Structural Attacks on Block Ciphers

Gregor Leander
Ruhr University Bochum
Germany

CTCrypt 2017

hgi
Horst Görtz Institute
for IT-Security

Outline

The Context

Lightweight Crypto

Ligthweight crypto tends to be

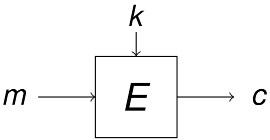
- more agressive
- less standard

Main advantage (besides the obvious):

New insights

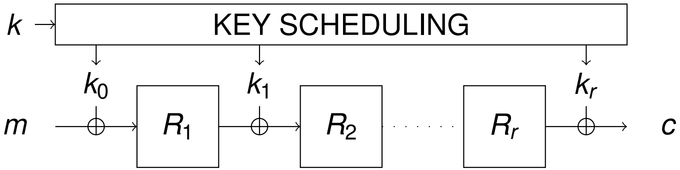
We learn more about the basics on how (not) to design secure ciphers.

A Block Cipher



Ideal block cipher: A random selection of permutations.

Key-Alternating Block Cipher

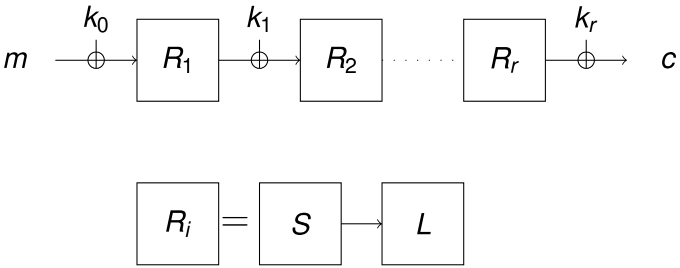


Remark

Many block ciphers are key-alternating.

The Advanced Encryption Standard is one of them.

Block Cipher (3/3): An SP-Network



- S: Sboxes
- L: Linear mapping

Attacks on Block Ciphers

Statistical Attacks

Differential attacks, linear attacks, etc.

- Involve probabilities
- Widely applicable
- Non-trivial to avoid

Structural Attacks

Integral attacks, high order differential attacks, etc

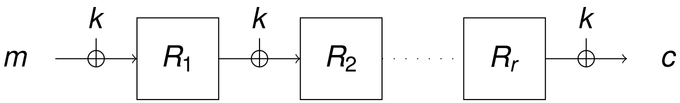
- Hold with prob. 1
- Once detected: Easy to avoid

Focus on special type of structural attacks: Symmetries

A recent trend

Simplify the Key-Scheduling

- Use the same key in every round
- add round constants



A recent trend

Question

Is this a good idea?

- When picking the round constants at random: This is sound.
- Otherwise: Beware of symmetries.

Symmetries

What you do not want (e.g.):

- A symmetric plain-text $p = (x||x)$
- with a symmetric key $k = (y||y)$
- produces always a symmetric cipher-text $c = (z||z)$

Here: It helps if all round keys are identical.
One possible abstraction:

Invariant Subspaces

A symmetry is an affine subspace that is (for weak keys) invariant under encryption.

Symmetries

Question

Are those symmetries attacks?

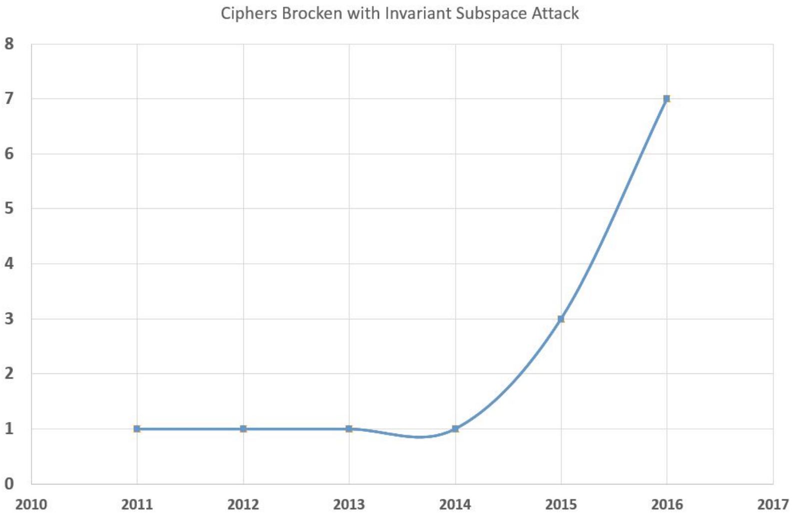
- In many cases debatable.
- In all cases something we do not want.

Do those things happen?

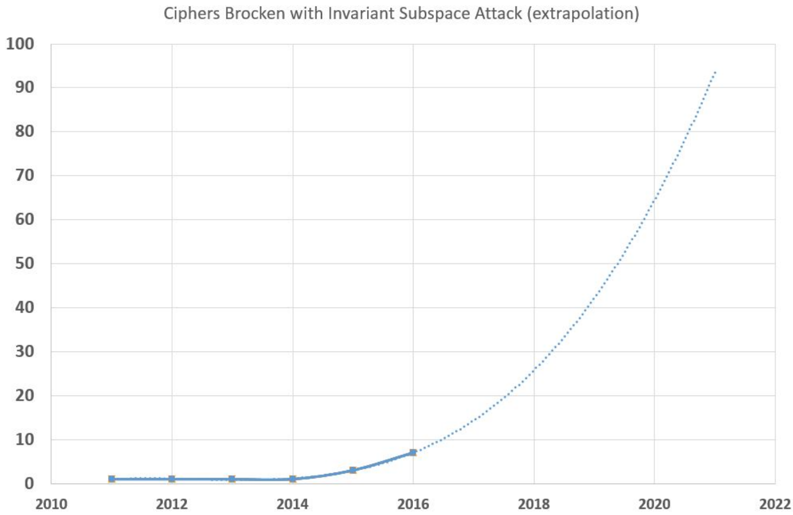
Examples

1. PRINTCipher ('11)
2. iSCREAM ('15)
3. Robin ('15)
4. Zorro ('15)
5. Midori ('16)
6. Haraka (v.0) ('16)
7. Simpira (v.1) ('16)
8. NORX (v 2.0) ('17)

