A Key Recovery Attack on SOBER-128

Risto Hakala and Kaisa Nyberg

Helsinki University of Technology
Laboratory for Theoretical Computer Science
P.O. Box 5400, FI-02015 TKK, FINLAND
risto.m.hakala@tkk.fi; kaisa.nyberg@tkk.fi

Abstract. In this paper, we consider how an unknown constant within a state update function or output function affects biases of linear approximations. This allows us to obtain information from an unknown constant within a T-function. We use this knowledge for mounting an attack against stream cipher SOBER-128 where we gain information from the key dependent secret constant using multiple linear approximations. One to four bits of information can be obtained from $2^{113.5}$ to $2^{124.6}$ keystream words, respectively. More bits can be covered by increasing the number of linear approximations used.

Keywords: linear approximations, correlation, linear cryptanalysis, key-recovery attack, SOBER-128.

1 Introduction

Stream ciphers are important cryptographic primitives and have many advantages over block ciphers due to their speed and flexibility in practical implementations. Therefore, they are widely used in practise and many proprietary designs are known. Unfortunately, many of them have severe weaknesses and are not suitable for general applications. Traditionally, cryptographic research of stream ciphers has been restricted to specific features such as period, linear complexity and correlation attacks. Only recently more general analysis methods originally developed for block cipher analysis have been used to analyze stream ciphers. Also new and different stream cipher constructions are being presented [1], which use a wide range of building blocks. In addition to traditional linear feedback shift registers (LFSRs) and correlation immune Boolean functions, stream cipher constructions often involve S-boxes, borrowed from block ciphers, algebraic operations and modular arithmetic.

In this paper, the focus is on linear cryptanalysis method. Linear cryptanalysis makes use of approximate linear relations over nonlinear components of the cipher and has been successfully applied to stream ciphers to distinguish the output keystream from a truly random sequence [2–5]. In this paper, we show that if the output function involves a secret constant, it is possible to get information of the constant using linear cryptanalysis in a similar manner than one gets information about round keys using linear cryptanalysis on block ciphers.

Dagstuhl Seminar Proceedings 07021
Symmetric Cryptography
http://drops.dagstuhl.de/opus/volltexte/2007/1018

We will apply our technique on SOBER-128 [6], which is a keystream generator for a stream cipher proposed by P. Hawkes and G.G. Rose. Originally, it had also integrity functionality, which was later removed due to discovered weaknesses in it [7]. The best known attack on SOBER-128 is due to J.Y. Cho and J. Pieprzyk [5]. It uses an application of linear cryptanalysis for LFSR based stream ciphers as was presented by D. Coppersmith, et al., in [2]. First, linear approximate relations over nonlinear functions are derived which involve terms from the LFSR state variables and key stream. Then a linear recursive relation originating from the LFSR feedback relation is used to cancel the internal LFSR state variables to obtain an approximate linear equation involving key stream variables only. The linear recursive relation involving six LFSR state variables used by Cho and Pieprzcyk in [5] is due to T. Johansson and P. Ekdahl [8]. The resulting linear distinguishing attack requires $2^{103.6}$ terms of the keystream.

In our analysis, we use the original 32-bit feedback recursion of the LFSR of SOBER-128. The middle part of the nonlinear filter function of SOBER-128 has two subsequent additions modulo $2^3 2$ with a key dependent secret constant xored to the data between the additions. The main observation our analysis is based on is that the biases of linear approximations over the middle part depend on the secret constant. We derive approximate linear relations over the filter function and show how the resulting approximate linear relation of the key stream variables can be used, not only to distinguish the output key stream from a purely random sequence but also to determine one bit of information of the secret constant. However, it seems that the complexity increases slightly compared to [5]. To our current estimates it takes on the average $2^{113.5}$ terms of the keystream to get one bit of information of the secret constant, and $2^{124.6}$ terms to get four bits of the secret constant.

The cryptanalysis technique developed in this paper is not specific to SOBER-128. It can be applied when ever linear approximations are taken over cryptographic functions involving secret constants. One linear approximation divides the constants upto three classes depending on whether the bias of the keystream relation is zero, positive or negative. We will also see that such a division into classes is not necessarily determined by a linear equation as is typically the case in linear cryptanalysis, for example, in the seminal work of M. Matsui in [9].

