

Improved Correlation Attacks on SOSEMANUK and SOBER-128

Joo Yeon Cho

Helsinki University of Technology
Department of Information and Computer Science,
Espoo, Finland

24th March 2009

Outline

SOSEMANUK

Attack Method

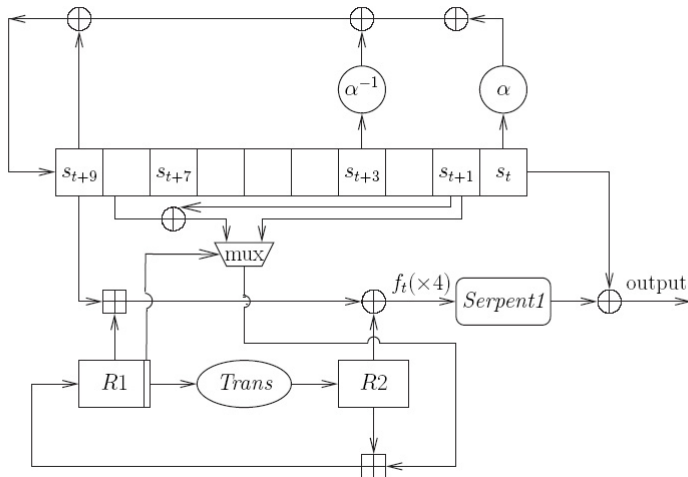
Searching Linear Approximations

SOBER-128

SOSEMANUK (from Wiki)

- A software-oriented stream cipher designed by Come Berbain, Olivier Billet, Anne Canteaut, Nicolas Courtois, Henri Gilbert, Louis Goubin, Aline Gouget, Louis Granboulan, Cédric Lauradoux, Marine Minier, Thomas Pornin and Hervè Sibert.
- One of the final four Profile 1 (software) ciphers selected for the eSTREAM Portfolio, along with HC-128, Rabbit, and Salsa20/12.
- Influenced by the stream cipher SNOW and the block cipher Serpent.
- The cipher key length can vary between 128 and 256 bits, but the guaranteed security is only 128 bits.
- The name means "snow snake" in the Cree Indian language because it depends both on SNOW and Serpent.

Overview



Structure

1. The states of LFSR : s_0, \dots, s_9 (320 bits)

$$s_{t+10} = s_{t+9} \oplus \alpha^{-1}s_{t+3} \oplus \alpha s_t, \quad t \geq 1$$

where α is a root of the primitive polynomial.

2. The Finite State Machine (FSM) : R_1 and R_2

$$R1_{t+1} = R2_t \boxplus (r_t s_{t+9} \oplus s_{t+2})$$

$$R2_{t+1} = Trans(R1_t)$$

$$f_t = (s_{t+9} \boxplus R1_t) \oplus R2_t$$

where r_t denotes the least significant bit of $R1_t$.

3. The trans function $Trans$ on $\mathbb{F}_{2^{32}}$:

$$Trans(R1_t) = (R1_t \times 0x54655307 \bmod 2^{32}) \lll 7$$

4. The output of the FSM :

$$(z_{t+3}, z_{t+2}, z_{t+1}, z_t) = Serpent1(f_{t+3}, f_{t+2}, f_{t+1}, f_t) \oplus (s_{t+3}, s_{t+2}, s_{t+1}, s_t)$$

Previous Attacks

- Authors state that "No linear relation holds after applying *Serpent1* and there are too many unknown bits...".
- In Asiacrypt'08, the best linear approximation with the correlation of $2^{-21.41}$ was derived as

$$FSM : \quad \Gamma \cdot f_t \oplus \Gamma \cdot f_{t+1} \oplus \Gamma \cdot s_{t+10} \oplus \Gamma \cdot s_{t+2} = 0$$

$$Serpent1 : \quad \Gamma \cdot f_t \oplus \Gamma \cdot f_{t+1} \oplus \Gamma \cdot (s_t \oplus z_t) \oplus \Gamma \cdot (s_{t+3} \oplus z_{t+3}) = 0$$

- Using this approximation, a correlation attack was applied, which is the similar attack applied to Grain stream cipher.
- The complexity of attack was estimated around $2^{140.5}$ data, 2^{148} computing time and 2^{147} memory.

