Journal of Hardware and Systems Security (2019) 3:64–77
https://doi.org/10.1007/s41635-018-0054-1

Certain Observations on ACORN v3 and Grain v1—Implications Towards TMDTO Attacks

Akhilesh Anilkumar Siddhanti[1] · Subhamoy Maitra[2] · Nishant Sinha[3]

Received: 28 June 2018 / Accepted: 16 October 2018 / Published online 25 October 2018

Abstract

It is known that for a stream cipher with state size less than 2.5 times the key size, it is possible to mount a Time-Memory-Data Trade-Off attack with an online complexity lower than the exhaustive key search. The search space is restricted by considering a fixed keystream prefix and deducing certain state bits by formulating equations. We show how by using SAT solving techniques one can automate this process of solving equations and obtain better parameters. This is demonstrated by mounting TMDTO attacks on ACORN v3 and Grain v1. We show that a TMDTO attack can be mounted on ACORN v3 with a preprocessing complexity 2^{171} and 2^{180} (without and with the help of a SAT solver) and the maximum of online time, memory and data complexities 2^{122} and 2^{120} respectively. For Grain v1, we show that it is possible to obtain parameters as T = $2^{68.06}$, M = 2^{64}, D = 2^{68} with a preprocessing complexity of 2^{96}. While our results do not refute any claim of the designers, these observations might be useful for further understanding of the ciphers.

Keywords ACORN v3 · CAESAR · Cryptanalysis · Grain v1 · Stream cipher

1 Introduction

Stream ciphers are particularly useful in resource-constrained environments because of their low gate counts. The designers are hence competing to model stream ciphers with as low gate count as possible. In fact, the eStream portfolio saw one of its finalist Grain v1 [6] with a circuit size of 960 GE. However, it has been noted that designing stream ciphers with state size less than twice the key size makes them weak against the well known Time-Memory-Data Trade-Off (TMDTO) Attacks. Hence, it was considered a thumb rule to design stream ciphers with state size more than twice the key size, only to be proved wrong by the introduction of BSW sampling [1, 2], which asks for a state size minimum of 2.5 times the key size (one may refer to [8] for more details). CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) [4] has recently announced its finalists, and ACORN v3 is one among those [13]. ACORN v3 is a lightweight authenticated stream cipher with a state size of 293, composed of 6 Linear Feedback Shift Registers (LFSRs) and four additional bits. It promises a 128-bit security using a 128-bit secret key and IV.

Given that the present ciphers are designed with well-informed efforts, refuting the designer's claim are quite challenging and sometimes even elusive. However, there are important observations discovered by the cryptanalysts that help in providing more robust ciphers. This is the reason ACORN has been revised twice and the current version is

This work is an extended and revised version of the paper presented at SPACE 2017 [12]. Section 5 has been added in this draft over the conference version.

✉ Akhilesh Anilkumar Siddhanti
akhileshsiddhanti@gmail.com

Subhamoy Maitra
subho@isical.ac.in

Nishant Sinha
nishantsinha.iitr@gmail.com

[1] Department of Computer Science and Mathematics, BITS Pilani, Goa Campus, Vasco-da-Gama, Goa, 403726, India

[2] Applied Statistics Unit, Indian Statistical Institute, 203, B. T. Road, Kolkata 700108, India

[3] Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, India

ACORN v3. In this paper, we try to see how well one can obtain certain portion of the state bits for ACORN v3 and Grain v1, given some keystream bits and the rest of the bits of the state. This is related to sampling resistance as noted in [1, 2]. We calculate the sampling resistance by writing down several equations and feeding them to a SAT solver. Using SAT solving techniques, we have formed a generalized attack on stream ciphers with a state size less than 2.5 times the key size. The online complexity can be best reached to $T = M = D = N^{2/5}$ in such scenarios. We then observe this attack on Grain v1, a stream cipher from eStream portfolio whose state size is less than 2.5 times the key size. We will apply similar techniques for mounting an attack on Grain v1 as we have done for ACORN v3. Finally, we will list the possible TMDTO parameters and compare them.

1.1 Overview of the Paper

The paper is divided under the following sections:

- In Section 2, we will describe ACORN v3 relevant to our work.
- In Section 3, we provide ways to recover a certain portion of ACORN v3 using a fixed keystream sequence and guessing the remaining state bits.
- In Section 4, we discuss the possible TMDTO parameters for ACORN v3.
- In Section 5, we discuss how the attack can be applied to Grain v1.
- In Section 6, we compare the two attacks on ACORN v3 and Grain v1, and comment on its feasibility.
- In Section 7, we conclude our work.

2 Description of ACORN v3

We briefly state here the description of ACORN v3 relevant to our work. We assume the plaintext message to be a stream of 0's, and we concentrate only on the Pseudo Random Generation Algorithm (PRGA) that provides the keystream. We omit the Key Loading Algorithm (KLA) and the Key Scheduling Algorithm (KSA) of the cipher that are available at [13]. This is because the recovery of secret state bits during the PRGA and further the TMDTO attack can be studied irrespective of the initialization process. As stated before, ACORN v3 has six LFSRs and four additional bits concatenated to form the 293 bit state. The block diagram of ACORN is represented in Fig. 1 where f_t represents the feedback bit and m_t represents the message bit at t^{th} step [13]. We denote the state of the cipher by $\mathscr{S}_t$ and its respective bits as $S_{t+0} \ldots S_{t+292}$. The cipher has the following three functions.

1. Output Function: The output bit z_t for any state t is generated as

$$\begin{aligned} z_t &= S_{t+12} \oplus S_{t+154} \\ &\oplus maj(S_{t+235}, S_{t+61}, S_{t+193}) \\ &\oplus ch(S_{t+230}, S_{t+111}, S_{t+66}) \end{aligned} \tag{1}$$

where $maj()$ and $ch()$ functions are defined as following:

$$maj(x, y, z) = x\&y \oplus y\&z \oplus z\&x \tag{2}$$

$$ch(x, y, z) = x\&y \oplus (\sim x)\&z \tag{3}$$

2. Feedback Function: The feedback bit f_t for any state t is generated as

$$\begin{aligned} f_t = {} & S_{t+0} \oplus (\sim S_{t+107}) \oplus maj(S_{t+244}, S_{t+23}, \\ & S_{t+160}) \oplus (ca_t \& S_{t+196}) \oplus (cb_t \& z_t) \end{aligned} \tag{4}$$

3. State Update Function: Before performing the shift, the bits $S_{t+289}, S_{t+230}, S_{t+193}, S_{t+154}, S_{t+107}, S_{t+61}$ are updated as follows:

$$S_{t+289} = S_{t+289} \oplus S_{t+235} \oplus S_{t+230} \tag{5}$$

$$S_{t+230} = S_{t+230} \oplus S_{t+196} \oplus S_{t+193} \tag{6}$$

$$S_{t+193} = S_{t+193} \oplus S_{t+160} \oplus S_{t+154} \tag{7}$$

$$S_{t+154} = S_{t+154} \oplus S_{t+111} \oplus S_{t+107} \tag{8}$$

$$S_{t+107} = S_{t+107} \oplus S_{t+66} \oplus S_{t+61} \tag{9}$$

$$S_{t+61} = S_{t+61} \oplus S_{t+23} \oplus S_{t+0} \tag{10}$$

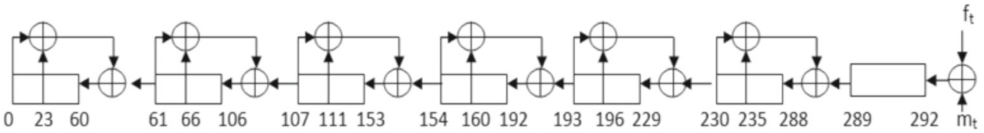

Fig. 1 The internal state of ACORN cipher

And then the next bit entering the state is initialized with the feedback bit:

$$S_{t+293} = f_t \tag{11}$$

3 Methods to Recover Certain Bits of the State for ACORN v3

The underlying motivation of BSW sampling [1, 2] is the fact that certain bits of the state can be recovered by observing the keystream sequence z_t and guessing the remaining part of the state. This reduces the search space and offers a wider range of parameters to choose from in TMDTO attack. We consider two approaches here. The first one is using the SAT solver, and the other one is by discovering the equations by hand using trial and error.

3.1 Using SAT Solver

Towards this, we first form a family of equations and then feeding them into a SAT solver. While building equations, the degree increases rapidly, which makes it very difficult to find solutions. Hence, we have to adopt a specific approach for formulating equations by introducing new variables. This is in line of [11]. Consider some PRGA round t of ACORN v3. The equations for the same round are:

1. 1 output bit equation,
2. 1 feedback bit equation, and
3. 6 state update equations.

At the beginning of PRGA, the adversary has 293 state variables $S_0, S_1, \ldots, S_{292}$. The adversary has access to an ℓ-length keystream $z_0,\ z_1,\ \ldots\ z_{\ell-1}$. We will now explain how the output equation is introduced into the system of equations. The output equation as mentioned in Eq. 1 is:

$$\begin{aligned} z_t \ &= \ S_{t+12} \oplus S_{t+154} \oplus maj(S_{t+235}, S_{t+61}, S_{t+193}) \\ &\oplus ch(S_{t+230}, S_{t+111}, S_{t+66}) \end{aligned} \tag{12}$$

To add an equation to the SAT solver, the equations are represented in a way such that it is zero in the ring of Boolean polynomials. That is, the output equation is written as

$$\begin{aligned} z_t \ &\oplus \ S_{t+12} \oplus S_{t+154} \oplus maj(S_{t+235}, S_{t+61}, S_{t+193}) \\ &\oplus ch(S_{t+230}, S_{t+111}, S_{t+66}) \equiv 0, \end{aligned} \tag{13}$$

for $t = 0, 1, 2, \ldots, \ell - 1$ and added to the system. Thus we have an array of output equations as

$$\begin{aligned} z_0 \ &\oplus \ S_{12} \oplus S_{154} \oplus maj(S_{235}, S_{61}, S_{193}) \\ &\oplus ch(S_{230}, S_{111}, S_{66}) \equiv 0 \\ z_1 \ &\oplus \ S_{13} \oplus S_{155} \oplus maj(S_{236}, S_{62}, S_{194}) \\ &\oplus ch(S_{231}, S_{112}, S_{67}) \equiv 0 \\ &\vdots \\ z_{\ell-1} \ &\oplus \ S_{\ell-1+12} \oplus S_{\ell-1+154} \\ &\oplus maj(S_{\ell-1+235}, S_{\ell-1+61}, S_{\ell-1+193}) \\ &\oplus ch(S_{\ell-1+230}, S_{\ell-1+111}, S_{\ell-1+66}) \equiv 0 \end{aligned}$$

