{88}, B_{89}\}$ since remaining instances can be directly inferred from the same. For this recovery, we fix $\mathscr{F}_6 = \mathscr{F}_2 \cup \mathscr{A}_3 \cup \mathscr{A}_4$ where $\mathscr{A}_3 = \{B_9, B_{10}, B_{12}, B_{13}, B_{14}, B_{15}, B_{18}, B_{19}, B_{20}, B_{21}\}$ and $\mathscr{A}_4 = \{B_{54}, B_{55}, B_{56}, B_{57}\}$. Initially, we recover $\mathscr{R}_2$ by fixing $\mathscr{F}_2$ (using our previous method). We fix $\mathscr{A}_3$ by 0's and $\mathscr{A}_4$ by 1's, for the same reason as mentioned before. We do not guess bits from $\mathscr{A}_3$ and $\mathscr{A}_4$ while recovering $\mathscr{R}_2$ since they are already fixed. By putting $i = 10, 11, 12, 13$ in equation (14), the internal state bits B_{86}, B_{87}, B_{88} and B_{89} are recovered by guessing all state bits on the right hand side of the equation. Note that all state bits on the right hand side of equation (14) are known beforehand either by guessing, fixing or recovering.

The state bits guessed in case of $\mathscr{R}_1, \mathscr{R}_2$ and $\mathscr{R}_6$ are listed in Table 2. We have also mentioned all the equations required to recover $\mathscr{R}_6$ in the Appendix A of the paper.

4 Recovery of State using Conditional TMDTO

It is well known from [7] that BSW sampling allows choice for a wider range of parameters of T, M, D with lower number of disk operations. We deduce some bits from the secret state by fixing certain key stream bit pattern. However, in case of Lizard, the output function has several nonlinear terms. Hence we resort to conditional BSW sampling as mentioned in Section 3. Fixing

- $\tau = |\mathscr{F}|$ bits of the secret state to a specific pattern,
- assuming a specific key stream bit pattern $\psi = |\mathscr{R}|$ bits and
- assigning values to the rest of the state bits $\nu - \psi - \tau$ (call this as a pattern ζ),

we deduce ψ bits of the secret state. As in TMDTO attack, the steps can be divided into two sections: (i) offline phase and (ii) online phase. The offline phase is used to compute and store tables by covering the search space of $P \approx \frac{N}{D}$, as usually done for stream ciphers [7]. The online phase involves the offline data to recover the state from the available data D during the attack. Note that the adversary can perform preprocessing P only once. While this technique is well known, we present most of the details for better explanation.

4.1 Preparing Tables for Offline Attack

Consider the state size of ν bits, i.e., the total search space $N = 2^{\nu}$. In conditional BSW sampling, we need to search for a ψ-bit key stream pattern and we need to fix τ state variables. The total search space thus decreases by $2^{\psi+\tau}$. Therefore, the total search space here is $N' = 2^{\nu-\psi-\tau}$. While we use the usual parameters P, T, M, D for TMDTO attack, for the reduced state we denote the parameters by P', T', M', D' and later connect those to the original parameters P, T, M, D. The table(s) that we will prepare during the offline phase will have m rows and t columns. As a part of our pre-computation (offline phase), we prepare $\frac{t}{D'}$ table(s), and taking $t = D'$, we can consider only one table. The preprocessing here is $P' = \frac{N'}{D'}$. Now $T'M'^2D'^2 = N'^2$ and we need to minimize the maximum of T, M, D finally. Towards this, we may consider $T' = D'^2$ and then satisfy $T'M'^2D'^2 = N'^2$ by $T' = 2^{\frac{\nu-\psi-\tau}{2}-x}$, $M' = 2^{\frac{\nu-\psi-\tau}{2}+x}$ and $D' = 2^{\frac{\nu-\psi-\tau}{4}-\frac{x}{2}}$ for some x.

