Distinguishing attacks on SOBER-t16 and t32

Patrik Ekdahl and Thomas Johansson

Dept. of Information Technology, Lund University,
P.O. Box 118, 221 00 Lund, Sweden.
Email: {Patrik,Thomas}@it.lth.se

Abstract. Two ways of mounting distinguishing attacks on two similar stream ciphers, SOBER-t16 and SOBER-t32, are proposed. It results in distinguishing attacks faster than exhaustive key search on full SOBER-t16 and on SOBER-t32 without stuttering.

1 Introduction

In the design of symmetric ciphers, security and performance are of outmost importance. For example, in the recent AES process we have seen a number of block ciphers competing in security and performance.

When choosing a symmetric encryption algorithm, the first choice is whether to choose a block cipher or a stream cipher. Most known block ciphers offer a sufficient security and a reasonably good performance. But a block cipher must usually be used in a "stream cipher" mode, which suggests that using a pure stream cipher primitive might be beneficial.

Modern stream ciphers will indeed offer an improved performance compared with block ciphers (typically a factor 4-5 if measured in speed). However, the security of stream ciphers is not as well understood as for block ciphers. Most proposed stream ciphers such as (alleged) RC4, A5/1, have security weaknesses [7, 1].

In the recent call for primitives in the NESSIE project, two similar stream ciphers were submitted from Qualcomm Australia, called SOBER-t16 and SOBER-t32, respectively. These are two shift register based stream ciphers developed from previous versions of stream ciphers under the name of SOBER. There has been no known attacks better than exhaustive key search on these two stream ciphers, which means that they have offered full security. By full security we roughly mean that there is no attack that is better than an exhaustive key search attack. It should be noted that not many proposed stream ciphers offer full security.

A stream cipher consists of a keyed generator, producing a pseudo-random sequence that is added to the plaintext. In cryptanalysis, we consider the pseudo-random sequence to be known (known plaintext attack) and try to either recover the key, called a *key recovery attack*, or we try to distinguish the pseudo-random sequence from a truly random sequence, called a *distinguishing attack*.

The SOBER-t16 and SOBER-t32 generators can roughly be described as being nonlinear filter generators with an additional "stuttering" step before producing the output. Because of the stuttering step, the output will be irregularly produced. It is known that because of this irregularity, one can use a power analysis attack or a timing attack to recover the input to the stuttering step [9]. However, the authors claim that the generator is secure even without the stuttering step [9].

In this paper we consider several new ways of mounting distinguishing attacks on SOBER-t16 and SOBER-t32. The attacks are based on combining linear approximations of the nonlinear filter with the linear recurrence, defined through the feedback polynomial. Linear approximations have previously been used in e.g. the BAA attack on stream ciphers [3] and in linear cryptanalysis on block ciphers [8]. In our case we mainly derive the distribution of the noise introduced through the linear approximations by simulations. We consider attacks on the ciphers both including and excluding the stuttering step.

The final results are as follows. For SOBER-t16 without stuttering, which uses a 128 bit key, the output can be distinguished from a random sequence using at most 2^{92} output words and the same complexity. For the full SOBER-t16 with stuttering, we need at most 2^{111} output words and the same complexity. For SOBER-t32, without stuttering, which uses a 256 bit key, the output can be distinguished from a random sequence using at most 2^{87} output words and the same complexity. For the full SOBER-t32 with stuttering we could not find an exact complexity expression, but the proposed methods indicate a strong attack also here.

We should also mention that the proposed methods are applicable to the stream cipher SNOW [4], another candidate in the NESSIE project. The strength of such an attack on SNOW is considered in a subsequent paper.

The paper is organized as follows. In Section 2 we shortly describe the stream ciphers SOBER-t16 and SOBER-t32. Then we start by explaining the attack on SOBER-t16 without stuttering in Section 3. This is generalized to an attack on the full SOBER-t16 in Section 4. In Section 5 we describe a simple attack on SOBER-t32 without stuttering. In Section 6 we then elaborate on different possibilities for mounting an attack on the full SOBER-t32. Finally, we give some concluding remarks.

2 A brief description of SOBER-t16 and t32

Both SOBER-t16 and SOBER-t32 are word oriented stream ciphers. The word size is 16 bits for t16 and 32 bits for t32. The structure of t16 and t32 are very similar and we will here describe them as one cipher. The specific parameters for both t16 and t32 will be given alongside. To simplify the description of the common parts of t16 and t32, we will use the notation W to denote the word size. Thus, W is either 16 or 32 bits, depending on which cipher we are looking at. The operations in the ciphers include both addition in an extension field $\mathbb{F}_{2^W}$ and addition modulo 2^W, and we will denote the field addition by $\oplus$ (also called

XOR) and the ring addition by $\boxplus$. In case there is no risk of confusion we will simply use the addition symbol $+$.

There are three main building blocks for the SOBER stream ciphers. The first is a word oriented linear feedback shift register (LFSR) which produces a LFSR sequence denoted $\{s_t, t \geq 0\}$. Secondly, a non-linear filter (NLF) takes some of these symbols as inputs and produces a new sequence $\{v_t, t \geq 0\}$. Finally, there is a so called stuttering unit. The stuttering unit takes $\{v_t, t \geq 0\}$ as input and produces an irregular output $\{z_n, n \geq 0\}$. The overall structure is pictured in Figure 1.

