On the SOSEMANUK Related Key-IV Sets

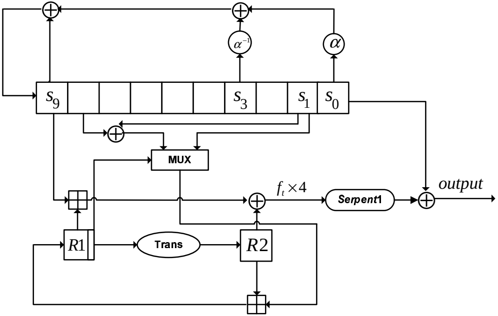
Aleksandar Kircanski and Amr Youssef
Concordia University
Montréal, Québec, Canada

LATINCRYPT 2012, Santiago de Chile, Chile
October 7-10, 2012

Talk outline

- Specification of SOSEMANUK
- A probabilistic argument for the existence of related key-IV pairs
- Inner-state recovery alg. for a set of key-IV pairs
- A distinguisher for another key-IV pair set
- Conclusions

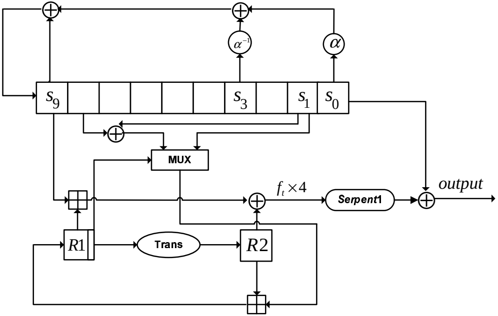
The SOSEMANUK stream cipher



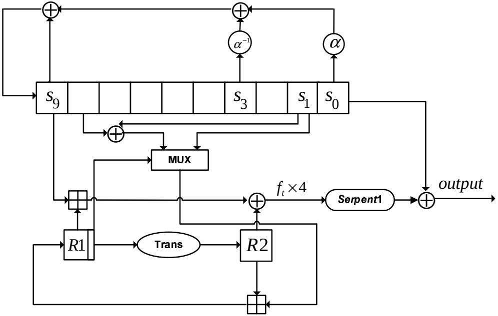
- eStream software portfolio member
- SOSEMANUK: a strengthened SNOW construction
- Components: LFSR and FSM (10 + 2 32-bit words)

  $(s_t, \ldots s_{t+9}, R1_t, R2_t)$, 384 bits altogether

The SOSEMANUK stream cipher



- LFSR taps: s_0 $(\times\alpha)$, s_3 $(\times\alpha^{-1})$ and s_9
- $GF(2^{32})$ constructed so that $\times$ and $/$ by α is:
  - $\ll 8$, then $\oplus$ with a 32-bit mask depending on the MSB
  - $\gg 8$, then $\oplus$ with a 32-bit mask depending on the LSB

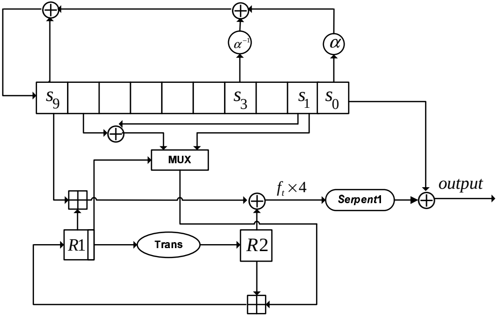


- FSM (after expanding mux):

$$R1_{t+1} = \begin{cases} R2_t \boxplus s_{t+1} & if\, lsb(R1_t) = 0 \\ R2_t \boxplus (s_{t+1} \oplus s_{t+8}) & if\, lsb(R1_t) = 1 \end{cases}$$