Now we demonstrate the procedure of formulating the table(s) during preprocessing. We take a random pattern ζ of length $\nu - \psi - \tau$ bits. This will be stored in the table during pre-computation. Then we fix τ bits of the state according to a fixed pattern as decided in Section 3. Thus, we now obtain $\nu - \psi - \tau + \tau = \nu - \psi$ bits of the state. Now considering the ψ-bit fixed pattern of the key stream, we obtain the rest ψ bits of the state, providing the complete information about ν bit state. Naturally, if we clock this state as in PRGA, we will first obtain ψ key stream bits of the specific pattern that we have used already. We will keep clocking for next $\nu - \psi - \tau$ bits and that pseudo-random pattern will be then considered as the next element in the table, which is referred as $f(\zeta)$. This process will be repeated τ times to obtain a row. Such an action will be repeated for m such randomly chosen $(\nu - \psi - \tau)$ bit string to obtain m rows. Thus, mt many elements are generated in the preprocessing phase, of which we will only store the first and last element of each row. This makes the memory complexity $o(m)$. According to the birthday paradox, with proper parameters, this table will have negligible collisions. At the same time, the online data (here key stream) should be of such an amount so that the attack becomes successful, i.e., we obtain the intended pattern in the table.

4.2 Online Phase

As we discussed, the entries of each row from index $2, \ldots, t-1$ are removed to save on memory. The first entry in each row is referred to as SP (start point) and the last element as the EP (end point), which are the only two elements stored in each row. Now we consider access to data of D key stream bits. The adversary linearly searches the D-bit stream for the specific pattern of ψ bits. Upon a hit, the following $(\nu - \psi - \tau)$ bits (name this string as ζ) are taken as the pattern to be searched in the offline table. The table is considered to be optimized for searching in constant time. If there is a match, it means the state is stored in $(t-1)^{th}$ position of that row and one should start from SP of that row to obtain the $(\nu - \psi - \tau)$-bit pattern at the $(t-1)$-th location. That is, the adversary has to operate the function $(t-1)$ times on the SP of the same row to obtain the state. If there is no match, the adversary performs an f operation on ζ, i.e., we obtain $f(\zeta)$ and search through the EPs in each table again. The process is repeated till a hit is obtained. Note than in the worst-case, the adversary has to go through the entire row of length t.

The probability of getting an ψ-bit key stream pattern is $\frac{1}{2^\psi}$. Further, once we obtain an $(\nu - \psi - \tau)$-bit ζ, we try to discover the complete ν-bit state. Now, τ bits of the actual state may not be of the pattern as decided during the preparation of the preprocessing table. Thus, obtaining the correct state has a probability of $\frac{1}{2^\tau}$. Hence, to make the attack successful, the data complexity will be $D = D' \times 2^{\psi+\tau}$. Only when a specific ψ bit pattern comes in the key stream, we access the preprocessing table. Thus, $T = T' \times 2^\tau$.

We have already considered a parameter set with $T' = 2^{\frac{\nu-\psi-\tau}{2}-x}$, $M' = 2^{\frac{\nu-\psi-\tau}{2}+x}$ and $D' = 2^{\frac{\nu-\psi-\tau}{4}-\frac{x}{2}}$. Thus, $T = T' \times 2^\tau = 2^{\frac{\nu-\psi-\tau}{2}-x+\tau}$, $M = M' = 2^{\frac{\nu-\psi-\tau}{2}+x}$ and $D = D' \times 2^{\psi+\tau} = 2^{\frac{\nu-\psi-\tau}{4}-\frac{x}{2}+\psi+\tau}$. Now to minimize the maximum of T, M, D, we need to have $T = M = D$ and $T = M$ provides $x = \frac{\tau}{2}$. Now from $T = D$, we obtain $5\psi + 2\tau = \nu$. This actually provides us the best parameters for the TMDTO attack under the conditional BSW sampling. The complexity here is $T = M = D = 2^{\frac{\nu-\psi}{2}}$. With these parameters, the size of the preprocessing table will be $P = P' = \frac{N'}{D'} = 2^{\frac{\nu+\psi}{2}}$.

This is the optimal parameter and thus, for Lizard, we get $\nu = 121, \psi = 13, \tau = 28$. This provides the TMDTO parameters $T = M = D = 2^{54}$ with $P = 2^{67}$. Note that, we could generate a scenario where $\psi = 13, \tau = 27$. However, in this case, we may fix one more bit in the state to make $\tau = 28$ for obtaining our parameters that provides $5\psi + 2\tau = 5 \cdot 13 + 2 \cdot 28 = 121 = \nu$. This actually provides us the result as in * marked row of Table 3. Following the data available in Table 1, different other possible combinations of parameters of the attack have been shown in Table 3.