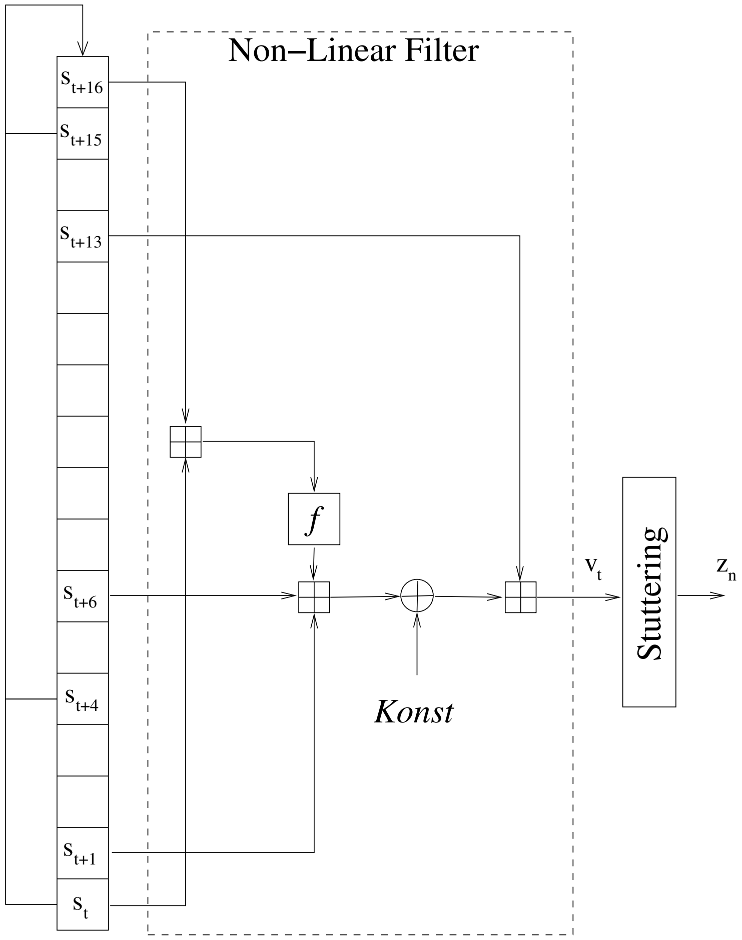

Fig. 1. Overall structure of SOBER-t16 and SOBER-t32.

2.1 The LFSR

The LFSR is a length 17 shift register, where each register element contains one word. Each word is considered as an element in an extension field ($\mathbb{F}_{2^W}$). The contents of the LFSR at time t is called the *state* of the LFSR at time t and will be denoted by a vector $\bar{S}_t = (s_t, s_{t+1}, \ldots, s_{t+16})$. The next state of the LFSR is obtained by shifting the previous state one step, and calculating a new word s_{t+17}. The new word is calculated as a certain linear combination of the contents of the previous state. Thus the next state will be $\bar{S}_{t+1} = (s_{t+1}, s_{t+2}, \ldots, s_{t+17})$ where

$$s_{t+17} = \sum_{i=0}^{16} c_i s_{t+i}, \qquad (1)$$

for some known constants $c_i \in \mathbb{F}_{2^W}, i = 0, 1 \ldots, 16$. The arithmetics in Eq. (1) is performed in the extension field $\mathbb{F}_{2^W}$. Equation (1) is called the linear recurrence equation. The specific extension fields and recurrence equations for t16 and t32 are summarized below:

SOBER-t16
Defining polynomial for $\mathbb{F}_{2^{16}}$: $x^{16} + x^{14} + x^7 + x^6 + x^4 + x^2 + x + 1$
Linear recurrence equation: $s_{t+17} \oplus \alpha s_{t+15} \oplus s_{t+4} \oplus \beta s_t = 0$
where $\alpha = 0xE382$ and $\beta = 0x67C3$.
SOBER-t32
Defining polynomial for $\mathbb{F}_{2^{32}}$: $x^{32} + (x^{24} + x^{16} + x^8 + 1)(x^6 + x^5 + x^2 + 1)$
Linear recurrence equation: $s_{t+17} \oplus s_{t+15} \oplus s_{t+4} \oplus \alpha s_t = 0$
where $\alpha = 0xC2DB2AA3$.

The field elements α and β have been given in a hexadecimal form, corresponding to a polynomial basis. See [5, 6] for more details.

2.2 The NLF function

At time t, the NLF function takes five words from the LFSR state, ($s_t, s_{t+1}, s_{t+6}, s_{t+13}, s_{t+16}$) and one constant value (*Konst*) as input, and produces through a nonlinear function an output word, denoted by v_t. The value of *Konst*$\in \mathbb{F}_{2^W}$ is determined during the initialization phase of the LFSR and is kept constant throughout the entire session. The operations involved in the NLF function are XOR (denoted $\oplus$), addition modulo 2^W (denoted $\boxplus$) and application of a substitution box (denoted SBOX).