Next, we discuss the inclusion of feedback bit equation into the system of equations. The equation as mentioned in Eq. 4 for PRGA is:

$$\begin{aligned} f_t \ &= \ S_{t+0} \oplus (\sim S_{t+107}) \oplus maj(S_{t+244}, \\ &S_{t+23}, S_{t+160}) \oplus S_{t+196} \end{aligned} \tag{14}$$

However, the feedback bit generated is not known. Thus directly substituting the state variable S_{t+293} by feedback equations increases non-linearity. Instead, the we introduce new variables $f_0,\ f_1, \ldots f_{\ell-1}$ and add these equations to the SAT solver in the following manner:

$$\begin{aligned} f_0 \ &\oplus \ S_0 \oplus (\sim S_{107}) \oplus maj(S_{244}, S_{23}, S_{160}) \\ &\oplus S_{196} \equiv 0 \\ f_1 \ &\oplus \ S_1 \oplus (\sim S_{108}) \oplus maj(S_{245}, S_{24}, S_{161}) \\ &\oplus S_{197} \equiv 0 \\ &\vdots \\ f_{\ell-1} \ &\oplus \ S_{\ell-1} \oplus (\sim S_{\ell-1+107}) \\ &\oplus maj(S_{\ell-1+244}, S_{\ell-1+23}, S_{\ell-1+160}) \\ &\oplus S_{\ell-1+196} \equiv 0 \end{aligned}$$

By now, 2ℓ new equations and ℓ new variables have been introduced into the system. The variables $S_{t+289}, S_{t+230}, S_{t+193}, S_{t+154}, S_{t+107}$, S_{t+61} are updated in Step 3 as mentioned earlier. For this, we introduce 6ℓ new variables $a_1^0, a_2^0, a_3^0, a_4^0, a_5^0, a_6^0, \ldots, a_1^{\ell-1}, a_2^{\ell-1}, a_3^{\ell-1}, a_4^{\ell-1}, a_5^{\ell-1}, a_6^{\ell-1}$ and add the following equations to the system (for $t = 0, 1, \ldots, \ell - 1$):

$$a_1^t \ \oplus \ S_{t+289} \oplus S_{t+235} \oplus S_{t+230} \equiv 0 \tag{15}$$

$$a_2^t \ \oplus \ S_{t+230} \oplus S_{t+196} \oplus S_{t+193} \equiv 0 \tag{16}$$

$$a_3^t \ \oplus \ S_{t+193} \oplus S_{t+160} \oplus S_{t+154} \equiv 0 \tag{17}$$

$$a_4^t \ \oplus \ S_{t+154} \oplus S_{t+111} \oplus S_{t+107} \equiv 0 \tag{18}$$

$$a_5^t \ \oplus \ S_{t+107} \oplus S_{t+66} \oplus S_{t+61} \equiv 0 \tag{19}$$

$$a_6^t \ \oplus \ S_{t+61} \oplus S_{t+23} \oplus S_{t+0} \equiv 0 \tag{20}$$

Since new variables have been introduced, new equations need to be introduced to maintain consistency of the system. That is, the following equations are added to the system:

$$a_1^t \oplus S_{t+288} \equiv 0 \tag{21}$$

$$a_2^t \oplus S_{t+229} \equiv 0 \tag{22}$$

$$a_3^t \oplus S_{t+192} \equiv 0 \tag{23}$$

$$a_4^t \oplus S_{t+153} \equiv 0 \tag{24}$$

$$a_5^t \oplus S_{t+106} \equiv 0 \tag{25}$$

$$a_6^t \oplus S_{t+60} \equiv 0 \tag{26}$$

for $t = 0, 1, \ldots, \ell - 1$. Finally, we substitute the feedback bit into the state variable:

$$S_{293+t} = f_t \qquad \forall t \in [0, \ell - 1]. \tag{27}$$

Therefore, the number of variables used are $293 + \ell + 6\ell = 293 + 7\ell$ and the number of equations formulated are $\ell + \ell + 6\ell = 8\ell$ equations. All the equations are collected and fed to the SAT solver.

We set the SAT solver to find all possible solutions for the above system of equations. In this way, we are guaranteed that if the SAT solver returns only one solution, no other solution exists for the system of equations, and hence, we can solve for the state. However, in few cases of our experiments, we could not achieve that. For example, when we consider recovery of 60 bits with the help of 70 keystream bits, we sometimes obtain two solutions. The reason for the same is that the number of keystream bits is not enough and thus the SAT solver provides more solutions instead of a unique solution.

We use the SAT solver Cryptominisat-2.9.6 available with Sage-7.6 [10]. The experiments were performed on a laptop having hardware configuration Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz and 8 GB RAM running with Ubuntu-16.10. A few experimental data are provided in Table 1 where each row is based on 2^{15} experiments.

Table 1 Experimental results for solving the equations

Keystream	State bits	Location of	# P	Average
Bits used	Recovered	Recovered bits		Time (sec)
47	47	$S_{107} \ldots S_{153}$	0	0.076
43	43	$S_{12} \ldots S_{54}$	0	0.067
72	60	$S_0 \ldots S_{59}$	$1/2^{10}$	0.127
60	53	$S_{107} \ldots S_{150}$, $S_{56}, \ldots S_{64}$	$1/2^{14}$	 0.097

The time required to run the PRGA for 293 clocks is 0.088 sec on an average. # P gives proportion of multiple (two) solutions

3.2 Formation of Equations by Observation, not using SAT Solver

In this section, we build the system of equations used to recover 49 bits of internal state by using first 49 bits of keystream. To perform this recovery, we need to fix 10 bits of internal state with a particular pattern and guess remaining state bits. The internal state bits to be recovered are represented by set $\mathscr{R} = \mathscr{R}_1 \cup \mathscr{R}_2$, where $\mathscr{R}_1 = \{S_{t+107} : t = 0, \ldots, 43\}$ and $\mathscr{R}_2 = \{S_{t+56} : t = 0, \ldots, 4\}$. The Eq. 1 for generating keystream can be written as

$$\begin{aligned} z_t \;=\; & S_{t+12} \oplus \overline{S_{t+154}} \oplus S_{t+235}\overline{S_{t+61}} \\ & \oplus\; S_{t+235}\overline{S_{t+193}} \oplus \overline{S_{t+193}}\overline{S_{t+61}} \\ & \oplus\; \overline{S_{t+230}}S_{t+111} \oplus \overline{S_{t+230}}S_{t+66} \oplus S_{t+66}. \end{aligned} \tag{28}$$

Note that in the above equation, over-lined bits are feedback bits. The state bits are updated according to the following equations before generating the output bit:

$$S_{t+289} = S_{t+289} \oplus S_{t+235} \oplus S_{t+230} \tag{29}$$

$$S_{t+230} = S_{t+230} \oplus S_{t+196} \oplus S_{t+193} \tag{30}$$

$$S_{t+193} = S_{t+193} \oplus S_{t+160} \oplus S_{t+154} \tag{31}$$

$$S_{t+154} = S_{t+154} \oplus S_{t+111} \oplus S_{t+107} \tag{32}$$

$$S_{t+107} = S_{t+107} \oplus S_{t+66} \oplus S_{t+61} \tag{33}$$

$$S_{t+61} = S_{t+61} \oplus S_{t+23} \oplus S_{t+0} \tag{34}$$

Thus, Eq. 28 can be written as

$$\begin{aligned} S_{t+107} \;=\; & z_t \oplus S_{t+12} \oplus S_{t+154} \oplus S_{t+111} \\ & \oplus\; S_{t+235}(S_{t+61} \oplus S_{t+23} \oplus S_{t+0}) \\ & \oplus\; S_{t+235}(S_{t+193} \oplus S_{t+160} \oplus S_{t+154}) \\ & \oplus\; (S_{t+193} \oplus S_{t+160} \oplus S_{t+154})(S_{t+61} \\ & \oplus\; S_{t+23} \oplus S_{t+0}) \oplus (S_{t+230} \\ & \oplus\; S_{t+196} \oplus S_{t+193})S_{t+111} \\ & \oplus\; (S_{t+230} \oplus S_{t+196} \oplus S_{t+193})S_{t+66} \\ & \oplus\; S_{t+66}, \end{aligned} \tag{35}$$

which makes the recovery simpler, because all the bits on the RHS of the equation are state bits (and not feedback bits) for $t = 0, \ldots, 32$. However when we place $t = 33, \ldots, 48$ in Eq. 35, feedback bits are also involved and need to be calculated.

Now we use Eq. 35 to recover internal state bits of set $\mathscr{R}_1$. The recovery of state bits is made in a certain order.

For example, if we attempt to recover S_{107} by placing $t = 0$ in Eq. 35, then S_{111} appears on the RHS of the equation and requires the knowledge of S_{111}. Thus, S_{111} is recovered before performing the recovery of S_{107}.

We define four sets $\mathscr{R}_3, \mathscr{R}_4, \mathscr{R}_5, \mathscr{R}_6$, where

$$\begin{aligned}
\mathscr{R}_3 &= \{S_{t+107} : t = 40, 36, \ldots, 0\}\\
\mathscr{R}_4 &= \{S_{t+107} : t = 41, 37, \ldots, 1\}\\
\mathscr{R}_5 &= \{S_{t+107} : t = 42, 38, \ldots, 2\}\\
\mathscr{R}_6 &= \{S_{t+107} : t = 43, 39, \ldots, 3\}
\end{aligned}$$

and each $\mathscr{R}_i \subset \mathscr{R}_1$, for $i = 3 \ldots, 6$. The order of recovery of state bits is $\mathscr{R}_3, \mathscr{R}_4, \mathscr{R}_5, \mathscr{R}_6$ and $\mathscr{R}_2$, respectively, i.e., the state bits of $\mathscr{R}_3$ are recovered first then $\mathscr{R}_4$ and so on. For each set $\mathscr{R}_i : i = 2 \ldots, 6$, the higher index elements are recovered first. We need not fix any internal state bits for recovering $\mathscr{R}_1$. However, to recover $\mathscr{R}_2$, the internal state bits are fixed according to Table 2. Let the set $\mathscr{F}$ represent the internal state bits which are fixed according to Table 2.