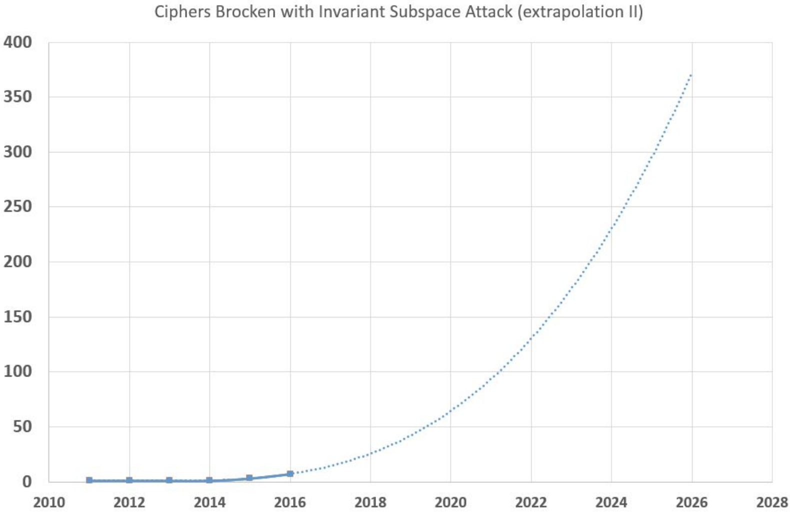
A trend- and were it might lead to (I/III)



A trend- and were it might lead to (II/III)



A trend- and were it might lead to (III/III)



Outline

Origin

Abdelraheem et al '12

Invariant Subspace Attack presented at CRYPTO 2012.

Idea

Make use of a

- weak keys
- that keep a subspace invariant

PRINTCIPHER-48 Attack

Summary

- Prob 1 distinguisher for full cipher
- 2^{50} out of 2^{80} keys weak.
- Similar for PRINTCIPHER-96

Abstraction:

$$F(U \oplus a) = U \oplus b$$

If $k \in U \oplus (a \oplus b)$

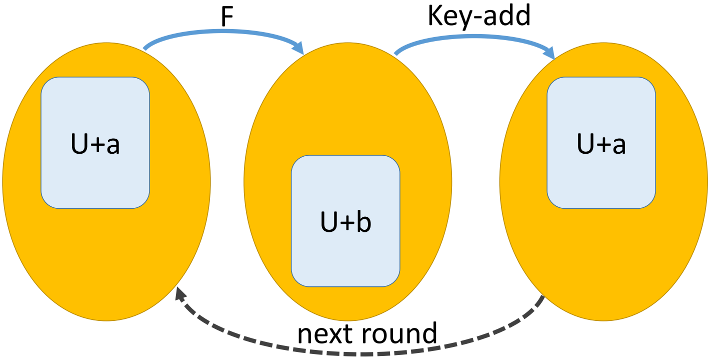
$$F_k(U \oplus a) = U \oplus a$$

Thus an invariant subspace

Question

How to detect it automatically?

The General Idea



- $F(U + a) = U + b$
- $k \in U + (a + b)$ then $U + b + k = U + a$
- Iterative for all rounds (for identical round keys).

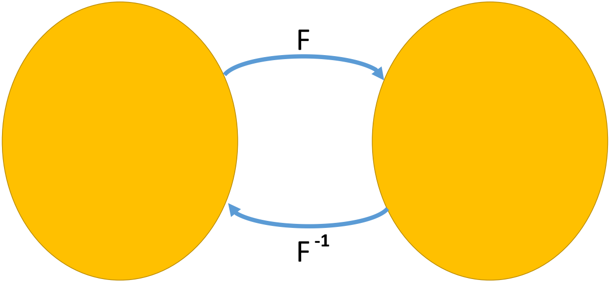
The General Idea

Generic Algorithm (Minaud, Rønjom, L, EC 2015)

Guess a subspace of U. Map it back and forth.

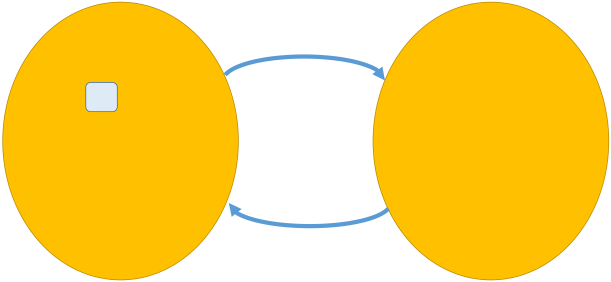
- If the guess was correct: Recovers U
- If not: Find trivial solution.

The General Idea



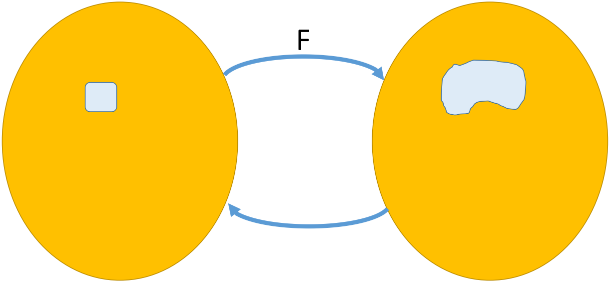
$F := \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ (permutation)