The output of the NLF function, v_t, at time t, can be written as:

$$v_t = ((s_{t+1} \boxplus s_{t+6} \boxplus f(s_t \boxplus s_{t+16})) \oplus Konst) \boxplus s_{t+13}, \qquad (2)$$

where $f(x)$ is a function, different for SOBER-t16 and SOBER-t32, which in both cases involves an SBOX application. The interior design of the function f is pictured in Fig. 2. First the input is partitioned into a high part containing the 8 most significant bits, and a low part containing the remaining bits. The high part addresses an SBOX with W bits of output. The 8 most significant bits

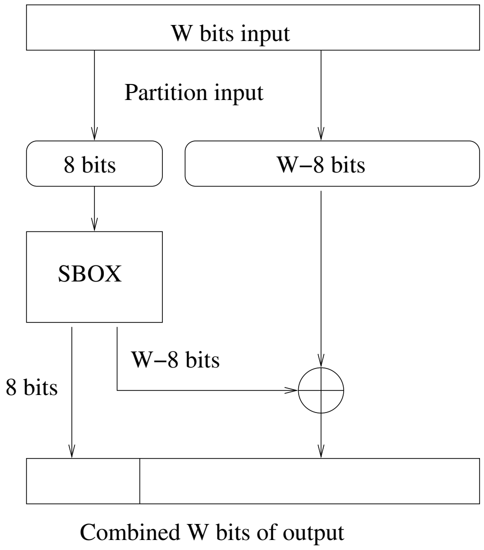

Fig. 2. The structure of the f-function in SOBER-t16 and SOBER-t32.

are directly taken as the f-function output, whereas the least significant part of the SBOX output is first XORed to the low part from the input, see Fig. 2.

2.3 Stuttering

Before producing the running key, the output from the NLF is passed through a stuttering unit. The stuttering decimates the NLF output, thus making e.g. a correlation attack harder. The first output from the NLF, v_0 is taken as the first *stutter control word* (SCW). The SCW is divided into pairs of bits (called dibits). Starting with the least significant dibit, the stuttering is determined from the value of these dibits. Actions are taken according to the value of the dibit, as listed in Table 1. The constant C has value 0x6996 for t16 and 0x6996C53A for t32, and $\sim C$ denotes the bitwise complement of C.

When all dibits in the SCW have been used, the LFSR is clocked once and a new SCW is read from the output of the NLF. This word determines the next 8 or 16 actions, depending on whether we are looking at SOBER-t16 or SOBER-t32. The resulting stream from the stuttering unit, denoted z_n, is the running key.

This concludes the brief description of SOBER-t16 and SOBER-t32. For a more detailed description, especially regarding the key initialization, we refer to [5] and [6].

Dibit	Action
00	1. Clock the LFSR, but do not output anything.
01	1. Clock the LFSR. 2. Set the value of the next key stream word to be the XOR between C and the NLF output. 3. Clock the LFSR again, but do not output anything.
10	1. Clock the LFSR, but do not output anything. 2. Clock the LFSR. 3. Set the value of the next key stream word to be the value of the NLF output.
11	1. Clock the LFSR. 2. Set the value of the next key stream word to be the XOR between $\sim C$ and the NLF output.

Table 1. The possible actions taken in the stuttering unit depending on the value of the dibit.

3 A distinguishing attack on SOBER-t16 without stuttering

We start by analyzing a version of SOBER-t16 where the stuttering unit has been removed. In this scenario each NLF output word is taken as a running key word. Thus we have $z_t = v_t$ for all $t \geq 0$. We also assume that we are given N words of the output key stream, so we have access to $z_0, z_1, \ldots, z_{N-1}$.

The first step in our attack is to approximate the NLF-function with a linear function and then argue that the noise introduced by the approximation possesses a nonuniform distribution. Recall the expression for the NLF output:

$$v_t = ((s_{t+1} \boxplus s_{t+6} \boxplus f(s_t \boxplus s_{t+16})) \oplus Konst) \boxplus s_{t+13}. \tag{3}$$

We now approximate this function with a linear function by replacing $\boxplus$ with $\oplus$, and f by the identity map. When we do this approximation we introduce a noise (an error), which we denoted by w_t. We also move the value of $Konst$ into the noise w_t. I.e., we write

$$v_t = s_{t+1} \oplus s_{t+6} \oplus s_t \oplus s_{t+16} \oplus s_{t+13} \oplus w_t, \tag{4}$$

where $w_t, t \geq 0$ denotes random variables that represent the error we get in the approximation at each time t. The distribution of w_t will be dependent on the value $Konst$. Also, w_t will have the same distribution for all t, and this distribution is denoted F.

