

Cryptanalysis of Clyde and Shadow

July 3rd, 2019

Horst Görtz Institut für IT Sicherheit, Ruhr-Universität Bochum

Gregor Leander, and *Friedrich Wiemer*

RUB



1 Invariant Attacks – Round Constants

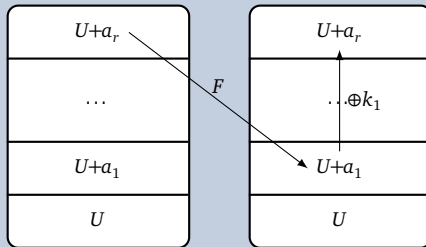
2 Subspace Trails

3 Division Property

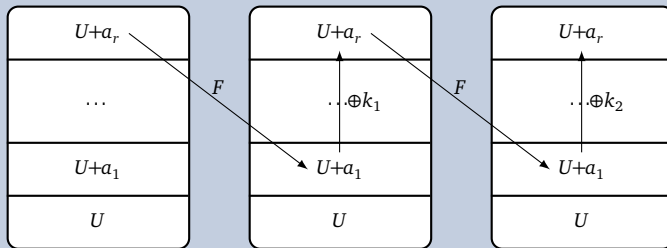
4 Results

Invariant Attacks

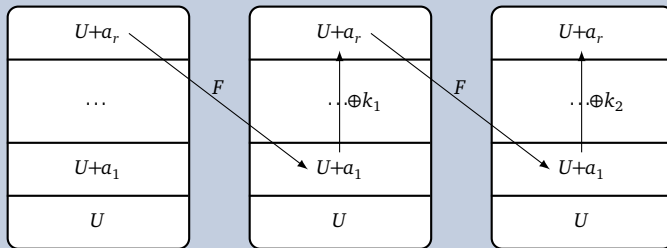
Main Idea: Invariant Subspaces



Main Idea: Invariant Subspaces



Main Idea: Invariant Subspaces



Invariant Subspace Attacks [Lea+11] (CRYPTO'11)

Let $U \subseteq \mathbb{F}_2^n$, $c, d \in U^\perp$, and $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$. Then U is an *invariant subspace* (IS) if and only if $F(U+c) = U+d$ and all round keys in $U+(c+d)$ are *weak keys*.

Invariant Attacks

A Short History



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. 2017. doi: 10.1007/978-3-319-63715-0_22. iacr: 2017/463.

What do we get from this?

- Non-existence of invariants for both parts of the round function (S-box and linear layer)

Issues

- Other partitionings of the round function might allow invariants (Christof B. found examples)
- Not clear how to prove the general absence of invariant attacks (best we can currently prove)
- All known attacks exploit exactly this structure (splitting in S-box and linear layer)

Invariant Attacks

Recap Security Argument (I)

Observation

- Invariants for the linear layer L and round key addition have to contain special linear structures.
- Denote by c_1, \dots, c_t the round constant differences for rounds with the same round key.
- Then the linear structures of any invariant have to contain $W_L(c_1, \dots, c_t)$.

Invariant Attacks

Recap Security Argument (I)

Observation

- Invariants for the linear layer L and round key addition have to contain special linear structures.
- Denote by c_1, \dots, c_t the round constant differences for rounds with the same round key.
- Then the linear structures of any invariant have to contain $W_L(c_1, \dots, c_t)$.

The smallest L -invariant subspace

$W_L(c_1, \dots, c_t)$ is the *smallest L -invariant subspace* of \mathbb{F}_2^n containing all c_i

$$\Leftrightarrow \forall x \in W_L(c_1, \dots, c_t) : L(x) \in W_L(c_1, \dots, c_t)$$

Invariant Attacks

Recap Security Argument (I)

Observation

- Invariants for the linear layer L and round key addition have to contain special linear structures.
- Denote by c_1, \dots, c_t the round constant differences for rounds with the same round key.
- Then the linear structures of any invariant have to contain $W_L(c_1, \dots, c_t)$.

