A note on related-key attacks on Saturnin*

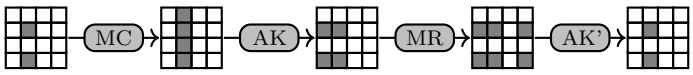
Anne Canteaut[1], Sébastien Duval[2], Gaëtan Leurent[1], María Naya-Plasencia[1], Léo Perrin[1], Thomas Pornin[3] and André Schrottenloher[1]

[1] Inria, France, {anne.canteaut,gaetan.leurent,maria.naya_plasencia,leo.perrin,andre.schrottenloher}@inria.fr
[2] UCL Crypto Group, Belgium, sebastien.pf.duval@gmail.com
[3] NCC Group, Canada, Thomas.pornin@nccgroup.com

1 A 10-round related key attack

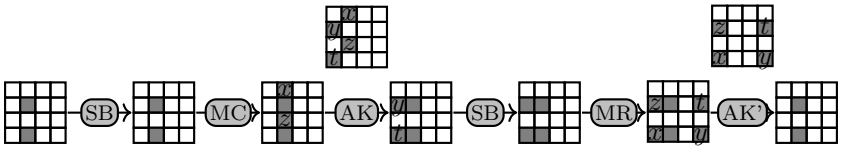
In order to provide the best related-key attack on a reduced-round version of Saturnin, we will consider the iterative trail that we proposed in [CDL+20] in Section 5.2.4, with a key-difference of $\delta\mathsf{K}$:

$$\delta\mathsf{K} = \begin{cases} \text{if } r = 2t+3 \text{ (master key } \mathsf{K}) \\ \text{if } r = 2t+1 \text{ (rotated key } \mathsf{K}') \end{cases}$$



1.1 Instantiating the truncated trail with concrete differences

For a given δK, we denote the active differences in the key as x, y, z, t. In order to satisfy the differential trail, several state differences must be equal to key differences, as show below:



In particular, the active rows for MixRows in the second round have two input words with a zero difference, one output word with a zero difference, and two output words with a difference that is fixed from the key difference ((z,t) or (x,y) respectively). Therefore there are only 2^{16} choices of (z,t) (respectively (x,y)) out of 2^{32} that are possible. Moreover for each choice of (x,y,z,t) we can compute all the differences before and after MR. Similarly, the active column for MixColumns in the first round has two input words with a zero difference, and two output words fixed a difference that is fixed from key conditions; therefore there is a single possibility for the differences before and after MC.

In order to find the best instantiation of this truncated trail, we just have to iterate over the 2^{32} possible choices of (x,y,z,t), deduce all the state differences, and multiply

*https://project.inria.fr/saturnin/

the probabilities of each differential transition. The best instantiation has a probability of $2^{-78.1}$ with

$$x = \texttt{0xe2e2}, \qquad y = \texttt{0xaaaa}, \qquad z = \texttt{0x3535}, \qquad t = \texttt{0x3f3f}.$$

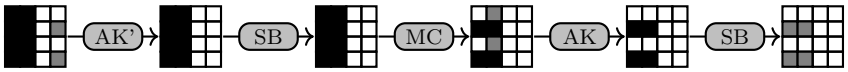
In [CDL+20] in Section 5.2.4 we gave a truncated differential with a reduced number of active S-Boxes in the first and last round. However, it turns out that this path can not be instantiated with concrete differences. Indeed, all the state differences can be computed directly from δK, and we can not have a cell that is active in the iterative rounds, and inactive in the first or last round.

1.2 Using a 6 Super-round distinguisher

If we choose the optimal δK, the probability for a related-key differential distinguisher over 6 super-rounds, repeating 3 times the previous figure over two will be $2^{-78.1\times 3} = 2^{-234.3}$, which can be detected with fewer pairs than the exhaustive search cost of 2^{256}.

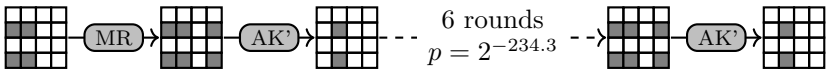
Our attack uses the 6-round related-key differential path, extended with 2 extra rounds at the top and two at the bottom, as a truncated differential. For simplicity, we remove the final MixRows and replace the final key K' by an equivalent key $K_{\text{eq}} = \text{MR}^{-1}(K')$. We also integrate the MixRows of the second round inside the 6-round trail. We use grey squares to denote a cell with a known difference, and black squares for cells with an unknown difference.

2-round truncated differential (top):

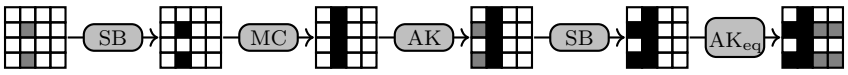


A random pair matching the input pattern follows the trail with probability 2^{-128}.