Now we describe recovery of $\mathscr{R}_3$. The internal state bit S_{147} is recovered by substituting $t = 40$ in Eq. 35. From this, we have

$$\begin{aligned}
S_{147} = {} & z_{40} \oplus S_{52} \oplus \overline{S_{194}} \oplus S_{151} \oplus S_{275}(S_{101}\\
& \oplus \overline{S_{63}} \oplus S_{40}) \oplus S_{275}(\overline{S_{233}} \oplus \overline{S_{200}}\\
& \oplus \overline{S_{194}}) \oplus (\overline{S_{233}} \oplus \overline{S_{200}} \oplus \overline{S_{194}})(S_{101}\\
& \oplus \overline{S_{63}} \oplus S_{40}) \oplus (S_{270} \oplus \overline{S_{236}} \oplus \overline{S_{233}})S_{151}\\
& \oplus (S_{270} \oplus \overline{S_{236}} \oplus \overline{S_{233}})S_{106} \oplus S_{106}. \qquad (36)
\end{aligned}$$

In Eq. 36, all the bits appearing on the RHS of the equation are guessed, except the over-lined bits. The over-lined bits are feedback bits and not internal state bits due to Eq. 34. Thus, we need to guess more internal state bits to calculate the value of S_{63}, S_{194}, S_{200}, S_{233}, and S_{236} using Eq. 34. In this way, we recover S_{147}.

Table 2 State bits fixed

Row #	State bits and value
1.	$S_{i+268} = 0 : i = 0 \ldots, 4$
2.	$S_{i+187} = S_{i+226} \oplus S_{i+193}$ $\oplus S_{i+160} \oplus S_{i+154} : i = 0 \ldots, 3$
3.	$S_{191} = S_{230} \oplus S_{196}$ $\oplus S_{193} \oplus S_{197} \oplus S_{164} \oplus S_{158}$

Now the internal state bit of S_{143} is recovered by placing $t = 36$ in Eq. 35 and we derive

$$\begin{aligned}
S_{143} = {} & z_{36} \oplus S_{48} \oplus S_{190} \oplus S_{147} \oplus S_{271}(S_{97}\\
& \oplus S_{59} \oplus S_{36}) \oplus S_{271}(S_{229} \oplus \overline{S_{196}} \oplus S_{190})\\
& \oplus (S_{229} \oplus \overline{S_{196}} \oplus S_{190})(S_{97} \oplus S_{59} \oplus S_{36})\\
& \oplus (S_{266} \oplus \overline{S_{232}} \oplus S_{229})S_{147}(\oplus S_{266} \oplus \overline{S_{232}}\\
& \oplus S_{229})S_{102} \oplus S_{102}. \qquad (37)
\end{aligned}$$

Similarly, in Eq. 37, all the state bits appearing on the right side of the equation need to be guessed, except S_{271}, S_{190} and the over-lined bits. The internal state bits S_{271} and S_{190} are fixed according to Table 2. The over-lined bits are calculated using Eq. 34. Thus, we need to guess more internal state bits to calculate the value of S_{196}, S_{232} and recover S_{143}.

The remaining state bits of $\mathscr{R}_3$ i.e. S_{139}, S_{135}, $\ldots$, S_{107} are recovered by substituting $t = 32, 28, \ldots, 0$, respectively, in Eq. 35. While placing $t = 32, 28, \ldots, 0$ in Eq. 35, the internal state bits appearing on the RHS of the equation are guessed, except state bits belonging to $\mathscr{R}$ and $\mathscr{F}$. Following the same methodology, the internal state bits of set $\mathscr{R}_4$, $\mathscr{R}_5$ and $\mathscr{R}_6$ are recovered.

To recover the state bits of set $\mathscr{R}_2$, a similar procedure is followed, except for Eq. 28, which is rewritten as

$$\begin{aligned}
S_{t+12} = {} & z_t \oplus S_{t+107} \oplus S_{t+154} \oplus S_{t+111}\\
& \oplus S_{t+235}(S_{t+61} \oplus S_{t+23} \oplus S_{t+0})\\
& \oplus S_{t+235}(S_{t+193} \oplus S_{t+160} \oplus S_{t+154})\\
& \oplus (S_{t+193} \oplus S_{t+160} \oplus S_{t+154})(S_{t+61}\\
& \oplus S_{t+23} \oplus S_{t+0}) \oplus (S_{t+230} \oplus S_{t+196}\\
& \oplus S_{t+193})S_{t+111} \oplus (S_{t+230} \oplus S_{t+196}\\
& \oplus S_{t+193})S_{t+66} \oplus S_{t+66}. \qquad (38)
\end{aligned}$$

Thus, the internal state bits $S_{56}, \ldots, S_{60}$ are recovered by using $t = 44, \ldots, 48$ in Eq. 38, respectively. Another difference between recovery of $\mathscr{R}_1$ and $\mathscr{R}_2$ is that it is not necessary to recover the higher indexed elements first (as done before).

In this way, we recover 49 bits of $\mathscr{R}$ by fixing the 10 internal state bits of set $\mathscr{F}$ and guessing the remaining 234 state bits. However, there are nine internal state bits, i.e., $S_{284}, \ldots, S_{292}$ which have not appeared in the equations used for recovery. However, these bits are also considered as guessed bits during application of TMDTO attack. The equations used for recovery of $\mathscr{R}$ have been mentioned in Table 6 in the Appendix. The over-lined state bits and underlined state bits in Table 6 in the Appendix are feedback bits and fixed state bits (according to Table 2), respectively.

4 Complexity of TMDTO Attack for ACORN v3

Now we will describe the TMDTO attack in complete detail. We have a state size of $n = 293$ bits. Thus, the standard TMDTO formula [1, 2] with a single table will be as follows:

- $TM^2D^2 = N^2$, where $N = 2^n$,
- $D^2 \leq T$,
- $P = \frac{N}{D}$.

During the preprocessing phase, we will prepare a table with m rows and t columns, where $mt^2 = N$ for a successful attack. The number of tables is $\frac{t}{D}$ and given a single table we have $t = D$. Each row of the table contains a chain of t elements. Consider that a specific state of $n = 293$ bits is ζ and f is the one way function. Here by one way function f, we mean that the cipher with the state ζ will be clocked for n times to generate n keystream bits. These n bits will be loaded into the new state, which is called η. That is $\eta = f(\zeta)$. We will start with a random state and then generate a row of t elements by this method. There will be m such rows. Thus, the total table size is mt. However, the complete row will not be saved. Only the starting and the final element will be kept. Thus, the storage requirement of the table will be $O(m)$, which is the memory parameter M.

4.1 Knowledge of 47 bits of State from 47 Keystream Bits

Now consider the case when we are able to recover ψ bits of the state from ψ consecutive keystream bits and the rest of the state bits. In this case, we consider a fixed pattern for the keystream bits and only when that pattern is found in the keystream, we try to search the state in the table. Thus, in this case, we consider a state size of $n - \psi$ bits and the parameters are referred as $N' = 2^{n-\psi}, P', M', T', D'$. Let us now consider the exact parameters referring to Table 1, where $\psi = 47$. Thus, $T'M'D'^2 = N'^2 = 2^{2(293-47)}$. Let us consider $D'^2 = T$. Thus, we have $T'M' = 2^{293-47} = 2^{246}$. Now, one can consider, $T' = M' = 2^{123}$ and $D' = 2^{61.5}$. However, as we have discussed that during the online phase, we can only mount the attack when a specific ψ-bit pattern comes, we have $D = 2^{\psi}D'$. Thus, finally, we will have the parameters $T = T' = 2^{123}$, $M = M' = 2^{123}$, $D = 2^{\psi}D' = 2^{47} \cdot 2^{61.5} = 2^{108.5}$, $P = P' = \frac{N'}{D'} = 2^{184.5}$. This provides the maximum of online parameters as 2^{123}, which is less than the exhaustive secret key search of complexity 2^{128}. However, as expected, the pre-processing time is much larger than the exhaustive key search.

At this point, we would like to explain the "unit" cost related to exact complexity. Such unit cost may involve several computations related to the cipher operations. In a most generic way, given a k-bit secret key, the exhaustive attack asks for the complexity of 2^k units, where each unit may require several CPU clocks. While mounting the TMDTO attack the same situation is valid. Thus, in our technique, we also consider all the operations as unit cost. However, we will point out a few cases when our calculations are most costly, and that should be taken care of in the complexity analysis. For example, simply generating a 293-bit keystream (that will become the state η) of ACORN v3 from a state ζ requires 0.088 sec in our computing facility. However, to recover the 47 bits of the state from 47 bits of keystream and the remaining state bits requires a time of 0.076 sec, which is almost as same as the time taken to generate ζ. Thus, no additional complexity is required for solving. Hence for this scenario, our parameters are as follows. We can take $T' = 2^{122}$, $M' = 2^{124}$ and $D' = 2^{61}$. Then, $T = T' \cdot 2^0 = 2^{122}$, $M = M' = 2^{124}$, $D = 2^{\psi} \cdot D' = 2^{47} \cdot 2^{61} = 2^{108}$, $P = P' = \frac{N'}{D'} = 2^{185}$.

4.2 Knowledge of 53 Bits of State from 60 Keystream Bits

We follow a similar procedure as mentioned in Section 4.1. However, when the SAT solver is populated with equations and is set to find all possible solutions for 53 state bits using only 53 keystream bits, the SAT solver fails to find a unique solution. Instead, we get multiple solutions, where each solution provides the same 53-bit keystream pattern. To combat this problem, we involve a new idea. Instead of searching for a 53-bit pattern (say 53 continuous 0's), we search for a 300-bit pattern where the first 53-bit sequence and the last 7-bit sequence are fixed (say to 0's). This is based on the fact that the keystream sequence generated by all solutions are different. The SAT solver identifies the difference between the sequence of last 7 bits and removes all additional solutions. However, this gives us an additional Data complexity of 2^7. Considering this constraint into the SAT solver in a similar fashion (as explained in Section 3.1), we get the data mentioned in Table 1. However, in very few cases, two solutions sets are possible which generate the same keystream. Since the proportion is very small and our success probability is $\frac{2^{14}-1}{2^{14}}$, the time complexity should be multiplied by $T' = T' \times \frac{2^{14}}{2^{14}-1} \approx T' \times 1 = T'$. However, we still attempt to deal with this edge case scenario of two solutions. The idea is to discard the second solution during the offline phase and continue with the first solution set. The matrix stopping rule ensures the entire search space is covered with a negligible collision. During the online phase, the adversary can access few more keystream bits following the fixed pattern in the keystream and hence conclude with the final solution. Our experiments show that 7 more keystream bits, i.e., 67 keystream bits in total are enough to find a unique solution.

Similar to Section 4.1, the time taken for solving equations is of the same order of generating ζ, hence $T = T'$.

- $T = T' = 2^{120}$ is the total time complexity of the attack,
- $M = M' = 2^{120}$ is the amount of memory required,
- $D = D' \times 2^{60} = 2^{120}$ where $D' = 2^{60}$, since the adversary must succeed in finding a 60-bit pattern,
- $P = \frac{N'}{D'} = 2^{180}$ is the preprocessing time for formulating tables.