Introduce the notation $\Omega_t = s_t \oplus s_{t+1} \oplus s_{t+6} \oplus s_{t+13} \oplus s_{t+16}$ for the XOR of the words from the LFSR that are inputs to the NLF function. Then we can write the output word v_t as

$$v_t = \Omega_t \oplus w_t. \tag{5}$$

By looking at the running key at time $t, t+4, t+15$ and $t+17$ in combination with Eq. (5) we can express $z_{t+17} \oplus \alpha z_{t+15} \oplus z_{t+4} \oplus \beta z_t$ in the following way,

$$\begin{aligned} & z_{t+17} \oplus \alpha z_{t+15} \oplus z_{t+4} \oplus \beta z_t = \\ & v_{t+17} \oplus \alpha v_{t+15} \oplus v_{t+4} \oplus \beta v_t = (\Omega_{t+17} \oplus w_{t+17}) \oplus \alpha(\Omega_{t+15} \oplus w_{t+15}) \oplus \\ & \qquad (\Omega_{t+4} \oplus w_{t+4}) \oplus \beta(\Omega_t \oplus w_t). \end{aligned} \tag{6}$$

Rearranging the terms of the right hand side of (6) we get

$$\begin{aligned} z_{t+17} \oplus \alpha z_{t+15} \oplus z_{t+4} \oplus \beta z_t = {} & \Omega_{t+17} \oplus \alpha \Omega_{t+15} \oplus \Omega_{t+4} \oplus \beta \Omega_t \oplus \\ & w_{t+17} \oplus \alpha w_{t+15} \oplus w_{t+4} \oplus \beta w_t. \end{aligned} \tag{7}$$

Recalling the linear recurrence relation for SOBER-t16,

$$s_{t+17} \oplus \alpha s_{t+15} \oplus s_{t+4} \oplus \beta s_t = 0, \tag{8}$$

we see that $\Omega_{t+17} \oplus \alpha \Omega_{t+15} \oplus \Omega_{t+4} \oplus \beta \Omega_t = 0$ and we can reduce (6) to

$$z_{t+17} \oplus \alpha z_{t+15} \oplus z_{t+4} \oplus \beta z_t = w_{t+17} \oplus \alpha w_{t+15} \oplus w_{t+4} \oplus \beta w_t, \tag{9}$$

where the multiplications with α and β are in the extension field $\mathbb{F}_{2^W}$.

3.1 Estimating the distribution of w_t

The noise $w_t, t \geq 0$ are random variables taken from $\mathbb{F}_{2^{16}}$ with a nonuniform but unknown distribution F. Let us write the distribution F in the form

$$F = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{2^{16}-1} \end{bmatrix}$$

where $P(w_t = x) = f_x$. We can not hope to find a closed expression for the distribution F, since it is computationally too complex to derive. However, we can run the cipher and estimate the distribution F.
In the simulations, we measure the frequency of different values for the noise w_t, calculated as

$$w_t = (((s_{t+1} \boxplus s_{t+6} \boxplus f(s_t \boxplus s_{t+16})) \oplus Konst) \boxplus s_{t+13}) \oplus \Omega_t. \tag{10}$$

Assume that we sample 2^ν values of w_t according to (10), and denote the measured frequencies by the vector $\hat{F} = (\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_{2^{16}-1})$. $\hat{F}$ is an estimation of F and we can write $F = \hat{F} + \bar{E}$, where $\bar{E}$ is a vector representing the error in the estimation. Focusing on one single component of $\bar{E}$, it will be approximately Gaussian distributed with zero mean and a standard deviation of $2^{-(\nu/2+8)}$. Simulations show that F is quite nonuniform. For example, in simulation with $Konst = 0$ and $\nu = 38$, the maximum value is $2^{-16} + 2^{-17.6}$. The error in this estimation is of order 2^{-28}. The distribution F has been tabulated for a number of different values of *Konst*.

3.2 Calculating the full noise distribution

Let us define

$$W_t = w_{t+17} \oplus \alpha w_{t+15} \oplus w_{t+4} \oplus \beta w_t,$$

i.e, $W_t, t \geq 0$ are the random variables corresponding to the full noise that we can sample from the running key. Looking at Eq. (9) we see that we must combine four F distributions (as above) to get the overall noise distribution, denoted $P(W)$. Since the samples are taken at different positions in time, we assume $w_t, t \geq 0$ to be independent variables.

The distribution $H = [h_i]$ of the XOR of two random variables with distribution $F = [f_i]$ and $G = [g_i]$ respectively, is obtained by

$$h_l = \sum_{i \oplus j = l} f_i g_j. \tag{11}$$

The distribution of αw_t is simply a permutation of the distribution F.

It can be shown that when combining distributions as done in (11), we sustain significance in the resulting distribution. So by estimating the F distribution by simulation and then combining the probabilities according to (11), we can estimate the distribution $P(W)$, of the right hand side of (9) for different values of $Konst$.

To be able to distinguish the full noise distribution, $P(W)$, from the uniform distribution we need have some N different keystream symbols. The theory of hypothesis testing [2] gives us a bound on N.

Let $Z_t = z_{t+17} \oplus \alpha z_{t+15} \oplus z_{t+4} \oplus \beta z_t$. For short, the optimal test for distinguishing between the two possible distributions ($P(W)$ and uniform distribution) is according to the Neyman-Pearson lemma to check if the likelihood ratio $\sum_{t=0}^{N} \log[P(W_t = Z_t)/2^{-16}]$ is smaller or larger than 0.