The General Idea



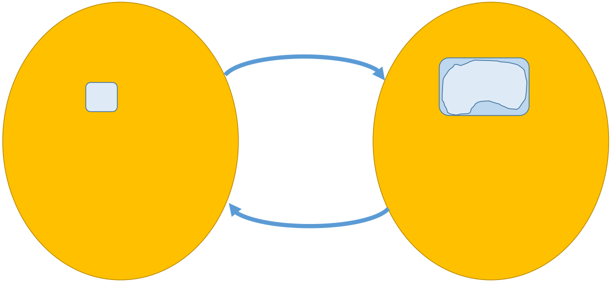
1) Guess a subspace of U

The General Idea



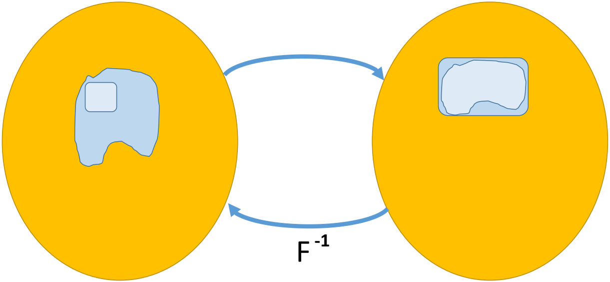
2) Map it using *F*

The General Idea



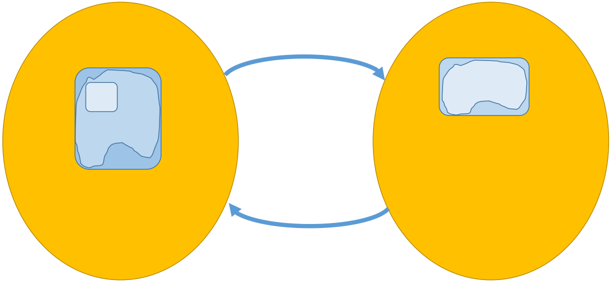
3) Compute the linear span

The General Idea



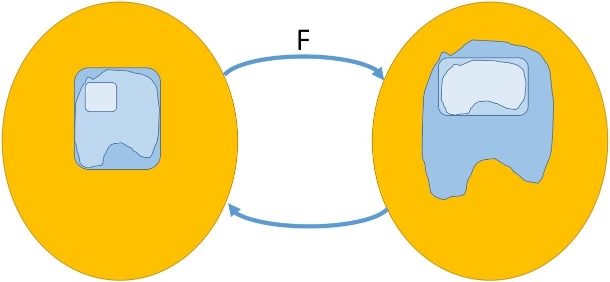
4) Map it using F^{-1}

The General Idea



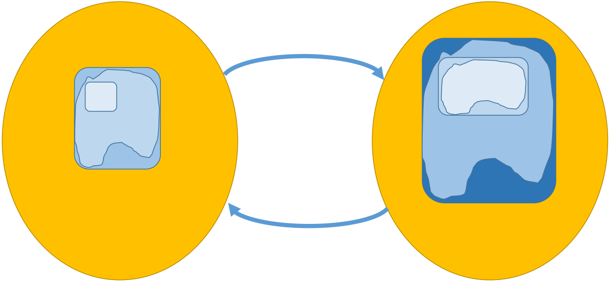
5) Compute the linear span

The General Idea



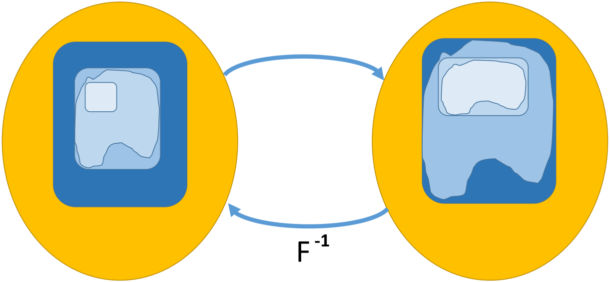
6) Map it using F

The General Idea



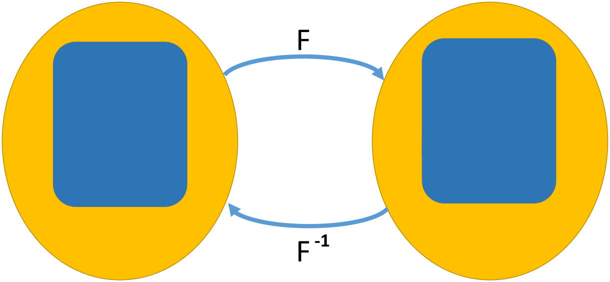
7) Compute the linear span

The General Idea



8) Map it using F^{-1}

The General Idea



9) ...until it stabilizes. Done.

Some Further Considerations

Running Time

Roughly $2^{3(n-d)}$ for the initial guess if an invariant subspace of dim. d exists.

Much better: Include round constants in the initial guess. Guess only the offset.

Reduced Running Time

2^{n-d} when an invariant subspace of dim. d exists.

One Application

FSE 2014: LS-Designs

A family of easy to mask block ciphers

Designed by UC-Louvain and INRIA

Main idea

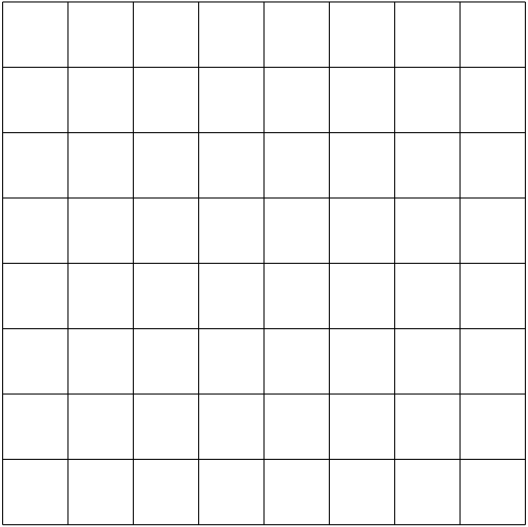
Opposite approach of what is done usually:

- Use tables for the linear-layer
- Use (few) logical operations for S-boxes

Two instances:

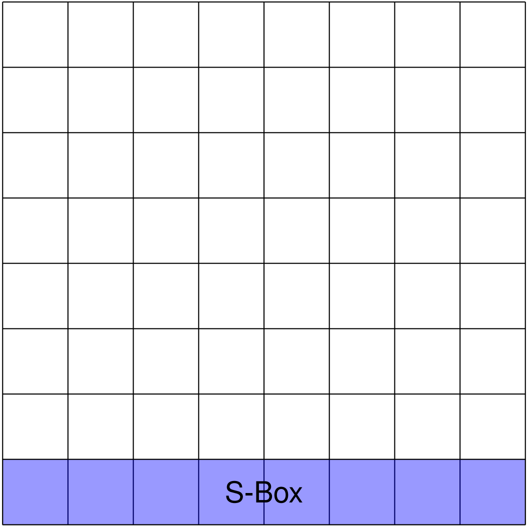
- Robin and iScream
- Fantomas and Scream

