Non-randomness of eSTREAM Candidates Salsa20 and TSC-4

Simon Fischer, Willi Meier,
Côme Berbain, Jean-François Biasse, and M.J.B. Robshaw

FHNW, Switzerland
FTRD, France

University of Applied Sciences
Northwestern Switzerland

Outline

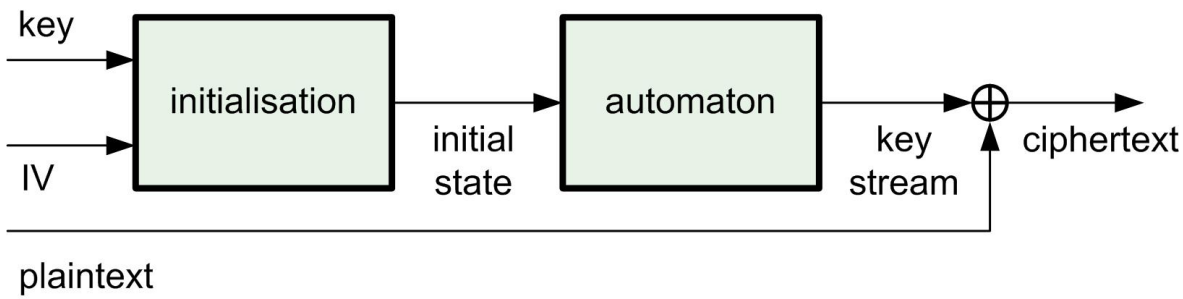
Part 1
Introduction

The general model

Stream Cipher: key, IV, plaintext $\longrightarrow$ ciphertext

Two Components:

Initialisation: key, IV $\longrightarrow$ initial state
Automaton: initial state $\longrightarrow$ keystream



Automaton:

Update: state $\longrightarrow$ new state
Output: state $\longrightarrow$ keystream

Security

Attack on automaton:
Given the keystream $\longrightarrow$ recover state (or distinguish from random).
Methods:
Analysis of update function and output function.
Assumption:
Initial state is uniformly random.

Initialisation function F should have good randomness:
good diffusion of key and IV.

Otherwise:
Chosen IV's or related keys $\longrightarrow$ non-random initial states.
Attack, if output function transfers non-randomness.

Efficiency

Initialisation function should be secure *and* efficient.

General model for initialisation function:

- Fill state with key and IV
- Run a number of simple warm-up rounds
- Warm-up round could be regular update + output feedback

Tradeoff: More rounds $\longrightarrow$ more randomness / reduced efficiency.

Question: What is a good tradeoff?

Non-randomness

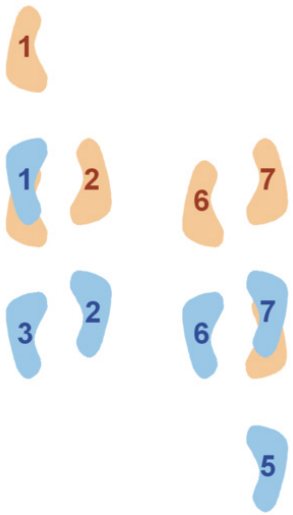
Determine randomness of $F(K, IV)$:

- Fix some bits in input, compute all outputs
- Mask some bits of outputs, compute bias ε_x
- Compute measure $\varepsilon^2 := \sum \varepsilon_x^2$

In practice:

- Choice of fixed input and masked output needs some insight
- Fixed key bits debatable
- Compute only $N \approx 1/\varepsilon^2$ outputs and χ^2 statistics
- χ^2 should be independent of the N inputs

Part 2
Non-randomness in Salsa20



Salsa20

eSTREAM focus candidate, based on hash function.

State: matrix of 16 words of 32 bits.
Update: increment a counter.
Output: modify below-diagonal word first...

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix}$$

Salsa20

eSTREAM focus candidate, based on hash function.

State: matrix of 16 words of 32 bits.
Update: increment a counter.
Output: modify below-diagonal word first...

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix}$$

Salsa20

eSTREAM focus candidate, based on hash function.

State: matrix of 16 words of 32 bits.
Update: increment a counter.
Output: modify below-diagonal word first...

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix}$$

$$+, \lll$$

Salsa20

eSTREAM focus candidate, based on hash function.