Motivation of Our Work

- We may obtain better approximations if we use different masks for FSM and Serpent1.
- We may reduce the data complexity of the attack by using multiple linear approximations with equal correlations.

LFSR and Linear Approximations

1. The linear recurrence of SOSEMANUK is expressed as

$$\begin{pmatrix} s'_0 \\ s'_1 \\ \cdots \\ s'_9 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ b_0 & b_1 & b_2 & \cdots & b_9 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \cdots \\ s_9 \end{pmatrix}$$

Since $s_{t+10} = s_{t+9} \oplus \alpha^{-1}s_{t+3} \oplus \alpha s_t$, we get

$(b_0 \ b_1 \ \cdots \ b_9) = (\alpha \ 0 \ 0 \ \alpha^{-1} \ 0 \ \cdots \ 1)$ where $s_i, b_i, \alpha \in GF(2^{32})$.

2. We can simply denote $S_{t+1} = AS_t$. Then, $S_t = A^t S_0$.
3. A linear approximation $U \cdot S_t \oplus W \cdot Z_t = 0$ is expressed as

$$U \cdot A^t S_0 \oplus W \cdot Z_t = 0, \quad t > 0.$$

Note that $U = (u_0 \ u_1 \ \cdots \ u_9)$ and $U \cdot S_t = u_0 \cdot s_t \oplus \cdots \oplus u_9 \cdot s_{t+9}$ where $u_i \in GF(2^{32})$. Similar for $W \cdot Z_t$.

Naive Attack

1. Assume $U \cdot S_t \oplus W \cdot Z_t = 0$ has the correlation of c_{sose} .
2. Observe N keystreams. Then, we obtain

$$\begin{pmatrix} U \cdot AS_0 \\ U \cdot A^2S_0 \\ \vdots \\ U \cdot A^N S_0 \end{pmatrix} \oplus \begin{pmatrix} W \cdot Z_1 \\ W \cdot Z_2 \\ \vdots \\ W \cdot Z_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where $S_0 = (s_0 \ s_1 \ \cdots \ s_9)^T$.

3. Guess S_0 . For each candidate, compute D which is defined as

$$D = \frac{1}{N} (\#\{U \cdot A^t S_0 \oplus W \cdot Z_t = 0\} - \#\{U \cdot A^t S_0 \oplus W \cdot Z_t = 1\})$$

If guessed S_0 is correct, D is close to c_{sose} . Otherwise, D is close to zero.

Fast Walsh Transform and Complexity

1. Assume $S_0 = (x_1 \ x_2 \ \cdots \ x_l)$ and $U \cdot A^t = (a_{1t} \ a_{2t} \ \cdots \ a_{lt})$ where $x_i, a_i \in \{0, 1\}$. Then,

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ & \vdots & & \\ a_{N1} & a_{N2} & \cdots & a_{Nl} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_l \end{pmatrix} \oplus \begin{pmatrix} W \cdot Z_1 \\ W \cdot Z_2 \\ \vdots \\ W \cdot Z_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

2. Since there are 2^l candidates for S_0 , the complexity is around $N \times 2^l$.
3. If Fast Walsh Transform is used, the complexity is reduced to around $N + 2^l \log 2^l = N + l \times 2^l$.
4. This is worse than state exhaustive search.

Simple Example on Fast Walsh Transform

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{array}{r} \begin{array}{ccccc} & x_1 & x_2 & x_3 & x_1 \oplus x_2 & \cdots \\ \hline (0) & 0 & 0 & 0 & 0 & \\ (1) & 1 & 0 & 0 & 1 & \\ (0) & 0 & 1 & 0 & 1 & \\ (0) & 1 & 1 & 0 & 0 & \\ (1) & 0 & 0 & 1 & 0 & \\ (2) & 1 & 0 & 1 & 1 & \\ (1) & 0 & 1 & 1 & 1 & \\ (1) & 1 & 1 & 1 & 0 & \end{array} \end{array}$$

