ESC 7.1. – 11. 1. 2007

Key Recovery with Probabilistic Neutral Bits

Simon Fischer[1], Shahram Khazaei[2] and Willi Meier[1]

[1]FHNW, Windisch, Switzerland
[2]EPFL, Lausanne, Switzerland

Outline

- Motivation
- Probabilistic Neutral Bits
- Description of Salsa
- Analysis of Salsa
- Attack based on polynomial description
- Application to Trivium
- Application to Grain-128
- Conclusions

Motivation

Given a Boolean function

$$F(K,V)\colon \{0,1\}^n \text{ x } \{0,1\}^m \rightarrow \{0,1\}$$

where K is the secret key and V is the initial vector of a stream cipher.

F: Key/IV mixing function, or function derived from it.

Oracle chooses random unknown key $K = (k_0,\ldots,k_{n-1})$ and returns (exact or biased) value

$$z = F(K,V)$$

for every query $V = (v_0,\ldots,v_{m-1})$ of our choice.

Goal: Determine key K in chosen IV attack.

If F mixes its inputs properly, we need to try all 2^n keys, by sending $O(n)$ queries to oracle.

Investigate methods which can lead to faster recovery of key in case mixing of inputs is not complete.

Existence of faster methods highly depends on structure of F.

1st direction: Analysis of reduced round variants of Salsa20.

Based on truncated differentials and approximate backwards computation.

2nd direction: Recent framework for chosen IV statistical analysis of stream ciphers (Saarinen, O'Neil, Englund-Johansson-Turan):

Based on polynomial description of F.

Open problem: Can distinguishers be exploited for key recovery?

Of interest for stream ciphers with round based initialization function F with sparse Boolean functions as components.

Examples: eSTREAM candidates Grain, Trivium.

Probabilistic Neutral Bits

Neutral bits: Known from hash function cryptanalysis (Biham-Chen, 2004).

Our goal:

Find functions approximating $F(K,V)$ that depend on less than all key bits.

Can sometimes be achieved, e.g., when V is restricted to a suitable subset W.

Function approximation conceivable if some key bits have no influence on value of z with high probability, i.e.,

if complementing these key bits is likely to leave value of z unchanged:

Probabilistic neutral key bits

Reduced complexity key recovery?

Formally: approximate function F,

$$F(K,W): \{0,1\}^n \times W \rightarrow \{0,1\}$$

by functions $A(L,W)$ that depend only on subset L of key bits, where W is a suitable subset of V.

Appropriate partitioning of key K as $K = (L,M)$, with L of t bits, and M of n-t bits.

Partitioning identified according to (probabilistic) neutral bits.

Single key bit k_i is called a neutral bit of function F if complementation of k_i does not change output of F, for all inputs in K and W.

Define approximations $A(L,W)$ to be $F(K,W)$, either with a fixed or randomly chosen value for non-significant key bits in M.

Approximations $A(L,W)$ of function $F(K,W)$ expected to hold with some probability.

If M consists of neutral key bits, get exact approximation $A(L,W)$ of function $F(K,W)$ that depends only on the (significant) key bits in L.

More generally, the neutrality measure of a key bit k_{i} is defined as γ_i

where $1/2(1+\gamma_i)$ is the probability that complementing k_{i} does not change the output of F.

Set threshold γ such that all key bits with $|\gamma_i| < \gamma$ are included in the subkey L: significant bits.

Probabilistic neutral bits often inexistent in original key/IV mixing procedure of a cipher, but:

Can occur in intermediate computation derived from mixing process.

Description of Salsa

State: matrix of 16 words of 32 bits, 256-bit key

Update: Increment of a counter

Output function: compression function, achieved through iteration of simple operation, called quarterround:

Input $y = (y_0, y_1, y_2, y_3)$, Output $z = (z_0, z_1, z_2, z_3)$

$$z_1 = y_1 \oplus ((y_0 + y_3) <<< 7)$$

$$z_2 = y_2 \oplus ((z_1 + y_0) <<< 9)$$

$$z_3 = y_3 \oplus ((z_2 + z_1) <<< 13)$$

$$z_0 = y_0 \oplus ((z_3 + z_2) <<< 18)$$

State as a matrix:

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

Update below diagonal words first.

Repeat for all words in columns, then in rows.

10 rounds columns, 10 rounds rows.

Output the keystream $X^0 + X^{20}$.

Nonce, counter, of 64 bits

Initialization of Salsa:
Fill state with (key, $counter$, $nonce$), $counter = 0$

Initial vector (known): IV = ($counter$, $nonce$)

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

Analysis of Salsa

Analysis of Salsa20 reduced to 8 rounds.