Bits fixed	Bits recovered	Search Space	Time Complexity	Memory	Data	Preprocessing Time
$(\tau = \lvert\mathscr{F}\rvert)$	$(\psi = \lvert\mathscr{R}\rvert)$	(N')	(T)	(M)	(D)	(P)
12	9	2^{100}	2^{56}	2^{56}	2^{43}	2^{78}
16	10	2^{95}	2^{56}	2^{55}	2^{46}	2^{75}
20	11	2^{90}	2^{56}	2^{54}	2^{49}	2^{72}
24	12	2^{85}	2^{56}	2^{53}	2^{52}	2^{69}
27	13	2^{81}	2^{55}	2^{53}	2^{54}	2^{67}
* 28	13	2^{80}	2^{54}	2^{54}	2^{54}	2^{67}
30	14	2^{77}	2^{52}	2^{55}	2^{55}	2^{66}

Table 3: The possible parameters for each value of $\lvert\mathscr{F}\rvert$ and $\lvert\mathscr{R}\rvert$.

It is important to clarify one issue at this point. While considering cryptanalysis we consider certain unit cost and that may involve several computations related to the cipher operations. In fact, given a k-bit secret key, the exhaustive attack requires complexity of 2^k units, where each unit may ask for several CPU clocks (might be of the order of 2^8 or more). While mounting the TMDTO attack the same situation is valid. However, we like to point out that our assumptions are as per to existing TMDTO attacks on stream ciphers as per the literature [7,12]. Note that, towards a generic TMDTO attack for state recovery in Lizard [11], we need to find out the image $f(\zeta)$ given ζ. This requires running the PRGA of Lizard 121 times to obtain the key stream bits and those 121 key stream bits are considered to be the next state in the chain. In our case, for the parameters $\tau = 28, \psi = 13$ requires 80-bit patterns to be the elements of the pre-processing table (instead of full 121 bits). From these 80 bits, we need to generate the full state of 121 bits according to the equations given in Appendix A. This requires some computation time. However, then we need to run the PRGA for $13 + 80 = 93$ bits instead of 121 bits. One may note that this advantage here will easily neutralize our computation for obtaining 121 bit state from 80 bit state. Thus, the time complexity for recovering state bits (using Appendix A) will not affect our online complexity.

We also like to point out the situation when $\tau = 0$ and $\psi = \frac{\nu}{5}$. That is, consider that from the key stream of length $\frac{\nu}{5}$ one may obtain $\frac{\nu}{5}$ unknown bits of the state by guessing rest $\frac{4\nu}{5}$ state bits. In that case, the preprocessing complexity will be $P = 2^{\frac{\nu+\psi}{2}} = 2^{\frac{3\nu}{5}}$ and $T = M = D = 2^{\frac{\nu-\psi}{2}} = 2^{\frac{2\nu}{5}}$. Thus, to have proper security against this kind of TMDTO attack based on conditional BSW sampling, the state size should be 2.5 times the secret key size. However, we are yet to obtain a recent well known stream cipher for which this can be achieved. Note that a closely related idea has been presented in [4, Page 10] while presenting the design of MICKEY 2.0.

5 Conclusion

In this paper, we present a TMDTO analysis on Lizard with pre-computation complexity 2^{67} and online complexity 2^{54} to recover the cipher state of 121 bits. Given that there are very serious advancement in hardware technology, these kind of complexities may be achieved in reasonable time. It should be noted that obtaining the state does not mean obtaining the secret key in Lizard due to its non-invertible KSA. That is, our attack provides only an inner state corresponding to one out of a large set of packets and there is no specific control for which packet the attacker can obtain the secret state. Thus, while our observation does not imply breaking Lizard, it should be considered as an insight towards security evaluation of this lightweight cipher.

Acknowledgments: The authors like to thank Willi Meier and Matthias Hamann for valuable comments on the initial drafts of this paper.