2 Preliminaries

In this section, we introduce the definitions and notation that are used throughout the paper. The terminology follows closely to the one used in [4], since it serves our purpose to examine linear approximations of functions that are composed of arithmetic and Boolean operations. We denote by x the n-bit vector $(x_0, \ldots, x_{n-1})$ in $\mathbb{F}_2^n$. The integers in $\{0, \ldots, 2^n - 1\}$ are identified with the vectors in $\mathbb{F}_2^n$ using the natural correspondence $x \leftrightarrow \sum_{j=0}^{n-1} x_j 2^j$.

Linear cryptanalysis [9] exploits correlations between certain linear combinations of the input and output bits of the components of the cipher. Let n and m be positive integers. In this paper, we consider a component of the cipher to be a mapping $f \colon \mathbb{F}_2^{m\times n} \to \mathbb{F}_2^n$, i.e., a mapping that takes m n-bit input words and maps them to a single n-bit output word. The following terminology is used to discuss linear approximations of components throughout the paper.

A constant vector or matrix that is used to select what input (output) bits will be used in a linear approximate relation of a function is called a *linear input (output) mask* of the function. Let m and n be positive integers. For vectors $x \in \mathbb{F}_2^n$ and $y \in \mathbb{F}_2^n$, let $x \cdot y$ denote the standard inner product $x \cdot y = x_0 y_0 \oplus \cdots \oplus x_{n-1} y_{n-1}$. A *linear approximation* of a functional dependency $f \colon \mathbb{F}_2^{m\times n} \to \mathbb{F}_2^n$ is an approximate relation of the form

$$\Gamma \cdot f(X) = \bigoplus_{i=0}^{m-1} \Lambda^{(i)} \cdot x^{(i)} \ ,$$

where the row vectors $\Lambda^{(0)}, \ldots, \Lambda^{(m-1)} \in \mathbb{F}_2^n$ are the linear input masks for the input words and $\Gamma \in \mathbb{F}_2^n$ is the linear output mask. The linear input mask for f is the matrix $\Lambda = (\Lambda_{i,j}) \in \mathbb{F}_2^{m\times n}$ with $\Lambda^{(0)}, \ldots, \Lambda^{(m-1)}$ as the rows. The efficiency of a linear approximation of f is measured by its *correlation*

$$\mathrm{cor}_f(\Gamma; \Lambda) = 2 \Pr \left[\Gamma \cdot f(X) = \bigoplus_{i=0}^{m-1} \Lambda^{(i)} \cdot x^{(i)} \right] - 1 \ ,$$

which is the probability that $\Gamma \cdot f(X) = \bigoplus_{i=0}^{m-1} \Lambda^{(i)} \cdot x^{(i)}$ taken over X and scaled between $[-1, 1]$. We use $\epsilon_f(\Gamma; \Lambda) = \mathrm{cor}_f(\Gamma; \Lambda)/2$ to denote the *bias* of a linear approximation of f. The linear approximation of f with the input mask Λ and the output mask Γ may be denoted by $(\Gamma; \Lambda)$ or by stating the input masks for each input word explicitly $(\Gamma; \Lambda^{(0)}, \ldots, \Lambda^{(m-1)})$. A semicolon is used for separating the output mask to the left and the input mask(s) to the right. Given a linear mask $\Gamma \in \mathbb{F}_2^n$ and an element $\alpha \in \mathbb{F}_2^n$, we denote by $\Gamma\alpha$ the linear mask, which satisfies the equality

$$\Gamma\alpha \cdot x = \Gamma \cdot \alpha x \quad \text{for all } x \in \mathbb{F}_2^n \ ,$$

where the product αx is taken in $\mathbb{F}_{2^n}$.

3 The Stream Cipher SOBER-128

SOBER-128 [6] is a synchronous stream cipher that generates a keystream of 32-bit words based on a 128-bit secret key. Originally, it also contained message authentication functionality, but it has been removed recently due to vulnerabilities to forgery attacks [6]. The structure of the SOBER-128 keystream generator is a traditional combination of a linear feedback shift register (LFSR) and a nonlinear filter (NLF). An illustration of this structure is depicted in Fig. 1.

