Improved Fast Correlation Attacks on the Sosemanuk Stream Cipher

Bin Zhang, Ruitao Liu, Xinxin Gong and Lin Jiao

Chinese Academy of Sciences & State Key Laboratory of Cryptology

2023-12-03

Outline

Background

- The European eSTREAM project, a multi-year effort running from 2004 to 2008, has a sustaining effect on the design and analysis of modern stream ciphers, which provides some typical design paradigms
- NFSR(nonlinear feedback shift register)-based: Grain v1 and Trivium
- ARX-based: Salsa20/12, Rabbit
- LFSR+FSM: Sosemanuk which combines SNOW 2.0 with Serpent block cipher
- The conclusion from the eSTREAM final report on Sosemanuk is that it offers a very considerable margin for security as well as reasonable performance trade-offs

Motivation

- Correlation attack is a classical cryptanalysis method for LFSR-based stream ciphers
- Fast correlation attacks (FCA) speeds up the exhaustive search of the involved LFSR state by some decoding algorithm from coding theory
- Sosemanuk follows the LFSR + FSM design strategy of the SNOW family of stream ciphers, thus a natural target to try FCA
- Huge efforts have been made to cryptanalyze Sosemanuk since 2004, e.g., Guess-and-determine attacks, Time/Memory/Data Tradeoff attacks, Correlation attacks and linear cryptanalysis, $\rightarrow$ almost 20 years of work

Motivation

- Given the correlation of the found linear approximation $p = \frac{1}{2} + 2^{-21.41}$ and the number of binary variables $l = 320$, the information-theoretical bound of the complexity for any attack aiming at the 320-bit entropy in the LFSR initial state is

$$\frac{4 \cdot l \cdot \ln 2}{1 - H(p)} \doteq 2^{51.08},$$

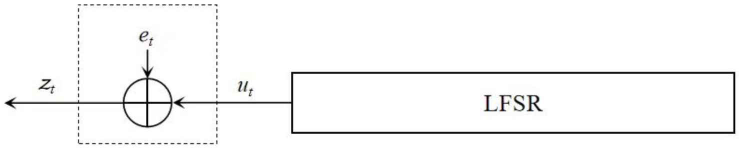
where $H(x) = -x \cdot \log(x) - (1 - x) \cdot \log(1 - x)$ is the binary entropy function. The gap between the theoretical value $2^{51.08}$ and the cryptanalysis practice on Sosemanuk 2^{154} is huge and a natural problem is whether we could narrow the gap by some improved algorithm ?

Our Contributions

- An improved algorithm for fast correlation attacks on stream ciphers is proposed with new algorithmic procedures
- Make the distribution transform to convert the distribution of the LFSR initial state from uniform to the same biased distribution as that of the found linear approximation by Gauss elimination, together with BKW reduction to substitute the previous algorithm based on generalized birthday problem and code-reduction to reduce the secret dimension
- Study the security of Sosemanuk by the newly developed algorithm, launch a new state recovery attack with a time complexity of around $2^{134.8}$, which is about 2^{20} times faster than the best previously known results at Asiacrypt 2008 and the fastest attack among all known attacks on Sosemanuk so far

Correlation Attacks

- Correlation attacks usually exploit the correlation between the keystream and the linear combinations of several LFSR sequences.
- The core problem is regarded as the decoding of a low-rate linear block code transmitted through a binary channel, usually symmetric (BSC), as depicted in the Figure below



Correlation Attacks

- The starting point is to look at the generator matrix $\mathbf{G}$ of the LFSR $[N, k]$ linear code, where N is the length of the codeword and k is the dimension of the information, i.e., the LFSR length. Let $\mathbf{u} = (u_0, u_1, \cdots, u_{N-1})$, then we have

$$\begin{aligned} \mathbf{u} &= (u_0, u_1, \cdots, u_{k-1}) \cdot \mathbf{G} \\ &= (u_0, u_1, \cdots, u_{k-1}) \cdot [\mathbf{g}_0, \mathbf{g}_1, \cdots, \mathbf{g}_{N-1}] \end{aligned} \tag{1}$$

Correlation Attacks

- Regard the column vectors $\mathbf{g}_i$ as random vectors over $\mathrm{GF}(2)^k$, construct the parity-checks by

$$\bigoplus_{j=0}^{v-1} u_{i_j} = (u_0, u_1, \cdots, u_{k-1}) \cdot \bigoplus_{j=0}^{v-1} \mathbf{g}_{i_j} = \bigoplus_{j=0}^{k_1-1} c_j u_j,$$