The probability that we make an incorrect decision, denoted P_e, when trying to distinguish between two distributions P_1 and P_2, given N samples from one of the distributions is bounded by

$$P_e \leq 2^{-N \cdot C(P_1, P_2)}, \tag{12}$$

where $C(P_1, P_2)$ is the Chernoff information between the two distributions. The Chernoff information is defined as

$$C(P_1, P_2) = - \min_{0 \leq \lambda \leq 1} \log_2 (\sum_x P_1^{\lambda}(x) P_2^{1-\lambda}(x)). \tag{13}$$

We get a lower bound on $C(P_1, P_2)$ by using e.g. $\lambda = 1/2$. We fix a probability of error of $P_e = 2^{-32}$. Then we need to choose $N \geq 32 C(P_1, P_2)^{-1}$.

3.3 Summarizing the results

The distribution $P(W)$ has been determined through simulation as previously described. The analysis in this section summarizes to the following attack:

For $t = 1 \ldots N$ do

1. Calculate $Z_t = z_{t+17} \oplus \alpha z_{t+15} \oplus z_{t+4} \oplus \beta z_t$.
2. Let $\hat{f}_{Z_t} = \hat{f}_{Z_t} + 1$.

end for.

Calculate $I = \sum_{x \in \mathbb{F}_{2^{16}}} \hat{f}_x \log_2 \left[\frac{P(W=x)}{2^{-16}} \right]$.

If $I > 0$ then output **SOBER** otherwise output **random**

We have calculated the combined $P(W)$ distribution using $2^{38}, (\nu = 38)$ outputs to generate the F distribution. Note that since F (and thus $P(W)$) is dependent on the unknown value of $Konst$, we actually need to determine the $P(W)$ distribution for all 2^{16} possible values of $Konst$.

The resulting Chernoff information between $P(W)$ and the uniform distribution, have been derived for 50 random values of $Konst$. They were all between 2^{-84} and 2^{-87}. We assume that calculating the Chernoff information for other values of $Konst$ will give similar results. In the worst case, we need at least $N = 32 \cdot 2^{87} = 2^{92}$ words from the running key to be able to distinguish a SOBER-t16 output sequence without stuttering from a uniform distribution with a probability of error $P_e = 2^{-32}$. The computational complexity of the attack is roughly 2^{92}.

Finally, the Neyman-Pearson test must be performed for each of the 2^{16} possible values of $Konst$. Still, the probability of error is smaller than 2^{-16}, which is small enough. Note that this step does not change the overall complexity.

4 A distinguishing attack on SOBER-t16 with stuttering

When the stuttering unit is present, not every NLF output, v_t, is used to produce a keystream symbol. Recalling the functionality of the stuttering unit, we see that each v_t can be either discarded, used as a new SCW, or used (possible XORed with a constant) as a keystream symbol z_n. To be able to use the results from Section 3, we must have access to the NLF output quadruple $(v_t, v_{t+4}, v_{t+15}, v_{t+17})$.

Assume that we look at one output symbol $z_n = \mathcal{C}_0 \oplus v_t$, where $\mathcal{C}_0 \in \{0, C, \sim C\}$ is the constant that is XORed to v_t in the stuttering unit to form z_n. Simulations show that the most probable position in the key stream for v_{t+4} to appear in is z_{n+2}. Similar simulations to determine the most probable position for v_{t+15} and v_{t+17} give the following results

$$\begin{aligned}
P(\mathcal{C}_1 \oplus v_{t+4} \to z_{n+2} | \mathcal{C}_0 \oplus v_t \to z_n) &= 0.31, \\
P(\mathcal{C}_2 \oplus v_{t+15} \to z_{n+7} | \mathcal{C}_1 \oplus v_{t+4} \to z_{n+2}) &= 0.19, \\
P(\mathcal{C}_3 \oplus v_{t+17} \to z_{n+8} | \mathcal{C}_2 \oplus v_{t+15} \to z_{n+7}) &= 0.40.
\end{aligned}$$

Having established the most probable positions in the keystream for (v_{t+4}, v_{t+15}, v_{t+17}), given an output $z_n = \mathcal{C}_0 \oplus v_t$, we are still faced with the problem of which constants $\mathcal{C}_i, i = 0, 1, 2, 3$ each NLF output is XORed with.

Denote by $\mathcal{E}$ the event that, given $\mathcal{C}_0 \oplus v_t \to z_n$, we have $\mathcal{C}_1 \oplus v_{t+4} \to z_{n+2}$, $\mathcal{C}_2 \oplus v_{t+15} \to z_{n+7}$, $\mathcal{C}_3 \oplus v_{t+17} \to z_{n+8}$. The probability of event $\mathcal{E}$, denoted p_0, is $p_0 \approx 2^{-5.5}$.

By looking at Table 1 we note that certain combinations of $(\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3)$ can not occur under the assumption $\mathcal{E}$. In general, the distribution is nonuniform and, for example, the five combinations, $(\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3) =$

$$\begin{array}{l} (0,0,0,0) \\ (C,C,C,\sim C) \\ (C,C,\sim C,0) \\ (C,\sim C,0,0) \\ (\sim C,0,0,0) \end{array}$$

are more likely to occur than others.

