Generalized Nonlinear Invariant Attack and a New Design Criterion for Round Constants

Yongzhuang Wei[1] Tao Ye[1] Wenling Wu[2] Enes Pasalic[3]

Presented by René Rodríguez[3]

Fast Software Encryption 2019, Paris, France

[1]Guilin University of Electronic Technology, P.R. China
[2]TCA Laboratory, SKLCS, Institute of Software, Chinese Academy of Sciences, P.R. China
[3]University of Primorska, FAMNIT, Slovenia

March 25th, 2019

Summary of the talk

Summary of the talk

Nonlinear invariant attack

Nonlinear invariant attack

Attack was introduced by Todo, Leander and Sasaki in 2016.

Nonlinear invariant attack

Attack was introduced by Todo, Leander and Sasaki in 2016.

Core idea

Considering an n-bit block cipher whose encryption function is $E(x, k)$, look for a non-linear Boolean function $g : GF(2)^n \to GF(2)$ such that

$$g(x) \oplus g(E(x, k)) = \text{constant} \quad \forall x.$$

Nonlinear invariant attack

Attack was introduced by Todo, Leander and Sasaki in 2016.

Core idea

Considering an n-bit block cipher whose encryption function is $E(x, k)$, look for a non-linear Boolean function $g : GF(2)^n \to GF(2)$ such that

$$g(x) \oplus g(E(x, k)) = \text{constant} \quad \forall x.$$

- We call g a **nonlinear invariant** for $E(x, k)$,

Nonlinear invariant attack

Attack was introduced by Todo, Leander and Sasaki in 2016.

Core idea

Considering an n-bit block cipher whose encryption function is $E(x, k)$, look for a non-linear Boolean function $g : GF(2)^n \to GF(2)$ such that

$$g(x) \oplus g(E(x, k)) = \text{constant} \quad \forall x.$$

- We call g a **nonlinear invariant** for $E(x, k)$,
- Those keys which admit a nonlinear invariant are called **weak keys**.

Nonlinear invariant attack

Attack was introduced by Todo, Leander and Sasaki in 2016.

Core idea

Considering an n-bit block cipher whose encryption function is $E(x,k)$, look for a non-linear Boolean function $g : GF(2)^n \to GF(2)$ such that

$$g(x) \oplus g(E(x,k)) = \text{constant} \quad \forall x.$$

- We call g a **nonlinear invariant** for $E(x,k)$,
- Those keys which admit a nonlinear invariant are called **weak keys**.

Why is it important?

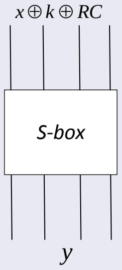
Commonly induce distinguishing attacks, especially lightweight block ciphers are susceptible to this kind of cryptanalysis.

Example

Let $g : F_2^4 \to F_2$ *be a nonlinear function defined as*

$$g(a_4, a_3, a_2, a_1) = a_4 a_3 \oplus a_3 \oplus a_2 \oplus a_1$$

$$a_i = x_i \oplus k_i \oplus RC_i$$



If $k_3 \oplus RC_3 = 0$ *and* $k_4 \oplus RC_4 = 0$,
then, $g(x_4, x_3, x_2, x_1) \oplus g(y_4, y_3, y_2, y_1) = c$ *for all* x

If $k_3 \oplus RC_3 \neq 0$ *or* $k_4 \oplus RC_4 \neq 0$, *then,*
$g(x_4, x_3, x_2, x_1) \oplus g(y_4, y_3, y_2, y_1) \neq c$ *for all* x

Vulnerable lightweight block ciphers

- PRINT-cipher [Leander et al. 2011]
- iSCREAM, Robin, Zorro [Leander, Minaud, Rønjom 2015]
- Midori-64 [Guo et al. 2016]
- iSCREAM, SCREAM, Midori-64 [Todo, Leander, Sasaki 2016]
- Simpira v1 [Rønjom 2016]
- Haraka v.0 [Jean 2016]
- NORX v2.0 [Chaigneau et al. 2017]

How to provide resistance?

How to provide resistance?

Beierle, Canteaut, Leander and Rotella (BCLR) in 2017 studied the mathematical nature of these invariants providing certain conditions under which an iterated block cipher could be resistant against invariant attacks.

How to provide resistance?

Beierle, Canteaut, Leander and Rotella (BCLR) in 2017 studied the mathematical nature of these invariants providing certain conditions under which an iterated block cipher could be resistant against invariant attacks.

Theorem

Let g be an **invariant** *of the substitution layer and of the linear parts $Add_{k_i} \circ L$ (including addition of the keys). Then $LS(g)$ must be a subspace invariant under L containing all the differences of the keys.*

How to provide resistance?