The LFSR consists of 17 registers, each containing a 32-bit word. We use the vector $(s_t, \ldots, s_{t+16})$ to define the state of the LFSR at time t. The new state at time $t+1$ is determined with the characteristic polynomial

$$x^{17} + x^{15} + x^4 + \gamma \in \mathbb{F}_{2^{32}}[x] \ , \tag{1}$$

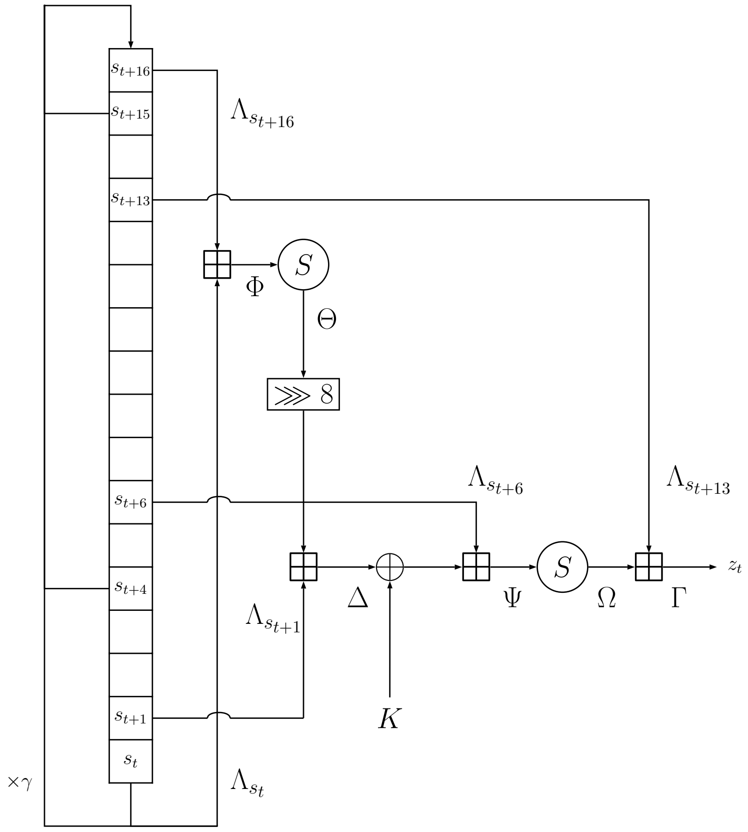

Fig. 1. The keystream generator SOBER-128

where $\gamma \in \mathbb{F}_{2^{32}}$ is a constant. The field $\mathbb{F}_{2^{32}}$ is realized using the isomorphic representation $\mathbb{F}_{(2^8)^4}$. If we denote the polynomials in $\mathbb{F}_{2^8}$ with their coefficients in a hexadecimal number, the constant γ is defined as $\texttt{0x01}x$ in $\mathbb{F}_{(2^8)^4}$. The NLF is a nonlinear function of the LFSR states and a key-dependent constant $K \in \mathbb{F}_2^{32}$. In this paper, however, we do not consider the constant as an argument but as an internal part of the NLF. At time t the NLF produces a 32-bit keystream word z_t as follows

$$\begin{aligned} z_t &= f_{\mathrm{NLF}}(s_t, s_{t+1}, s_{t+6}, s_{t+13}, s_{t+16}) \\ &= S((((S(s_t \boxplus s_{t+16}) \ggg 8) \boxplus s_{t+1}) \oplus K) \boxplus s_{t+6}) \boxplus s_{t+13} \ . \end{aligned}$$

The function $S \colon \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ is defined as $S(x) = S_1(x_{31,\dots,24}) \,\|\, (S_2(x_{31,\dots,24}) \oplus x_{23,\dots,0})$, where $\|$ denotes concatenation and S_1 and S_2 are used to denote the Skipjack S-box [10] and a tailor-designed ISRC S-box [11] respectively. Again, we define $f_K \colon \mathbb{F}_2^{3\times n} \to \mathbb{F}_2^n$ as $f_K(X) = ((x^{(0)} \boxplus x^{(1)}) \oplus K) \boxplus x^{(2)}$ and set $n = 32$. For a more detailed specification of SOBER-128, refer to [6].