References

1. F. Armknecht and V. Mikhalev. On Lightweight Stream Ciphers with Shorter Internal States. FSE 2015, pp. 451–470, LNCS 9054, 2015. http://eprint.iacr.org/2015/131
2. ASIC Bitcoin Miner. https://www.amazon.com/Antminer-S9-16nm-Bitcoin-Miner/dp/B01HFXQ7AG
3. S. Babbage. A Space/Time Tradeoff in Exhaustive Search Attacks on Stream Ciphers. European Convention on Security and Detection, IEE Conference Publication No. 408, May 1995.

4. S. Babbage and M. Dodd. The stream cipher MICKEY 2.0. http://www.ecrypt.eu.org/stream/p3ciphers/mickey/mickey_p3.pdf, 30 June, 2006.
5. S. Banik and T. Isobe. Some cryptanalytic results on Lizard. http://eprint.iacr.org/2017/346
6. A. Biryukov, A. Shamir, and D. Wagner. Real Time Cryptanalysis of A5/1 on a PC. FSE 2000, pp. 1–18, LNCS 1978, 2000.
7. A. Biryukov and A. Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. Asiacrypt 2000, pp. 1–13, LNCS 1976, 2000.
8. Blockchain Hash Rate. https://blockchain.info/charts/hash-rate
9. M. F. Esgin and O. Kara. Practical cryptanalysis of full sprout with TMD tradeoff attacksr. International Conference on Selected Areas in Cryptography. Springer International Publishing, 2015.
10. J. Golic. Cryptanalysis of Alleged A5 Stream Cipher. EUROCRYPT'97, pp. 239–255, LNCS 1233, 1997.
11. M. Hamann, M. Krause and W. Meier. LIZARD - A Lightweight Stream Cipher for Power-constrained Devices. FSE 2017, http://eprint.iacr.org/2016/926, http://tosc.iacr.org/index.php/ToSC/article/view/584
12. M. Hamann and M. Krause and W. Meier and B. Zhang. Time-Memory-Data Tradeoff Attacks against Small-State Stream Ciphers. http://eprint.iacr.org/2017/384, 2017
13. M. E. Hellman. A Cryptanalytic Time-Memory Trade-Off, IEEE Transactions on Information Theory, Vol. IT-26, N 4, pp. 401–406, July 1980.
14. J. Hong and P. Sarkar. New applications of time memory data tradeoffs. ASIACRYPT 2005, pp. 353–372, LNCS 3788, 2005.
15. S. Maitra, A. Siddhanti and S. Sarkar. A differential fault attack on plantlet. IEEE Transactions on Computers (online 2017). http://ieeexplore.ieee.org/document/7917296/
16. V. Mikhalev, F. Armknecht and C. Müller. On ciphers that continuously access the non-volatile key. FSE 2017. TOSC, Volume 2016, Issue 2, pp. 52-79, 2016. http://tosc.iacr.org/index.php/ToSC/article/view/565/507
17. V. Lallemand and M. Naya-Plasencia. Cryptanalysis of Full Sprout. CRYPTO 2015, pp. 663–682, LNCS 9215, 2015. http://eprint.iacr.org/2015/232
18. B. Zhang and X. Gong. Another Tradeoff Attack on Sprout-Like Stream Ciphers. Asiacrypt 2015, pp. 561–585, LNCS 9453, 2015.

Appendix A: Recovery of bits

Here we summarise the equations used to recover state variables and the bits guessed for the same. We denote bits fixed with 1's by an underline (e.g. $\underline{S_{16}}$) and bits fixed by 0's by a bar (e.g. $\overline{B_{11}}$).