Robin and iScream



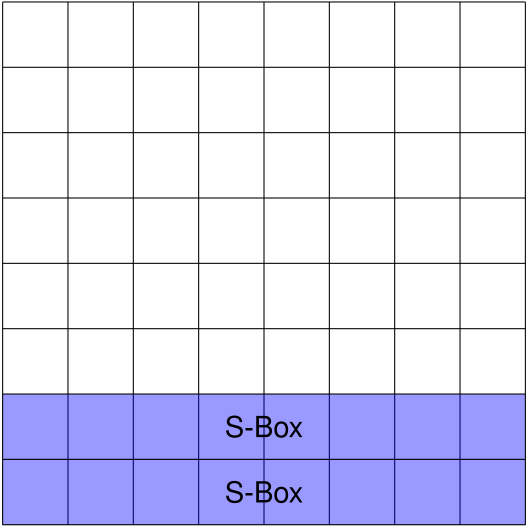
One square is a bit. Columns are stored in registers

Robin and iScream



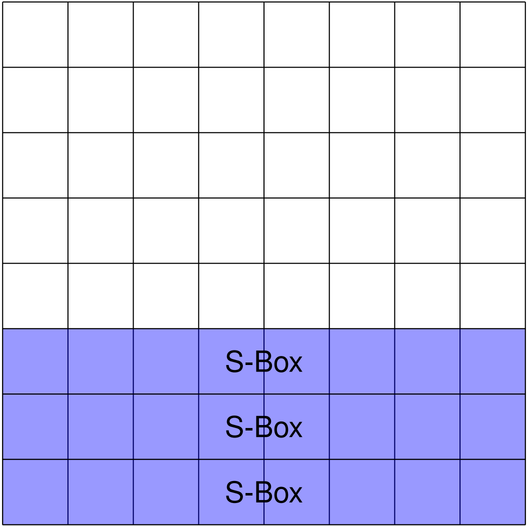
One square is a bit. Columns are stored in registers

Robin and iScream



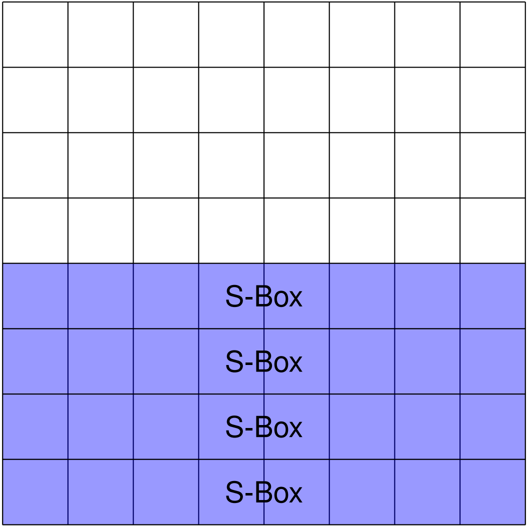
One square is a bit. Columns are stored in registers

Robin and iScream



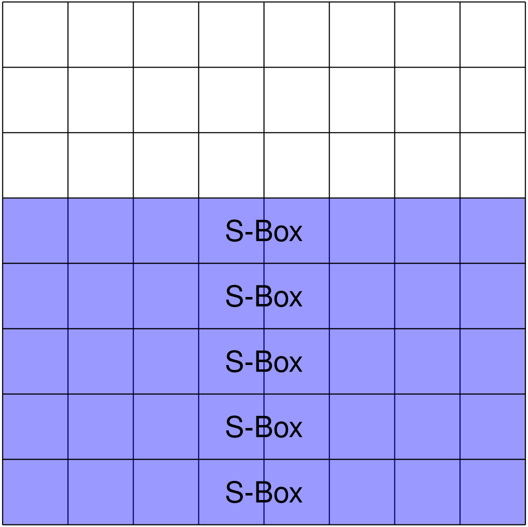
One square is a bit. Columns are stored in registers

Robin and iScream



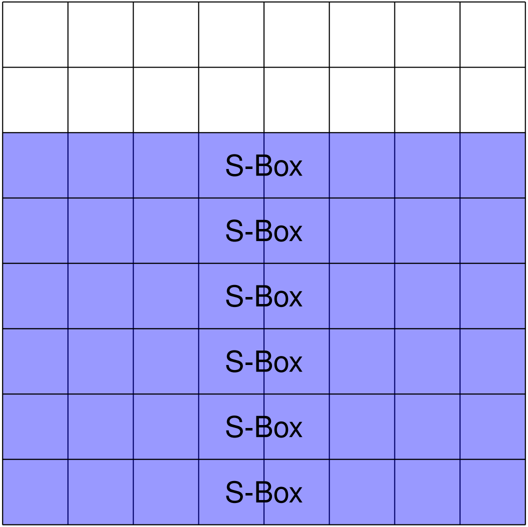
One square is a bit. Columns are stored in registers

Robin and iScream



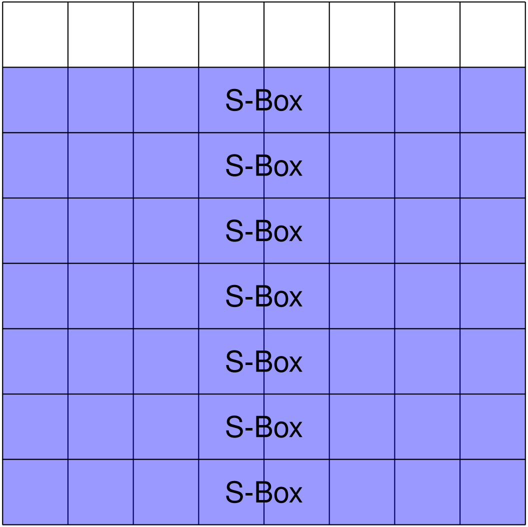
One square is a bit. Columns are stored in registers