The smallest L -invariant subspace

$W_L(c_1, \dots, c_t)$ is the *smallest L -invariant subspace* of \mathbb{F}_2^n containing all c_i

$$\Leftrightarrow \forall x \in W_L(c_1, \dots, c_t) : L(x) \in W_L(c_1, \dots, c_t)$$

The simple case

If $W_L(c_1, \dots, c_t)$ contains the whole \mathbb{F}_2^n , only trivial invariants for L and the round key addition are possible (constant 0 and 1 function).

Invariant Attacks

Recap Security Argument (II)

Bounding the dimension of W_L , [Bei+17, Theorem 1]

Given a linear layer L . Denote by Q_i its *invariant factors*. Then

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

Invariant Attacks

Recap Security Argument (II)

Bounding the dimension of W_L , [Bei+17, Theorem 1]

Given a linear layer L . Denote by Q_i its *invariant factors*. Then

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

Application to Clyde

- Compute invariant factors of linear layer:
- This gives a first lower bound on the number of rounds:
- Find the important round constant differences:
- Due to its tweakkey schedule, every tweakkey equals the fourth next tweakkey: $TK_i = TK_{i+3}$.
- After each step (two rounds), a tweakkey is added.
- We need at least four round constant differences: looking at the round constant additions, this

Invariant Attacks

Recap Security Argument (II)

Application to Clyde

- Compute invariant factors of linear layer: $4 \times (x^{32} + 1)$
 - This gives a first lower bound on the number of rounds: 3 steps/6 rounds
 - Find the important round constant differences: Set of RC differences D
- Due to its tweakey schedule, every tweakey equals the fourth next tweakey: $TK_i = TK_{i+3}$.
 - 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:

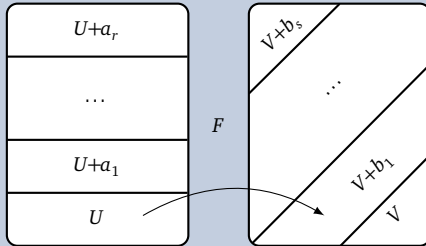
$$D = D_{TK_0} \cup D_{TK_1} \cup D_{TK_2} \cup D_0$$

$$D_{TK_0} = \{0 + W(5), 0 + W(11), W(5) + W(11)\}$$

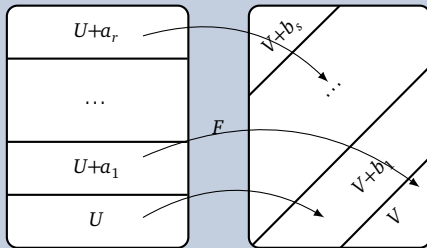
$$D_{TK_1} = \{W(1) + W(7)\}$$

Subspace Trails

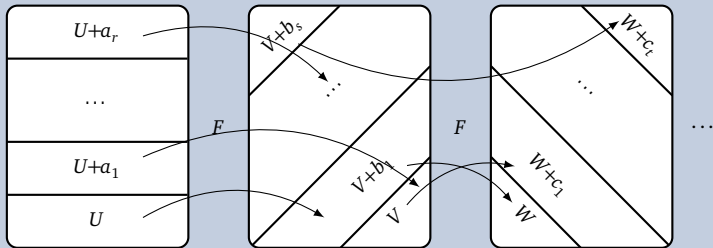
Main Idea: Subspace Trails



Main Idea: Subspace Trails



Main Idea: Subspace Trails



Subspace Trail Cryptanalysis [GRR16] (FSE'16)

Let $U_0, \dots, U_r \subseteq \mathbb{F}_2^n$, and $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$. Then these form a *subspace trail* (ST), $U_0 \xrightarrow{F} \dots \xrightarrow{F} U_r$, iff

$$\forall a \in U_i^\perp : \exists b \in U_{i+1}^\perp : F(U_i + a) \subseteq U_{i+1} + b$$

Computing Subspace Trails

Given a starting subspace U , we can efficiently compute the corresponding longest subspace trail.

Lemma

Let $U \xrightarrow{F} V$ be a ST. Then for all $u \in U$ and all $x: F(x) + F(x + u) \in V$.