Steps:

Identify optimal choices for truncated differentials (over 1st 4 rounds)

Search for probabilistic neutral key bits to approximate backwards computation from 8th round to 4th round, so that bias in 4th round is still detectable.

For approximation to hold need to guess only significant key bits. Enables reduced complexity search of these key bits.

Choosing a differential:

Consider truncated differentials with 1-bit input difference in the nonce and 1-bit difference in a specified word after 4 rounds.

Probabilistic backwards computation:

Assume differential with known bias is fixed.
Corresponding outputs Z and $Z‘$ are observed.

If full key is known, can invert operations in $Z = X + X^{\mathrm{R}}$ and $Z‘ = X‘ + (X‘)^{\mathrm{R}}$ to observe r-round differential ($R > r$) with its bias, by computing $R - r$ rounds backwards.

If only subkey of $m = 256 - n$ bits is known, could approximate inversion by fixing remaining n key bits (e.g., by 0) and invert R - r rounds. However, observable bias depends on n and positions of these n bits.

Probabilistic neutral key bits

Identify a large subset of key bits which can be replaced by fixed bits so that detectable bias after approximate backwards computation is still significant.

Precomputation

Find high probability differential with difference in nonce.

Identify subset of n key bits which are PNB‘s for this differential.

Determine bias of differential with respect to subset of PNB‘s, as observed after working R - r rounds backwards from pairs of keystream blocks, a correct key portion, and randomly chosen values for remaining key bits (PNB‘).

Effective attack

Collect N pairs of keystream blocks generated with selected input difference (N to be determined according to optimal Neyman-Pearson distinguisher).

For each choice of the $m = 256 - n$ remaining key bits, use the N keystream blocks to filter candidate keys with respect to optimal distinguisher.

For each filtered key, check correctness by performing exhaustive search over n remaining bits.

Experimental results

Attack on Salsa20/7

Use 4-round differential optimal for 3-round backwards computation. Find 125 key bits with neutrality measure greater than 0.6. Take these as PNB‘s.
Build attack in time 2^{153} and data 2^{23} (best previous attack: Tsunoo et. al., 2^{190} trials and 2^{12} data).

Attack on Salsa20/8

Use 4-round differential optimal for 4-round backwards computation. Identify 28 key bits with neutrality measure greater than 0.2, which are taken as PNB‘s. Have to guess $m = 256 – 28 = 228$ bits.
Get attack in 2^{249} time and 2^{21} data.

Attack based on polynomial description

Algebraic description of key/IV mixing function F too complex.

Derive simpler Boolean functions $C(K,W)$ with help of oracle, where W is a subset of V.

If $C(K,W)$ has imbalance in algebraic structure, e.g., for high degree monomials, this can be exploited in cryptanalysis.

Example Partition IV as $V=(U,W)$ with U of l bits and W of m-l bits.
$C(K,W)$: Coefficient in ANF of a function deduced from F by varying over bits in U only.

Scenarios:

1. If algebraic structure of $C(K,W)$ is imbalanced for chosen set W and many fixed values of unknown K: Stream cipher can be distinguished from random (Saarinen, O'Neil, EJT).

2. If $C(K,W)$ is evaluated for some fixed W: $C(K,W)$ is an expression in key bits only. Sometimes does involve not all key bits.

3. More generally, if for $C(K,W)$ many key bits have only a limited influence on values of $C(K,W)$: Suitable approximations may be identified that enable reduced complexity key recovery.

Scenario 2:

Find relation $C(K,W)$, evaluated for some W, that depends on subset of $t < n$ key bits only.

Determine functional form with 2^t evaluations of $C(K,W)$.

Can filter those keys which don‘t satisfy relation.

Example (Vielhaber)
$C(K,W)$ sometimes depends on only few key bits and can be a linear expression for well chosen IV part W:

Trivium initialization reduced to 576 iterations.

Allows reduced complexity key recovery in simplified Trivium.

Scenario 3:

Idea is to find function $A(L,W)$ that depends on sub window L of $t < n$ key bits only, and which is correlated to $C(W,K)$.

Ask oracle N queries to get information about t bits of key in time $N2^{t}$ (N to be specified).

Problem: Find suitable function C and approximating function A.

Derived Functions

Partition IV according to $V = (U,W)$ with U of l bits and W of m-l bits.

Write F as

$$F(K,V) = \sum_{\alpha,\beta,\kappa} C_{(\alpha,\beta),\kappa} U^{\alpha} W^{\beta} K^{\kappa} = \sum_{\alpha} C_{\alpha}(K,W) U^{\alpha}$$

where α, β, κ are multi-indices.

