Supplementary Material

Experiments on round-reduced key-recovery attacks.

Since according to Eq. (12) the data to collect and store is at least $2^{n/2}/pq$. In the case where block size $n = 64$ and distinguisher with probability 1, the complexity of 2^{32} is usually hard for our computers. So due to the limited computing power, we make experiments on round-reduced Simon32, whose block size is 32. We give an example on `Simon32/64` using a 6-round distinguisher with probability 2^{-2} in single-key setting. Appending 1-round E_b and 3-round E_f, we attack 10-round `Simon32/64` as Table 1, where ΔX_r is the input difference in round r.

Table 1: The 10-round rectangle attack on `Simon32/64`.

ΔX_0	0100 0000 0000 0000 ?000 0000 0?00 0001
$\Delta X_1(\alpha)$	0000 0000 0000 0000 0100 0000 0000 0000
...	...
$\Delta X_7(\delta)$	0100 0000 0000 0000 0000 0000 0000 0000
ΔX_8	?000 0000 0?00 0001 0100 0000 0000 0000
ΔX_9	0?00 000? ?000 01?? ?000 0000 0?00 0001
ΔX_{10}	?000 0??? 0?01 ???? 0?00 000? ?000 01??

Following the single-key model in the submitted paper, we choose the expected number of right quartets is $s = 4$, and construct 2^{18} pairs (P_1, P_2) satisfying the input difference α. With $m_b = m'_f = 0$, the time complexity of generating quartets is also 2^{18}. We invert the final round to derive ΔX_9 as filters to generate quartets. Assuming the differences are uniformly distributed, the remaining quartets is $s \cdot 2^{18+18-2\cdot 25}$. But in fact, the differences of ΔX_9 are not absolute uniformly distributed and there will be s quartets remains. The time complexity to generate the quartets is 2^{18}. Using the remaining quartets, we first guess $STK_9[0, 3-9, 12-15]$ to check whether ΔX_8 satisfies the ?000 0000 0?00 0001 0100 0000 0000 0000 for both (C_1, C_3) and (C_2, C_4). $s \cdot 2^{12-2\cdot 5} = s \cdot 2^2$ quartets remains. Then we guess $STK_8[5, 7, 14]$ to check whether ΔX_8 satisfies δ. $s \cdot 2^2 \cdot 2^{3-2\cdot 2} = 2s$ quartets remains. So the time complexity of the key recovery process is $s \cdot 2^{12} + s \cdot 2^5$. The size of key counters is 2^{15}.

In total, the data complexity is 2^{18} and the memory complexity is $\cdot 2^{18} + 2^{15} \approx 2^{18.17}$. The time complexity is $2^{18} + 2^{18} + 2^{14} + 2^7 \approx 2^{19.04}$ (We don't make experiments on the exhaustive search process). Set $h = 4$, and the success probability is 97.6%.

Experiments result. Testing with 100 different mater keys, if the right key candidate is in the top 2^{15-h} key counters, we consider the attack succeeds to gain a $h = 4$-bit advantage than the exhaustive search. The experiment need

about 1 minutes on one computer and the success rate is 100%. The code of the experiment can be found in https://github.com/key-guess-rectangle/key-guess-rectangle