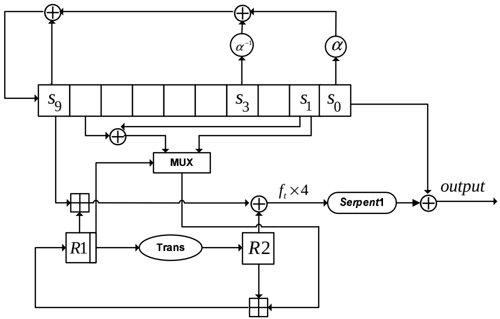
$R2_{t+1} = Trans(R1_t)$, with $Trans(x) = (M \times x) \lll 7$

The SOSEMANUK stream cipher



- SOSEMANUK iteration: update FSM, then LFSR
- Unlike SNOW 2.0/SNOW 3G/ZUC, SOSEMANUK has a *delay mechanism* (bitslice, defend against GD attacks)
- Every 4 steps, a 128-bit word is produced

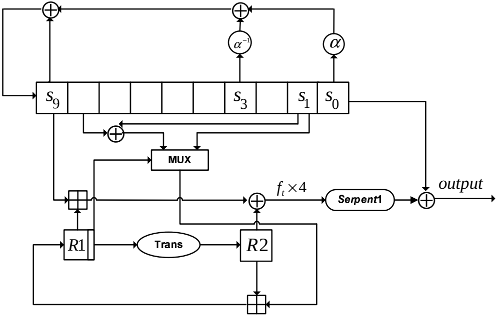
The SOSEMANUK stream cipher



Delay mechanism:

- Each FSM/LFSR step, one 32-bit f_t word is generated
- Instead of outputting the word, save it

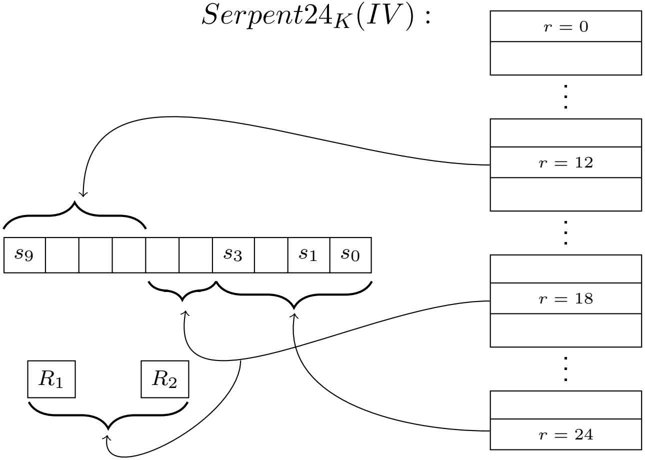
The SOSEMANUK stream cipher



- Every 4 steps, apply Serpent S-box in bitslice mode to $(f_t, f_{t+1}, f_{t+2}, f_{t+3})$
- Add to $(s_t, s_{t+1}, s_{t+2}, s_{t+3})$ and send to output
- Makes GD attacks more difficult (at least 4 LFSR words have to be guessed at the start)

Sosemanuk initialization procedure

$Serpent24_K(IV)$:



Probabilistic argument for related-key existence

A particularity of SOSEMANUK

- Small inner state/key-IV size ratio
- 384 bits vs. 256 bits
- Design goal, fit registers in the processor's cache

It follows as a simple fact that if we..

- Pick a relation between starting SOSEMANUK states
- Birthday paradox: corresponding key-IV pairs
- Even key-IV pairs that produce identical inner states are expected to exist
  - $\#$ of key-IV pairs: $\binom{2^{256}}{2} \times 2^{-384} \approx 2^{127}$
- Birthday-type argument similar for any other state relation

Probabilistic argument for related-key existence

In this work:

- Are there exist inner state relations that allow inner state recovery or distinguishing?