4.3 Knowledge of 49 Bits of State From 49 Keystream Bits and Fixing 10 State Bits

Here, we consider the third approach which is similar to what has been recently considered in [8] for a TMDTO attack against Lizard [5]. We consider that ψ state bits can be recovered from ψ many keystream bits and rest of the state bits, but τ many state bits has to be fixed to a specific pattern. This follows the idea mentioned in Section 3.2. In this case, we go back to the single preprocessing table. We will consider $\psi = 49$ here, with $\tau = 10$. That is from ψ bits of keystream and the remaining $(n - \psi)$ state bits (out of which τ are fixed to a specific pattern), we will be able to solve for the ψ bits of the state. The initial table preparation goes as follows. We start with a $(n - \psi - \tau)$ bit random pattern and then take a specific pattern for ψ. Also, the fixed pattern of ϕ is known. Now, using the equations as described in Section 3.2, we solve for the rest ψ bits of the state. This gives the complete state. Then we run the PRGA for $n - \tau$ times. The initial ψ bits will be as fixed. The remaining $(n - \psi - \tau)$ pseudorandom bits will be considered as the part of the next state bits. Thus, we have $T'M' = 2^{n-\psi-\tau} = 2^{293-49-10} = 2^{234}$. Let us take $T' = 2^{112}$ and $M' = 2^{122}$, which also gives, $D' = \sqrt{T'} = 2^{56}$. Thus, we will now have the following parameters.

- $D = D' \cdot 2^{\psi+\tau} = 2^{56+49+10} = 2^{115}$, as the specific pattern ψ should come towards consulting the table, and also for a good success rate to have the specific τ bit pattern in the state we need to try 2^{τ} many times,
- $M = M' = 2^{122}$,
- $T = T' \cdot 2^{\tau} = 2^{112+10} = 2^{122}$, as we only consult the preprocessing table when the specific ψ bit pattern appears in the keystream, but we need to try 2^{τ} times as we have that more data and here the solution time can be estimated from the operations in the equations and that can subsumed in the PRGA effort,
- $P = P' = \frac{N'}{D'} = 2^{234-56} = 2^{178}$.

A similar online parameter in this respect can be obtained considering the equation $5\psi + 2\tau = n$. Here, $\psi = 49$, by fixing $\tau = 10$. However, we can easily increase τ to 24 to satisfy the equation $5\psi + 2\tau = 5\cdot49 + 2\cdot24 = 293 = n$. That is we will fix 24 state bits to a specific pattern. In this case, the online complexity becomes $T = M = D = 2^{\frac{n-\psi}{2}} = 2^{\frac{293-49}{2}} = 2^{122}$. However, the preprocessing becomes less, which is $P = 2^{\frac{n+\psi}{2}} = 2^{\frac{293+49}{2}} = 2^{171}$.

5 Using SAT Solving Techniques to Mount TMDTO on Grain v1

Using SAT solver and a given keystream sequence, we could recover a portion of the state of ACORN v3. Now we would like to see, how this new approach can help us mount an unconditional TMDTO attack on one of the finalists of eStream portfolio, Grain v1, and compare it with the existing attacks on Grain v1. We will begin by describing Grain v1.

5.1 Description of Grain v1

Grain v1 constitutes of an LFSR and an NFSR, each of size 80 bits, and supports an 80-bit secret key with a 64-bit IV. The encryption process is divided into 3 phases: KLA (Key Loading Algorithm), KSA (Key Scheduling Algorithm), and PRGA (Pseudo Random Generating Algorithm).

Notation The LFSR bits are represented by $l_0, l_1, \ldots, l_{79}$ while the NFSR bits are represented as $n_0, n_1, \ldots, n_{79}$. Similarly, the key bits are represented as $k_0, k_1, \ldots, k_{79}$ and the IV bits as $v_0, v_1, \ldots, v_{63}$.

1. **KLA:** The lower 64 bits of LFSR are initialized with the IV concatenated by a 16-bit string of 1's, while the NFSR is initialized with the key bits:

$$l_i = v_i \text{ for } 0 \le i \le 63. \quad (39)$$

$$l_i = 1 \text{ for } 64 \le i \le 79, \quad (40)$$

$$n_i = k_i \text{ for } 0 \le i \le 79, \quad (41)$$

2. **KSA:** The Key Scheduling Algorithm is carried out for 160 rounds, where in each round, the LFSR and NFSR are shifted by 1 bit and a new bit generated by the feedback routine enters the state. The feedback routine for LFSR can be described as

$$\begin{aligned} l_{i+80} &= l_{i+62} \oplus l_{i+51} \oplus l_{i+38} \oplus l_{i+23} \\ &\oplus\ l_{i+13} \oplus l_i \oplus z_i \end{aligned} \tag{42}$$

   And the NFSR feedback routine is described as follows:

$$\begin{aligned} n_{i+80} &= z_i \oplus l_i \oplus n_{i+62} \oplus n_{i+60} \oplus n_{i+52} \\ &\oplus\ n_{i+45} \oplus n_{i+37} \oplus n_{i+33} \oplus n_{i+28} \oplus \\ n_{i+9} &\oplus\ n_i \oplus n_{i+63}n_{i+60} \oplus n_{i+37}n_{i+33} \\ &\oplus\ n_{i+15}n_{i+9} \oplus n_{i+60}n_{i+52}n_{i+45} \\ &\oplus\ n_{i+33}n_{i+28}n_{i+21} \\ &\oplus\ n_{i+63}n_{i+45}n_{i+28}n_{i+9} \\ &\oplus\ n_{i+60}n_{i+52}n_{i+37}n_{i+33} \\ &\oplus\ an_{i+63}n_{i+60}n_{i+52}n_{i+45}n_{i+37} \\ &\oplus\ n_{i+33}n_{i+28}n_{i+21}n_{i+15}n_{i+9} \\ &\oplus\ n_{i+52}n_{i+45}n_{i+37}n_{i+33}n_{i+28}n_{i+21} \end{aligned} \tag{43}$$

   where z_i is an output bit (used internally during KSA) computed as

$$\begin{aligned} z_i &= n_{i+1} \oplus n_{i+2} \oplus n_{i+4} \oplus n_{i+10} \\ &\oplus\ n_{i+31} \oplus n_{i+43} \oplus n_{i+56} \oplus h(x) \end{aligned} \tag{44}$$

   and $h(x)$ is a non-linear function:

$$\begin{aligned} h(x) &= x_1 \oplus x_4 \oplus x_0x_3 \oplus x_2x_3 \oplus x_3x_4 \oplus x_0x_1x_2 \\ &\oplus\ x_0x_2x_3 \oplus x_0x_2x_4 \oplus x_1x_2x_4 \oplus x_2x_3x_4 \end{aligned} \tag{45}$$

   where x_0, x_1, x_2, x_3, x_4 correspond to the tap positions $l_{i+3}, l_{i+25}, l_{i+46}, l_{i+64}, n_{i+63}$.

3. **PRGA:** After 160 rounds of KSA, the cipher is ready to produce keystream bits. The output function remains the same from KSA. The LFSR and NFSR registers are updated in a similar fashion as KSA, except that the output bits are not XORed into the state, but produced externally. Note that the output bit is generated prior to a state update.

   Now that we have discussed the structure of Grain v1, we shall discuss how by guessing a certain portion of the state bits and assuming a fixed-length keystream sequence, we can recover the remaining state bits.

5.2 Formulating Equations for Grain v1

The procedure of formulating equations for Grain v1 is very similar to that of ACORN v3 as mentioned in Section 3.1. First, 160 variables are introduced into the system, for representing the internal state bits. Next, we assume that we know a portion of the internal state and directly substitute the known values instead of the state variables. We also have a fixed keystream $z_0, z_1, \ldots, z_{\psi-1}$. In our case, we do not include any equations of the LFSR, since we are guessing the values of LFSR bits in our attack. Hence, the LFSR bits are clocked as mentioned in Eq. 42 without the keystream bit (since we are talking about PRGA rounds here). For updating the NFSR bits, we substitute the known values directly into the NFSR state update routine using Eq. 43 (without the keystream bit) and directly substitute the newly generated bit into the state (instead of introducing an extra variable as done in Section 3.1); since this is giving us a considerable speed boost.

We clock the state ψ times, and for each round, we calculate the output bit (which will be a boolean polynomial) using Eq. 44 and XOR the polynomial with the corresponding keystream bit z_i before adding it to the set of equations. We first convert this system of ANF equations into CNF using the popular dense strategy (available with SAGE). Now the equations are expressible in terms of input to SAT solvers, we can use any SAT solver to solve for a solution. One popular SAT solver is Cryptominisat 5.6, which we have used for performing our experiments.

Now we will describe our attack on Grain v1.

5.3 Recovering 25 State Bits using 30 Keystream Bits

We follow a similar idea as mentioned in Section 4.2. The equations are formulated as mentioned in the Section 5.2. Here, we are considering NFSR bits $n_0, n_1, \ldots, n_{24}$ for recovery, by guessing the remaining state bits and using the fixed length keystream pattern. Like in Section 4.2, we are faced with a situation of multiple solutions for the system of equations. Hence, we resort to increasing the number of keystream bits from 25 to 30. The chance of two solutions in this scenario is $1/2^5$. However, we can tackle this situation

Table 3 The possible TMDTO parameters inclusive of the time complexity of the SAT solver

$T = T' \times 2^{2.90}$	$M = M'$	$D = D' \times 2^{30}$	$P = N'/D'$
$2^{69.90}$	2^{68}	$2^{63.5}$	$2^{101.5}$
$2^{67.90}$	2^{70}	$2^{62.5}$	$2^{102.5}$
$2^{72.90}$	2^{66}	2^{64}	2^{101}
$2^{71.90}$	2^{68}	$2^{62.5}$	$2^{102.5}$

Table 4 The possible TMDTO parameters inclusive of the time complexity of the SAT solver (Recovering 32 state bits using 36 keystream bits)

$T = T' \times 2^{4.06}$	$M = M'$	$D = D' \times 2^{36}$	$P = N'/D'$
$2^{72.06}$	2^{60}	2^{70}	2^{94}
$2^{70.06}$	2^{62}	2^{69}	2^{95}
$\mathbf{2^{68.06}}$	$\mathbf{2^{64}}$	$\mathbf{2^{68}}$	$\mathbf{2^{96}}$
$2^{66.06}$	2^{66}	2^{67}	2^{97}

The bold entry is the best attack parameter

by following a similar approach as followed in Section 4.2. We search for a 160 bit pattern in the keystream sequence where the first 25 bits and the last 5 bits are a pattern of our choice, say all 0's. This will increase the data complexity by 2^{30}, while our search space is now reduced to $N' = 2^{160-25} = 2^{135}$. We have to find appropriate parameters for T', M' and D' such that $T'M'^2D'^2 \geq N'^2$. We can choose $T' = M' = 2^{67.5}$ and $D' = 2^{33.75}$ such that $T = M = 2^{67.5}$ and $D = 2^{63.75}$. The offline complexity of the attack will stand at $P = 2^{101.25}$.