Robin and iScream



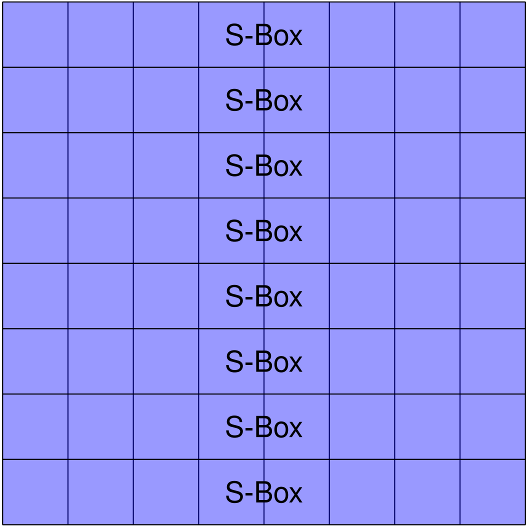
One square is a bit. Columns are stored in registers

Robin and iScream



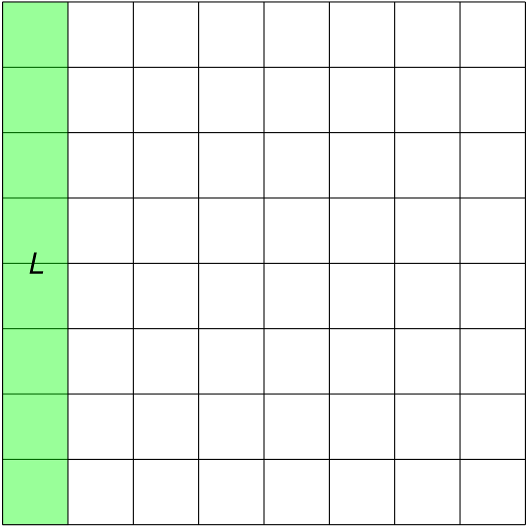
One square is a bit. Columns are stored in registers

Robin and iScream



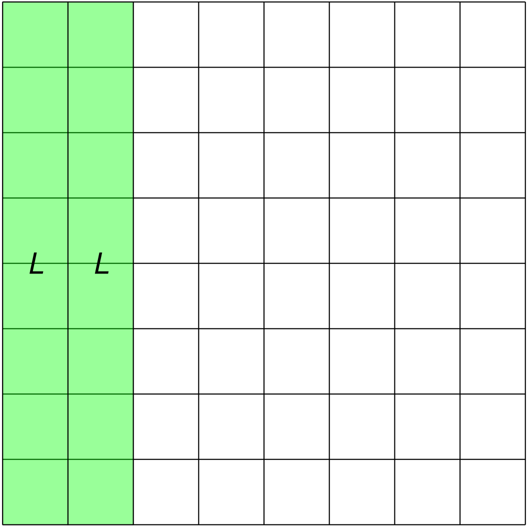
One square is a bit. Columns are stored in registers

Robin and iScream



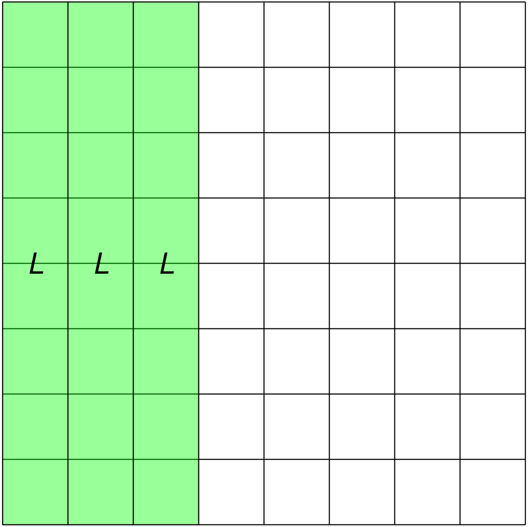
One square is a bit. Columns are stored in registers

Robin and iScream



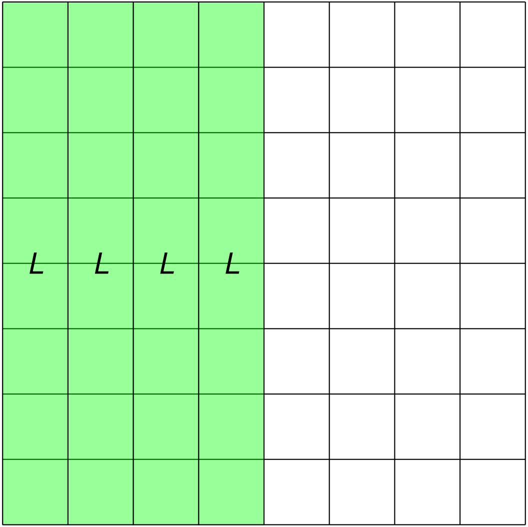
One square is a bit. Columns are stored in registers

Robin and iScream



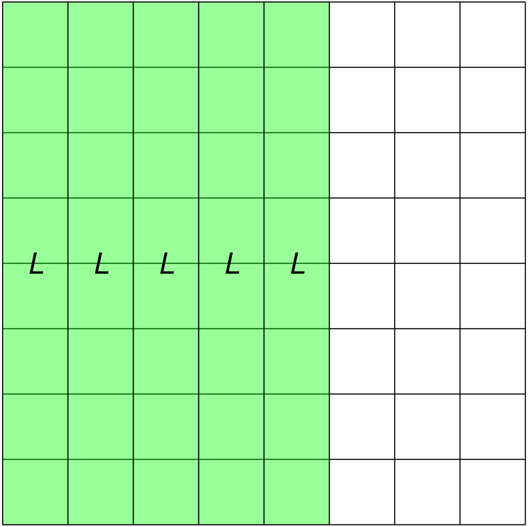
One square is a bit. Columns are stored in registers