From Eq. (6) and (9) in Section 3, we know that

$$v_{t+17} \oplus \alpha v_{t+15} \oplus v_{t+4} \oplus \beta v_t = w_{t+17} \oplus \alpha w_{t+15} \oplus w_{t+4} \oplus \beta w_t. \tag{14}$$

Given event $\mathcal{E}$, we can write

$$z_{n+8} \oplus \alpha z_{n+7} \oplus z_{n+2} \oplus \beta z_n = W_t \oplus \mathcal{C}_3 \oplus \alpha \mathcal{C}_2 \oplus \mathcal{C}_1 \oplus \beta \mathcal{C}_0, \tag{15}$$

where $W_t = w_{t+17} \oplus \alpha w_{t+15} \oplus w_{t+4} \oplus \beta w_t$ and has known distribution $P(W)$.

Again, we derive the distribution of the right hand side of (15) and denote this distribution by $P(W')$, assuming $W'_t = W_t \oplus \mathcal{C}_3 \oplus \alpha \mathcal{C}_2 \oplus \mathcal{C}_1 \oplus \beta \mathcal{C}_0$. The Chernoff information between $P(W')$ and the uniform distribution, is calculated to be $C(P(W'), P_U) \approx 2^{-95}$, where P_U is the uniform distribution.

Sampling the keystream output sequence at $(z_n, z_{n+2}, z_{n+7}, z_{n+8})$ will give us a sample of the noise from the distribution $P(W')$ with probability $p_0 = 2^{-5.5}$. With probability $1-p_0$ the assumption was wrong and it is reasonable to assume that we then get a uniform distribution. Write the distribution $P(W')$ as a vector

$$P(W') = \begin{bmatrix} 2^{-16} + \xi_0 \\ 2^{-16} + \xi_1 \\ \vdots \\ 2^{-16} + \xi_{2^{16}-1} \end{bmatrix}, \tag{16}$$

where each element $2^{-16} + \xi_x$ represents $P(W' = x) = 2^{-16} + \xi_x$.

Let $Y = z_{n+8} \oplus \alpha z_{n+7} \oplus z_{n+2} \oplus \beta z_n$. The distribution of Y, denoted $P(Y)$, can then be calculated to be

$$P(Y) = \begin{bmatrix} 2^{-16} + \xi_0 p_0 \\ 2^{-16} + \xi_1 p_0 \\ \vdots \\ 2^{-16} + \xi_{2^{16}-1} p_0 \end{bmatrix}. \tag{17}$$

The resulting Chernoff information between Y and the uniform distribution, is finally calculated to be $C(P(Y), P_U) \approx p_0^2 C(P(W'), P_U)$, where P_U is the uniform distribution.

From the discussion in Section 3.3, we conclude that we need at most $N = 32 \cdot p_0^{-2} 2^{95} \approx 2^{111}$ keystream symbols to be able to distinguish the output of SOBER-t16 with stuttering from a uniform source. The complexity is of the same size. We summarize the results in this section in the following attack.

> For $t = 1 \ldots N$ do
>
> 1. Calculate $Z_n = z_{n+8} \oplus \alpha z_{n+7} \oplus z_{n+2} \oplus \beta z_n$.
> 2. Let $\hat{f}_{Z_n} = \hat{f}_{Z_n} + 1$.
>
> end for.
>
> Calculate $I = \sum_{x \in \mathbb{F}_{2^{16}}} \hat{f}_x \log_2 \left[\frac{P(Y=x)}{2^{-16}} \right]$.
>
> If $I > 0$, then output **SOBER** otherwise output **random**

Again, we should note that $P(Y)$ is dependent on *Konst*, and a full attack includes testing against 2^{16} different distributions.

5 A distinguishing attack on SOBER-t32 without stuttering

The attack on SOBER-t16 was possible because we could compute the noise distribution F by simulation. From F we could derive $P(W)$.

Obtaining significance in simulation was possible because of the small word size of 16 bits. In SOBER-t32 we cannot directly use the same method to obtain a similar distribution F, due to our computational limitations. We note, however, that *if* we could simulate and find a noise distribution, then the attack on t32 would probably be strong. This is due to the fact that the linear recurrence relation in t32 has only one constant α different from one, whereas t16 has two, α and β. The multiplications by these constants tend to smooth out the distribution.

However, in this section we present another attack, based on a bitwise linear approximation through the NLF function. Using the same notation as before, we denote the XOR of the input words to the NLF at time t, by $\Omega_t = s_t \oplus s_{t+1} \oplus s_{t+6} \oplus s_{t+13} \oplus s_{t+16}$. The output from the NLF at time t, is denoted v_t. Since the stuttering unit is removed, we have $z_t = v_t$ for all $t \geq 0$. Each word is 32 bits, and we will denote a specific bit i, $0 \leq i \leq 31$, in a word x, with $x[i]$. Let k denote the value of *Konst*.