Relation	#	Dist.	Recovery
Slid pairs $d = 1$	2^{128}	$< 2^8$ words	2^{32} time, $2^{23.8}$ space
Slid pairs $d = 2$	2^{128}	$< 2^8$ words	$2^{47.7}$ time, $2^{23.8}$ space
Zero-diff LFSR	2^{192}	2^{52} words	-

- Particular examples appear difficult to find
  - Related to attacking 12-round Serpent in hash mode
- The existence of such pairs is shown probabilistically (birthday paradox, assuming randomness of 6-round Serpent cascades) and their expected sizes are calculated

Inner state recovery for slid pairs

How to recover the inner state for the set of *slid states*? Recall that every SOSEMANUK iteration contains 4 *steps*.

Definition: Slid states are SOSEMANUK inner states on d steps away, where d is *not* a multiple of 4.

Suppose that the keystream output generated by slid states, e.g, at times

$$t, t+4, t+8, \dots$$
$$t+1, t+5, t+9, \dots$$

is available. How to use such outputs for inner state recovery in practical complexity?

Inner state recovery for slid pairs

Consider a 128-bit SOSEMANUK keystream word.

- Pick out the 4 bits corresponding to one S-box application
- This will correspond to f_3^i, f_2^i, f_1^i, f_0^i and $s_3^i, s_2^i, s_1^i, s_0^i$
- Write down the equation in inner state words:
  $S(f_3^i f_2^i f_1^i f_0^i) \oplus s_3^i s_2^i s_1^i s_0^i = w_{0,i}$

Do this for the first 4 kst. words (0, 4, 8 and 12):

$$S(f_3^i f_2^i f_1^i f_0^i) \oplus s_3^i s_2^i s_1^i s_0^i = w_{0,i}$$
$$S(f_7^i f_6^i f_5^i f_4^i) \oplus s_7^i s_6^i s_5^i s_4^i = w_{4,i}$$
$$S(f_{11}^i f_{10}^i f_9^i f_8^i) \oplus s_{11}^i s_{10}^i s_9^i s_8^i = w_{8,i}$$
$$S(f_{15}^i f_{14}^i f_{13}^i f_{12}^i) \oplus s_{15}^i s_{14}^i s_{13}^i s_{12}^i = w_{12,i}$$

- "Normal feature", no eq's include the same bit-variable

Consider a 128-bit Sosemanuk keystream word.

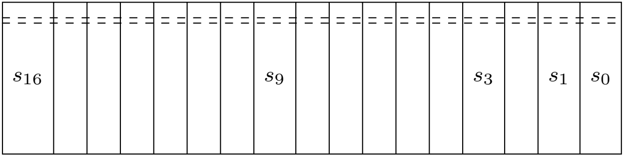
- Pick out the 4 bits corresponding to one S-box application
- This will correspond to f_3^i, f_2^i, f_1^i, f_0^i and $s_3^i, s_2^i, s_1^i, s_0^i$
- Write down the equation in inner state words: $S(f_3^i f_2^i f_1^i f_0^i) \oplus s_3^i s_2^i s_1^i s_0^i = w_{0,i}$

Do this for the first 4 kst. words (0, 4, 8 and 12):

$$
\begin{aligned}
&S(f_3^i f_2^i f_1^i f_0^i) \oplus s_3^i s_2^i s_1^i s_0^i = w_{0,i}, && S(f_4^i f_3^i f_2^i f_1^i) \oplus s_4^i s_3^i s_2^i s_1^i = w_{1,i}\\
&S(f_7^i f_6^i f_5^i f_4^i) \oplus s_7^i s_6^i s_5^i s_4^i = w_{4,i}, && S(f_8^i f_7^i f_6^i f_5^i) \oplus s_8^i s_7^i s_6^i s_5^i = w_{5,i}\\
&S(f_{11}^i f_{10}^i f_9^i f_8^i) \oplus s_{11}^i s_{10}^i s_9^i s_8^i = w_{8,i}, && S(f_{12}^i f_{11}^i f_{10}^i f_9^i) \oplus s_{12}^i s_{11}^i s_{10}^i s_9^i = w_{9,i}\\
&S(f_{15}^i f_{14}^i f_{13}^i f_{12}^i) \oplus s_{15}^i s_{14}^i s_{13}^i s_{12}^i = w_{12,i}, && S(f_{16}^i f_{15}^i f_{14}^i f_{13}^i) \oplus s_{16}^i s_{15}^i s_{14}^i s_{13}^i = w_{13,i}
\end{aligned}
$$