Beierle, Canteaut, Leander and Rotella (BCLR) in 2017 studied the mathematical nature of these invariants providing certain conditions under which an iterated block cipher could be resistant against invariant attacks.

Theorem

Let g be an **invariant** *of the substitution layer and of the linear parts $Add_{k_i} \circ L$ (including addition of the keys). Then $LS(g)$ must be a subspace invariant under L containing all the differences of the keys.*

$LS(g)$ is a subspace of **linear structures** and $W_L(c)$ is the **minimal L-invariant subspace** containing c.

How to provide resistance?

Beierle, Canteaut, Leander and Rotella (BCLR) in 2017 studied the mathematical nature of these invariants providing certain conditions under which an iterated block cipher could be resistant against invariant attacks.

Theorem

Let g be an **invariant** *of the substitution layer and of the linear parts $Add_{k_i} \circ L$ (including addition of the keys). Then $LS(g)$ must be a subspace invariant under L containing all the differences of the keys.*

$LS(g)$ is a subspace of **linear structures** and $W_L(c)$ is the **minimal** L**-invariant subspace** containing c.

Need that $W_L(D) \subseteq LS(g)$ where D is a set of differences of keys.

Main conclusions of BCLR are:

Main conclusions of BCLR are:

- Assuming $\dim W_L(D) \geq n-1$ implies $\deg(g)$ is trivial (constant function), [Independent of the S-Layer]

Main conclusions of BCLR are:

- Assuming $\dim W_L(D) \geq n-1$ implies $\deg(g)$ is trivial (constant function), [Independent of the S-Layer]
- In some cases when $n - \dim W_L(D)$ is small, they found certain structure of the S-Layer which allows to conclude that there are no non-trivial invariants.

Main conclusions of BCLR are:

- Assuming $\dim W_L(D) \geq n-1$ implies $\deg(g)$ is trivial (constant function), [Independent of the S-Layer]
- In some cases when $n - \dim W_L(D)$ is small, they found certain structure of the S-Layer which allows to conclude that there are no non-trivial invariants.

The following lightweight ciphers are resistant against invariant attacks.

- *Skinny*-64,
- *Prince*,
- $Mantis_7$

Nonlinear invariant attacks can lead not only to distinguishing attacks but sometimes to more dangerous scenarios (ciphertext-only attack in certain modes of operation).

Nonlinear invariant attacks can lead not only to distinguishing attacks but sometimes to more dangerous scenarios (ciphertext-only attack in certain modes of operation).
Two natural questions arising here are:

Nonlinear invariant attacks can lead not only to distinguishing attacks but sometimes to more dangerous scenarios (ciphertext-only attack in certain modes of operation).
Two natural questions arising here are:

- Are there more similar attacks?

Nonlinear invariant attacks can lead not only to distinguishing attacks but sometimes to more dangerous scenarios (ciphertext-only attack in certain modes of operation).
Two natural questions arising here are:

- Are there more similar attacks?
- Moreover, how can we protect ciphers against them?

Nonlinear invariant attacks can lead not only to distinguishing attacks but sometimes to more dangerous scenarios (ciphertext-only attack in certain modes of operation).
Two natural questions arising here are:

- Are there more similar attacks?
- Moreover, how can we protect ciphers against them?

Goal of the paper

Provide useful generalizations of nonlinear invariant attacks.

Summary of the talk

Generalized Nonlinear Invariants

Main idea

Look for a nonlinear Boolean function g and a pair $a_1, a_2 \in GF(2)^n$, such that $g(x \oplus a_1) \oplus g(F_{k_i}(x) \oplus a_2) = \text{const.} \quad \forall x.$

Generalized Nonlinear Invariants

Main idea

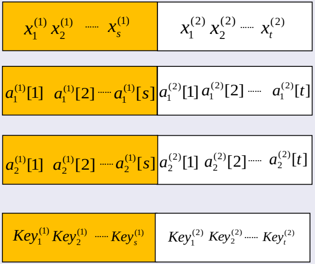
Look for a nonlinear Boolean function g and a pair $a_1, a_2 \in GF(2)^n$, such that $g(x \oplus a_1) \oplus g(F_{k_i}(x) \oplus a_2) = \text{const.} \quad \forall x$.

These are called, **generalized nonlinear invariants**, where $F_{k_i}(x)$ is a round function. For any function F, let us denote by

$$U(F, a_1, a_2) := \{g : F_2^m \to F_2 | g(x \oplus a_1) = g(F(x) \oplus a_2) \oplus c\}$$

We assume that the nonlinear terms of $g(x)$ only cover the first s input variables, and the remaining t variables have a linear relation, i.e $g(x) = f(x^{(1)}) \oplus l(x^{(2)})$.