- Note that the above procedure is similar to the BKW collision in LPN solvers, and the new noise variable is $e = \bigoplus_{j=0}^{v-1} e_{i_j}$ with a bias ϵ^v when $\Pr\{e_{i_j} = 0\} = \frac{1}{2}(1+\epsilon)$

Linear Cryptanalysis

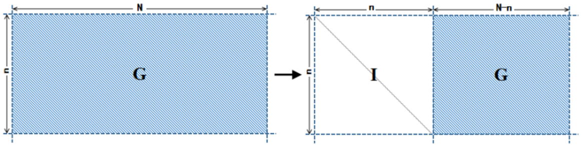
- Linear cryptanalysis (LC) is a known-plaintext attack proposed by Matsui in 1993 to break DES, but can be seen as a more generic and closely related method to correlation attacks in symmetric key cryptanalysis
- LC looks for bitwise linear approximations of the nonlinear components with a deviation from $\frac{1}{2}$ as much as possible and connect them together to build some probabilistic linear equations between several input/output bits and the key material
- The key recovery routine in LC is unnecessarily be confined to the above algorithmic routine. In general, we can loose the restrictions and to establish a probabilistic linear system with the involved key as variables

Solving LPN: Informal Definition

- The LPN problem is believed to be hard even given quantum computers, though no formal reduction from hard lattice problems exists unlike the case of Learning with Errors (LWE) problem
- The search LPN problem is as follows.
- **Definition** : For a secret vector $\mathbf{s} \in \mathsf{GF}(2)^k$, the adversary is given many pairs of the form $(g, \langle s, g\rangle \oplus e)$, where $g \in \mathsf{GF}(2)^k$ is randomly generated and $e \leftarrow Ber_\eta$, the task of the adversary is to recover $\mathbf{s}$ from the many given pairs.
- FCA, LC and LPN solvers share the same mathematical model, and thus we could consider and exploit the links to get improved results

Solving LPN: Gauss Elimination

- **Gauss Elimination** : As in the LPN solvers, Gauss elimination can be adopted here to transform the distribution of the secret $(u_0, u_1, \cdots, u_{k-1})$ into the same distribution as that of the noise variables e_i.



Solving LPN: BKW Collision-reduction

- **BKW Collision-reduction** : The BKW collision-reduction algorithm is shown in Algorithm 1 below

Algorithm 1 BKW Reduction

```
Input:   The matrix $\mathbf{G}_0 = \mathbf{G} = [\mathrm{g}_0, \mathrm{g}_1, \ldots, \mathrm{g}_{n-1}]$, the parameters $t$ and $b$
Online: Reduction phase in BKW algorithm
 1: for $i = 1$ to $t$ do
 2:    Partition the columns of $\mathbf{G}_{i-1}$ according to the last $b \cdot i$ bits
       Form pairs of the columns in each partition to obtain $\mathbf{G}_i$
 2a: LF1. Partition $\mathbf{G}_{i-1} = V_0 \cup V_1 \cup \cdots \cup V_{2^b-1}$.
          Randomly choose $\mathbf{v}^* \in V_j$ as the representative.
          For $\mathbf{v} \in V_j, \mathbf{v} \neq \mathbf{v}^*$, $\mathbf{G}_i = \mathbf{G}_i \cup (\mathbf{v} \oplus \mathbf{v}^*)$
 2b: LF2. Partition $\mathbf{G}_{i-1} = V_0 \cup V_1 \cup \cdots \cup V_{2^b-1}$.
          For each pair $(\mathbf{v}, \mathbf{v}') \in V_j, \mathbf{v} \neq \mathbf{v}'$, $\mathbf{G}_i = \mathbf{G}_i \cup (\mathbf{v} \oplus \mathbf{v}')$
Output: The matrix $\mathbf{G}_t$
```

Solving LPN: Code-reduction

- **Code-reduction** : Another algorithmic procedure is the code-reduction that reduces the dimension of the secret information without having the piling-up lemma penalty of the folded noise.