Note that the complexity of SAT solver also needs to be accounted for. Similar to Section 4.2, we will compare the average time to solve one system of equations with the time taken to compute 160 keystream bits in SAGE 8.2. The average time taken to solve a system of equations is 0.059 seconds, whereas the time taken to compute 160 keystream bits is 0.0079 seconds. From this, we can assume that the complexity of solving equations to be $2^{2.90}$ Grain v1 encryptions.

The different possible TMDTO parameters are mentioned in Table 3.

5.4 Recovering 32 State Bits using 36 Keystream Bits

Here, we will recover NFSR bits $n_0, n_1, \ldots, n_{31}$ by guessing remaining state bits and using a 36-bit fixed keystream pattern. The chance of two solutions in this case would be 2^{-5}. The search space in this case would be $N' = 2^{128}$. A suitable parameter in this case would be $T' = 2^{64}$, $M' = 2^{64}$, $D' = 2^{32}$ and consequently $P = N'/D' = 2^{96}$.

The average time taken to solve the system of equations in this case is 0.42 seconds. Considering the time taken for producing 160 keystream bits from the previous Section 5.3, the complexity of solving equations in this case would be $2^{4.06}$. The various possible TMDTO parameters are given in Table 4.

6 Comparison and Viability of the Two Attacks

We have seen TMDTO attacks on ACORN v3 and Grain v1 in Section 4 and Section 5, respectively. Even in the best case scenario for our attack on ACORN v3, we need a preprocessing complexity of at least 2^{171} (Section 4.3) which is considerably higher than exhaustive key search. However, a preprocessing complexity higher than exhaustive key search has often been allowed since it is a one-time cost.

Note that for Grain v1, the offline complexity is close to the exhaustive key search. We cannot claim that we have broken the cipher since it is still slower than exhaustive key search. But since the tables need to be prepared only once and then can repeatedly be used during the online phase with a complexity lower than exhaustive key search, the attack does seem practical in future. A comparison of our work for Grain v1 with existing works has been mentioned in Table 5.

7 Conclusion

In this paper, we have studied a generalized TMDTO attack by using SAT solver on stream ciphers with state size less than 2.5 times its key size. We have presented a TMDTO attack on ACORN v3 and Grain v1. We have shown how we can formulate equations and by using SAT solver, we can deduce a portion of the state using a fixed keystream pattern and guessing the remaining state bits. Then we have enlisted the possible TMDTO parameters for each case. We observe that the online parameters are lower than exhaustive key search, while the pre-processing complexity still remains higher than exhaustive key search. While our observations do not refute any security claim of the cipher, the study adds certain insight towards the cryptanalysis and may lead to further research in this area.

Acknowledgments The authors like to thank Dr. Dibyendu Roy, School of Mathematical Science, NISER, India, for an excellent review on an initial version of this paper.

Table 5 Comparison of complexities with existing TMDTO attacks on Grain-v1

Algorithm	State bits recovered	State bits fixed	Keystream bits used	Time (T)	Memory (M)	Keystream (D)	Preprocessing (P)
Bjørstad [3]	21	0	21	2^{70}	2^{69}	2^{56}	2^{104}
Mihaljević et al. [9]	18	54	18	2^{54}	2^{70}	2^{72}	2^{88}
Jiao et al. [7]	28	51	28	2^{61}	2^{71}	2^{79}	2^{81}
Our approach	32	0	36	$2^{68.06}$	2^{64}	2^{64}	2^{96}

Appendix

Table 6 Recovery of 49 bits of the internal state after fixing 10 bits

Steps	Equations used for recovery	Guessed bits
0	$S_{147} = z_{40} \oplus S_{52} \oplus \overline{S_{194}} \oplus S_{151} \oplus S_{275}(S_{101} \oplus \overline{S_{63}} \oplus S_{40})$ $\oplus S_{275}(\overline{S_{233}} \oplus \overline{S_{200}} \oplus \overline{S_{194}}) \oplus (\overline{S_{233}} \oplus \overline{S_{200}} \oplus \overline{S_{194}})$ $(S_{101} \oplus \overline{S_{63}} \oplus S_{40}) \oplus (\underline{S_{270}} \oplus \overline{S_{236}} \oplus \overline{S_{233}})S_{151}$ $\oplus(\underline{S_{270}} \oplus \overline{S_{236}} \oplus \overline{S_{233}})S_{106} \oplus S_{106}$	$S_{52}, S_{101}, S_{63}, S_{25}, S_{2},$ $S_{40}, S_{275}, S_{233}, S_{199},$ $S_{196}, S_{200}, S_{167}, S_{161},$ $S_{194}, S_{155}, S_{236}, S_{202},$ S_{151}, S_{106}
1	$S_{143} = z_{36} \oplus S_{48} \oplus \underline{S_{190}} \oplus S_{147} \oplus \underline{S_{271}}(S_{97} \oplus S_{59} \oplus S_{36})$ $\oplus \underline{S_{271}}(S_{229} \oplus \overline{S_{196}} \oplus \underline{S_{190}}) \oplus (S_{229} \oplus \overline{S_{196}} \oplus \underline{S_{190}})$ $(S_{97} \oplus S_{59} \oplus S_{36}) \oplus (S_{266} \oplus \overline{S_{232}} \oplus S_{229})S_{147}$ $\oplus(S_{266} \oplus \overline{S_{232}} \oplus S_{229})S_{102} \oplus S_{102}$	$S_{48}, S_{97}, S_{36}, S_{229},$ $S_{163}, S_{157}, S_{266}, S_{232},$ $S_{198}, S_{195}, S_{102}$
2	$S_{139} = z_{32} \oplus S_{44} \oplus S_{186} \oplus S_{143} \oplus S_{267}(S_{93} \oplus S_{55} \oplus S_{32})$ $\oplus S_{267}(S_{225} \oplus S_{192} \oplus S_{186}) \oplus (S_{225} \oplus S_{192} \oplus S_{186})$ $(S_{93} \oplus S_{55} \oplus S_{32}) \oplus (S_{262} \oplus S_{228} \oplus S_{225})S_{143}$ $\oplus(S_{262} \oplus S_{228} \oplus S_{225})S_{98} \oplus S_{98}$	$S_{44}, S_{93}, S_{55}, S_{32},$ $S_{267}, S_{225}, S_{192}, S_{186}$ S_{262}, S_{228}, S_{98}
3	$S_{135} = z_{28} \oplus S_{40} \oplus S_{182} \oplus S_{139} \oplus S_{263}(S_{89} \oplus S_{51} \oplus S_{28})$ $\oplus S_{263}(S_{221} \oplus \underline{S_{188}} \oplus S_{182}) \oplus (S_{221} \oplus \underline{S_{188}} \oplus S_{182})$ $(S_{89} \oplus S_{51} \oplus S_{28}) \oplus (S_{258} \oplus S_{224} \oplus S_{221})S_{139}$ $\oplus(S_{258} \oplus S_{224} \oplus S_{221})S_{94} \oplus S_{94}$	$S_{89}, S_{51}, S_{28}, S_{263},$ $S_{221}, S_{182}, S_{258}, S_{224},$ S_{94}
4	$S_{131} = z_{24} \oplus S_{36} \oplus S_{178} \oplus S_{135} \oplus S_{259}(S_{85} \oplus S_{47} \oplus S_{24})$ $\oplus S_{259}(S_{217} \oplus S_{184} \oplus S_{178}) \oplus (S_{217} \oplus S_{184} \oplus S_{178})$ $(S_{85} \oplus S_{47} \oplus S_{24}) \oplus (S_{254} \oplus S_{220} \oplus S_{217})S_{135}$ $\oplus(S_{254} \oplus S_{220} \oplus S_{217})S_{90} \oplus S_{90}$	$S_{85}, S_{47}, S_{24}, S_{259},$ $S_{217}, S_{184}, S_{178}, S_{254},$ S_{220}, S_{90}
5	$S_{127} = z_{20} \oplus S_{32} \oplus S_{174} \oplus S_{131} \oplus S_{255}(S_{81} \oplus S_{43} \oplus S_{20})$ $\oplus S_{255}(S_{213} \oplus S_{180} \oplus S_{174}) \oplus (S_{213} \oplus S_{180} \oplus S_{174})$ $(S_{81} \oplus S_{43} \oplus S_{20}) \oplus (S_{250} \oplus S_{216} \oplus S_{213})S_{131}$ $\oplus(S_{250} \oplus S_{216} \oplus S_{213})S_{86} \oplus S_{86}$	$S_{81}, S_{43}, S_{20}, S_{255},$ $S_{213}, S_{180}, S_{174},$ S_{250}, S_{216}, S_{86}
6	$S_{123} = z_{16} \oplus S_{28} \oplus S_{170} \oplus S_{127} \oplus S_{251}(S_{77} \oplus S_{39} \oplus S_{16})$ $\oplus S_{251}(S_{209} \oplus S_{176} \oplus S_{170}) \oplus (S_{209} \oplus S_{176} \oplus S_{170})$ $(S_{77} \oplus S_{39} \oplus S_{16}) \oplus (S_{246} \oplus S_{212} \oplus S_{209})S_{127}$ $\oplus(S_{246} \oplus S_{212} \oplus S_{209})S_{82} \oplus S_{82}$	$S_{77}, S_{39}, S_{16}, S_{251},$ $S_{209}, S_{176}, S_{170}, S_{246},$ S_{212}, S_{82}
7	$S_{119} = z_{12} \oplus S_{24} \oplus S_{166} \oplus S_{123} \oplus S_{247}(S_{73} \oplus S_{35} \oplus S_{12})$ $\oplus S_{247}(S_{205} \oplus S_{172} \oplus S_{166}) \oplus (S_{205} \oplus S_{172} \oplus S_{166})$ $(S_{73} \oplus S_{35} \oplus S_{12}) \oplus (S_{242} \oplus S_{208} \oplus S_{205})S_{123}$ $\oplus(S_{242} \oplus S_{208} \oplus S_{205})S_{78} \oplus S_{78}$	$S_{73}, S_{35}, S_{12}, S_{247},$ $S_{205}, S_{172}, S_{166}, S_{242},$ S_{208}, S_{78}
8	$S_{115} = z_{8} \oplus S_{20} \oplus S_{162} \oplus S_{119} \oplus S_{243}(S_{69} \oplus S_{31} \oplus S_{8})$ $\oplus S_{243}(S_{201} \oplus S_{168} \oplus S_{162}) \oplus (S_{201} \oplus S_{168} \oplus S_{162})$ $(S_{69} \oplus S_{31} \oplus S_{8}) \oplus (S_{238} \oplus S_{204} \oplus S_{201})S_{119}$ $\oplus(S_{238} \oplus S_{204} \oplus S_{201})S_{74} \oplus S_{74}$	$S_{69}, S_{31}, S_{8}, S_{243},$ $S_{201}, S_{168}, S_{162}, S_{238},$ S_{204}, S_{74}
9	$S_{111} = z_{4} \oplus S_{16} \oplus S_{158} \oplus S_{115} \oplus S_{239}(S_{65} \oplus S_{27} \oplus S_{4})$ $\oplus S_{239}(S_{197} \oplus S_{164} \oplus S_{158}) \oplus (S_{197} \oplus S_{164} \oplus S_{158})$ $(S_{65} \oplus S_{27} \oplus S_{4}) \oplus (S_{234} \oplus S_{200} \oplus S_{197})S_{115}$ $\oplus(S_{234} \oplus S_{200} \oplus S_{197})S_{70} \oplus S_{70}$	$S_{65}, S_{27}, S_{4}, S_{239},$ $S_{197}, S_{164}, S_{158}, S_{234},$ S_{70}
10	$S_{107} = z_{0} \oplus S_{12} \oplus S_{154} \oplus S_{111} \oplus S_{235}(S_{61} \oplus S_{23} \oplus S_{0})$ $\oplus S_{235}(S_{193} \oplus S_{160} \oplus S_{154}) \oplus (S_{193} \oplus S_{160} \oplus S_{154})$ $(S_{61} \oplus S_{23} \oplus S_{0}) \oplus (S_{230} \oplus S_{196} \oplus S_{193})S_{111}$ $\oplus(S_{230} \oplus S_{196} \oplus S_{193})S_{66} \oplus S_{66}$	$S_{61}, S_{23}, S_{0}, S_{235},$ $S_{193}, S_{160}, S_{154}, S_{230},$ S_{66}