- Introduce the eq's due to slid state: dependencies

Inner state recovery for slid pairs



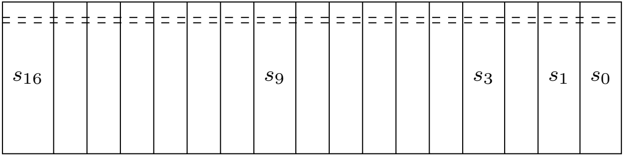
What do we get from these equations?

- The LFSR specified "bitwise", i.e. candidates for $(s_{16}^i, \ldots, s_0^i)$ for $0 \leq i \leq 31$
- Denote by S_i the respective sets. Then $E(|S_i|)$ is $2^{3.34}$

It should be mentioned that

- Equations involve $(s_{16}, \ldots s_0)$ rather than $(s_9, \ldots s_0)$
- We have the extended LFSR specified "bit-wise"
- Goal: use this redudancy to prune the incorrect LFSRs

Inner state recovery for slid pairs



- Fixing an element from some S_i adds linear equations to the system.
- After fixing elements from some number of sets S_i, $0 \leq i \leq 31$, the system becomes contradictory
- Goal: minimize the number of S_i sets to be chosen from
- Q: which S_i to choose so that a contradiction will occur as early as possible?
- A: Use the specifics of multiplication by α and α^{-1}

Inner state recovery for slid pairs

Represent:

$$\alpha x = (x \ll 8) \oplus T_\alpha(x \gg 24),$$
$$\alpha^{-1} x = (x \gg 8) \oplus T_{\alpha^{-1}}(x \& 0xFF)$$

Then, when the LFSR update

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

is presented bitwise:

$$s^i_{t+10} = s^i_{t+9} \oplus s^{i+8}_{t+3} \oplus T^i_{\alpha^{-1}}(s_{t+3,0}) \oplus s^{i-8}_t \oplus T^i_\alpha(s_{t,3})$$

For bit i, we have bits $\{i, i-8, i+8\}$ participating.

For example, if $i = 0$ (choosing the candidate from S_0)

$$s^0_{t+10} \quad = s^0_{t+9} \oplus s^8_{t+3} \oplus T^0(s_{t+3,0}s_{t,3})$$

(here s^0_{t+10} and s^0_{t+9} are known). For the goal of causing a contradiction, we choose now a candidate from S_8

$$s^8_{t+10} \quad = s^8_{t+9} \oplus s^{16}_{t+3} \oplus s^0_t \oplus T^8(s_{t+3,0}s_{t,3})$$

and then similarly, candidates from S_{16} and S_{24}.

$$\begin{aligned} s^{16}_{t+10} &= s^{16}_{t+9} \oplus s^{24}_{t+3} \oplus s^8_t \oplus T^{16}(s_{t+3,0}s_{t,3}) \\ s^{24}_{t+10} &= s^{24}_{t+9} \oplus s^{16}_t \oplus T^{24}(s_{t+3,0}s_{t,3}) \end{aligned}$$

Inner state recovery for slid pairs

We obtained the system

$$
\begin{aligned}
s^0_{t+10} &= s^0_{t+9} \oplus s^8_{t+3} \oplus T^0(s_{t+3,0}s_{t,3}) \\
s^8_{t+10} &= s^8_{t+9} \oplus s^{16}_{t+3} \oplus s^0_t \oplus T^8(s_{t+3,0}s_{t,3}) \\
s^{16}_{t+10} &= s^{16}_{t+9} \oplus s^{24}_{t+3} \oplus s^8_t \oplus T^{16}(s_{t+3,0}s_{t,3}) \\
s^{24}_{t+10} &= s^{24}_{t+9} \oplus s^{16}_t \oplus T^{24}(s_{t+3,0}s_{t,3})
\end{aligned}
$$

which constraints the input-output values for the T table for $t = 0, \ldots 6$ since all the bits outside the T table are known.