If the round subkeys Key_j and the constants $a_i, (i = 1, 2)$ satisfy any one of the following two conditions:



(1) $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$;

(2) $a_1^{(1)} \neq \mathbf{0}, a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$,

The generalized nonlinear invariant attack can work on the full-round block cipher.

Case 1: for $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

Case 1: for $a_1^{(1)} = \mathbf{0}$, $a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$g(C) \quad = \quad g(x_r \oplus a_2 \oplus a_2)$$

Case 1: for $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$\begin{aligned} g(C) &= g(x_r \oplus a_2 \oplus a_2) \\ &= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_2 \oplus a_2) \end{aligned}$$

Case 1: for $a_1^{(1)} = \mathbf{0}$, $a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$\begin{aligned} g(C) &= g(x_r \oplus a_2 \oplus a_2) \\ &= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_2 \oplus a_2) \\ &= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \end{aligned}$$

Case 1: for $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$\begin{aligned}
g(C) &= g(x_r \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \oplus c_{r-1}
\end{aligned}$$

Case 1: for $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$
\begin{aligned}
g(C) &= g(x_r \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \oplus c_{r-1} \\
&= g(x_{r-1} \oplus a_2 \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1}
\end{aligned}
$$

Case 1: for $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$
\begin{aligned}
g(C) &= g(x_r \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \oplus c_{r-1} \\
&= g(x_{r-1} \oplus a_2 \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1} \\
&\dots
\end{aligned}
$$

Case 1: for $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$
\begin{aligned}
g(C) &= g(x_r \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \oplus c_{r-1} \\
&= g(x_{r-1} \oplus a_2 \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1} \\
&\dots \\
&= g(P) \oplus \sum_{i=0}^{r-1} l(Key_i^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus \sum_{j=0}^{r-1} c_j
\end{aligned}
$$

Case 1: for $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$
\begin{aligned}
g(C) &= g(x_r \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \oplus c_{r-1} \\
&= g(x_{r-1} \oplus a_2 \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1} \\
&\dots \\
&= g(P) \oplus \sum_{i=0}^{r-1} l(Key_i^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus \sum_{j=0}^{r-1} c_j
\end{aligned}
$$

Case 1: for $a_1^{(1)} = \mathbf{0}, a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$
\begin{aligned}
g(C) &= g(x_r \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)}) \oplus c_{r-1} \\
&= g(x_{r-1} \oplus a_2 \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1} \\
&\dots \\
&= g(P) \oplus \sum_{i=0}^{r-1} l(Key_i^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus \sum_{j=0}^{r-1} c_j
\end{aligned}
$$

Moreover, we have

$$g(P) \oplus g(C) = Constant'.$$

Case 2: for $a_1^{(1)} \neq \mathbf{0}, a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

Case 2: for $a_1^{(1)} \neq \mathbf{0}$, $a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$g(C \oplus a_1) \quad = \quad g(x_r \oplus a_1 \oplus a_2 \oplus a_2)$$

Case 2: for $a_1^{(1)} \neq \mathbf{0}$, $a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$\begin{aligned} g(C \oplus a_1) &= g(x_r \oplus a_1 \oplus a_2 \oplus a_2) \\ &= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_1 \oplus a_2 \oplus a_2) \end{aligned}$$

Case 2: for $a_1^{(1)} \neq \mathbf{0}, a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$\begin{aligned} g(C \oplus a_1) &= g(x_r \oplus a_1 \oplus a_2 \oplus a_2) \\ &= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_1 \oplus a_2 \oplus a_2) \\ &= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \end{aligned}$$

Case 2: for $a_1^{(1)} \neq \mathbf{0}, a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$
\begin{aligned}
g(C \oplus a_1) &= g(x_r \oplus a_1 \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_1 \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1}
\end{aligned}
$$

Case 2: for $a_1^{(1)} \neq \mathbf{0}, a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$\begin{aligned}
g(C \oplus a_1) &= g(x_r \oplus a_1 \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_1 \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus I(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus I(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1} \\
&\dots
\end{aligned}$$

Case 2: for $a_1^{(1)} \neq \mathbf{0}, a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$\begin{aligned}
g(C \oplus a_1) &= g(x_r \oplus a_1 \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_1 \oplus a_2 \oplus a_2) \\
&= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \\
&= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1} \\
\dots \\
&= g(P \oplus a_1) \oplus \sum_{i=0}^{r-1} l(Key_i^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus \sum_{j=0}^{r-1} c_j
\end{aligned}$$