Step	Key bits	Equations used for recovery	Guessed bits	Feedback bits calculated	Recovered bits
0	z_0	$S_3\underline{S_{16}} = z_0 \oplus B_7 \oplus \overline{B_{11}} \oplus B_{30} \oplus B_{40} \oplus B_{45} \oplus \underline{B_{54}} \oplus B_{71}$ $\oplus B_4\overline{B_{21}} \oplus \overline{B_9}B_{52} \oplus \overline{B_{18}}B_{37} \oplus B_{44}B_{76} \oplus B_5 \oplus \overline{B_8}B_{82}$ $\oplus B_{34}B_{67}B_{73} \oplus B_2B_{28}B_{41}B_{65} \oplus \overline{B_{13}}B_{29}\overline{B_{50}}B_{64}B_{75}\oplus$ $B_6\overline{B_{14}}B_{26}B_{32}B_{47}B_{61} \oplus B_1\overline{B_{19}}B_{27}B_{43}\underline{B_{57}}B_{66}B_{78}\oplus$ $\underline{S_{23}} \oplus S_9S_{13}\overline{B_{48}} \oplus S_1\underline{S_{24}}B_{38}B_{63}$	$B_1, B_2, B_4, B_5, B_6, B_7, B_{26}, B_{27}, B_{28}, B_{29},$ $B_{30}, B_{32}, B_{34}, B_{37}, B_{38}, B_{40}, B_{41}, B_{43},$ $B_{44}, B_{45}, B_{47}, B_{52}, B_{61}, B_{63}, B_{64},$ $B_{65}, B_{66},$ $B_{67}, B_{71}, B_{73}, B_{75}, B_{76}, B_{78}, B_{82}, S_1,$	–	S_3
1	z_1	$S_4\underline{S_{17}} = z_1 \oplus \overline{B_8} \oplus \overline{B_{12}} \oplus B_{31} \oplus B_{41} \oplus B_{46} \oplus \underline{B_{55}} \oplus B_{72}$ $\oplus B_5B_{22} \oplus \overline{B_{10}}\underline{B_{53}} \oplus \overline{B_{19}}B_{38} \oplus B_{45}B_{77} \oplus B_6 \oplus \overline{B_9}B_{83}$ $\oplus B_{35}B_{68}B_{74} \oplus B_3B_{29}B_{42}B_{66} \oplus \overline{B_{14}}B_{30}B_{51}B_{65}B_{76}\oplus$ $B_7\overline{B_{15}}B_{27}B_{33}\overline{B_{48}}B_{62} \oplus B_2\overline{B_{20}}B_{28}B_{44}B_{58}B_{67}B_{79}\oplus$ $\underline{S_{24}} \oplus S_{10}S_{14}\overline{B_{49}} \oplus S_2S_{25}B_{39}B_{64}$	$B_3, B_{22}, B_{31},$ $B_{33}, B_{35}, B_{39}, B_{42}, B_{46}, B_{51},$ $B_{58}, B_{62}, B_{68}, B_{72}, B_{74},$ $B_{77}B_{79}, B_{83}, S_2, S_{25}$	–	S_4
2	z_2	$S_5\underline{S_{18}} = z_2 \oplus \overline{B_9} \oplus \overline{B_{13}} \oplus B_{32} \oplus B_{42} \oplus B_{47} \oplus \underline{B_{56}} \oplus B_{73}$ $\oplus B_6B_{23} \oplus \overline{B_{11}}\underline{B_{54}} \oplus \overline{B_{20}}B_{39} \oplus B_{46}B_{78} \oplus B_7 \oplus \overline{B_{10}}B_{84}$ $\oplus B_{36}B_{69}B_{75} \oplus B_4B_{30}B_{43}B_{67} \oplus \overline{B_{15}}B_{31}B_{52}B_{66}B_{77}\oplus$ $\overline{B_8}B_{16}B_{28}B_{34}\overline{B_{49}}B_{63} \oplus B_3\overline{B_{21}}B_{29}B_{45}B_{59}B_{68}B_{80}\oplus$ $S_{25} \oplus S_{11}S_{15}\overline{B_{50}} \oplus S_3S_{26}B_{40}B_{65}$	$B_{16}, B_{23}, B_{36}, B_{59}, B_{69}, B_{80}, B_{84}, S_{26}$	–	S_5