4 Linear Masking of SOBER-128

The linear masking method for stream ciphers is generally based on finding linear approximations with high bias over nonlinear components of the cipher. These approximations are then applied multiple times using a linear relation in the keystream generator, which results into an approximation consisting only of output words. The bias of this approximation can be calculated using the *piling-up lemma* [9]. We apply linear masking as usual to the SOBER-128 keystream generator. Our purpose is, however, to search multiple approximations that partition the constant $K \in \mathbb{F}_2^{32}$ into different classes based on the correlation. This would allow us to gain information from K using independent distinguishers as described in Sect. 3. We are especially interested in how the constants partition based on the sign of the correlation, since it is possible to get larger correlation differences between constant classes. We will discuss (i) how linear approximations of f_K partition constants K based on their correlation, (ii) what distinguishers were found, (iii) what heuristic was used for searching linear masks, and (iv) how does the constant γ in (1) affect attacks based on linear masking.

Let $\mathit{\Gamma}$ and $\mathit{\Lambda}$ denote the output and input mask of a linear approximation of f_{NLF} respectively. The characteristic polynomial (1) for the LFSR yields the linear relation

$$s_{t+17} \oplus s_{t+15} \oplus s_{t+4} \oplus \gamma s_t = 0 \ ,$$

which can be used for forming the main distinguishing equation that consists of output words from the keystream generator. To form this distinguishing equation, we use a linear approximation of f_{NLF} four times: three times with the masks $\mathit{\Gamma}, \mathit{\Lambda}$ at times $t+4, t+15, t+17$ and one time with the masks $\mathit{\Gamma}_\gamma, \mathit{\Lambda}\gamma$ at time t. This results in the following equation:

$$\mathit{\Gamma} z_{t+17} \oplus \mathit{\Gamma} z_{t+15} \oplus \mathit{\Gamma} z_{t+4} \oplus \mathit{\Gamma}_\gamma z_t = 0 \ . \tag{2}$$

We let $\epsilon_{\mathrm{NLF}}(\Gamma;\Lambda)$ denote the bias of a linear approximation $(\Gamma;\Lambda)$ of f_{NLF}. Hence, the total bias $\epsilon(\Gamma,\Gamma_\gamma)$ of the distinguishing equation is

$$\epsilon(\Gamma,\Gamma_\gamma) = 8\epsilon_{\mathrm{NLF}}(\Gamma;\Lambda)^3\epsilon_{\mathrm{NLF}}(\Gamma_\gamma;\Lambda\gamma)\ .$$

4.1 Linear Approximations of f_K

Consider the function $f_K : \mathbb{F}_2^{3\times n} \to \mathbb{F}_2^n$ given as $f_K(X) = ((x^{(0)} \boxplus x^{(1)}) \oplus K) \boxplus x^{(2)}$, where $K \in \mathbb{F}_2^n$ is a constant. We can compute $\mathrm{cor}_{f_K}(\Gamma;\Lambda)$ using similar methods as those used in [4]. Suppose that a linear approximation $(\Gamma;\Lambda)$ of f_K is fixed. It is now possible to determine the exact correlation $\mathrm{cor}_{f_K}(\Gamma;\Lambda)$ for a given constant $K \in \mathbb{F}_2^n$. Intuitively, it is clear that the correlation $\mathrm{cor}_{f_K}(\Gamma;\Lambda)$ might change, if the constant in f_K is changed.

To show that constants really do partition into several classes with some linear approximations, we consider correlations of f_K with two linear approximations for 5-bit words. Tables of the constant classes for these cases are presented in Appendix A. Since it is possible to classify constants into certain classes based on their correlation, we have an obvious way for gaining information from an unknown constant from some T-functions based on one or more linear approximations. For stream ciphers, it means that if an unknown constant is applied within a T-function in the nonlinear function of the keystream generator, we can use linear distinguishers for determining possible values for the constant. One suitable stream cipher is SOBER-128, which we will examine in the following sections.