Reducing Time Complexity

1. Let $\Omega_m = \{(x_1 \ x_2 \ \dots \ x_l) | x_i \in \{0, 1\}, x_{m+1} = \dots = x_l = 0\}$ for $1 \leq m \leq l$. Clearly, $|\Omega_m| = 2^m$.
2. Among N approximations, take $U \cdot A^t S_0 \oplus W \cdot Z_t = 0$ such that $U \cdot A^t S_0 \in \Omega_m$.

$$\begin{pmatrix} U \cdot A^{\tau_1} S_0 \\ U \cdot A^{\tau_2} S_0 \\ \vdots \\ U \cdot A^{\tau'_N} S_0 \end{pmatrix} \oplus \begin{pmatrix} W \cdot Z_{\tau_1} \\ W \cdot Z_{\tau_2} \\ \vdots \\ W \cdot Z_{\tau'_N} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

3. The probability that such approximation occurs is $2^m/2^l$. Hence, we obtain around $N' \approx N \times 2^m/2^l$ 'good' approximations.
4. By Fast Walsh Transform, time complexity is reduced to $N' + m \times 2^m$.

Second LFSR Derivative Technique

1. Used for the attack on Grain Version 0 by Berbain et al.
2. Obtain more "good" approximations without further the keystream observations.
3. Perform pairwise combinations of N approximations as

$$(U \cdot A^i \oplus U \cdot A^j)S_0 \oplus (W \cdot Z_i \oplus W \cdot Z_j) = 0, \quad 1 \leq i, j \leq N$$

4. Choose combined approximations such as $(U \cdot A^i S_0 \oplus U \cdot A^j S_0) \in \Omega_m$. with the correlation of c_{sose}^2 .
5. The number of approximations that satisfy this condition is expected to be $N' = 2^{m-l} \binom{N}{2} \approx 2^{m-l} \times N^2$.

Sorting and Combining

1. A simple pairing requires $\binom{N}{2} \approx N^2$ operations.
2. The number of operations can be reduced by applying sorting-and-combining technique.
3. First, N approximations are sorted out according to the value of $(l - m)$ state bits.
4. Let the sorted approximations be represented by X_1, X_2, \dots, X_N . Then, two consecutive approximations X_i and X_{i+1} are checked whether their $(l - m)$ state bits are same.
5. If they are same, we know $X_i \oplus X_{i+1} \in \Omega_m$.
6. The fastest sorting algorithm takes $O(N \log N)$.
7. Time complexity : $T = N \times \log(N) + m \times 2^m$.

Linear Approximations of FSM

- Using five masks $(\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5)$, we get

$$\Gamma_2 \cdot R2_{t+1} = \Phi \cdot R1_t$$

$$\Lambda \cdot R1_{t+1} = \Gamma_1 \cdot R2_t \oplus \Gamma_4 \cdot (s_{t+2} \oplus r_i s_{t+9})$$

$$\Gamma_1 \cdot f_t = \Gamma_3 \cdot s_{t+9} \oplus \Phi \cdot R1_t \oplus \Gamma_1 \cdot R2_t$$

$$\Gamma_2 \cdot f_{t+1} = \Gamma_5 \cdot s_{t+10} \oplus \Lambda \cdot R1_{t+1} \oplus \Gamma_2 \cdot R2_{t+1}$$