- After specifying 3 such systems, the constrains overdefine the T table
- If any of the systems is contradictory, discard the guess
- Thus, contradictions will be spotted after going through candidates from around $3 \times 4 = 12$ (instead of all 32) S_i sets

Inner state recovery for slid pairs

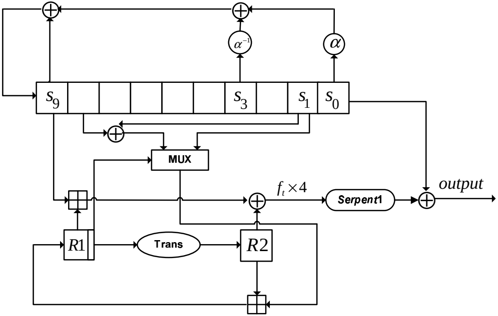
What is the cost of going through candidates of $3 \times 4 = 12$ S_i sets? The attacker can *choose* which sets to pick candidates from, so he chooses the smallest ones.

Inner state recovery:

- Testing whether $2^{9.517+10.602+11.372} = 2^{31.491}$ T table constraints are contradictory
- Storage space: $2^{23.8}$ bits (yes/no information on the T table constraints)
- 4×2 keysteram words

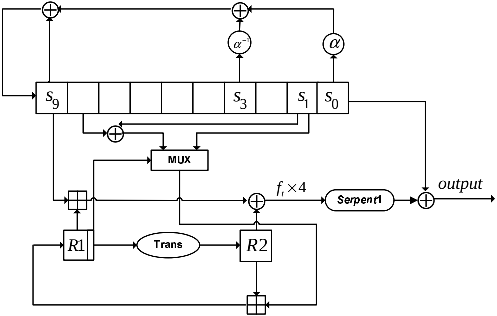
The attack was implemented on a PC (2.4 GHz AMD) and recovered the inner state in less than a day.

Distinguisher: $< 2^8$ keystream words (main idea: slid pairs reuse the inner state bits which introuces correlation).



- Expected size: 2^{192} key-IV pairs
- Zero-difference preserved during iterations
- We get a distinguisher requiring 2^{52} keystream outputs

Distinguisher for zero-LFSR-difference instances

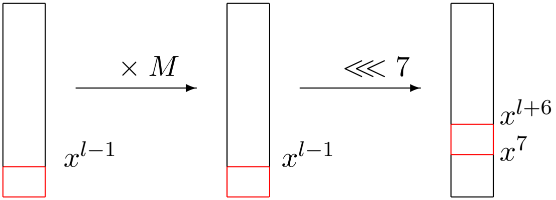


Reasonable to try:

- Wait for zero-difference on some bits in $R1_t$, $R2_t$
- $Trans()$ preserves some zero-diff.
- Expect consecutive zero-diff. in the output

Property of $Trans()$ function

$$Trans(x) = (M \times x) \lll 7$$



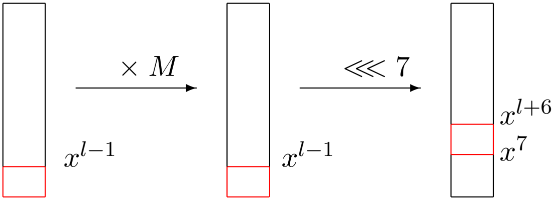
Problem: rotation of zero differences due to $Trans()$ f.

- Due to bit-slicing, the S-boxes pick up bits with non-zero differences

Denote by $(0, 0)$ the event that $R1$ and $R2$ have zero-diff. on bits $0..l-1$ at some step t.