We examine linear approximations of the nonlinear function f_K with two purposes: (i) to find out the average, minimum, and maximum of $|\mathrm{cor}_{f_K}(\Gamma;\Lambda)|$ when $(\Gamma;\Lambda)$ is fixed, and (ii) to to find out how the constants $K \in \mathbb{F}_2^{32}$ partition into classes based on the sign of the correlation when $(\Gamma;\Lambda)$ is fixed. We try to find linear approximations for which the average of $|\mathrm{cor}_{f_K}(\Gamma;\Lambda)|$ (over all $K \in \mathbb{F}_2^{32}$) is the highest, since experimentation shows that sign of the correlation varies often. Hence, this approach gives a large correlation difference between the constant classes that correspond to negative and positive correlations. Another approach would be to ignore the sign of the correlation and search those approximations whose constant classes have large correlation difference between their absolute values. This approach can also be used in cases, when it is not possible to examine sign of the correlation as is observed in Sect. 4.4. We compute average correlations by considering the constant K as a uniformly distributed random variable. In this case, the additions modulo 2^{32} are completely independent of each other. Using the piling-up lemma, the average correlation can be defined by simply multiplying the correlations of independent additions together. For computing $\mathrm{cor}_{f_K}(\Gamma;\Lambda)$ with a certain K we need to consider f_K as a single function. Therefore, we can determine the maximum and minimum values for $|\mathrm{cor}_{f_K}(\Gamma;\Lambda)|$ by enumerating 2^{32} constants at maximum. The constant classes can be formed at the same time when constants are enumerated. So far we have not been able to find an analytic method to define the constant classes.

4.2 Our Results

We determined four linearly independent linear approximations for the NLF. The algorithm that was used to search the chain of approximations for the NLF is given in the next section. Each linear approximation partitions the constants $K \in \mathbb{F}_2^{32}$ into two classes of the same size based on the sign of the correlation. Hence, each distinguisher allows extracting one bit of information from K. These constant classes are also pairwise orthogonal, which means that we get 16 constant classes of the same size, that is, four bits of information of the secret constant. The approximations are presented in Table 1 with their maximum, minimum, and average biases, and the relations on the constant $K = (k_0, k_1, \ldots, k_{31})$ which determine the splittings into classes. All four splitting relations in Table 1 are linear. However, this is not necessarily the case in general. An example of a nonlinear division relation is given in the Appendix.

Table 1. Distinguishers and their biases.

Γ	Γ_γ	$\lvert\epsilon(\Gamma, \Gamma_\gamma)\rvert$			constant class with bias > 0
		max	avg	min	
`0x01980000`	`0x00011000`	$2^{-53.288}$	$2^{-56.735}$	$2^{-62.001}$	$k_{12} + k_{16} + k_{19} + k_{20} + k_{23} + k_{24} = 0$
`0x00000181`	`0x24000001`	$2^{-55.385}$	$2^{-58.290}$	$2^{-62.385}$	$k_2 + k_3 + k_{13} + k_{14} + k_{22} + k_{26} + k_{27} = 0$
`0x0040000c`	`0x08006000`	$2^{-57.701}$	$2^{-61.155}$	$2^{-66.112}$	$k_7 + k_8 + k_{24} + k_{30} = 0$
`0x000000c0`	`0x21000000`	$2^{-58.959}$	$2^{-62.279}$	$2^{-66.638}$	$k_6 + k_7 + k_{22} + k_{29} + k_{31} = 0$

4.3 Mask Search Methods

We searched for the linear masks using similar techniques as in [4]. In particular we took advantage of the possibility to generate all linear masks with a given correlation for one addition modulo 2^{32}. In this section, the term correlation is used to refer to the absolute value of correlation unless otherwise specified. We split the NLF into components that can reasonably be assumed to be independent. Then an approximation is created for one nonlinear component at a time. We progress to the next component, when we have found an approximation with correlation that is higher than the preset limits. During this process, we keep track of the total correlation using the piling-up lemma. Used masks are depicted in Fig. 1. The subscript γ is used to denote masks that work when each input s_i from the LFSR has been multiplied with γ. We start by generating masks for the addition with s_{t+1} as an input. All masks $\Lambda_{s_{t+1}}, \Theta$, and Δ are generated with a correlation $\geq 2^{-3}$. For each $\Lambda_{s_1}\gamma$, we generate Θ_γ and Δ_γ with a correlation $\geq 2^{-4}$. The three least significant bytes of Θ and Θ_γ are also the three least significant bytes of Φ and Φ_γ. Previous experiences show that large correlations are achieved with masks that have a low Hamming weight [12, 3]. Hence, we iterate all values with a Hamming weight ≤ 4 for the most significant byte of Φ and generate Λ_{s_t} and $\Lambda_{s_{t+16}}$ with a correlation $\geq 2^{-3}$. We continue with masks that have a nonzero correlation over S. For the input masks $\Lambda_{s_t}\gamma$ and $\Lambda_{s_{t+16}}\gamma$, we iterate all values with a Hamming weight ≤ 4 for the most significant byte of Φ_γ and compute the correlation for the addition and S. We continue with masks that have a correlation $\geq 2^{-6}$ over the addition and a nonzero correlation over S. We continue from the addition with s_{t+6} as an input. Using Δ we generate all masks $\Lambda_{s_{t+6}}$ and Ψ with a correlation $\geq 2^{-3}$. For each $\Lambda_{s_{t+6}}\gamma$ and Δ_γ, we generate Ψ_γ with a correlation $\geq 2^{-4}$. These approximations fix the three least significant bytes of Ω and Ω_γ. We iterate all values for the most significant byte with a Hamming weight ≤ 4 and generate $\Lambda_{s_{t+13}}$ and Γ with a correlation $\geq 2^{-3}$. For the $\Lambda_{s_{t+13}}\gamma$, we generate Γ_γ with a correlation $\geq 2^{-4}$ by iterating again all values with a Hamming weight ≤ 4 for the most significant byte of Ω_γ. A chain of approximations for the NLF has now been created.