- By combining above approximations

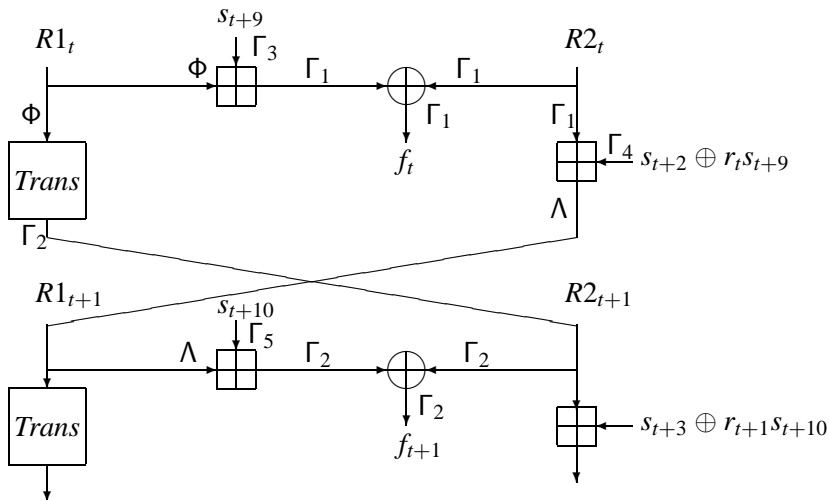
$$\Gamma_1 \cdot f_t \oplus \Gamma_2 \cdot f_{t+1} = \Gamma_3 \cdot s_{t+9} \oplus \Gamma_5 \cdot s_{t+10} \oplus \Gamma_4 \cdot (s_{t+2} \oplus r_i s_{t+9})$$

- The correlation is $c_{FSM} = c_{TranPlus} \times c_{PlusPlus}$ where

$$c_{TransPlus} = \sum_{\Phi=1}^{2^{32}-1} c_+(\Gamma_3, \Phi; \Gamma_1) c_{Trans}(\Phi; \Gamma_2)$$

$$c_{PlusPlus} = \frac{1}{2} \sum_{\Lambda=1}^{2^{32}-1} c_+(\Gamma_1, \Gamma_4; \Lambda) c_+(\Gamma_5, \Lambda; \Gamma_2)$$

Linear Masking of FSM



Observations on Trans Function

1. Recall $Trans(R1) = (R1 \times 0x54655307 \bmod 2^{32}) \lll 7$.
2. Multiplication : 14 consecutive modular additions
($Ham(0x54655307) = 14$)

$$\begin{aligned} & (R1 \times 0x54655307 \bmod 2^{32}) \\ &= R1 \boxplus (R1 \lll 1) \boxplus (R1 \lll 2) \boxplus (R1 \lll 8) \boxplus \dots \boxplus (R1 \lll 30) \end{aligned}$$

3. Due to the rotation $\lll 7$, Linear masks must have ones in the bit positions of $\{i + 25\}$, $i = 0, 1, \dots$, or 6. In particular, Γ_2 must have one in the bit positions of $\{i + 25, \dots, i\}$, $i = 0, 1, \dots$, or 6.
4. Provided $x \boxplus y = z$, let a linear approximation be $\Psi_1 \cdot x \oplus \Psi_2 \cdot y = \Psi_3 \cdot z$. Then, the positions of most significant effective bit of Ψ_1, Ψ_2, Ψ_3 are same.

Linear Approximations of Serpent1

$$(z_{t+3}, z_{t+2}, z_{t+1}, z_t) = \text{Serpent1}(f_{t+3}, f_{t+2}, f_{t+1}, f_t) \oplus (s_{t+3}, s_{t+2}, s_{t+1}, s_t)$$

$$\Rightarrow \Gamma_1 \cdot f_t \oplus \Gamma_2 \cdot f_{t+1} = \bigoplus_{i=0}^3 \zeta_i \cdot (s_{t+i} \oplus z_{t+i}).$$

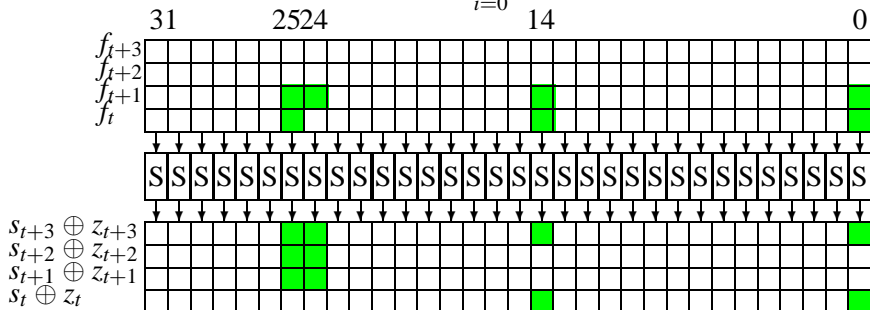


Figure: $\Gamma_1 = 0x02004001, \Gamma_2 = 0x03004001$

Correlation of Serpent1

1. One of the best approximations is $\zeta_0 = 0x00004001$, $\zeta_1 = 0x03000000$, $\zeta_2 = 0x03000000$, $\zeta_3 = 0x03004001$
2. The correlation is