Robin and iScream



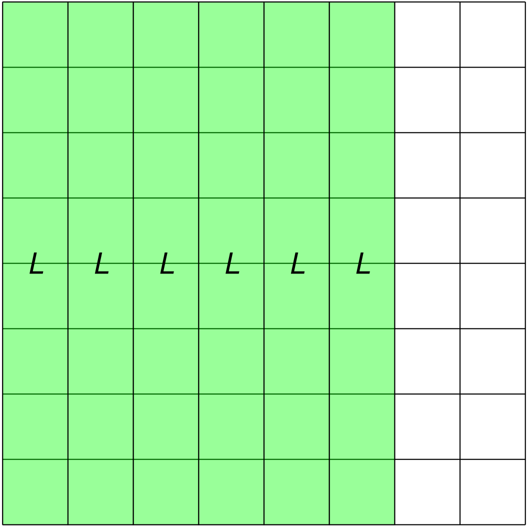
One square is a bit. Columns are stored in registers

Robin and iScream



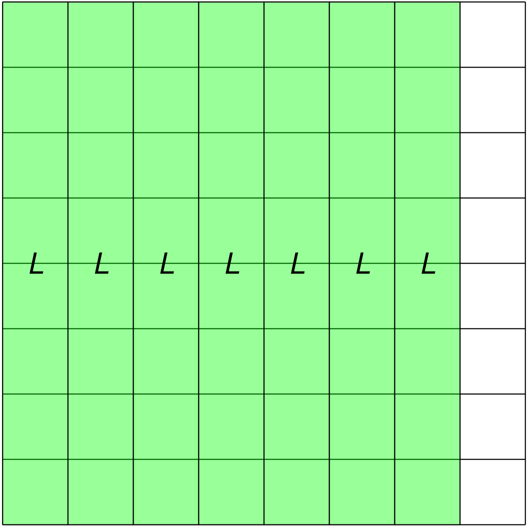
One square is a bit. Columns are stored in registers

Robin and iScream



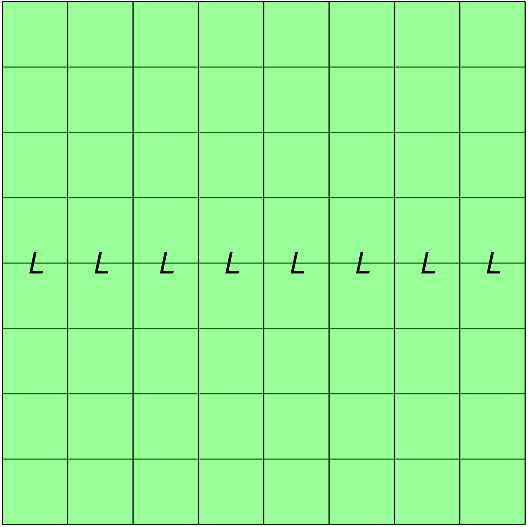
One square is a bit. Columns are stored in registers

Robin and iScream



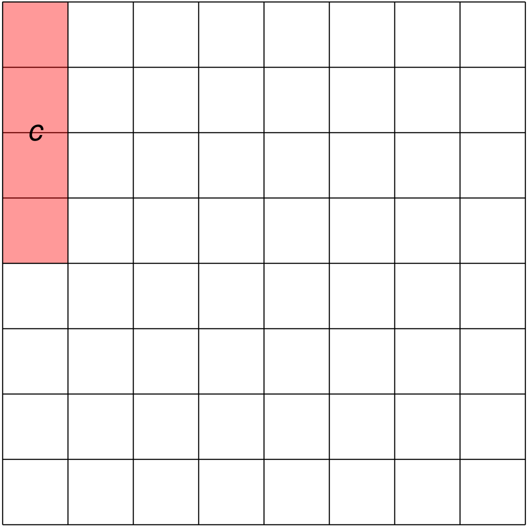
One square is a bit. Columns are stored in registers

Robin and iScream



One square is a bit. Columns are stored in registers

Robin and iScream



One square is a bit. Columns are stored in registers

Applications to Zorro, Robin and iScream

Apply the algorithm to Robin and iScream.

Easy but Powerful

Allows to detect some things

- 32 dim subspace for Robin
- ...and for Zorro

Improve Afterwards

The tool detects a (minimal) invariant subspace. Careful analysis increases attack and understanding.

The Robin Sbox

$$00000000 \to 00000000$$
$$10000000 \to 10100001$$
$$01100100 \to 01100100$$
$$11100100 \to 11000101$$
$$00100001 \to 00100001$$
$$10100001 \to 10000000$$
$$01000101 \to 01000101$$
$$11000101 \to 11100100$$

$$S(*, a, b, 0, 0, a, 0, a \oplus b) = (*, \alpha, \beta, 0, 0, \alpha, 0, \alpha \oplus \beta)$$

A Symmetry in Robin and iScream

*	a_7	b_7	0	0	a_7	0	c_7
*	a_6	b_6	0	0	a_6	0	c_6
*	a_5	b_5	0	0	a_5	0	c_5
*	a_4	b_4	0	0	a_4	0	c_4
*	a_3	b_3	0	0	a_3	0	c_3
*	a_2	b_2	0	0	a_2	0	c_2
*	a_1	b_1	0	0	a_1	0	c_1
*	a_0	b_0	0	0	a_0	0	c_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	a_7	b_7	0 S-Box	0	a_7	0	c_7
*	a_6	b_6	0 S-Box	0	a_6	0	c_6
*	a_5	b_5	0 S-Box	0	a_5	0	c_5
*	a_4	b_4	0 S-Box	0	a_4	0	c_4
*	a_3	b_3	0 S-Box	0	a_3	0	c_3
*	a_2	b_2	0 S-Box	0	a_2	0	c_2
*	a_1	b_1	0 S-Box	0	a_1	0	c_1
*	a_0	b_0	0 S-Box	0	a_0	0	c_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	a_7	b_7	S-Box		a_7	0	c_7
*	a_6	b_6	S-Box		a_6	0	c_6
*	a_5	b_5	S-Box		a_5	0	c_5
*	a_4	b_4	S-Box		a_4	0	c_4
*	a_3	b_3	S-Box		a_3	0	c_3
*	a_2	b_2	S-Box		a_2	0	c_2
*	a_1	b_1	S-Box		a_1	0	c_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	a_7	b_7	S-Box		a_7	0	c_7
*	a_6	b_6	S-Box		a_6	0	c_6
*	a_5	b_5	S-Box		a_5	0	c_5
*	a_4	b_4	S-Box		a_4	0	c_4
*	a_3	b_3	S-Box		a_3	0	c_3
*	a_2	b_2	S-Box		a_2	0	c_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$