3	z_3	$S_6\underline{S_{19}} = z_3 \oplus \overline{B_{10}} \oplus \overline{B_{14}} \oplus B_{33} \oplus B_{43} \oplus B_{44} \oplus \underline{B_{57}} \oplus B_{74}$ $\oplus B_7B_{24} \oplus \overline{B_{12}}\underline{B_{55}} \oplus \overline{B_{21}}B_{40} \oplus B_{47}B_{79} \oplus \overline{B_8} \oplus \overline{B_{11}}B_{85}$ $\oplus B_{37}B_{70}B_{76} \oplus B_5B_{31}B_{44}B_{68} \oplus B_{16}B_{32}\underline{B_{53}}B_{67}B_{78}\oplus$ $\overline{B_9}\overline{B_{17}}B_{29}B_{35}\overline{B_{50}}B_{64} \oplus B_4B_{22}B_{30}B_{46}B_{60}B_{69}B_{81}\oplus$ $S_{26} \oplus S_{12}\underline{S_{16}}B_{51} \oplus S_4S_{27}B_{41}B_{66}$	$B_{24}, B_{60}, B_{70}, B_{81}, S_{12}, S_{27}$	–	S_6
4	z_4	$S_7\underline{S_{20}} = z_4 \oplus \overline{B_{11}} \oplus \overline{B_{15}} \oplus B_{34} \oplus B_{44} \oplus B_{45} \oplus B_{58} \oplus B_{75}$ $\oplus \overline{B_8}B_{25} \oplus \overline{B_{13}}\underline{B_{56}} \oplus B_{22}B_{41} \oplus \overline{B_{48}}B_{80} \oplus \overline{B_9} \oplus \overline{B_{12}}B_{86}$ $\oplus B_{38}B_{71}B_{77} \oplus B_6B_{32}B_{45}B_{69} \oplus \overline{B_{17}}B_{33}\underline{B_{54}}B_{68}B_{79}\oplus$ $\overline{B_{10}}\overline{B_{18}}B_{30}B_{36}B_{51}B_{65} \oplus B_5B_{23}B_{31}B_{47}B_{61}B_{70}B_{82}\oplus$ $S_{27} \oplus S_{13}\underline{S_{17}}B_{52} \oplus S_5S_{28}B_{42}B_{67}$	$B_{25}, B_{86}, S_{13}, S_{28}$	–	S_7
5	z_5	$S_8\underline{S_{21}} = z_5 \oplus \overline{B_{12}} \oplus B_{16} \oplus B_{35} \oplus B_{45} \oplus B_{46} \oplus B_{59} \oplus B_{76}$ $\oplus \overline{B_9}B_{26} \oplus \overline{B_{14}}\underline{B_{57}} \oplus B_{23}B_{42} \oplus \overline{B_{49}}B_{81} \oplus \overline{B_{10}} \oplus \overline{B_{13}}B_{87}$ $\oplus B_{39}B_{72}B_{78} \oplus B_7B_{33}B_{46}B_{70} \oplus \overline{B_{18}}B_{34}\underline{B_{55}}B_{69}B_{80}\oplus$ $\overline{B_{11}}\overline{B_{19}}B_{31}B_{37}B_{52}B_{66} \oplus B_6B_{24}B_{32}\overline{B_{48}}B_{62}B_{71}B_{83}\oplus$ $S_{28} \oplus S_{14}\underline{S_{18}}\underline{B_{53}} \oplus S_6S_{29}B_{43}B_{68}$	S_{14}, S_{29}	–	S_8
6	z_6	$S_9\underline{S_{22}} = z_6 \oplus \overline{B_{13}} \oplus \overline{B_{17}} \oplus B_{36} \oplus B_{46} \oplus B_{47} \oplus B_{60} \oplus B_{77}$ $\oplus \overline{B_{10}}B_{27} \oplus \overline{B_{15}}B_{58} \oplus B_{24}B_{43} \oplus \overline{B_{50}}B_{82} \oplus \overline{B_{11}} \oplus \overline{B_{14}}B_{88}$ $\oplus B_{40}B_{73}B_{79} \oplus \overline{B_8}B_{34}B_{47}B_{71} \oplus \overline{B_{19}}B_{35}\underline{B_{56}}B_{60}B_{81}\oplus$ $\overline{B_{12}}\overline{B_{20}}B_{32}B_{38}\underline{B_{53}}B_{67} \oplus B_7B_{25}B_{33}\overline{B_{49}}B_{63}B_{72}B_{84}\oplus$ $S_{29} \oplus S_{15}\underline{S_{19}}\underline{B_{54}} \oplus S_7S_{30}B_{44}B_{69}$	S_{15}, S_{30}	–	S_9