$$c_S(3; 14) \times c_S(2; 14) \times c_S(3; 9) \times c_S(3; 9) = 2^{-4}$$

where $c_S(\gamma_i; \lambda_j)$ denote a correlation of a single S-box induced by the input mask γ_i and the output mask λ_j .

Multiple Approximations

1. Since $c_S(3; 9) = c_S(3; 14) = 2^{-1}$ and $c_S(2; 7) = c_S(2; 14) = 2^{-1}$, we obtain 2^4 approximations.
2. Since we have

$$\Gamma_1 \cdot f_{t+1} \oplus \Gamma_2 \cdot f_{t+2} = \bigoplus_{i=0}^3 \zeta'_i \cdot (s_{t+i} \oplus z_{t+i})$$

$$\Gamma_1 \cdot f_{t+2} \oplus \Gamma_2 \cdot f_{t+3} = \bigoplus_{i=0}^3 \zeta''_i \cdot (s_{t+i} \oplus z_{t+i})$$

we get $2^4 + 2^8 + 2^4 = 288$ approximations. Note that $c_S(6, i) = c_S(12, j) = 2^{-1}$ for $i = 3, 5, 11, 13$ and $j = 12, 13$.

3. In addition, the approximation with $r_t = 1$ is

$$(\Gamma_3 \oplus \Gamma_4) \cdot s_{t+9} \oplus \Gamma_5 \cdot s_{t+10} \oplus \Gamma_4 \cdot s_{t+2} = \bigoplus_{i=0}^3 \zeta_i \cdot (s_{t+i} \oplus z_{t+i}).$$

4. Hence, we can obtain $288 \times 2 = 576$ approximations with the same correlations for each approximation of FSM.

Combining Approximations of FSM and Serpent1

1. Approximations of FSM :

$$\Gamma_1 \cdot f_t \oplus \Gamma_2 \cdot f_{t+1} = \Gamma_3 \cdot s_{t+9} \oplus \Gamma_5 \cdot s_{t+10} \oplus \Gamma_4 \cdot (s_{t+2} \oplus r_i s_{t+9})$$

Approximations of Serpent1 :

$$\Gamma_1 \cdot f_t \oplus \Gamma_2 \cdot f_{t+1} = \bigoplus_{i=0}^3 \zeta_i \cdot (s_{t+i} \oplus z_{t+i}).$$

By combining two approximations,

$$\Gamma_3 \cdot s_{t+9} \oplus \Gamma_5 \cdot s_{t+10} \oplus \Gamma_4 \cdot s_{t+2} = \bigoplus_{i=0}^3 \zeta_i \cdot (s_{t+i} \oplus z_{t+i}).$$

with the correlation of $\sum_{\Gamma_1, \Gamma_2} c_{FSM} \times c_{Serpent1}$.

2. The strongest correlations is $2^{-21.8}$.

Searching Linear Masks

source	$ C_{FSM} $	$ C_{Serpent1} $	$ C_{sose} $	M
Lee et al.'s attack	$2^{-17.41}$	2^{-4}	$2^{-21.41}$	2^3
this paper	$2^{-17.41}$	2^{-4}	$2^{-21.41}$ 2^{-22}	$2^{11.2}$ 2^{16}

Table: $C_{sose} = C_{FSM} \times C_{Serpent1}$ and M is the number of approximations

Correlation Attack using Multiple Approximations

1. Assume we have M approximations :

$$U_i \cdot AS_0 \oplus W_i \cdot Z_i = 0, i = 1, \dots, M.$$

2. By N keystreams, we get $N \times M$ approximations :

$$\begin{pmatrix} U_1 \cdot A^1 S_0 & U_2 \cdot A^1 S_0 & \cdots & U_M \cdot A^M S_0 \\ U_1 \cdot A^2 S_0 & U_2 \cdot A^2 S_0 & \cdots & U_M \cdot A^M S_0 \\ \vdots & & & \\ U_1 \cdot A^N S_0 & U_2 \cdot A^N S_0 & \cdots & U_M \cdot A^M S_0 \end{pmatrix}$$