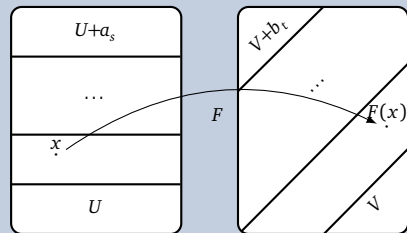
Computing Subspace Trails

Given a starting subspace U , we can efficiently compute the corresponding longest subspace trail.

Lemma

Let $U \xrightarrow{F} V$ be a ST. Then for all $u \in U$ and all $x: F(x) + F(x + u) \in V$.

Proof



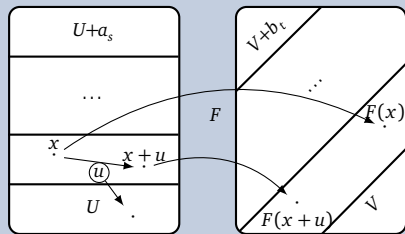
Computing Subspace Trails

Given a starting subspace U , we can efficiently compute the corresponding longest subspace trail.

Lemma

Let $U \xrightarrow{F} V$ be a ST. Then for all $u \in U$ and all $x: F(x) + F(x+u) \in V$.

Proof



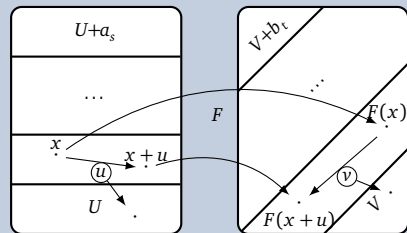
Computing Subspace Trails

Given a starting subspace U , we can efficiently compute the corresponding longest subspace trail.

Lemma

Let $U \xrightarrow{F} V$ be a ST. Then for all $u \in U$ and all $x: F(x) + F(x+u) \in V$.

Proof



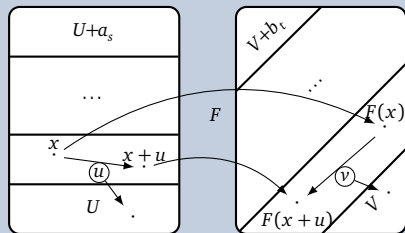
Computing Subspace Trails

Given a starting subspace U , we can efficiently compute the corresponding longest subspace trail.

Lemma

Let $U \xrightarrow{F} V$ be a ST. Then for all $u \in U$ and all $x: F(x) + F(x+u) \in V$.

Proof



Computing the subspace trail

- To compute the next subspace, we have to compute the image of the derivatives.

Computing Subspace Trails

Algorithm

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 ST starting in U over F .

```

1 function Compute Trail( $F, U$ )
2   if  $\dim(U) = n$  then
3     return  $U$ 
4    $V \leftarrow \emptyset$ 
5   for  $u_i$  basis vectors of  $U$  do
6     for enough  $x \in_{\mathbb{R}} \mathbb{F}_2^n$  do           ▷ e. g.  $n + 20$   $x$ 's are enough
7        $V \leftarrow V \cup \Delta_{u_i}(F)(x)$       ▷  $\Delta_a(F)(x) := F(x) + F(x + a)$ 
8    $V \leftarrow \text{span}(V)$ 
9   return the subspace trail  $U \rightarrow \text{Compute Trail}(F, V)$ 
```


Goal: Apply security argument from

G. Leander, C. Tezcan, and F. Wiemer. "Searching for Subspace Trails and Truncated Differentials". In: ToSC 2018.1 (2018). doi: 10.13154/tosc.v2018.i1.74-100.

What do we get from this?

- (Tight) upper bound on the length of any ST for an SPN construction

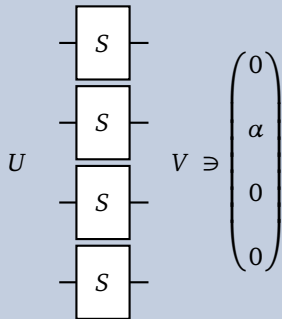
Why is the Compute Trail algorithm not enough?

- Exhaustively checking all possible starting points is too costly.