6-round trail (distinguisher):



2-round truncated differential (bottom):



Key words: $K = \begin{array}{|c|c|c|c|}\hline k_0 & k_1 & k_2 & k_3 \\ \hline k_4 & k_5 & k_6 & k_7 \\ \hline k_8 & k_9 & k_{10} & k_{11} \\ \hline k_{12} & k_{13} & k_{14} & k_{15} \\ \hline \end{array}$, $K' = \begin{array}{|c|c|c|c|}\hline k_5 & k_6 & k_7 & k_8 \\ \hline k_9 & k_{10} & k_{11} & k_{12} \\ \hline k_{13} & k_{14} & k_{15} & k_0 \\ \hline k_1 & k_2 & k_3 & k_4 \\ \hline \end{array}$, $K_{\text{eq}} = \begin{array}{|c|}\hline \text{MR}^{-1}(k_5, k_6, k_7, k_8) \\ \hline \text{MR}^{-1}(k_9, k_{10}, k_{11}, k_{12}) \\ \hline \text{MR}^{-1}(k_{13}, k_{14}, k_{15}, k_0) \\ \hline \text{MR}^{-1}(k_1, k_2, k_3, k_4) \\ \hline \end{array}$

1.3 Attack procedure

We use structures of 2^{128} plaintexts encrypted under key K and $K \oplus \delta K$. Each structure is defined by a fixed value u; we encrypt all 2^{128} plaintexts with value u in the two rightmost columns under key K, and all 2^{128} plaintexts with value $u \oplus \delta P$ in the two rightmost columns under key $K' \oplus \delta K'$, where δP refers to the key difference in the rightmost column of K'. A pair of structures defines 2^{256} pairs that match the input pattern.

We initially encrypt $N = 2^{107}$ structures, so that we expect $N \times 2^{256} \times 2^{-128} \times 2^{-234.3} = 2^{0.7}$ pairs that follow the full trail; with high probability there is one right pair. For each structure we sort the ciphertexts according to the value in the 10 output cells with a known difference, and locate collisions. After this filtering we are left with $N \times 2^{256} \times 2^{-160} = N \times 2^{96}$ pairs that match the input and output pattern. We explain below how to recover 224 bits of key from each candidate pair, for a cost of $N \times 2^{128}$.

This suggest $N \times 2^{96}$ candidates for 224 bits of the key. We can verify each candidate by exhaustive search of the missing 32 bits, for a total cost of $N \times 2^{128}$. With high probability there is a pair following the path in the initial data, and the attack is successful. The total complexity will be about $2 \times N \times 2^{128} = 2^{236}$.

For each of the $N \times 2^{96}$ pair candidates, we perform the following steps:

Step 1. We first guess k_{10} and k_2, corresponding to positions 5 and 13 of K', with a cost of 2^{32}. Therefore we can compute SuperBoxes 5 and 13 in round 0. Since two differences are fixed after MixColumns, we can recover the difference in cells 1 and 9 after the first SuperBoxes. Since the input difference of those SuperBoxes is known from the plaintext pair, we recover the values and finally k_6 and k_{14}. After this step we have $N \times 2^{128}$ candidates, and the following steps are therefore repeated $N \times 2^{128}$ times.

Step 2. The second column of K' is known, therefore we can compute the second column of the state through AK', SB, and MC. In particular, we obtain the difference at the input of SuperBoxes 5 and 13 of round 1. Since the output differences are fixed by the differential path, we recover the values and finally k_5 and k_{13}.

Step 3. Similarly to step 1, but without needing to guess any key bits now, we can compute SuperBoxes 0 and 8 in round 0 using k_5 and k_{13}, previously recovered. Since two differences are fixed after MixColumns, we can recover the difference in cells 4 and 12 after the first SuperBoxes, hence the values and finally k_9 and k_1.

Step 4. Similarly to step 2, the first column of K' is now known, therefore we obtain the differences at the input of SuperBoxes 4 and 12 of round 1. Since the output differences are fixed by the differential path, we recover the values and finally k_4 and k_{12}. Up to now, we have determined a candidate value for 10 key words, that is of 160 bits.

Step 5. We now focus on the last round. We know the output differences of SuperBoxes 4 and 12 from the ciphertext pairs, and the input differences are fixed by the differential trail. Therefore we can recover the values, and we deduce cells 4 and 12 of the equivalent key $K_{\text{eq}} = \text{MR}^{-1}(K')$. Since three cells are already known in the second and fourth rows of K', we deduce both full rows, in particular cells 11 and 3 (we also have the second and fourth rows of K_{eq}). The total number of determined keybits rises to 192.

Step 6. We can now compute the values and differences through the inverse SuperBoxes 5 and 13 of the last round. Since two differences are fixed to zero before MixColumns in round 8, we can recover the differences of the full column. In particular, we obtain the input differences of SuperBoxes 1 and 9 in the last round. The corresponding output differences are given by the ciphertext, therefore we recover cells 1 and 9 of K_{eq}, with a total of 224 key bits of information determined.

Step 7. Finally, we have the full second column of K_{eq}. We can decrypt the second column of the state through SB, AK (because the second column is also known), MC and the SuperBox of round 8. We can verify that the difference in cell 5 and 13 matches the difference fixed by the differential trail, which is satisfied by a fraction 2^{-32} of the pairs.

To conclude, we obtain $N \times 2^{96}$ candidate pairs, each with a determined valued for 224 bits of keys on average. We can test the remaining 32 bits of the key for each candidate, and keep only the one that follows the full path. The complexity of this step is $N \times 2^{128}$ encryptions.

1.4 Quantized version of this attack

A quantized version of this attack is expected to reach less rounds, as the generation of the pairs with structures may reach less than a quadratic speedup in the quantum setting (thus going below than exhaustive search of the key). We leave the best quantum related-key attack as an open problem.

References

[CDL+20] Anne Canteaut, Sébastien Duval, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, Thomas Pornin, and André Schrottenloher. Saturnin: a suite of lightweight symmetric algorithms for post-quantum security. *IACR Trans. Symm. Cryptol.*, 2020(S1):160–207, 2020.