3. Take $U_i \cdot A^t S_0 \oplus W_i \cdot Z_t = 0$ such that $U_i \cdot A^t S_0 \in \Omega_m$. Then,

$$\begin{pmatrix} U_1 \cdot A^{\tau_1} S_0 & U_2 \cdot A^{\tau_2} S_0 & \cdots & U_M \cdot A^{\tau_M} S_0 \\ U_1 \cdot A^{\tau_{M+1}} S_0 & U_2 \cdot A^{\tau_{M+2}} S_0 & \cdots & U_M \cdot A^{\tau_{M^2}} S_0 \\ \vdots & & & \\ U_1 \cdot A^{\tau_{N'} - M + 1} S_0 & U_2 \cdot A^{\tau_{N'} - M + 2'} S_0 & \cdots & U_M \cdot A^{\tau_{N'}} S_0 \end{pmatrix}$$

Complexity

1. Data complexity :

$$N' = (N \times M)^2 \times 2^m / 2^l = c_{sose}^{-4}$$
$$\Rightarrow N = 2^{\frac{l-m}{2}} / (M \times c_{sose}^2)$$

2. Time complexity : $N \log N + m \times 2^m$.

Attack Complexity

- Set $m = 124$.
- Since $l = 320$, $M = 2^{16}$ and $c_{sose}^2 = 2^{-44}$, the data complexity is computed as $N = 2^{\frac{l-m}{2}} / (M \times c_{sose}^2) \approx 2^{126}$.
- The time complexity is computed as
$$T = m \times 2^m + N \times \log N \approx 2^{133}$$
- The memory complexity is around $l \times N + 2^m \log N = 2^{134}$.
- Repeat our attack to another set of m bits and recover $2m = 244$ bits of the initial states. The rest of the state bits ($320 - 244 = 76$ bits) are recovered by exhaustive search.

Correlation Attacks against SOBER-128

History of SOBER family

1. 1998 SOBER
2. 2000 SOBER-t32/-t16 : NESSIE candidates
⇒ Algebraic attacks on SOBER-t32/-t16 without stuttering
3. 2003 SOBER-128
⇒ Distinguishing attacks on SOBER-128 with linear masking
4. 2005 NLS (Non-Linear SOBER) : eSTREAM candidate
⇒ Crossword Puzzle Attack on NLS
5. 2006 NLSv2 : tweak version
⇒ Crossword Puzzle Attack on NLSv2
6. 2007 Shannon
⇒ Distinguishing Attack on Shannon

Brief Description of SOBER-128

- Key size : 128 bits
- It consists of a 17-word (544-bit) LFSR and a nonlinear filter (NLF).
- The connection polynomial of LFSR :

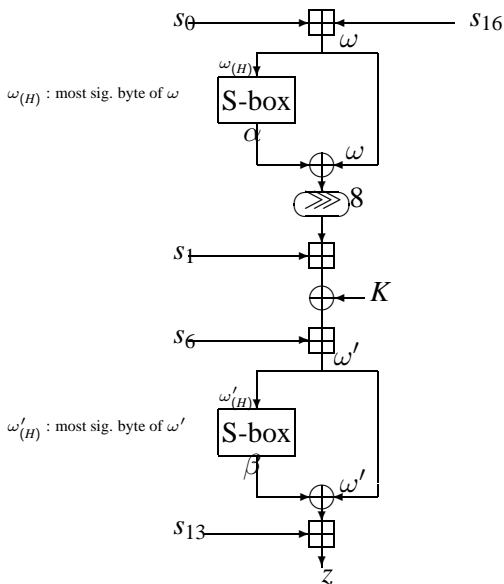
$$s_{t+17} = s_{t+15} \oplus s_{t+4} \oplus \gamma s_t, \quad \gamma = 0x00000100$$

- Output filter is described as

$$z_t = f((((f(s_t \boxplus s_{t+16}) \ggg 8) \boxplus s_{t+1}) \oplus K) \boxplus s_{t+6}) \boxplus s_{t+13},$$

- The function f is defined as $f(a) = \text{S-box}(a_H) \oplus a$, where the S-box takes 8-bit inputs and generates 32-bit outputs and a_H is the most significant 8 bits of 32-bit word a .