A Symmetry in Robin and iScream

*	a_7	b_7	S-Box		a_7	0	c_7
*	a_6	b_6	S-Box		a_6	0	c_6
*	a_5	b_5	S-Box		a_5	0	c_5
*	a_4	b_4	S-Box		a_4	0	c_4
*	a_3	b_3	S-Box		a_3	0	c_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$

A Symmetry in Robin and iScream

*	a_7	b_7	S-Box		a_7	0	c_7
*	a_6	b_6	S-Box		a_6	0	c_6
*	a_5	b_5	S-Box		a_5	0	c_5
*	a_4	b_4	S-Box		a_4	0	c_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	a_7	b_7	S-Box		a_7	0	c_7
*	a_6	b_6	S-Box		a_6	0	c_6
*	a_5	b_5	S-Box		a_5	0	c_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \qquad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	a_7	b_7	0 S-Box	0	a_7	0	c_7
*	a_6	b_6	0 S-Box	0	a_6	0	c_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$

A Symmetry in Robin and iScream

*	a_7	b_7	0 S-Box	0	a_7	0	c_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
L	L	L	L	L	L	L	L
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
	L	L	L	L	L	L	L
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
		L	L	L	L	L	L
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
			L	L	L	L	L
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
				L	L	L	L
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4 L	0 L	γ_4 L
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

L L

$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

L

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

c

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$$

A Symmetry in Robin and iScream

*	α_7	β_7	0	0	α_7	0	γ_7
*	α_6	β_6	0	0	α_6	0	γ_6
*	α_5	β_5	0	0	α_5	0	γ_5
*	α_4	β_4	0	0	α_4	0	γ_4
*	α_3	β_3	0	0	α_3	0	γ_3
*	α_2	β_2	0	0	α_2	0	γ_2
*	α_1	β_1	0	0	α_1	0	γ_1
*	α_0	β_0	0	0	α_0	0	γ_0

$c_i = a_i \oplus b_i \quad \gamma_i = \alpha_i \oplus \beta_i$

Generalization

Question

Can we generalize this attack?

Possible directions:

- Statistical Variant
- Add key-recovery
- Not focus on subspaces only

Outline

Non-linear Invariant Attacks

- ASIACRYPT 2016
- joint work with Yosuke Todo and Yu Sasaki (NTT)
- Developed not like the storyline suggests.

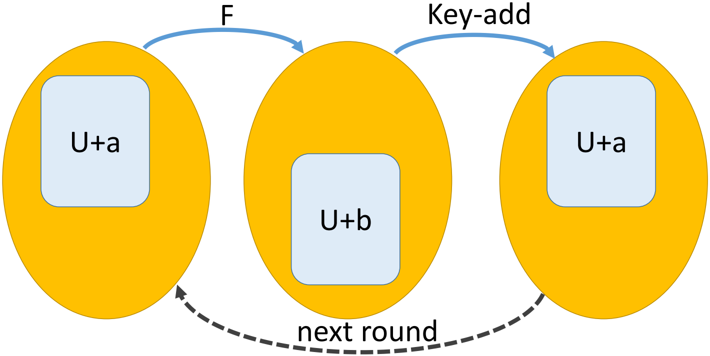
Nonlinear Invariant Attack

Practical Attack on Full SCREAM, iSCREAM, and Midori64

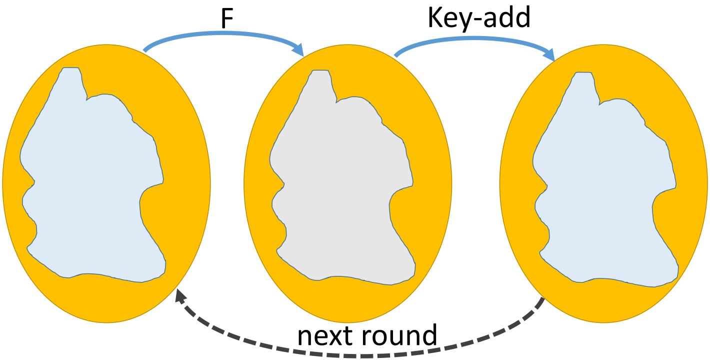
Yosuke Todo and Gregor Leander and Yu Sasaki

Abstract. In this paper we introduce a new type of attack, called *non-linear invariant attack*. As application examples, we present new attacks that are able to distinguish the full versions of the (tweakable) block ciphers Scream, iScream and Midori64 in a weak-key setting. Those attacks require only a handful of plaintext-ciphertext pairs and have minimal computational costs. Moreover, the nonlinear invariant attack on the underlying (tweakable) block cipher can be extended to a ciphertext-

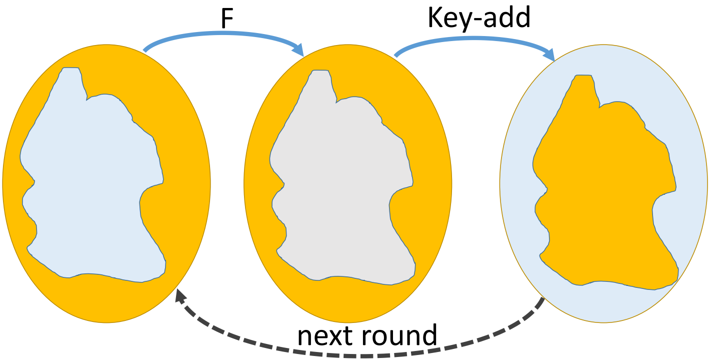
Invariant Subspace Attacks



Nonlinear Invariant Attack (I/II)



Invariant Subspace Attacks (II/II)



Basics

Definition

Given a permutation $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$. A Boolean function $g : \mathbb{F}_2^n \to \mathbb{F}_2$ is called a *non linear invariant of F* if

$$g(F(x)) = g(x) + c \quad \forall x$$

where $c \in \mathbb{F}_2$ is a constant.