State: matrix of 16 words of 32 bits.
Update: increment a counter.
Output: modify below-diagonal word first...

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix}$$

$$\oplus$$

Salsa20

eSTREAM focus candidate, based on hash function.

State: matrix of 16 words of 32 bits.
Update: increment a counter.
Output: modify below-diagonal word first...

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix}$$

$$\oplus$$

- Repeat for all words in columns, then in rows
- 10 rounds columns, 10 rounds rows
- Output the keystream $X^0 + X^{20}$

Initialisation

Initialisation:
Trivial, fill state with (key, counter, nonce) where counter $= 0$

$$X = \begin{pmatrix} \text{const} & \text{key} & \text{key} & \text{key} \\ \text{key} & \text{const} & \text{nonce} & \text{nonce} \\ \text{counter} & \text{counter} & \text{const} & \text{key} \\ \text{key} & \text{key} & \text{key} & \text{const} \end{pmatrix}$$

Alternative interpretation:

- Let $IV =$ (counter, nonce)
- Evolved state corresponds to initial state filled with (K, IV)
- Output function mixes $(K, IV) \rightarrow$ initialisation function $F(K, IV)$

Question: Good diffusion of $F(K, IV)$ after r rounds?

Differential Attack

Differential attacks on r rounds (previous work by Crowley):

$$\begin{array}{rcl} (K, IV) & \rightarrow & F(K, IV) \\ (K', IV') & \rightarrow & F(K', IV') \\ \hline \Delta & \rightarrow & \Delta^r \end{array}$$

Our framework:
Many inputs (K, IV) with fixed $\Delta \rightarrow$ bias in Δ^r with χ^2.

1. Find optimal input-difference Δ
2. Find optimal inputs (K, IV)
3. Find optimal mask in Δ^r

Linear Salsa20

Step 1: Optimal input-difference Δ

Consider linear version of Salsa20 $\rightarrow$ depends only on Δ.
Good approximation, if active words have low weight.

$\Delta_2 = \texttt{0x00000100}$, $\Delta_6 = \texttt{0x00001000}$, and $\Delta_{14} = \texttt{0x80080000}$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix} \xrightarrow{\text{col}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \xrightarrow{\text{row}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 3 & 4 \end{pmatrix} \xrightarrow{\text{col}}$$

$$\begin{pmatrix} 4 & 1 & 3 & 4 \\ 1 & 2 & 4 & 8 \\ 1 & 0 & 7 & 10 \\ 3 & 1 & 3 & 14 \end{pmatrix} \xrightarrow{\text{row}} \begin{pmatrix} 13 & 1 & 6 & 7 \\ 11 & 14 & 5 & 7 \\ 7 & 4 & 14 & 5 \\ 14 & 21 & 18 & 17 \end{pmatrix} \xrightarrow{\text{col}} \begin{pmatrix} 13 & 16 & 17 & 17 \\ 6 & 16 & 19 & 23 \\ 14 & 13 & 18 & 15 \\ 18 & 16 & 15 & 15 \end{pmatrix}$$

Linear Salsa20

Step 1: Optimal input-difference Δ

Consider linear version of Salsa20 $\rightarrow$ depends only on Δ.
Good approximation, if active words have low weight.

$\Delta_2 = \texttt{0x00000100}$, $\Delta_6 = \texttt{0x00001000}$, and $\Delta_{14} = \texttt{0x80080000}$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix} \xrightarrow{\text{col}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \xrightarrow{\text{row}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 3 & 4 \end{pmatrix} \xrightarrow{\text{col}}$$

$$\begin{pmatrix} 4 & 1 & 3 & 4 \\ 1 & 2 & 4 & 8 \\ 1 & 0 & 7 & 10 \\ 3 & 1 & 3 & 14 \end{pmatrix} \xrightarrow{\text{row}} \begin{pmatrix} 13 & 1 & 6 & 7 \\ 11 & 14 & 5 & 7 \\ 7 & 4 & 14 & 5 \\ 14 & 21 & 18 & 17 \end{pmatrix} \xrightarrow{\text{col}} \begin{pmatrix} 13 & 16 & 17 & 17 \\ 6 & 16 & 19 & 23 \\ 14 & 13 & 18 & 15 \\ 18 & 16 & 15 & 15 \end{pmatrix}$$

Differences in True Salsa20

Step 2: Optimal inputs (K, IV)

- Integer addition in true Salsa20 $\longrightarrow \Delta^r$ depends on input
- Conditions on input, such that Salsa20 is close to LinSalsa20
- Previous Δ + simple condition on $IV \longrightarrow$ first round is linear

Step 3: Optimal mask in Δ^r

- Word 9 of Δ^5 is expected to be biased
- Examine each bit with χ^2 test

Experimental results:
Randomly fixed key, weak nonce, incremented counter, fixed Δ.
With $N = 2^{24}$ samples, average $\chi^2 = 383$ instead of $\chi^2 = 1$.