$$\begin{aligned}\mathbf{G}_t &= [I^{(t)}_{k_1\times k_1}, A^{(t)}_{k_1\times(N_1-k_1)}] = [I^{(t)}_{k_1\times k_1}, \mathrm{g}^{(t)}_{k_1+1}, \ldots, \mathrm{g}^{(t)}_{N_1-1}] \\ &= [I^{(t)}_{k_1\times k_1}, c_{k_1+1}\oplus \bar{e}_{k_1+1}, \ldots, c_{N_1-1}\oplus \bar{e}_{N_1-1}],\end{aligned} \tag{2}$$

- We have

$$\begin{aligned}\hat{\mathbf{z}} &= \hat{\mathbf{u}}\cdot\mathbf{G}\oplus\mathbf{e} \\ &= \hat{\mathbf{u}}\cdot[I_{k_1\times k_1}, c_{k_1+1}\oplus\bar{e}_{k_1+1}, \ldots, c_{N_1-1}\oplus\bar{e}_{N_1-1}]\oplus\mathbf{e},\end{aligned}$$

Construct parity-check equations

- Look for some t-tuple column vectors $(\mathbf{g}_{i_0}, \mathbf{g}_{i_1}, \ldots, \mathbf{g}_{i_{t-1}})$ such that

$$\bigoplus_{j=0}^{t-1} \hat{u}_{i_j} = (\hat{u}_0, \hat{u}_1, \cdots, \hat{u}_{k-1}) \cdot \bigoplus_{j=0}^{t-1} \mathbf{g}_{i_j} = \bigoplus_{j=0}^{k_1-1} c_j \hat{u}_j \oplus \bigoplus_{j=v}^{1} c_{k-j} \hat{u}_{k-j},$$

where $\bigoplus_{j=0}^{t-1} \mathbf{g}_{i_j} = (c_0, c_1, \cdots, c_{k_1-1}, 0, \cdots, 0, \underbrace{*, \ldots, *}_{v})^t$ with $*$ being an arbitrary value in GF(2) and $k_1 < k - v$.

Construct parity-check equations

- Accordingly, we have

$$\begin{aligned}\bigoplus_{j=0}^{t-1} \hat{z}_{i_j} &= (\hat{u}_0, \hat{u}_1, \cdots, \hat{u}_{k-1}) \cdot \bigoplus_{j=0}^{t-1} \mathbf{g}_{i_j} \oplus \bigoplus_{j=0}^{t-1} \mathbf{e}_{i_j} \\ &= \bigoplus_{j=0}^{k_1-1} c_j \hat{u}_j \oplus \bigoplus_{j=v}^{1} c_{k-j} \hat{u}_{k-j} \oplus \bigoplus_{j=0}^{t-1} \mathbf{e}_{i_j},\end{aligned}$$

  where Pr$\{\bigoplus_{j=0}^{t-1} \mathbf{e}_{i_j} = 0\} = \frac{1}{2}(1 + \epsilon^t)$.

- To simplify the notations, let $\mathbf{z}'_i = \bigoplus_{j=0}^{t-1} \hat{z}_{i_j} \oplus \bigoplus_{j=v}^{1} c_{k-j} \hat{u}_{k-j}$, $\mathbf{g}'_i = \bigoplus_{j=0}^{t-1} \mathbf{g}_{i_j}$ the truncated $k_1 \times 1$ column vector and $\mathbf{e}'_i = \bigoplus_{j=0}^{t-1} \mathbf{e}_{i_j}$ and taking the form of $\mathbf{g}'_i$ into account, we have

$$\mathbf{z}'_i = (\hat{u}_0, \hat{u}_1, \cdots, \hat{u}_{k_1-1}) \cdot \mathbf{g}'_i \oplus \mathbf{e}'_i,$$

Code reduction

- Our aim is to construct a $[k_1, k_c]$ linear code $\mathcal{K}$ with covering radius d_c to regroup the columns in $\mathbf{G}'$, i.e., express $\mathbf{g}'_i = c_i \oplus \bar{e}_i$ where $c_i \in \mathcal{K}$ is the nearest codeword to the random column vector $\mathbf{g}'_i$

Code reduction

- It is well-known that there is a very limited types of perfect codes in binary domain, which is listed in the following table

Table: All the binary perfect codes with their codeword/information length and the covering radius

Code	Codeword	Information	Covering radius
`Hamming`	$2^i - 1$	$2^i - i - 1$	1
`Repetition`	$2i + 1$	1	i
`Golay`	23	12	3
$\{0\}^i$	i	0	i
$\{0,1\}^i$	i	i	0