For every $\alpha \in \{0,1\}^l$ the function $C_\alpha(K,W)$ can serve as

a function $C = C(K,W)$ derived from F.

Adversary with help of oracle evaluates $C_\alpha(K,W)$

for the unknown key K at chosen input W and for any chosen α by sending at most 2^l queries.

For l small enough, this is a feasible computation.

Attack: Description

Assume suitable function C and partitioning of $K = (L,M)$ for setting of approximation A.

Probabilistic guess and determine: Find small set of candidate subkeys in L.

Filter set of all 2^t subkeys in L into smaller set: Need to distinguish correct guess L' from incorrect ones.

Depends on correlation coefficient between $A(L',W)$ and $C(K,W)$ with $K = (L,M)$ under hypotheses

H_o: guessed part L' is incorrect
H_1: guessed part L' is correct

$$\Pr_W\{A(L',W)\}=C(K,W)\big|K=(L,M)\}=\frac{1}{2}(1+\varepsilon_0)$$

$$\Pr_{L',W}\{A(L',W)=C(K,W)\big|K=(L',M)\}=\frac{1}{2}(1+\varepsilon_1)$$

Both correlation coefficients ε_0 and ε_1 are random variables depending on the key.

If distributions of ε_0 and ε_1 well separated, can achieve small non detection probability p_{mis} and false alarm probability p_{fa} at most 2^{-c}.

If ε_0 and ε_1 assumed to be constants with $\varepsilon_0 < \varepsilon_1$, the optimum distinguisher is Neyman-Pearson.

Determine required number N of values $C(K,W)$ for different W to achieve prescribed p_{fa} and p_{mis}.

Complexity of attack:

For each guess of L' of subkey, correlation ε of

$$A(L',W) \oplus C(K,W)$$

is computed.
Requires computation of coefficients $A(L',W)$ by adversary, and computation of coefficient $C(K,W)$ through oracle, for N values of W. Has cost $N2^l$.

Repeat for all 2^t guesses for L'.

Set of candidates for subkey L has size $2^t\, p_{fa} = 2^{t-c}$.

Entire key verified by exhaustive search over key part M with cost $2^{t-c}2^{n-t}$ evaluations of function F.

Total complexity: $N2^l2^t + 2^{t-c}2^{n-t} = N2^{l+t} + 2^{n-c}$.

Application to Trivium

Trivium has 288 bit internal state consisting of three shift registers of different lengths.

Initialization: $n = 80$ key bits and $m = 80$ IV bits are written into two shift registers with remaining part being set to fixed pattern.

Cipher state updated $R = 18 \times 64 = 1152$ times without producing output.

Consider Boolean function $F(K,V)$ which computes 1st key stream bit after r rounds of initialization.

Derived functions $C(K,W)$.

Previous results:

Distinguisher based on monomial tests (Englund-Johansson-Turan), for r up to 11.5×64 rounds, and l up to 33 variable IV bits.

Key recovery (Vielhaber), similar to scenario 2, for $r = 9 \times 64$.

New results:

Provide examples with respect to scenario 3 for $r = 10 \times 64$ as well as for $r = 10.5 \times 64$.

Example 1

Number of rounds $r = 10 \times 64$, variable IV part U with $l = 10$ (non-consecutive) bit positions. Index α of coefficient $C = C_\alpha(K,W)$ is 1023. C (virtually) only depends on $t = 10$ key bits L. Leads to exact approximating function $A(L,W)$.

65 equivalence classes for L with respect to A: one with 512 members, and 64 classes with 8 members, i.e., get $\frac{1}{2} \times 1 + \frac{1}{2} \times 7 = 4$ bits of information about key.

Example 2

$r = 672$ rounds, $l = 11$ bit positions. Consider W's of weight 5 and compute neutrality measure of key bits: A set of $t = 29$ key bits ruled out as significant.

Correct subkey of 29 bits can be detected using approximations with time complexity 2^{55}.

Application to Grain-128

Grain-128 consists of a LFSR and a NFSR, and an output function h. It has $n = 128$ key bits, $m = 96$ IV bits and full initialization takes $R = 256$ rounds.

Example

$r = 180$, $l = 7$ suitable bit positions. Identify $t = 110$ significant key bits for L. Can detect these in estimated time complexity 2^{124}, i.e., improvement factor 2^4.

Conclusions

- Have introduced technique of probabilistic neutral bits.
- Useful in analysis of initialization of stream ciphers.
- Contributes to applicability of recent chosen IV statistical distinguishers.
- Key recovery with complexity lower than exhaustive key search for simplified versions of three phase 3 eStream candidates.