Attacks

Statistical weakness after 5 rounds $\rightarrow$ can be detected 2 rounds later:

- Observe $X + X^7$, guess 5 key words and compute word 9 in X^5
- Repeat for N pairs with input-difference Δ
- Right guess of subkey $\rightarrow$ large χ^2

Attack on 7 rounds ?
Complexity about 2^{217} using 2^{24} pairs of keystream.
Input-difference in key $\rightarrow$ related key scenario / non-randomness in F.

Attack on 6 rounds:
Complexity about 2^{177} using 2^{16} pairs of keystream.
Crowley's differential, input-difference in IV only.

Part 3
Non-randomness in TSC-4

Description of TSC-4

eSTREAM candidate, multiword T-function with single cycle.

State: two matrices of 128 bits each.
Update: two 32 bit parameters α, two S-boxes ...
Output: combine bytes of both states, output 1 byte.

$$X = \begin{pmatrix} 0 & \dots & 1 & 1 \\ 0 & \dots & 1 & 0 \\ 1 & \dots & 1 & 1 \\ 0 & \dots & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & \dots & 1 & 0 \\ 1 & \dots & 0 & 0 \\ 1 & \dots & 0 & 1 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

Description of TSC-4

eSTREAM candidate, multiword T-function with single cycle.

State: two matrices of 128 bits each.
Update: two 32 bit parameters α, two S-boxes ...
Output: combine bytes of both states, output 1 byte.

$$X = \begin{pmatrix} 0 & \dots & 1 & 1 \\ 0 & \dots & 1 & 0 \\ 1 & \dots & 1 & 1 \\ 0 & \dots & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & \dots & 1 & 0 \\ 1 & \dots & 0 & 0 \\ 1 & \dots & 0 & 1 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

$$\alpha_X = \begin{pmatrix} 1 & \dots & 0 & 1 \end{pmatrix} \quad \alpha_Y = \begin{pmatrix} 0 & \dots & 1 & 1 \end{pmatrix}$$

Description of TSC-4

eSTREAM candidate, multiword T-function with single cycle.

State: two matrices of 128 bits each.
Update: two 32 bit parameters α, two S-boxes ...
Output: combine bytes of both states, output 1 byte.

$$X = \begin{pmatrix} 0 & \dots & 1 & 1 \\ 0 & \dots & 1 & 0 \\ 1 & \dots & 1 & 1 \\ 0 & \dots & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & \dots & 1 & 0 \\ 1 & \dots & 0 & 0 \\ 1 & \dots & 0 & 1 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

$$\alpha_X = \begin{pmatrix} 1 & \dots & 0 & 1 \end{pmatrix} \quad \alpha_Y = \begin{pmatrix} 0 & \dots & 1 & 1 \end{pmatrix}$$

$$S(1010) = 1100 \qquad S(0011) = 1111$$

Description of TSC-4

eSTREAM candidate, multiword T-function with single cycle.

State: two matrices of 128 bits each.
Update: two 32 bit parameters α, two S-boxes ...
Output: combine bytes of both states, output 1 byte.

$$X = \begin{pmatrix} 0 & \dots & 1 & 1 \\ 0 & \dots & 1 & 1 \\ 1 & \dots & 1 & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & \dots & 1 & 1 \\ 1 & \dots & 0 & 1 \\ 1 & \dots & 0 & 1 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

$$\alpha_X = \begin{pmatrix} 1 & \dots & 0 & 1 \end{pmatrix} \quad \alpha_Y = \begin{pmatrix} 0 & \dots & 1 & 1 \end{pmatrix}$$

$$S'(1110) = 0010 \qquad S(1000) = 1010$$

Description of TSC-4

eSTREAM candidate, multiword T-function with single cycle.