Property of $Trans()$ function

$$Trans(x) = (M \times x) \lll 7$$



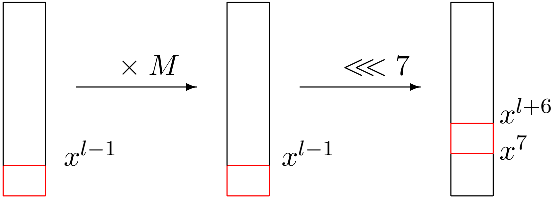
Let $P[(0,0) \to (0,0)]$ probability that the event will be preserved in the next step.

- How does $P[(0,0) \to (0,0)]$ depend on l?

The probability does not decrease regularly as l grows

Property of $Trans()$ function

$$Trans(x) = (M \times x) \lll 7$$



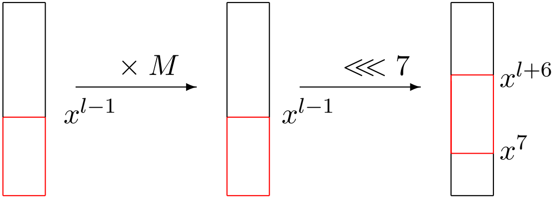
What is the $P[(0,0) \rightarrow (0,0)]$? The property is preserved in $R1$ (since $lsb(R1) = lsb(R1'))$. As for $R2$, we have

Observation:

$$P[(0,0) \rightarrow (0,0)] = \begin{cases} 2^{-l} & \text{if } l \leq 7 \\ 2^{-7} & \text{if } l > 7 \end{cases}$$

Property of $Trans()$ function

$$Trans(x) = (M \times x) \lll 7$$



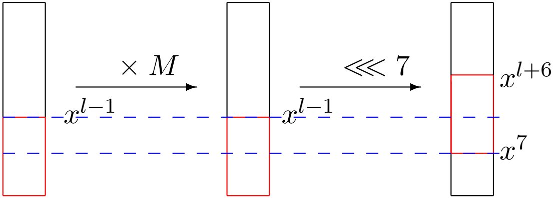
What is the $P[(0,0) \to (0,0)]$? The property is preserved in $R1$ (since $lsb(R1) = lsb(R1'))$. As for $R2$, we have

Observation:

$$P[(0,0) \to (0,0)] = \begin{cases} 2^{-l} & \text{if } \; l \leq 7 \\ 2^{-7} & \text{if } \; l > 7 \end{cases}$$

Property of $Trans()$ function

$$Trans(x) = (M \times x) \lll 7$$



What is the $P[(0,0) \rightarrow (0,0)]$? The property is preserved in $R1$ (since $lsb(R1) = lsb(R1'))$. As for $R2$, we have

Observation:

$$P[(0,0) \rightarrow (0,0)] = \begin{cases} 2^{-l} & \text{if } l \leq 7 \\ 2^{-7} & \text{if } l > 7 \end{cases}$$

Property of $Trans()$ function

Putting $l = 14$ turns out to work well for the distinguisher:

- Step t: $p = 2^{-14\times 2=-28}$
- Step $t+1$: $p = 2^{-7}$
- Step $t+2$: $p = 2^{-7}$
- Step $t+3$: $p = 2^{-7}$

Then,

- All four conditions $2^{-28-3\times 7} = 2^{-49}$
- 56-bit zero-diff. occurs more often than it should
- Random: $2^{-14\times 4=-56}$, SOSEMANUK: 2^{-49}
- Distinguisher requiring 2^{52} words

Conclusions and future work

Conclusions:

- We provided a "roadmap" for SOSEMANUK related keys
- Classes of key-IV pairs that allow inner state recovery and distinguishing identified
- No practical impact on the security of SOSEMANUK

Future work:

- Find particular (K, IV) (K', IV') pairs that yield correlated SOSEMANUK keystreams
- The problem related to finding collisions and near-collisions of 12-round cascades of Serpent in hashing mode