We start by considering the linear recurrence relation of t32 given by

$$s_{t+17} \oplus s_{t+15} \oplus s_{t+4} \oplus \alpha s_t = 0, \tag{18}$$

and the corresponding characteristic polynomial for the recurrence

$$x^{17} + x^{15} + x^4 + \alpha. \tag{19}$$

Repeated squaring of this polynomial will still yield a valid linear recurrence equation for the considered linear recurrence of t32. Specifically, exponentiation with 2^{32} gives

$$x^{17\cdot 2^{32}} + x^{15\cdot 2^{32}} + x^{4\cdot 2^{32}} + \alpha^{2^{32}}. \tag{20}$$

Since $\alpha \in \mathbb{F}_{2^{32}}$ we have $\alpha^{2^{32}} = \alpha$ and addition of (19) and (20) gives

$$x^{17} + x^{15} + x^4 + x^{17\cdot 2^{32}} + x^{15\cdot 2^{32}} + x^{4\cdot 2^{32}}. \tag{21}$$

Here we can divide with x^4, and the resulting linear recurrence is given by

$$s_{t+17\cdot 2^{32}-4} \oplus s_{t+15\cdot 2^{32}-4} \oplus s_{t+4\cdot 2^{32}-4} \oplus s_{t+13} \oplus s_{t+11} \oplus s_t = 0,$$

which is written

$$s_{t+\tau_5} \oplus s_{t+\tau_4} \oplus s_{t+\tau_3} \oplus s_{t+\tau_2} \oplus s_{t+\tau_1} \oplus s_t = 0, \tag{22}$$

by introducing the constants $\tau_1 = 11$, $\tau_2 = 13$, $\tau_3 = 4 \cdot 2^{32} - 4$, $\tau_4 = 15 \cdot 2^{32} - 4$ and $\tau_5 = 7 \cdot 2^{32} - 4$. Note that in Eq. (22) we have derived a linear recurrence equation that holds *for each single bit position.*

Consider the XOR between two adjacent bits, i and $i-1$, $i \geq 1$, in the running key z_t. As before, we use a linear approximation of the NLF function, $z_t = \Omega_t \oplus w_t$, where the value of *Konst* is merged into the binary random variable w_t representing the noise. We can now write

$$z_t[i] \oplus z_t[i-1] = \Omega_t[i] \oplus \Omega_t[i-1] \oplus w_t[i]. \tag{23}$$

where $w_t[i]$ denotes the noise in bit position i introduced by the linear approximation. Let $F[i]$ be the distribution of $w_t[i]$. We can estimate the distribution $F[i]$ by simulation and the result shows that the distribution is quit nonuniform for many positions $0 \leq i \leq 31$. We can write the correlation between the XOR of bit i and $i-1$ of the input and output as

$$\begin{aligned} P(z_t[i] \oplus z_t[i-1] = \Omega_t[i] \oplus \Omega_t[i-1]) = \\ P(F[i] = 0) = \frac{1}{2} + \varepsilon_i, \end{aligned} \tag{24}$$

for each bit position $0 < i \leq 31$.

The largest correlation we have found is for the XOR of bit 29 and bit 30 (i.e. $F[30]$) in the input and output words. Simulations with 2^{30} samples for 100 random values of k, indicates that the correlation in (24) for $i = 30$ is only dependent on the two corresponding bits in k, i.e. $k[30]$ and $k[29]$. We have the following result,

$$\varepsilon_{30} \approx \begin{cases} -0.0086 \text{ if } k[30] = 0 \text{ and } k[29] = 0 \\ -0.0052 \text{ if } k[30] = 1 \text{ and } k[29] = 1 \\ +0.0086 \text{ if } k[30] = 1 \text{ and } k[29] = 0 \\ +0.0052 \text{ if } k[30] = 0 \text{ and } k[29] = 1. \end{cases} \tag{25}$$

Now, given a key stream output, $z_0, z_1, \ldots, z_{N-1}$, of length N, we can use the linear recurrence relation (22) to calculate

$$\begin{aligned} z_{t+\tau_5} \oplus z_{t+\tau_4} \oplus z_{t+\tau_3} \oplus z_{t+\tau_2} \oplus z_{t+\tau_1} \oplus z_t = {} & \Omega_{t+\tau_5} \oplus w_{t+\tau_5} \oplus \Omega_{t+\tau_4} \oplus w_{t+\tau_4} \oplus \\ & \Omega_{t+\tau_3} \oplus w_{t+\tau_3} \oplus \Omega_{t+\tau_2} \oplus w_{t+\tau_2} \oplus \\ & \Omega_{t+\tau_1} \oplus w_{t+\tau_1} \oplus \Omega_t \oplus w_t \oplus \end{aligned}$$

where the sum of all the Ω_j terms will equal zero because of Eq. (22). Thus, we have

$$z_{t+\tau_5} \oplus z_{t+\tau_4} \oplus z_{t+\tau_3} \oplus z_{t+\tau_2} \oplus z_{t+\tau_1} \oplus z_t = \bigoplus_{j=0}^{5} w_{t+\tau_j}. \tag{26}$$

Introduce the notation $Z_t = z_{t+\tau_5} \oplus z_{t+\tau_4} \oplus z_{t+\tau_3} \oplus z_{t+\tau_2} \oplus z_{t+\tau_1} \oplus z_t$ for the left hand side of (26), and $W_t = \bigoplus_{j=0}^{5} w_{t+\tau_j}$ for the right hand side. We can calculate the probability that

$$\begin{aligned} P(Z_t[i] \oplus Z_t[i-1] = 0) = \\ P(W_t[i] \oplus W_t[i-1] = 0) = \frac{1}{2} + 2^5 \varepsilon_i^6, \end{aligned} \tag{27}$$

where the last equality comes from combining the six independent noise distributions of $w_t[i]$, each with probability $1/2 + \varepsilon_i$ of being zero.