(Continued on next page.)

Step	Key bits	Equations used for recovery	Guessed bits	Feedback bits calculated	Recovered bits
7	z_7	$S_{10}\underline{S_{23}} = z_7 \oplus \overline{B_{14}} \oplus \overline{B_{18}} \oplus B_{37} \oplus B_{47} \oplus \overline{B_{48}} \oplus B_{61} \oplus B_{78}$ $\oplus \overline{B_{11}}B_{28} \oplus B_{16}B_{59} \oplus B_{25}B_{44} \oplus B_{51}B_{83} \oplus \overline{B_{12}} \oplus \overline{B_{15}}B_{89}$ $\oplus B_{41}B_{74}B_{80} \oplus \overline{B_9}B_{35}\overline{B_{48}}B_{72} \oplus \overline{B_{20}}B_{36}\underline{B_{57}}B_{61}B_{82}\oplus$ $\overline{B_{13}}\overline{B_{21}}B_{33}B_{39}\underline{B_{54}}B_{68} \oplus \overline{B_8}B_{26}B_{34}\overline{B_{50}}B_{64}B_{73}B_{85}\oplus$ $S_{30} \oplus \underline{S_{16}S_{20}B_{55}} \oplus S_8S_{31}B_{45}B_{70}$	S_0	S_{31}	S_{10}
8	z_8	$S_{11}\underline{S_{24}} = z_8 \oplus \overline{B_{15}} \oplus \overline{B_{19}} \oplus B_{38} \oplus \overline{B_{48}} \oplus \overline{B_{49}} \oplus B_{62} \oplus B_{79}$ $\oplus \overline{B_{12}}B_{29} \oplus \overline{B_{17}}B_{60} \oplus B_{26}B_{45} \oplus B_{52}B_{84} \oplus \overline{B_{13}} \oplus B_{16}B_{90}$ $\oplus B_{42}B_{75}B_{81} \oplus \overline{B_{10}}B_{36}\overline{B_{49}}B_{73} \oplus \overline{B_{21}}B_{37}B_{58}B_{62}B_{83}\oplus$ $\overline{B_{14}}B_{22}B_{34}B_{40}\underline{B_{55}}B_{69} \oplus \overline{B_9}B_{27}B_{35}B_{51}B_{65}B_{74}B_{86}\oplus$ $S_{31} \oplus \underline{S_{17}S_{21}B_{56}} \oplus S_9S_{32}B_{46}B_{71}$	–	S_{31}, S_{32}, B_{90}	S_{11}
9	z_9	$\underline{B_{53}}B_{85} = z_9 \oplus B_{16} \oplus \overline{B_{20}} \oplus B_{39} \oplus \overline{B_{49}} \oplus \overline{B_{50}} \oplus B_{63} \oplus B_{80}$ $\oplus \overline{B_{13}}B_{30} \oplus \overline{B_{18}}B_{61} \oplus B_{27}B_{46} \oplus \overline{B_{14}} \oplus \overline{B_{17}}B_{91}$ $\oplus B_{43}B_{76}B_{82} \oplus \overline{B_{17}}B_{37}\underline{B_{54}}B_{74} \oplus B_{22}B_{38}B_{59}B_{63}B_{84}\oplus$ $\overline{B_{15}}B_{23}B_{35}B_{41}\underline{B_{56}}B_{70} \oplus \overline{B_{10}}B_{28}B_{36}B_{52}B_{66}B_{75}B_{87}\oplus$ $S_{32} \oplus S_{12}S_{25} \oplus \underline{S_{18}S_{22}B_{57}} \oplus S_{10}S_{33}B_{47}B_{72}$	–	S_{32}, S_{33}, B_{91}	B_{85}