State: two matrices of 128 bits each.
Update: two 32 bit parameters α, two S-boxes ...
Output: combine bytes of both states, output 1 byte.

$$X = \begin{pmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 0 & 1 \\ 1 & \dots & 1 & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & \dots & 1 & 1 \\ 1 & \dots & 0 & 1 \\ 1 & \dots & 1 & 1 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

$$\alpha_X = \begin{pmatrix} 1 & \dots & 0 & 1 \end{pmatrix} \quad \alpha_Y = \begin{pmatrix} 0 & \dots & 1 & 1 \end{pmatrix}$$

$$S(0010) = 1011 \qquad S'(0110) = 0001$$

Initialisation

Fill states with (K, IV):

$$X = \begin{pmatrix} \text{key} & \text{key} & \text{key} & \text{key} \\ \text{key} & \text{key} & \text{key} & \text{key} \\ \text{IV} & \text{IV} & \text{IV} & \text{IV} \\ \text{IV} & \text{IV} & \text{IV} & \text{IV} \end{pmatrix} \quad Y = \begin{pmatrix} \text{IV} & \text{IV} & \text{IV} & \text{IV} \\ \text{IV} & \text{IV} & \text{IV} & \text{IV} \\ \text{key} & \text{key} & \text{key} & \text{key} \\ \text{key} & \text{key} & \text{key} & \text{key} \end{pmatrix}$$

Apply 8 warm-up rounds:

- Regular update
- Rotate row 2 in X, row 1 in Y
- XOR keystream byte to rotated rows

Question: Large diffusion with output feedback?

Statistical Model

What to measure?
Fix single bit-slice of input, compute bias of this bit-slice after r rounds.

Statistical model of initialisation:

- Parameter α and feedback are uniformly random
- Compute exact bias ε^2 after r rounds
- Find IV with maximum bias

$$\begin{pmatrix} \dots & 0 & \dots \\ \dots & 0 & \dots \\ \dots & 1 & \dots \\ \dots & 0 & \dots \end{pmatrix} \xrightarrow{S \text{ or } S'} \begin{pmatrix} \dots & 0 & \dots \\ \dots & 0 & \dots \\ \dots & 1 & \dots \\ \dots & 1 & \dots \end{pmatrix} \xrightarrow{0 \text{ or } 1} \begin{pmatrix} \dots & 0 & \dots \\ \dots & 1 & \dots \\ \dots & 1 & \dots \\ \dots & 1 & \dots \end{pmatrix}$$

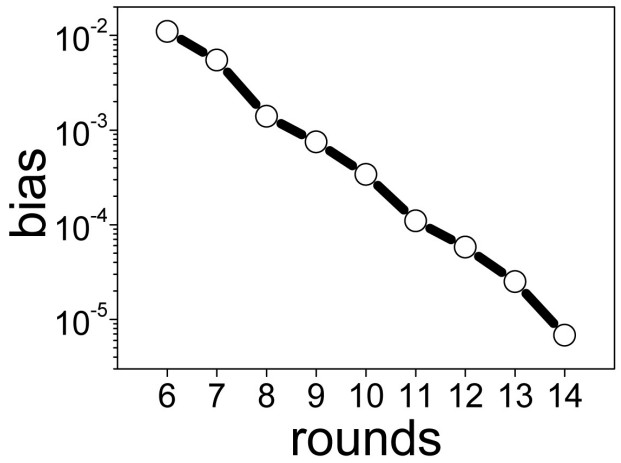
Results:
For $r = 8$ rounds observe at most $\varepsilon^2 = 2^{-9.8}$.

Experiments

Setup:
Fixed random key, partially fixed IV.

Results:

- 8 rounds $\longrightarrow$ 1000 inputs to detect bias
- In each round, bias reduced by factor 2.5
- Non-randomness persists for many more *regular* updates



Attack?
Non-random initial states are not accessible to attacker. Strong output function prevents from attack.

Part 4
Conclusions

Conclusions

Our framework...
Measure non-randomness of initialisation function
for different number of rounds.

Salsa20...
Observe imbalance for 7 round, attack for 6 rounds.
Salsa20/8 may not offer adequate security in future.
Salsa20 appears to be a conservative design.

TSC-4 ...
Diffusion of output feedback is limited.
Large imbalance for 8 rounds and more.
Strong output function prevents from attack.

Questions ?