4.4 Effect of γ in the Characteristic Polynomial

Without γ in the characteristic polynomial (1), the distinguishing equation (2) is formed using the same linear approximation $(\Gamma; \Lambda)$ four times. Hence, we get the same equation as (2) but with Γ_γ replaced with Γ. The bias is determined as

$$\epsilon(\Gamma) = 8\epsilon_{\mathrm{NLF}}(\Gamma; \Lambda)^4 \ .$$

In this case, the sign of $\epsilon_{\mathrm{NLF}}(\Gamma; \Lambda)$ would cancel out, which makes it harder to find constant classes that have large correlation differences. On the other hand, a stronger distinguishing attack could be launched. The best linear mask we found was $\Gamma =$ `0x03000001` with (average) bias $\epsilon(\Gamma) = 2^{-36.771}$.

4.5 Data Complexity

Using the best linear approximation on the NLF of SOBER-128, one bit of information of K can be obtained using $2^{113.5}$ keystream words on average. For obtaining two bits of information, another mask is needed. Therefore, we need $2^{116.6}$ keystream words on average for gaining two bits of information. This is a conservative estimate assuming that the two linear approximations are statistically independent.

Previously, Kaliski and Robshaw [13] and Biryukov et al. [14] have investigated the data complexity when multiple linear approximations are used. In both papers it is assumed that the linear approximations are statistically independent. However, this is very unlikely to be the case in reality, when statistical dependencies, while being diluted, appear almost everywhere.

A more accurate estimate of the bias is achieved by considering the joint statistical distribution of multiple linear approximative relations. We have not computed joint distributions for the linear approximations of SOBER-128, yet. It is left for future work. For each constant, the theoretical statistical distribution must be derived. It is expected that many constants are going to have the same distribution and are, in this sense, equivalent. The correct equivalence class is then found by comparing the empirical distribution with the theoretical distributions.

5 Conclusions

We have proposed new techniques how to analyze secret constants in key stream generators using linear approximations. It turns out that the value of the correlation depends on the constant. In this manner the constants are divided into up to three classes depending on whether the resulting correlation is zero, positive or negative. Given sufficient amount of observed data computed using this function, one to two bits of information of the secret constant can be obtained. As the constant occurs between mutually dependent functions the piling-up lemma by Matsui in [9] cannot be used. For the same reason, division of the constants into classes is not always determined by linear relations. We applied this technique to stream cipher SOBER-128 which involves a secret, key-dependent constant in its output filter function. We presented a linear cryptanalysis method using which a number of bits can be recovered from the secret constant.