10	z_{10}	$\underline{B_{54}}B_{86} = z_{10} \oplus \overline{B_{17}} \oplus \overline{B_{21}} \oplus B_{40} \oplus \overline{B_{50}} \oplus \overline{B_{10}} \oplus B_{64} \oplus B_{81}$ $\oplus \overline{B_{14}}B_{31} \oplus \overline{B_{19}}B_{62} \oplus B_{28}B_{47} \oplus \overline{B_{15}} \oplus \overline{B_{18}}B_{92}$ $\oplus B_{44}B_{77}B_{83} \oplus \overline{B_{18}}B_{38}\underline{B_{55}}B_{75} \oplus B_{23}B_{39}B_{60}B_{64}B_{85}\oplus$ $B_{16}B_{24}B_{36}B_{42}\underline{B_{57}}B_{71} \oplus \overline{B_{11}}B_{29}B_{37}\underline{B_{53}}B_{67}B_{76}B_{88}\oplus$ $S_{33} \oplus S_{13}S_{26} \oplus \underline{S_{19}S_{23}}B_{58} \oplus S_{11}S_{34}\overline{B_{48}}B_{73}$	–	S_{33}, S_{34}, B_{92}	B_{86}
11	z_{11}	$\underline{B_{55}}B_{87} = z_{11} \oplus \overline{B_{18}} \oplus B_{22} \oplus B_{41} \oplus B_{51} \oplus \overline{B_{11}} \oplus B_{65} \oplus B_{82}$ $\oplus \overline{B_{15}}B_{32} \oplus \overline{B_{20}}B_{63} \oplus B_{29}\overline{B_{48}} \oplus B_{16} \oplus \overline{B_{19}}B_{93}$ $\oplus B_{45}B_{78}B_{84} \oplus \overline{B_{19}}B_{39}\underline{B_{56}}B_{76} \oplus B_{24}B_{40}B_{61}B_{65}B_{86}\oplus$ $\overline{B_{17}}B_{25}B_{37}B_{43}B_{58}B_{72} \oplus \overline{B_{12}}B_{30}B_{38}\underline{B_{54}}B_{68}B_{77}B_{89}\oplus$ $S_{34} \oplus S_{14}S_{27} \oplus \underline{S_{20}S_{24}}B_{59} \oplus S_{12}S_{35}\overline{B_{49}}B_{74}$	–	S_{34}, S_{35}, B_{93}	B_{87}
12	z_{12}	$\underline{B_{56}}B_{88} = z_{12} \oplus \overline{B_{19}} \oplus B_{23} \oplus B_{42} \oplus B_{52} \oplus \overline{B_{12}} \oplus B_{66} \oplus B_{83}$ $\oplus B_{16}B_{33} \oplus \overline{B_{21}}B_{64} \oplus B_{30}\overline{B_{49}} \oplus \overline{B_{17}} \oplus \overline{B_{20}}B_{94}$ $\oplus B_{46}B_{79}B_{85} \oplus \overline{B_{20}}B_{40}\underline{B_{57}}B_{77} \oplus B_{25}B_{41}B_{62}B_{66}B_{87}\oplus$ $\overline{B_{18}}B_{26}B_{38}B_{44}B_{59}B_{73} \oplus \overline{B_{13}}B_{31}B_{39}\underline{B_{55}}B_{69}B_{78}B_{90}\oplus$ $S_{35} \oplus S_{15}S_{28} \oplus \underline{S_{21}}S_{25}B_{60} \oplus S_{13}S_{36}\overline{B_{50}}B_{75}$	–	$S_{35}, S_{36}, B_{90}, B_{94}$	B_{88}
13	z_{13}	$\underline{B_{57}}B_{89} = z_{13} \oplus \overline{B_{20}} \oplus B_{24} \oplus B_{43} \oplus \underline{B_{53}} \oplus \overline{B_{13}} \oplus B_{67} \oplus B_{84}$ $\oplus \overline{B_{17}}B_{34} \oplus B_{22}B_{65} \oplus B_{31}\overline{B_{50}} \oplus \overline{B_{18}} \oplus \overline{B_{21}}B_{95}$ $\oplus B_{47}B_{80}B_{86} \oplus \overline{B_{21}}B_{41}B_{58}B_{78} \oplus B_{26}B_{42}B_{63}B_{67}B_{88}\oplus$ $\overline{B_{19}}B_{27}B_{39}B_{45}B_{60}B_{74} \oplus \overline{B_{14}}B_{32}B_{40}\underline{B_{56}}B_{70}B_{79}B_{91}\oplus$ $S_{36} \oplus \underline{S_{16}}S_{29} \oplus \underline{S_{22}}S_{26}B_{61} \oplus S_{14}S_{37}B_{51}B_{76}$	–	$S_{36}, S_{37}, B_{91}, B_{95}$	B_{89}

Table 4: Recovery of 14 bits of the internal state after fixing 30 bits.