Cryptanalysis of Clyde and Shadow

July 2nd, 2019

Horst Görtz Institut für IT Sicherheit, Ruhr-Universität Bochum

Gregor Leander, and *Friedrich Wiemer*

Invariant Attacks

What are Invariant Attacks

Invariant Attacks

Proving Resistance

- Goal: apply security argument from

  C. Beierle, A. Canteaut, G. Leander, and Y. Rotella. "Proving Resistance Against Invariant Attacks: How to Choose the Round Constants". In: CRYPTO 2017, Part II. ed. by J. Katz and H. Shacham. Vol. 10402. LNCS. Springer, Heidelberg, Aug. 2017, pp. 647–678. doi: 10.1007/978-3-319-63715-0_22. iacr: 2017/463.

- This argument proves that there is no invariant for both, the S-box and linear layer, parts of the round function.
- However, there might be other partitionings of the round function, for which there are invariants (in particular, Christof Beierle found some examples).
- It is not clear how to prove the general absence of invariant attacks; this is the best we can currently prove.
- All known attacks exploit exactly this structure (that is, splitting in S-box and linear layer).

Invariant Attacks

Recap Security Argument

- The argument bases on the observation that all published invariant attacks use invariant functions which are invariant for the S-box layer *and* invariant for the linear layer.
- Furthermore, invariants over the linear layer L and the round key addition have to be invariant over $W_L(c_1, \ldots, c_t)$.
- $W_L(c_1, \ldots, c_t)$ is the smallest L-invariant subspace of $\mathbb{F}_2^n$ containing all c_i, where these are the round constant differences from rounds that add the same round key.
- $W_L(c_1, \ldots, c_t)$ is L-invariant if and only if: $\forall x \in W_L(c_1, \ldots, c_t) : L(x) \in W_L(c_1, \ldots, c_t)$.
- Thus, if $W_L(c_1, \ldots, c_t)$ contains the whole $\mathbb{F}_2^n$, only the trivial invariants for L and the key addition remain (the constant 0 and 1 functions).
- There is a link between the invariant factors Q_i of the linear layer and the dimension of W_L, [Bei+17, Theorem 1]:

$$\max_{c_1,\ldots,c_t \in \mathbb{F}_2^n} \dim W_L(c_1, \ldots, c_t) = \sum_{i=1}^{t} \deg Q_i \ .$$

Invariant Attacks

Recap Security Argument

For Clyde:

- The linear layer has four invariant factors ($4 \times (x^{32}+1)$).
- Due to its tweakey schedule, every tweakey equals the fourth next tweakey: $\mathrm{TK}_i = \mathrm{TK}_{i+3}$.
- After each step (two rounds), a tweakey is added.
- We need at least four round constant differences; looking at the round constant additions, this implies at least three steps (six rounds), so that W_L can achieve full dimension.
- In particular, the set of round constant differences, for the six steps Clyde uses, is:

$$D = D_{\mathrm{TK}_0} \cup D_{\mathrm{TK}_1} \cup D_{\mathrm{TK}_2} \cup D_0$$

$$\begin{aligned} D_{\mathrm{TK}_0} &= \{0 + W(5), 0 + W(11), W(5) + W(11)\} \\ D_{\mathrm{TK}_1} &= \{W(1) + W(7)\} \\ D_{\mathrm{TK}_2} &= \{W(3) + W(9)\} \\ D_0 &= \{a + b \mid a, b \in \{W(0), W(2), W(4), W(6), W(8), W(10)\}, a \neq b\} \end{aligned}$$

- This gives us 20 round constant differences.

Invariant Attacks

Recap Security Argument

For Clyde (cont.):

- Computing W_L is efficiently doable (takes ≈ 10 seconds on my laptop).
- For the round constants chosen for Clyde, $\dim W_L(D) = 128 = n$.
- Thus, we can apply:

Proposition 2 ([Bei+17])

Suppose that the dimension of $W_L(D)$ is at least $n-1$. Then any invariant g is linear or constant. As a consequence, there is no non-trivial invariant g of the S-box layer, unless the S-box layer has a component of degree 1.