References

1. ECRYPT: The homepage for eSTREAM. (`http://www.ecrypt.eu.org/stream/`)
2. Coppersmith, D., Halevi, S., Jutla, C.: Cryptanalysis of stream ciphers with linear masking. In: Advances in Cryptology – CRYPTO 2002. Volume 2442 of Lecture Notes in Computer Science., Springer-Verlag (2002) 515–532
3. Watanabe, D., Biryukov, A., Cannière, C.D.: A distinguishing attack of SNOW 2.0 with linear masking method. In: Selected Areas in Cryptography (SAC). Volume 3006 of Lecture Notes in Computer Science., Springer-Verlag (2004) 222–233
4. Nyberg, K., Wallén, J.: Improved linear distinguishers for SNOW 2.0. In: Fast Software Encryption (FSE). Volume 4047 of Lecture Notes in Computer Science., Springer-Verlag (2006) 144–162
5. Cho, J.Y., Pieprzyk, J.: Distinguishing attack on SOBER-128 with linear masking. In: Information Security and Privacy (ACISP). Volume 4058 of Lecture Notes in Computer Science., Springer-Verlag (2006) 29–39
6. Hawkes, P., Paddon, M., Rose, G.G.: Primitive specification for SOBER-128. Technical report, Qualcomm Australia (2003)
7. QUALCOMM Australia: The homepage for SOBER-128. (`http://www.qualcomm.com.au/Sober128.html`)
8. Ekdahl, P., Johansson, T.: Distinguishing attacks on SOBER-t16 and t32. In: Fast Software Encryption (FSE). Volume 2365 of Lecture Notes in Computer Science., Springer-Verlag (2002) 210–224
9. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Advances in Cryptology – EUROCRYPT 1993. Volume 765 of Lecture Notes in Computer Science., Springer-Verlag (1994) 386–397
10. National Institute of Standards and Technology (NIST): Escrowed encryption standard. Federal Information Processing Standards Publication (FIPS PUB) 185 (1994)
11. Dawson, E., Millan, W., Burnett, L., Carter, G.: On the design of 8*32 S-boxes. Unpublished report, Information Systems Research Centre (ISRC), Queensland University of Technology (QUT) (1999)
12. Wallén, J.: Linear approximations of addition modulo 2^n. In: Fast Software Encryption (FSE). Volume 2887 of Lecture Notes in Computer Science., Springer-Verlag (2003) 261–273
13. Jr., B.S.K., Robshaw, M.J.B.: Linear cryptanalysis using multiple approximations. In: CRYPTO. (1994) 26–39
14. Biryukov, A., Cannière, C.D., Quisquater, M.: On multiple linear approximations. In: CRYPTO. (2004) 1–22

A Examples of Constant Partitions for f_K

We give two examples of how the constants $K \in \mathbb{F}_2^n$ partition into classes with a fixed linear approximation $(\Gamma; \Lambda)$ of f_K. The partitions are presented in Tables 2 and 3, where the constants under certain correlation belong to the same class. For clarity, we denote the elements in $\mathbb{F}_2^n$ as binary numbers.

The constants are divided into classes according to the following relations. In Table 2, the constants belong to a class with a zero or nonzero correlation depending on whether $k_0 = 0$ or 1. Furthermore, depending on whether $k_1 = 0$ or 1, the constants belong to the class with a negative or positive correlation. In Table 3, the constants belong to the class with a zero correlation if the nonlinear relation $k_1 \oplus k_3 \oplus k_1k_2 \oplus k_0k_1k_2 \oplus k_1k_2k_3 \oplus k_0k_1k_2k_3 = 0$ applies. The rest of the constants belong to the class with a negative or positive correlation depending whether $k_0 \oplus k_1 \oplus k_2 \oplus k_3 \oplus k_4 = 1$ or 0. Hence, the classes are not always determined by linear relations, when the constant is within a T-function.

Table 2. The constant classes for the linear approximation with masks $\Gamma = 00011$, $\Lambda^{(0)} = 00011$, $\Lambda^{(1)} = 00011$, $\Lambda^{(2)} = 00011$ of f_K when $n = 5$.

cor_{f_K}	-2^{-1}	0	2^{-1}
K	00001	00000, 10000	00011
	00101	00010, 10010	00111
	01001	00100, 10100	01011
	01101	00110, 10110	01111
	10001	01000, 11000	10011
	10101	01010, 11010	10111
	11001	01100, 11100	11011
	11101	01110, 11110	11111

Table 3. The constant classes for the linear approximation with masks $\Gamma = 11101$, $\Lambda^{(0)} = 10111$, $\Lambda^{(1)} = 11110$, $\Lambda^{(2)} = 11101$ of f_K when $n = 5$.

cor_{f_K}	-2^{-4}	0	2^{-4}
K	00010	00000, 10000	00011
	00111	00001, 10001	00110
	01000	00100, 10100	01001
	01101	00101, 10101	01100
	10011	01010, 11010	10010
	10110	01011, 11011	10111
	11001	01110, 11110	11000
	11100	01111, 11111	11101