Non-Linear Filter (NLF) of SOBER-128

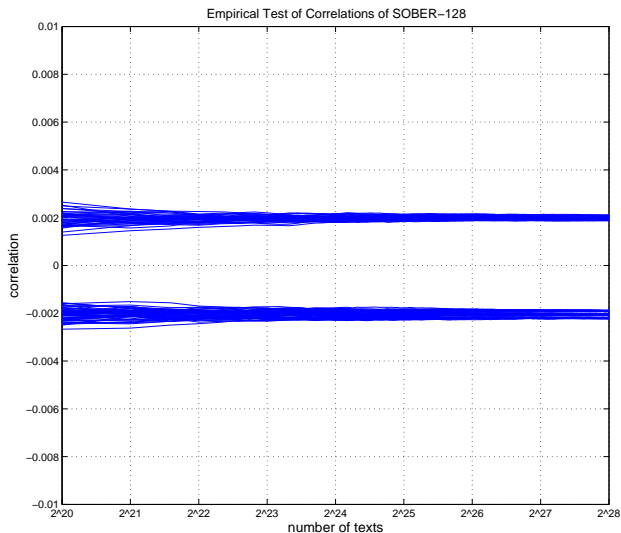


Linear Approximations of SOBER-128

source	$ c_{Sober} $	# of approx.(M)
Previous	$2^{-8.8}$	8
this paper	$2^{-8.8}$	16
	$2^{-8.9}$	24
	$2^{-9.0}$	56

Verifying Correlations

$$2^{-9} = 0.001953$$



Complexity of State Recovery Attack

- Set $l = 32 \times 17 = 544$, $c_{sober}^2 = 2^{-18}$, $m = 180$ and $M = 96$.
- Data complexity : $N = 2^{\frac{l-m}{2}} / (M \times c_{sober}^2) \approx 2^{194}$
- Time complexity : $T = m \times 2^m + N \times \log N \approx 2^{201.6}$
- Memory complexity : $l \times N + 2^m \log N = 2^{203}$

Improved Distinguishing Attack using Multiple Approximations

- The LFSR of SOBER-128 has the following relation:

$$s_{t+\tau_1} \oplus s_{t+\tau_2} \oplus s_{t+\tau_3} \oplus s_{t+\tau_4} \oplus s_{t+\tau_5} \oplus s_{t+\tau_6} = 0$$

$$\tau_1 = 0, \tau_2 = 11, \tau_3 = 13, \tau_4 = 4 \cdot 2^{32} - 4, \tau_5 = 15 \cdot 2^{32} - 4, \tau_6 = 17 \cdot 2^{32} - 4$$

- Assume that we have $U_i \cdot S \oplus W_i \cdot Z = 0$, $i = 1, \dots, 96$.
- Then, $\sum_{t=\tau_1}^{\tau_6} U_i \cdot S_t \oplus W_i \cdot Z_t = \sum_{t=\tau_1}^{\tau_6} W_i \cdot Z_t = 0$ with correlation of c_{sober}^6 .
- Data complexity for distinguisher :

$$\sum_{i=1}^{96} c_{sober,i}^{-12} \approx 96 \times 2^{106} = 2^{99.4}$$

Concluding Remarks

- Combination of two encryption blocks induce multiple linear approximations.
- The Rotation plays an important role to remove the linearity of modular addition.
- Our analysis shows that SOSEMANUK and SOBER-128 have multiple linear approximations with strong correlations, by which the complexity of the attack can be reduced.
Note that SNOW 2.0 has a single strong linear approximation.
- In a similar way, we may analyze other software-oriented stream ciphers such as HC-128, Rabbit or Salsa.

Thank You