Table 6 (continued)

Steps	Equations used for recovery	Guessed bits
11	$S_{148} = z_{41} \oplus S_{53} \oplus \overline{S_{195}} \oplus S_{152} \oplus S_{276}(S_{102} \oplus \overline{S_{64}} \oplus S_{41})$ $\oplus S_{276}(\overline{S_{234}} \oplus \overline{S_{201}} \oplus \overline{S_{195}}) \oplus (\overline{S_{234}} \oplus \overline{S_{201}} \oplus \overline{S_{195}})$ $(S_{102} \oplus \overline{S_{64}} \oplus S_{41}) \oplus (\underline{S_{271}} \oplus \overline{S_{237}} \oplus \overline{S_{234}})S_{152}$ $\oplus(\underline{S_{271}} \oplus \overline{S_{237}} \oplus \overline{S_{234}})\overline{S_{107}} \oplus \overline{S_{107}}$	$S_{53}, S_{64}, S_{26}, S_3,$ $S_{41}, S_{276}, S_{156}, S_{237},$ S_{203}, S_{152}
12	$S_{144} = z_{37} \oplus S_{49} \oplus \underline{S_{191}} \oplus S_{148} \oplus \underline{S_{272}}(S_{98} \oplus S_{60} \oplus S_{37})$ $\oplus \underline{S_{272}}(\overline{S_{230}} \oplus \overline{S_{197}} \oplus \underline{S_{191}}) \oplus (\overline{S_{230}} \oplus \overline{S_{197}} \oplus \underline{S_{191}})$ $(S_{98} \oplus S_{60} \oplus S_{37}) \oplus (S_{267} \oplus \overline{S_{233}} \oplus \overline{S_{230}})S_{148}$ $\oplus(S_{267} \oplus \overline{S_{233}} \oplus \overline{S_{230}})S_{103} \oplus S_{103}$	S_{49}, S_{37}, S_{103}
13	$S_{140} = z_{33} \oplus S_{45} \oplus \underline{S_{187}} \oplus S_{144} \oplus \underline{S_{268}}(S_{94} \oplus S_{56} \oplus S_{33})$ $\oplus \underline{S_{268}}(S_{226} \oplus \overline{S_{193}} \oplus \underline{S_{187}}) \oplus (S_{226} \oplus \overline{S_{193}} \oplus \underline{S_{187}})$ $(S_{94} \oplus S_{56} \oplus S_{33}) \oplus (S_{263} \oplus S_{229} \oplus S_{226})S_{144}$ $\oplus(S_{263} \oplus S_{229} \oplus S_{226})S_{99} \oplus S_{99}$	$S_{45}, S_{94}, S_{33},$ S_{226}, S_{99}
14	$S_{136} = z_{29} \oplus S_{41} \oplus S_{183} \oplus S_{140} \oplus S_{264}(S_{90} \oplus S_{52} \oplus S_{29})$ $\oplus S_{264}(S_{222} \oplus \underline{S_{189}} \oplus S_{183}) \oplus (S_{222} \oplus \underline{S_{189}} \oplus S_{183})$ $(S_{90} \oplus S_{52} \oplus S_{29}) \oplus (S_{259} \oplus S_{225} \oplus S_{222})S_{140}$ $\oplus(S_{259} \oplus S_{225} \oplus S_{222})S_{95} \oplus S_{95}$	$S_{29}, S_{264}, S_{222},$ S_{183}, S_{95}
15	$S_{132} = z_{25} \oplus S_{37} \oplus S_{179} \oplus S_{136} \oplus S_{260}(S_{86} \oplus S_{48} \oplus S_{25})$ $\oplus S_{260}(S_{218} \oplus S_{185} \oplus S_{179}) \oplus (S_{218} \oplus S_{185} \oplus S_{179})$ $(S_{86} \oplus S_{48} \oplus S_{25}) \oplus (S_{255} \oplus S_{221} \oplus S_{218})S_{136}$ $\oplus(S_{255} \oplus S_{221} \oplus S_{218})S_{91} \oplus S_{91}$	$S_{260}, S_{218}, S_{185},$ S_{179}, S_{91}
16	$S_{128} = z_{21} \oplus S_{33} \oplus S_{175} \oplus S_{132} \oplus S_{256}(S_{82} \oplus S_{44} \oplus S_{21})$ $\oplus S_{256}(S_{214} \oplus S_{181} \oplus S_{175}) \oplus (S_{214} \oplus S_{181} \oplus S_{175})$ $(S_{82} \oplus S_{44} \oplus S_{21}) \oplus (S_{251} \oplus S_{217} \oplus S_{214})S_{132}$ $\oplus(S_{251} \oplus S_{217} \oplus S_{214})S_{87} \oplus S_{87}$	$S_{21}, S_{256}, S_{214},$ S_{181}, S_{175}, S_{87}
17	$S_{124} = z_{17} \oplus S_{29} \oplus S_{171} \oplus S_{128} \oplus S_{252}(S_{78} \oplus S_{40} \oplus S_{17})$ $\oplus S_{252}(S_{210} \oplus S_{177} \oplus S_{171}) \oplus (S_{210} \oplus S_{177} \oplus S_{171})$ $(S_{78} \oplus S_{40} \oplus S_{17}) \oplus (S_{247} \oplus S_{213} \oplus S_{210})S_{128}$ $\oplus(S_{247} \oplus S_{213} \oplus S_{210})S_{83} \oplus S_{83}$	$S_{17}, S_{252}, S_{210},$ S_{177}, S_{171}, S_{83}
18	$S_{120} = z_{13} \oplus S_{25} \oplus S_{167} \oplus S_{124} \oplus S_{248}(S_{74} \oplus S_{36} \oplus S_{13})$ $\oplus S_{248}(S_{206} \oplus S_{173} \oplus S_{167}) \oplus (S_{206} \oplus S_{173} \oplus S_{167})$ $(S_{74} \oplus S_{36} \oplus S_{13}) \oplus (S_{243} \oplus S_{209} \oplus S_{206})S_{124}$ $\oplus(S_{243} \oplus S_{209} \oplus S_{206})S_{79} \oplus S_{79}$	$S_{13}, S_{248}, S_{206},$ S_{173}, S_{79}
19	$S_{116} = z_9 \oplus S_{21} \oplus S_{163} \oplus S_{120} \oplus S_{244}(S_{70} \oplus S_{32} \oplus S_9)$ $\oplus S_{244}(S_{202} \oplus S_{169} \oplus S_{163}) \oplus (S_{202} \oplus S_{169} \oplus S_{163})$ $(S_{70} \oplus S_{32} \oplus S_9) \oplus (S_{239} \oplus S_{205} \oplus S_{202})S_{120}$ $\oplus(S_{239} \oplus S_{205} \oplus S_{202})S_{75} \oplus S_{75}$	$S_9, S_{244}, S_{169}, S_{75}$
20	$S_{112} = z_5 \oplus S_{17} \oplus S_{159} \oplus S_{116} \oplus S_{240}(S_{66} \oplus S_{28} \oplus S_5)$ $\oplus S_{240}(S_{198} \oplus S_{165} \oplus S_{159}) \oplus (S_{198} \oplus S_{165} \oplus S_{159})$ $(S_{66} \oplus S_{28} \oplus S_5) \oplus (S_{235} \oplus S_{201} \oplus S_{198})S_{116}$ $\oplus(S_{235} \oplus S_{201} \oplus S_{198})S_{71} \oplus S_{71}$	$S_5, S_{240}, S_{165},$ S_{159}, S_{71}
21	$S_{108} = z_1 \oplus S_{13} \oplus S_{155} \oplus S_{112} \oplus S_{236}(S_{62} \oplus S_{24} \oplus S_1)$ $\oplus S_{236}(S_{194} \oplus S_{161} \oplus S_{155}) \oplus (S_{194} \oplus S_{161} \oplus S_{155})$ $(S_{62} \oplus S_{24} \oplus S_1) \oplus (S_{231} \oplus S_{197} \oplus S_{194})S_{112}$ $\oplus(S_{231} \oplus S_{197} \oplus S_{194})S_{67} \oplus S_{67}$	$S_{62}, S_1, S_{231}, S_{67}$
22	$S_{149} = z_{42} \oplus S_{54} \oplus \overline{S_{196}} \oplus S_{153} \oplus S_{277}(S_{103} \oplus \overline{S_{65}} \oplus S_{42})$ $\oplus S_{277}(\overline{S_{235}} \oplus \overline{S_{202}} \oplus \overline{S_{196}}) \oplus (\overline{S_{235}} \oplus \overline{S_{202}} \oplus \overline{S_{196}})$ $(S_{103} \oplus \overline{S_{65}} \oplus S_{42}) \oplus (\underline{S_{272}} \oplus \overline{S_{238}} \oplus \overline{S_{235}})S_{153}$ $\oplus(\underline{S_{272}} \oplus \overline{S_{238}} \oplus \overline{S_{235}})\overline{S_{108}} \oplus \overline{S_{108}}$	$S_{54}, S_{42}, S_{277}, S_{153}$