Recalling the measured correlation for bits 29 XOR 30 from (25), we see that ε_{30} takes four possible values. If we want to distinguish the distribution of w from a uniform source, the worst case is the smallest value of ε_{30}. Thus, using $\varepsilon_{30} = 0.0052$ and combining the six noise distribution according to (27) we derive the final correlation probability for the six independent key stream positions as

$$p_0 = P(Z_t[i] \oplus Z_t[i-1] = 0) = \frac{1}{2} + 2^5 (0.0052)^6 \approx \frac{1}{2} + 2^{-40.5}. \tag{28}$$

5.1 Summarizing the results

To be able to distinguish this nonuniform distribution, denoted P_0, from a uniform source, denoted P_U, we again calculate the Chernoff information between the two distributions,

$$C(P_0, P_U) = -\min_{0 \le \lambda \le 1} \log_2 \sum_x P_0^{\lambda}(x) P_U^{1-\lambda}(x) \approx 2^{-81.5}. \tag{29}$$

Settling for an error probability of $P_e = 2^{-32}$ we see that we need $N = 2^{86.5}$ samples from the key stream. Each sample spans a distance of $\tau_5 = 17 \cdot 2^{32} - 4 \approx 2^{36}$ positions, so all in all we need $N + \tau_5 \le 2^{87}$ key stream output words, to distinguish an output sequence from SOBER-t32 without stuttering unit, from a uniform source. The attack presented in this section summarizes as follows.

For $t = 1 \dots N$ do

1. Calculate $Z_t = z_{t+\tau_5} \oplus z_{t+\tau_4} \oplus z_{t+\tau_3} \oplus z_{t+\tau_2} \oplus z_{t+\tau_1} \oplus z_t$.
2. Let $\hat{f} = \hat{f} + (1 - (Z_t[i] \oplus Z_t[i-1]))$.

end for.
Calculate $I = \hat{f} \log \left[\frac{\frac{1}{2}+2^{-40.5}}{1/2} \right] + (2^N - \hat{f}) \log \left[\frac{\frac{1}{2}-2^{-40.5}}{1/2} \right]$.
If $I > 0$, then output **SOBER** otherwise output **random**

6 Some remarks on SOBER-t32 with stuttering

The obvious extension of the attack in the previous section would be to guess the most probable key stream positions for $v_{t+\tau_5}, \dots, v_{t+\tau_1}$, given $z_n = v_t$. Since τ_3, τ_4, τ_5 are all in the order of 2^{32}, the probability of guessing the positions of $v_{t+\tau_5}, \dots, v_{t+\tau_3}$ in the output will be very small. However, it might be possible to get an attack using $N < 2^{256}$ words, in this way.

Another approach would be to repeat the attack on SOBER-t16 but consider only a specific subset of the bit positions of the words. Then we can simulate the distribution of the selected bit positions of w_t as well as the same bit positions of αw_t. If these distributions show a non-uniformity of similar magnitude as SOBER-t16, we can distinguish the full SOBER-t32 using about the same method as for t16.

7 Conclusions

We have derived a distinguishing attack, based on a linear approximation of the NLF function, on SOBER-t16 with and without stuttering unit. We can distinguish the output sequence from a random source using at most 2^{92} keystream words and same complexity in the case of no stuttering, and using at most 2^{111} key stream words and same complexity for full SOBER-t16. For SOBER-t32 without the stuttering unit we can, due to a fairly strong bit correlation in the NLF function, distinguish the output from a random source using 2^{87} key stream output words and same complexity.

References

1. A. Biryukov, A. Shamir, and D. Wagner. Real time crypanalysis of A5/1 on a PC. In *Proceeding of Fast Software Encryption Workshop*, LNCS vol.1978, pp 1-18, Springer-Verlag, 2001.
2. T. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley series in Telecommunication. Wiley, 1991.
3. C. (Cunsheng) Ding, G. Xiao, and W. Shan. *The stability theory of stream ciphers*, volume 561. Springer-Verlag Inc., New York, NY, USA, 1991.
4. P. Ekdahl and T. Johansson. SNOW - a new stream cipher. In *Proceedings of the First Open NESSIE Workshop*, 13-14 November 2000, Heverlee, Belgium.

5. P. Hawkes and G. Rose. Primitive specification and supporting documentation for SOBER-t16 submission to NESSIE. In *Proceedings of the First Open NESSIE Workshop*, 13-14 November 2000, Heverlee, Belgium.
6. P. Hawkes and G. Rose. Primitive specification and supporting documentation for SOBER-t32 submission to NESSIE. In *Proceedings of the First Open NESSIE Workshop*, 13-14 November 2000, Heverlee, Belgium.
7. I. Mantin and A. Shamir. Practical attack on broadcast RC4. In *Preproceedings of Fast Software Encryption*, 2001.
8. M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology — Eurocrypt '93*, volume 765, pages 386–397, Berlin, 1994. Springer-Verlag.
9. M. Schafheutle. A first report on the stream ciphers SOBER-t16 and SOBER-t32. NESSIE document NES/DOC/SAG/WP3/025/2, NESSIE, 2001.