Code reduction

Algorithm 3 Computing Pr$\{\bar{e}^t_{i,j} = 1\}$ for a fixed position t

```
Parameter: $[n_j, k'_j, d_j]$ code $\mathcal{C}_j$ with its parity-check matrix $\mathbf{H}$, $t$
 1:  Initialize the array $A[n_j] = \{0\}$
 2:  for each $0 \le x \le 2^{n_j} - 1$ do
 3:      compute the syndrome $\mathbf{H} \cdot x^t$
 4:      decode $x$ into the nearest codeword $\mathbf{c}_x$ by syndrome-decoding
 5:      $A[x] = x \oplus c_x$
 6:  Initialize a counter $c = 0$
 7:  for each $0 \le x \le 2^{n_j} - 1$ do
 8:      if $A[x]_t = 1$ then $c \leftarrow c + 1$
Output: Pr$\{\bar{e}^t_{i,j} = 1\} = \frac{c}{2^{n_j}}$
```

Code reduction

- We found that for a $[2^i - 1, 2^i - i - 1, 3]$ Hamming code, Pr$\{\bar{e}^t_{i,j} = 1\} = \frac{1}{2^i}$ for $1 \leq t \leq 2^i - 1$; for the $[23, 12, 7]$ Golay code and each t-th coordinate in the error vector, Pr$\{\bar{e}^t_{i,j} = 1\} = \frac{127}{1024}$; for a repetition code, the relevant value shown in the following table

Table: Coordinate distribution in the error vector of the Repetition Codes

Repetition Codes							
i	2	3	4	5	6	7	8
Pr$\{\bar{e}^t_{i,j} = 1\}$	$\frac{5}{16}$	$\frac{11}{32}$	$\frac{95}{256}$	$\frac{193}{512}$	$\frac{793}{2048}$	$\frac{1619}{4096}$	$\frac{26333}{65536}$

Code reduction

- The optimal configuration of the direct sum is equivalent to the maximized solution of

$$\max \sum_{\substack{i:2^i-1\le k_1\\ 2^i-i-1\le k_c}} \lambda^i_{\mathtt{Hamming}} \log \epsilon^i_{\mathtt{Hamming}} + \sum_{i:2i+1\le k_1} \lambda^i_{\mathtt{Repet}} \log \epsilon^i_{\mathtt{Repet}} + \lambda_{\mathtt{Golay}} \log \epsilon_{\mathtt{Golay}} + \lambda_{\{\mathtt{0}\}^{\mathtt{i}}} \log \epsilon_{\{\mathtt{0}\}^{\mathtt{i}}} + \lambda_{\{\mathtt{0,1}\}^{\mathtt{i}}} \log \epsilon_{\{\mathtt{0,1}\}^{\mathtt{i}}}$$

- under the constraint of

$$\begin{cases} \sum\limits_{\substack{i:2^i-1\le k_1\\ 2^i-i-1\le k_c}} (2^i-1)\lambda^i_{\mathtt{Hamming}} + \sum\limits_{i:2i+1\le k_1} (2i+1)\lambda^i_{\mathtt{Repet}} + 23\lambda_{\mathtt{Golay}} + \lambda_{\{\mathtt{0}\}^{\mathtt{i}}} + \lambda_{\{\mathtt{0,1}\}^{\mathtt{i}}} = k_1 \\ \sum\limits_{\substack{i:2^i-1\le k_1\\ 2^i-i-1\le k_c}} (2^i-i-1)\lambda^i_{\mathtt{Hamming}} + \sum\limits_{i:2i+1\le k_1} \lambda^i_{\mathtt{Repet}} + 12\lambda_{\mathtt{Golay}} + \lambda_{\{\mathtt{0,1}\}^{\mathtt{i}}} = k_c \end{cases}$$

New Framework of FCA

Algorithm 2 Improved FCA

```
Input:  A keystream $\mathbf{z} = (z_0, z_1, \ldots, z_{k-1}, z_k, \ldots, z_{N-1})$
Online: Recover (partial) LFSR initial state which is consistent with $\mathbf{z}$
 1: Apply Gauss elimination to derive the equivalent model
 2: Select a dimension reduction strategy from Section 4 and
        build a new linear code accordingly
 3: Decode the new code via FWHT
Output: The partial LFSR initial state involved in the new code
```