Link to the picture:

1. Split $\mathbb{F}_2^n$ into two sets

$$\begin{aligned} A &:= \{x \mid g(x) = 1\} \\ B &:= \{x \mid g(x) = 0\} \end{aligned}$$

2. $F(A) = A$ and $F(B) = B$ $(c = 0)$
3. $F(A) = B$ and $F(B) = A$ $(c = 1)$

Applications

Applications

This leads to attacks on

- iSCREAM
- Midori64
- SCREAM (v.3)

Can be extended to a cipher-text only attack

- when used in almost all modes (e.g. CBC, CTR) mode
- same message encrypted multiple times

with very low complexity.

Results

	weak keys	recovered bits	data	time
SCREAM (v.3)	2^{96}	1/4	33 CT	32^3
iSCREAM	2^{96}	1/4	33 CT	32^3
Midori64	2^{64}	1/2	33 CT	32^3

More details in the paper. In particular

- The details
- An explanation why that attack works on those ciphers

Outline

To appear: CRYPTO 2017

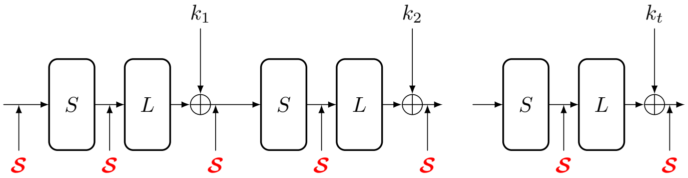
Proving Resistance against Invariant Attacks: How to Choose the Round Constants

Christof Beierle[1], Anne Canteaut[2], Gregor Leander[1], and Yann Rotella[2]

[1] Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
{christof.beierle, gregor.leander}@rub.de
[2] Inria, Paris, France
{anne.canteaut, yann.rotella}@inria.fr

Abstract. Many lightweight block ciphers apply a very simple key schedule in which the round keys only differ by addition of a round-specific constant. Generally, there is not much theory on how to choose appropriate constants. In fact, several of those schemes were recently broken using invariant attacks, i.e. invariant subspace or nonlinear invariant attacks. This work analyzes the resistance of such ciphers against invariant attacks and reveals the precise mathematical properties that render those attacks applicable. As a first practical consequence, we prove that some ciphers including Prince, Skinny-64 and Mantis_7 are not vulnerable to invariant attacks. Also, we show that the invariant factors of the linear

Invariants under L and S



Focus on invariants that are

- Invariant for S-Layer
- Invariant for all $\text{Add}_{k_i} \circ L$

Not much of a restriction!?

Known attacks are of this form.

Implication

$$g(L(x)+k_i) = g(x)+\varepsilon_i \text{ and } g(L(x)+k_j) = g(x)+\varepsilon_j$$
$$\Rightarrow g(L(x)+k_i) = g(L(x)+k_j)+(\varepsilon_i+\varepsilon_j)$$
$$\Leftrightarrow g(y+k_i+k_j) = g(y)+(\varepsilon_i+\varepsilon_j)$$

Linear Structure

$(k_i + k_j)$ is a linear structure of g.

Recall:

Linear space of a Boolean function g

$$\mathrm{LS}(g) := \{\alpha \in \mathbb{F}_2^n : x \mapsto g(x+\alpha)+g(x) \text{ is constant}\}$$

More Implications

Lemma

Let g be an invariant

- *for S-Layer*
- *for all* $Add_{k_i} \circ L$

then

- $\mathrm{LS}(g)$ *contains* $k_i + k_j$
- $\mathrm{LS}(g)$ *is invariant under L.*

Focus on the simplest key-scheduling:

$$k_i = k + c_i$$

That is

$$k_i + k_j = c_i + c_j$$

Existence of Non-Trivial Non-linear Invariant

Given

$$D := \{(c_i + c_j) \mid i, j \in \{1, \ldots, r\}\}$$

we define

$$W_L(D) := \text{smallest L-invariant subspace containing } D$$

Question

Is there a non-trivial invariant g for the S-Layer such that

$$W_L(D) \subseteq \mathrm{LS}(g)?$$

Dimension of $W_L(D)$

Corollary

If $\dim(W_L(D)) \geq n - 1$ *than such a g does not exist.*

Proof.

Otherwise S-Layer has linear component. □

Proves that the attack does not work for e.g.

- LED
- Skinny-64-64

More General

Theorem

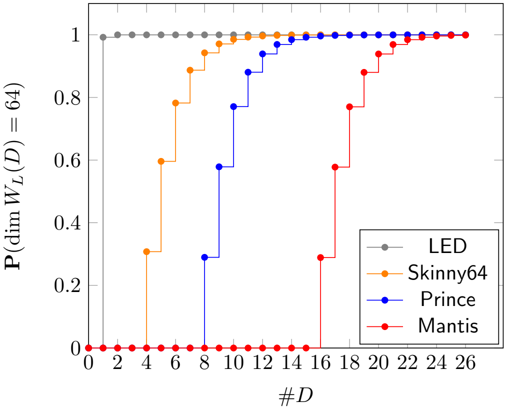
Let $Q_1, \ldots Q_r$ be the invariant factors of L. For any $t \leq r$

$$\max_{c_1,\ldots,c_t} \dim W_L(\{c_1, \ldots, c_t\}) = \sum_{i=1}^{t} \deg Q_i$$

Study the invariant factors of the linear layer!

- Explains required number of constants
- Explains how to choose them
- Works independent of S-layer.

Examples


$\mathbf{P}(\dim W_L(D) = 64)$
0
0.2
0.4
0.6
0.8
1
0 2 4 6 8 10 12 14 16 18 20 22 24 26
$\#D$
LED
Skinny64
Prince
Mantis

The End

Thank you for your attention.