Subspace Trails

How to bound the length of any subspace trail

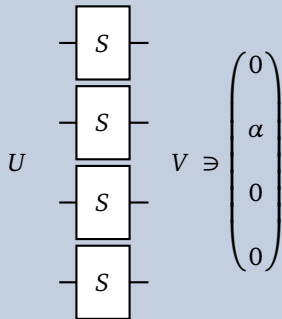
Observation



Subspace Trails

How to bound the length of any subspace trail

Observation



Algorithm Idea

Compute the subspace trails for any starting point $W_{i,\alpha} \in \mathcal{W}$, with

$$W_{i,\alpha} := \underbrace{(0, \dots, 0)}_{i-1}, \alpha, 0, \dots, 0$$

Complexity (Size of \mathcal{W})

For an S-box layer $S : \mathbb{F}_2^{kn} \rightarrow \mathbb{F}_2^{kn}$ with k S-boxes, each n -bit:
 $|\mathcal{W}| = k \cdot (2^n - 1)$

Generic Subspace Trail Search

Input: A linear layer matrix $M : \mathbb{F}_2^{n \cdot k \times n \cdot k}$, and an S-box $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$.

Output: A bound on the length of all STs over $F = M \circ S^k$.

```

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

```

▶ Overall $k \cdot (2^n - 1)$ iterations
 ▶ S^k denotes the S-box layer

Overall Complexity

Algorithm Complexity	Compute Trail $\mathcal{O}(k^2 n^2)$	Generic Subspace Trail Length $\mathcal{O}(k 2^n)$	Overall $\mathcal{O}(k^3 n^2 2^n)$
----------------------	--------------------------------------	--	------------------------------------

Division Property

Goal: Apply security argument from

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. 2016. doi: 10.1007/978-3-662-53887-6_24. iacr: 2016/857.

What do we get from this?

bla

Approach

Model division trail propagations as MILP, find solutions for this over increasing number of rounds.

Results

Results

Thanks for your attention!

Number of rounds for which a distinguisher exist

Cipher	Subspace Trails	Division Property
Clyde	2 (+1)	8
Shadow	4 (+1)	not yet testet

Future Work/Cryptanalysis

- Cryptograph [HV18]



References I

- [Lea+11] G. Leander, M. A. Abdelraheem, H. AlKhazimi, and E. Zenner. "A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack". In: *CRYPTO 2011*. 2011. doi: 10.1007/978-3-642-22792-9_12.
- [LMR15] G. Leander, B. Minaud, and S. Rønjom. "A Generic Approach to Invariant Subspace Attacks: Cryptanalysis of Robin, iSCREAM and Zorro". In: *EUROCRYPT 2015, Part I*. 2015. doi: 10.1007/978-3-662-46800-5_11.
- [GRR16] L. Grassi, C. Rechberger, and S. Rønjom. "Subspace Trail Cryptanalysis and its Applications to AES". In: *ToSC 2016.2* (2016). doi: 10.13154/tosc.v2016.i2.192-225.
- [Guo+16] J. Guo, J. Jean, I. Nikolic, K. Qiao, Y. Sasaki, and S. M. Sim. "Invariant Subspace Attack Against Midori64 and The Resistance Criteria for S-box Designs". In: *ToSC 2016.1* (2016). doi: 10.13154/tosc.v2016.i1.33-56.
- [TLS16] Y. Todo, G. Leander, and Y. Sasaki. "Nonlinear Invariant Attack - Practical Attack on Full SCREAM, iSCREAM, and Midori64". In: *ASIACRYPT 2016, Part II*. 2016. doi: 10.1007/978-3-662-53890-6_1.
- [Xia+16] 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*. 2016. doi: 10.1007/978-3-662-53887-6_24. iacr: 2016/857.
- [Bei+17] 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*. 2017. doi: 10.1007/978-3-319-63715-0_22. iacr: 2017/463.
- [HV18] M. Hall-Andersen and P. S. Vejre. "Generating Graphs Packed with Paths". In: *ToSC 2018.3* (2018). doi: 10.13154/tosc.v2018.i3.265-289.
- [LTW18] G. Leander, C. Tezcan, and F. Wiemer. "Searching for Subspace Trails and Truncated Differentials". In: *ToSC 2018.1* (2018). doi: 10.13154/tosc.v2018.i1.74-100.