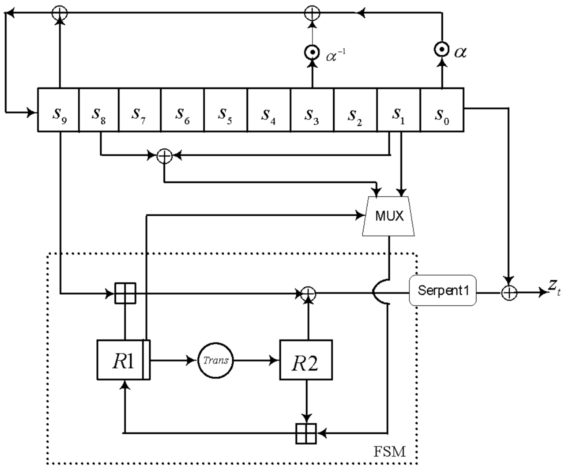
Complexity Analysis

Theorem

Let ε_g, ε_c be the correlations introduced in the BKW collision-reduction and the code-reduction procedure, respectively. Denote $\varepsilon_f = \varepsilon_g \cdot \varepsilon_c$, then Algorithm 2 has the following complexities.

- *The time complexity of one-round BKW collision-reduction is $\mathcal{O}(N + (\frac{4 \cdot k_1 \cdot \ln 2}{\varepsilon^2}) \cdot (k + 1 - b))$.*
- *The online decoding time complexity for recovering the k_c-bit part of the LFSR initial state is $\mathcal{O}\left(\frac{4 \cdot k_c \cdot \ln 2}{\varepsilon_f^2} + k_c 2^{k_c}\right)$. The other bits in the LFSR initial state can be recovered similarly with a much lower complexity.*
- *The required length N of the observed keystream segment for Algorithm 2 to have a non-negligible success probability higher than 0.5 has to satisfy the relation $m_v = \binom{N/2^b}{2} \cdot 2^b = \frac{4 \cdot k_c \cdot \ln 2}{\varepsilon_f^2}$.*

Brief Description of the Sosemanuk Stream Cipher



Application to the Sosemanuk Stream Cipher

- The FSM state at time t is denoted by $(R1_t, R2_t)$. The FSM is updated as follows.

$$R1_t = R2_{t-1} \boxplus (s_{t+1} \oplus \mathtt{lsb}(\mathtt{R1_{t-1}})\mathtt{s_{t+8}}),$$
$$R2_t = \mathtt{Trans}(\mathtt{R1_{t-1}}) = (\mathtt{M} \times \mathtt{R1_{t-1}})^{\lll 7},$$

- It is a 2-round approximation of the FSM i.e.

$$\langle \Gamma, f_t \rangle \oplus \langle \Gamma, R2_t \rangle = \langle \Gamma, s_{t+9} \rangle \oplus \langle \Gamma, R2_{t+1} \rangle,$$
$$\langle \Lambda, f_{t+1} \rangle \oplus \langle \Lambda, R2_{t+1} \rangle = \langle \Lambda, s_{t+10} \rangle \oplus \langle \Lambda, R2_t \rangle \oplus \langle \Lambda, s_{t+2} \rangle \oplus \langle \Lambda a_t, s_{t+9} \rangle,$$

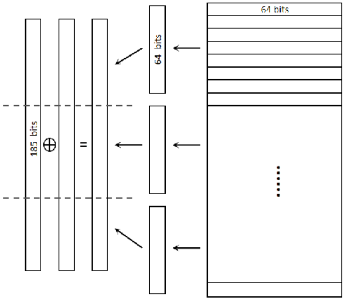
  where $a_t = \mathtt{lsb}(R1_t)$ and $\Gamma, \Lambda \in \mathrm{GF}(2)^{32}$ are the corresponding linear masks.

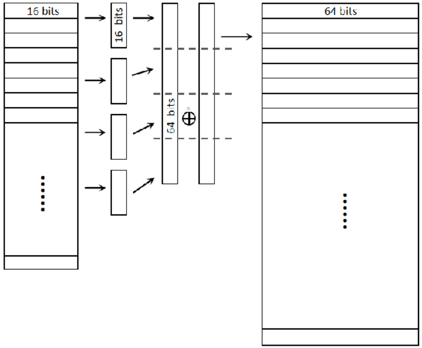
- When $\Gamma = \Lambda =$ 0x03004001, the best found linear approximation is of the correlation $2^{-21.41}$,

$$\langle \Gamma, z_t \rangle \oplus \langle \Gamma, z_{t+3} \rangle = \langle \Gamma, s_t \rangle \oplus \langle \Gamma, s_{t+2} \rangle \oplus \langle \Gamma, s_{t+3} \rangle \oplus \langle \Gamma, s_{t+10} \rangle.$$