- Such an S-box would be attackable by linear cryptanalysis.
- We conclude that we cannot find any g for Clyde which is at the same time invariant for the S-box layer and for the linear layer.

Subspace Trails

What are Subspace Trails

Subspace Trails

Proving Resistance

- Goal: apply security argument from

  G. Leander, C. Tezcan, and F. Wiemer. "Searching for Subspace Trails and Truncated Differentials". In: IACR Trans. Symm. Cryptol. 2018.1 (2018), pp. 74–100. ISSN: *2519-173X.* DOI: 10.13154/tosc.v2018.i1.74-100*.*

Subspace Trails

Recap Security Argument

RUB

- Basically: Exhaustive Search of possible subspace trails
- Reduce tested subspace trails to a minimal set, so that all subspace trails are still covered
- For SPN constructions using S-boxes with linear structures, this is the set

$$\mathcal{W} := \left\{ W_{i,\alpha} := \{0\}^{i-1} \times \{0, \alpha\} \times \{0\}^{k-i} \;\middle|\; \alpha \in \mathbb{F}_2^n, 1 \leq i \leq k \right\}.$$

  where the round function applies k S-boxes in parallel and each S-box permutes $\mathbb{F}_2^n$
- That is, we check for each candidate starting subspace $\{W_{i,\alpha}\}$, the length of the corresponding subspace trail, using the $\mathrm{Generic\ Subspace\ Trail\ Length}$ algorithm from Leander, Tezcan, and Wiemer [LTW18].
- Intuitively, the $W_{i,\alpha}$ capture all possible output values after the first S-box layer, when only one S-box is active, the algorithm then checks the longest possible subspace trail length from this point on.

Subspace Trails

Recap Security Argument – The algorithms

RUB

Notation

- $\Delta_\alpha(F) := x \mapsto F(x) + F(x+\alpha)$, the derivative of F in direction α
- $F^k := x \mapsto (\underbrace{F(x), \ldots, F(x)}_{k \text{ times}})$, the k-th parallel application of F (e. g. an S-box layer)

Subspace Trails

Recap Security Argument – The algorithms

Compute subspace trails

Input: A nonlinear, bijective function $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ and a subspace U.
Output: The longest subspace trail starting in U over F.

function $\mathrm{Compute\ Trail}(F, U)$
  if $\dim(U) = n$ **then**
    return U
  $V \leftarrow \emptyset$
  for u_i basis vectors of U **do**
    for enough $x \in_R \mathbb{F}_2^n$ **do**
      $V \leftarrow V \cup \Delta_{u_i}(F)(x)$
  $V \leftarrow \mathrm{span}(V)$
  return the subspace trail $U \rightarrow \mathrm{Compute\ Trail}(F, V)$

Subspace Trails

Recap Security Argument – The algorithms

Generic Subspace Trail Search

Input: A linear layer matrix $M : \mathbb{F}_2^{n\cdot k \times n\cdot k}$, and an S-box $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$.
Output: A bound on the length of all subspace trails over $F = M \circ S^k$.

function $\mathrm{Generic\ Subspace\ Trail\ Length}(M, S)$
    empty list L
    for possible initial subspaces represented by $W_{i,\alpha} \in \mathcal{W}$ **do**
        $L.\mathrm{append}(\mathrm{Compute\ Trail}(S^k \circ M, \{W_{i,\alpha}\}))$
    return $\max \{\mathrm{len}(t) \mid t \in L\}$

Division Property

RUB

- Goal: apply security argument fro

  Z. Xiang, W. Zhang, Z. Bao, and D. Lin. "Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers". In: ASIACRYPT 2016, Part I. ed. by J. H. Cheon and T. Takagi. Vol. 10031. LNCS. Springer, Heidelberg, Dec. 2016, pp. 648–678. doi: 10.1007/978-3-662-53887-6_24. iacr: 2016/857.

- Approach: model division trail propagations as MILP, find solutions for this over increasing number of rounds.

Results

Number of rounds for which a distinguisher exist

Cipher	Subspace Trails	Division Property
Clyde	2 (+1)	8
Shadow	4 (+1)	???