Table 6 (continued)

Steps	Equations used for recovery	Guessed bits
23	$S_{145} = z_{38} \oplus S_{50} \oplus S_{192} \oplus S_{149} \oplus S_{273}(S_{99} \oplus \overline{S_{61}} \oplus S_{38})$ $\oplus S_{273}(\overline{S_{231}} \oplus \overline{S_{198}} \oplus S_{192}) \oplus (\overline{S_{231}} \oplus \overline{S_{198}} \oplus S_{192})$ $(S_{99} \oplus \overline{S_{61}} \oplus S_{38}) \oplus (\underline{S_{268}} \oplus \overline{S_{234}} \oplus \overline{S_{231}})S_{149}$ $\oplus(\underline{S_{268}} \oplus \overline{S_{234}} \oplus \overline{S_{231}})S_{104} \oplus S_{104}$	$S_{50}, S_{38}, S_{273}, S_{104}$
24	$S_{141} = z_{34} \oplus S_{46} \oplus \underline{S_{188}} \oplus S_{145} \oplus \underline{S_{269}}(S_{95} \oplus S_{57} \oplus S_{34})$ $\oplus\underline{S_{269}}(S_{227} \oplus \overline{S_{194}} \oplus \underline{S_{188}}) \oplus (S_{227} \oplus \overline{S_{194}} \oplus \underline{S_{188}})$ $(S_{95} \oplus S_{57} \oplus S_{34}) \oplus (S_{264} \oplus \overline{S_{230}} \oplus S_{227})S_{145}$ $\oplus(S_{264} \oplus \overline{S_{230}} \oplus S_{227})S_{100} \oplus S_{100}$	S_{46}, S_{34}, S_{227} S_{100}
25	$S_{137} = z_{30} \oplus S_{42} \oplus S_{184} \oplus S_{141} \oplus S_{265}(S_{91} \oplus S_{53} \oplus S_{30})$ $\oplus S_{265}(S_{223} \oplus \underline{S_{190}} \oplus S_{184}) \oplus (S_{223} \oplus \underline{S_{190}} \oplus S_{184})$ $(S_{91} \oplus S_{53} \oplus S_{30}) \oplus (S_{260} \oplus S_{226} \oplus S_{223})S_{141}$ $\oplus(S_{260} \oplus S_{226} \oplus S_{223})S_{96} \oplus S_{96}$	$S_{30}, S_{265}, S_{223}, S_{96}$
26	$S_{133} = z_{26} \oplus S_{38} \oplus S_{180} \oplus S_{137} \oplus S_{261}(S_{87} \oplus S_{49} \oplus S_{26})$ $\oplus S_{261}(S_{219} \oplus S_{186} \oplus S_{180}) \oplus (S_{219} \oplus S_{186} \oplus S_{180})$ $(S_{87} \oplus S_{49} \oplus S_{26}) \oplus (S_{256} \oplus S_{222} \oplus S_{219})S_{137}$ $\oplus(S_{256} \oplus S_{222} \oplus S_{219})S_{92} \oplus S_{92}$	S_{261}, S_{219}, S_{92}
27	$S_{129} = z_{22} \oplus S_{34} \oplus S_{176} \oplus S_{133} \oplus S_{257}(S_{83} \oplus S_{45} \oplus S_{22})$ $\oplus S_{257}(S_{215} \oplus S_{182} \oplus S_{176}) \oplus (S_{215} \oplus S_{182} \oplus S_{176})$ $(S_{83} \oplus S_{45} \oplus S_{22}) \oplus (S_{252} \oplus S_{218} \oplus S_{215})S_{133}$ $\oplus(S_{252} \oplus S_{218} \oplus S_{215})S_{88} \oplus S_{88}$	$S_{22}, S_{257}, S_{215}, S_{88}$
28	$S_{125} = z_{18} \oplus S_{30} \oplus S_{172} \oplus S_{129} \oplus S_{253}(S_{79} \oplus S_{41} \oplus S_{18})$ $\oplus S_{253}(S_{211} \oplus S_{178} \oplus S_{172}) \oplus (S_{211} \oplus S_{178} \oplus S_{172})$ $(S_{79} \oplus S_{41} \oplus S_{18}) \oplus (S_{248} \oplus S_{214} \oplus S_{211})S_{129}$ $\oplus(S_{248} \oplus S_{214} \oplus S_{211})S_{84} \oplus S_{84}$	$S_{18}, S_{253}, S_{211}, S_{84}$
29	$S_{121} = z_{14} \oplus S_{26} \oplus S_{168} \oplus S_{125} \oplus S_{249}(S_{75} \oplus S_{37} \oplus S_{14})$ $\oplus S_{249}(S_{207} \oplus S_{174} \oplus S_{168}) \oplus (S_{207} \oplus S_{174} \oplus S_{168})$ $(S_{75} \oplus S_{37} \oplus S_{14}) \oplus (S_{244} \oplus S_{210} \oplus S_{207})S_{125}$ $\oplus(S_{244} \oplus S_{210} \oplus S_{207})S_{80} \oplus S_{80}$	$S_{14}, S_{249}, S_{207}, S_{80}$
30	$S_{117} = z_{10} \oplus S_{22} \oplus S_{164} \oplus S_{121} \oplus S_{245}(S_{71} \oplus S_{33} \oplus S_{10})$ $\oplus S_{245}(S_{203} \oplus S_{170} \oplus S_{164}) \oplus (S_{203} \oplus S_{170} \oplus S_{164})$ $(S_{71} \oplus S_{33} \oplus S_{10}) \oplus (S_{240} \oplus S_{206} \oplus S_{203})S_{121}$ $\oplus(S_{240} \oplus S_{206} \oplus S_{203})S_{76} \oplus S_{76}$	S_{10}, S_{245}, S_{76}
31	$S_{113} = z_{6} \oplus S_{18} \oplus S_{160} \oplus S_{117} \oplus S_{241}(S_{67} \oplus S_{29} \oplus S_{6})$ $\oplus S_{241}(S_{199} \oplus S_{166} \oplus S_{160}) \oplus (S_{199} \oplus S_{166} \oplus S_{160})$ $(S_{67} \oplus S_{29} \oplus S_{6}) \oplus (S_{236} \oplus S_{202} \oplus S_{199})S_{117}$ $\oplus(S_{236} \oplus S_{202} \oplus S_{199})S_{72} \oplus S_{72}$	S_{6}, S_{241}, S_{72}
32	$S_{109} = z_{2} \oplus S_{14} \oplus S_{156} \oplus S_{113} \oplus S_{237}(S_{63} \oplus S_{25} \oplus S_{2})$ $\oplus S_{237}(S_{195} \oplus S_{162} \oplus S_{156}) \oplus (S_{195} \oplus S_{162} \oplus S_{156})$ $(S_{63} \oplus S_{25} \oplus S_{2}) \oplus (S_{232} \oplus S_{198} \oplus S_{195})S_{113}$ $\oplus(S_{232} \oplus S_{198} \oplus S_{195})S_{68} \oplus S_{68}$	S_{68}
33	$S_{150} = z_{43} \oplus S_{55} \oplus \overline{S_{197}} \oplus \overline{S_{154}} \oplus S_{278}(S_{104} \oplus \overline{S_{66}} \oplus S_{43})$ $\oplus S_{278}(\overline{S_{236}} \oplus \overline{S_{203}} \oplus \overline{S_{197}}) \oplus (\overline{S_{236}} \oplus \overline{S_{203}} \oplus \overline{S_{197}})$ $(S_{104} \oplus \overline{S_{66}} \oplus S_{43}) \oplus (S_{273} \oplus \overline{S_{239}} \oplus \overline{S_{236}})\overline{S_{154}}$ $\oplus(S_{273} \oplus \overline{S_{239}} \oplus \overline{S_{236}})\overline{S_{109}} \oplus \overline{S_{109}}$	S_{278}
34	$S_{146} = z_{39} \oplus S_{51} \oplus \overline{S_{193}} \oplus S_{150} \oplus S_{274}(S_{100} \oplus \overline{S_{62}} \oplus S_{39})$ $\oplus S_{274}(\overline{S_{232}} \oplus \overline{S_{199}} \oplus \overline{S_{193}}) \oplus (\overline{S_{232}} \oplus \overline{S_{199}} \oplus \overline{S_{193}})$ $(S_{100} \oplus \overline{S_{62}} \oplus S_{39}) \oplus (\underline{S_{269}} \oplus \overline{S_{235}} \oplus \overline{S_{232}})S_{150}$ $\oplus(\underline{S_{269}} \oplus \overline{S_{235}} \oplus \overline{S_{232}})S_{105} \oplus S_{105}$	S_{274}, S_{105}

Table 6 (continued)