Application to Sosemanuk

- Adopt the table-based strategy to save the sum of 2 columns, each of which has f dimensions. The n-dimensional xor is divided into $\lceil \frac{n}{f} \rceil$ parts, and store a table of all possible xors of the f-dimensional vectors and read it up to $\lceil \frac{n}{f} \rceil$ times.





Application to Sosemanuk

- Construct a covering code with the parameters $n = 185, l = 127$ that have

$$\begin{cases} \sum\limits_{\substack{i:2^i-1\leq 185\\ 2^i-i-1\leq 127}} (2^i-1)\lambda^i_{\mathtt{Hamming}} + \sum\limits_{i:2i+1\leq 185} (2i+1)\lambda^i_{\mathtt{Repet}} + 23\lambda_{\mathtt{Golay}} + \lambda_{\{0\}^{\mathtt{i}}} + \lambda_{\{0,1\}^{\mathtt{i}}} = 185 \\ \sum\limits_{\substack{i:2^i-1\leq 185\\ 2^i-i-1\leq 127}} (2^i-i-1)\lambda^i_{\mathtt{Hamming}} + \sum\limits_{i:2i+1\leq 185} \lambda^i_{\mathtt{Repet}} + 12\lambda_{\mathtt{Golay}} + \lambda_{\{0,1\}^{\mathtt{i}}} = 127 \end{cases}$$

- we can construct a $[185, 127]$ linear code

$$\mathcal{C} = 2 \cdot \mathcal{H}_4 \ \dot{+} \ \mathcal{H}_6 \ \dot{+} \ 4 \cdot \mathcal{G}$$

Application to Sosemanuk

- The bias introduced in the covering code phase is:

$$\begin{aligned}
\varepsilon_c &= \texttt{bias}(15, 11, 1, \tfrac{1}{16})^2 \cdot \texttt{bias}(63, 57, 1, \tfrac{1}{64}) \cdot \texttt{bias}(23, 12, 3, \tfrac{127}{1024})^4 \\
&= \left[\sum_{\substack{0 \le i \le 1 \\ i \,\&\, 0x1 = 1}} \binom{15}{i} \left(\frac{1}{16}\right)^i \cdot \left(\frac{15}{16}\right)^{15-i}\right]^2 \cdot \left[\sum_{\substack{0 \le i \le 1 \\ i \,\&\, 0x1 = 1}} \binom{63}{i} \left(\frac{1}{64}\right)^i \cdot \left(\frac{63}{64}\right)^{63-i}\right] \\
&\quad \cdot \left[\sum_{\substack{0 \le i \le 3 \\ i \,\&\, 0x1 = 1}} \binom{23}{i} \left(\frac{127}{1024}\right)^i \cdot \left(\frac{897}{1024}\right)^{23-i}\right]^4 \\
&= 2^{-19.53215483015486}
\end{aligned}$$

- we get about $\frac{4 \cdot l \cdot ln2}{\varepsilon^2} = 2^{133.16}$ parity check equations and the data complexity is $N = 2^{135}$. By constructing the table, the time complexities of pre-processing phase is $\lceil \frac{n+1}{f} \rceil \cdot \frac{4k \ln 2}{\varepsilon^2} + f2^{f-1}(2^f - 2) = 2^{134.798}$ and the computational complexity of the online decoding phase is $\frac{4l \ln 2}{\varepsilon^2} + l2^l = 2^{134.634}$

Conclusions

- Present a new framework for fast correlation attack on stream ciphers with the integrated algorithmic procedures Gauss elimination, BKW collision reduction and code-reduction
- Improved cryptanalysis results on Sosemanuk with a time complexity of $2^{134.8}$ which is 2^{20} times faster than achievable before by FCA with much less data complexity and is the fastest one among all known attacks so far, which is experimentally verified on the essential covering code step and on the reduced version of Sosemanuk
- Our results indicate that the security margin of Sosemanuk is around 2^{8} for the 128-bit security for the first time, somewhat contrary to the conclusion of the European eSTREAM final report in 2008

Thank you!

Q & A