Steps	Equations used for recovery	Guessed bits
35	$S_{142} = z_{35} \oplus S_{47} \oplus \underline{S_{189}} \oplus S_{146} \oplus \underline{S_{270}}(S_{96} \oplus S_{58} \oplus S_{35})$ $\oplus\underline{S_{270}}(S_{228} \oplus \overline{S_{195}} \oplus \underline{S_{189}}) \oplus (S_{228} \oplus \overline{S_{195}} \oplus \underline{S_{189}})$ $(S_{96} \oplus S_{58} \oplus S_{35}) \oplus (S_{265} \oplus \overline{S_{231}} \oplus S_{228})S_{146}$ $\oplus(S_{265} \oplus \overline{S_{231}} \oplus S_{228})S_{101} \oplus S_{101}$	–
36	$S_{138} = z_{31} \oplus S_{43} \oplus S_{185} \oplus S_{142} \oplus S_{266}(S_{92} \oplus S_{54} \oplus S_{31})$ $\oplus S_{266}(S_{224} \oplus \underline{S_{191}} \oplus S_{185}) \oplus (S_{224} \oplus \underline{S_{191}} \oplus S_{185})$ $(S_{92} \oplus S_{54} \oplus S_{31}) \oplus (S_{261} \oplus S_{227} \oplus S_{224})S_{142}$ $\oplus(S_{261} \oplus S_{227} \oplus S_{224})S_{97} \oplus S_{97}$	–
37	$S_{134} = z_{27} \oplus S_{39} \oplus S_{181} \oplus S_{138} \oplus S_{262}(S_{88} \oplus S_{50} \oplus S_{27})$ $\oplus S_{262}(S_{220} \oplus \underline{S_{187}} \oplus S_{181}) \oplus (S_{220} \oplus \underline{S_{187}} \oplus S_{181})$ $(S_{88} \oplus S_{50} \oplus S_{27}) \oplus (S_{257} \oplus S_{223} \oplus S_{220})S_{138}$ $\oplus(S_{257} \oplus S_{223} \oplus S_{220})S_{93} \oplus S_{93}$	–
38	$S_{130} = z_{23} \oplus S_{35} \oplus S_{177} \oplus S_{134} \oplus S_{258}(S_{84} \oplus S_{46} \oplus S_{23})$ $\oplus S_{258}(S_{216} \oplus S_{183} \oplus S_{177}) \oplus (S_{216} \oplus S_{183} \oplus S_{177})$ $(S_{84} \oplus S_{46} \oplus S_{23}) \oplus (S_{253} \oplus S_{219} \oplus S_{216})S_{134}$ $\oplus(S_{253} \oplus S_{219} \oplus S_{216})S_{89} \oplus S_{89}$	–
39	$S_{126} = z_{19} \oplus S_{31} \oplus S_{173} \oplus S_{130} \oplus S_{254}(S_{80} \oplus S_{42} \oplus S_{19})$ $\oplus S_{254}(S_{212} \oplus S_{179} \oplus S_{173}) \oplus (S_{212} \oplus S_{179} \oplus S_{173})$ $(S_{80} \oplus S_{42} \oplus S_{19}) \oplus (S_{249} \oplus S_{215} \oplus S_{212})S_{130}$ $\oplus(S_{249} \oplus S_{215} \oplus S_{212})S_{85} \oplus S_{85}$	S_{19}
40	$S_{122} = z_{15} \oplus S_{27} \oplus S_{169} \oplus S_{126} \oplus S_{250}(S_{76} \oplus S_{38} \oplus S_{15})$ $\oplus S_{250}(S_{208} \oplus S_{175} \oplus S_{169}) \oplus (S_{208} \oplus S_{175} \oplus S_{169})$ $(S_{76} \oplus S_{38} \oplus S_{15}) \oplus (S_{245} \oplus S_{211} \oplus S_{208})S_{126}$ $\oplus(S_{245} \oplus S_{211} \oplus S_{208})S_{81} \oplus S_{81}$	S_{15}
41	$S_{118} = z_{11} \oplus S_{23} \oplus S_{165} \oplus S_{122} \oplus S_{246}(S_{72} \oplus S_{34} \oplus S_{11})$ $\oplus S_{246}(S_{204} \oplus S_{171} \oplus S_{165}) \oplus (S_{204} \oplus S_{171} \oplus S_{165})$ $(S_{72} \oplus S_{34} \oplus S_{11}) \oplus (S_{241} \oplus S_{207} \oplus S_{204})S_{122}$ $\oplus(S_{241} \oplus S_{207} \oplus S_{204})S_{77} \oplus S_{77}$	S_{11}
42	$S_{114} = z_{7} \oplus S_{19} \oplus S_{161} \oplus S_{118} \oplus S_{242}(S_{68} \oplus S_{30} \oplus S_{7})$ $\oplus S_{242}(S_{200} \oplus S_{167} \oplus S_{161}) \oplus (S_{200} \oplus S_{167} \oplus S_{161})$ $(S_{68} \oplus S_{30} \oplus S_{7}) \oplus (S_{237} \oplus S_{203} \oplus S_{200})S_{118}$ $\oplus(S_{237} \oplus S_{203} \oplus S_{200})S_{73} \oplus S_{73}$	–
43	$S_{110} = z_{3} \oplus S_{15} \oplus S_{157} \oplus S_{114} \oplus S_{238}(S_{64} \oplus S_{26} \oplus S_{3})$ $\oplus S_{238}(S_{196} \oplus S_{163} \oplus S_{157}) \oplus (S_{196} \oplus S_{163} \oplus S_{157})$ $(S_{64} \oplus S_{26} \oplus S_{3}) \oplus (S_{233} \oplus S_{199} \oplus S_{196})S_{114}$ $\oplus(S_{233} \oplus S_{199} \oplus S_{196})S_{69} \oplus S_{69}$	–
44	$S_{56} = z_{44} \oplus S_{151} \oplus \overline{S_{198}} \oplus \overline{S_{155}} \oplus S_{279}(S_{105} \oplus \overline{S_{67}} \oplus S_{44})$ $\oplus S_{279}(\overline{S_{237}} \oplus \overline{S_{204}} \oplus \overline{S_{198}}) \oplus (\overline{S_{237}} \oplus \overline{S_{204}} \oplus \overline{S_{198}})$ $(S_{105} \oplus \overline{S_{67}} \oplus S_{44}) \oplus (S_{274} \oplus \overline{S_{240}} \oplus \overline{S_{237}})\overline{S_{155}}$ $\oplus(S_{274} \oplus \overline{S_{240}} \oplus \overline{S_{237}})\overline{S_{110}} \oplus \overline{S_{110}}$	S_{279}
45	$S_{57} = z_{45} \oplus S_{152} \oplus \overline{S_{199}} \oplus \overline{S_{156}} \oplus S_{280}(S_{106} \oplus \overline{S_{68}} \oplus S_{45})$ $\oplus S_{280}(\overline{S_{238}} \oplus \overline{S_{205}} \oplus \overline{S_{199}}) \oplus (\overline{S_{238}} \oplus \overline{S_{205}} \oplus \overline{S_{199}})$ $(S_{106} \oplus \overline{S_{68}} \oplus S_{45}) \oplus (S_{275} \oplus \overline{S_{241}} \oplus \overline{S_{238}})\overline{S_{156}}$ $\oplus(S_{275} \oplus \overline{S_{241}} \oplus \overline{S_{238}})\overline{S_{111}} \oplus \overline{S_{111}}$	S_{280}
46	$S_{58} = z_{46} \oplus S_{153} \oplus \overline{S_{200}} \oplus \overline{S_{157}} \oplus S_{281}(\overline{S_{107}} \oplus \overline{S_{69}} \oplus S_{46})$ $\oplus S_{281}(\overline{S_{239}} \oplus \overline{S_{206}} \oplus \overline{S_{200}}) \oplus (\overline{S_{239}} \oplus \overline{S_{206}} \oplus \overline{S_{200}})$ $(\overline{S_{107}} \oplus \overline{S_{69}} \oplus S_{46}) \oplus (S_{276} \oplus \overline{S_{242}} \oplus \overline{S_{239}})\overline{S_{157}}$ $\oplus(S_{276} \oplus \overline{S_{242}} \oplus \overline{S_{239}})\overline{S_{112}} \oplus \overline{S_{112}}$	S_{281}

Table 6 (continued)

Steps	Equations used for recovery	Guessed bits
47	$S_{59} = z_{47} \oplus \overline{S_{154}} \oplus \overline{S_{201}} \oplus \overline{S_{158}} \oplus S_{282}(\overline{S_{108}} \oplus \overline{S_{70}} \oplus S_{47})$ $\oplus S_{282}(\overline{S_{240}} \oplus \overline{S_{207}} \oplus \overline{S_{201}}) \oplus (\overline{S_{240}} \oplus \overline{S_{207}} \oplus \overline{S_{201}})$ $(\overline{S_{108}} \oplus \overline{S_{70}} \oplus S_{47}) \oplus (S_{277} \oplus \overline{S_{243}} \oplus \overline{S_{240}})\overline{S_{158}}$ $\oplus(S_{277} \oplus \overline{S_{243}} \oplus \overline{S_{240}})\overline{S_{113}} \oplus \overline{S_{113}}$	S_{282}
48	$S_{60} = z_{48} \oplus \overline{S_{155}} \oplus \overline{S_{202}} \oplus \overline{S_{159}} \oplus S_{283}(\overline{S_{109}} \oplus \overline{S_{71}} \oplus S_{48})$ $\oplus S_{283}(\overline{S_{241}} \oplus \overline{S_{208}} \oplus \overline{S_{202}}) \oplus (\overline{S_{241}} \oplus \overline{S_{208}} \oplus \overline{S_{202}})$ $(\overline{S_{109}} \oplus \overline{S_{71}} \oplus S_{48}) \oplus (S_{278} \oplus \overline{S_{244}} \oplus \overline{S_{241}})\overline{S_{159}}$ $\oplus(S_{278} \oplus \overline{S_{244}} \oplus \overline{S_{241}})\overline{S_{114}} \oplus \overline{S_{114}}$	S_{283}

References

1. Biryukov A, Shamir A, Wagner D Real time cryptanalysis of A5/1 on a PC. FSE 2000, pp. 1–18, LNCS 1978, 2000. Available at: https://link.springer.com/chapter/10.1007/3-540-44706-7_1
2. Biryukov A, Shamir A Cryptanalytic time/memory/data tradeoffs for stream ciphers. Asiacrypt 2000, pp. 1–13, LNCS 1976, 2000. Available at: https://link.springer.com/chapter/10.1007/3-540-44448-3_1
3. Bjrstad TE Cryptanalysis of grain using time/memory/data tradeoffs. Estream Phase 3 (2013). Available at: www.ii.uib.no/tor/pdf/grain.pdf
4. Competition CAESAR, Hosted at: http://competitions.cr.yp.to/caesar.html
5. Hamann M, Krause M, Meier W LIZARD - A lightweight stream cipher for power-constrained devices. FSE 2017. Available at: http://tosc.iacr.org/index.php/ToSC/article/view/584
6. Hell M, Johansson T, Meier W (2007) Grain: a stream cipher for constrained environments. Int J Wirel Mob Comput 2(1):86–93. Available at: https://dl.acm.org/citation.cfm?id=1358401
7. Jiao L, Zhang B, Wang M Two generic methods of analyzing stream ciphers. ISC 2015, Lecture Notes in Computer Science, pp. 379–396, 2015. Available at: https://dl.acm.org/citation.cfm?id=2966308
8. Maitra S, Sinha N, Siddhanti A, Anand R, Gangopadhyay S (2018) A TMDTO attack against lizard. IEEE Trans Comput 67(5):733–739. Available at: https://ieeexplore.ieee.org/abstract/document/8107499/
9. Mihaljević MJ, Gangopadhyay S, Paul G, Imai H (2012) Internal state recovery of Grain-v1 employing normality order of the filter function. IET Inf Secur 6(2):55–64. Available at: ieeexplore.ieee.org/document/6230812/
10. 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)
11. Sarkar S, Banik S, Maitra S (2015) Differential Fault Attack against Grain family with very few faults and minimal assumptions. IEEE Trans Comput 64(6):1647–1657. Available at: https://ieeexplore.ieee.org/document/6857997/
12. Siddhanti AA, Maitra S, Sinha N Certain Observations on ACORN v3 and the Implications to TMDTO Attacks. International Conference on Security, Privacy, and Applied Cryptography Engineering, pp. 264-280, LNCS 10662, Springer. Available at: https://link.springer.com/chapter/10.1007/978-3-319-71501-8_15
13. Wu H ACORN: A Lightweight Authenticated Cipher (v3). Available at: https://competitions.cr.yp.to/round3/acornv3.pdf