Case 2: for $a_1^{(1)} \neq \mathbf{0}, a_1^{(1)} \oplus a_2^{(1)} \oplus Key_j^{(1)} = \mathbf{0}$, we have

$$\begin{aligned} g(C \oplus a_1) &= g(x_r \oplus a_1 \oplus a_2 \oplus a_2) \\ &= g(F(x_{r-1}) \oplus Key_{r-1} \oplus a_1 \oplus a_2 \oplus a_2) \\ &= g(F(x_{r-1}) \oplus a_2) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \\ &= g(x_{r-1} \oplus a_1) \oplus l(Key_{r-1}^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus c_{r-1} \\ &\dots \\ &= g(P \oplus a_1) \oplus \sum_{i=0}^{r-1} l(Key_i^{(2)} \oplus a_2^{(2)} \oplus a_1^{(2)}) \oplus \sum_{j=0}^{r-1} c_j \end{aligned}$$

Moreover, we have

$$g(P \oplus a_1) \oplus g(C \oplus a_1) = Constant'.$$

Generalized Nonlinear Invariant Attack

Distinguishing Attack by using *Generalized Nonlinear Invariant Attack*

Assume that $(P_i, C_i), (i = 1, ..., N)$ *are* N *pairs of plaintexts and ciphtexts. In a known-plaintext attack scenario, the adversary can easily determine whether* $g(P) \oplus g(C)$ *(or* $g(P \oplus a_1) \oplus g(C \oplus a_1)$ *) is constant or not for all pairs. It is clear that any random permutation has this property with a probability of* 2^{1-N} *if* g *is balanced.*

Standard procedure:

Standard procedure:

- Viewing the round function as $\mathcal{L} \circ \mathcal{S}$, one first finds a set of generalized invariants $U(S, a_1, a_2)$ for a single S-box.

Standard procedure:

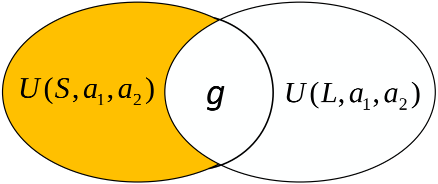
- Viewing the round function as $\mathcal{L} \circ \mathcal{S}$, one first finds a set of generalized invariants $U(S, a_1, a_2)$ for a single S-box.
- Combine these to get an invariant of the entire S-box layer $\mathcal{S}$ as $g_{\mathcal{S}} = \sum_{i=1}^{m} \beta_i g_i$, with $\beta_i \in \{0, 1\}$.

Standard procedure:

- Viewing the round function as $\mathcal{L} \circ \mathcal{S}$, one first finds a set of generalized invariants $U(S, a_1, a_2)$ for a single S-box.

- Combine these to get an invariant of the entire S-box layer $\mathcal{S}$ as $g_{\mathcal{S}} = \sum_{i=1}^{m} \beta_i g_i$, with $\beta_i \in \{0, 1\}$.

- If $\mathcal{L}$ can be viewed as an orthogonal matrix and $\deg(g) = 2$ then one can easily specify invariant for a whole round.



Assumptions-more formally

Theorem

Assume that $g_i \in U(S, a_1, a_2)$, $i = 1, \ldots, m$ are arbitrary generalized nonlinear invariants of a given S-box. Define

$$g_S(x_1, \ldots, x_m) = \sum_{i=1}^{m} \beta_i g_i(x_i), \quad \beta_i \in GF(2),$$

which is a **generalized nonlinear invariant of entire S-box layer**.

Assumptions-more formally

Theorem

Assume that $g_i \in U(S, a_1, a_2)$, $i = 1, \ldots, m$ are arbitrary generalized nonlinear invariants of a given S-box. Define

$$g_S(x_1, \ldots, x_m) = \sum_{i=1}^{m} \beta_i g_i(x_i), \quad \beta_i \in GF(2),$$

which is a **generalized nonlinear invariant of entire S-box layer**.

Theorem

For SPN network if L is an **orthogonal matrix** *$M \in GF(2)^{m \times m}$ and $g' \in U(S, a_1, a_2)$ is* **quadratic**, *then $g(x_1, \ldots, x_m) = \sum_{i=1}^{m} g'(x_i)$ is also a generalized nonlinear invariant for the round function $L \circ \mathcal{S}$.*

Are generalized invariants useful?

Are generalized invariants useful?

- Lead to an efficient distinguishing attack on iSCREAM under weak key assumption (identifying $2^{96} + 2^{80}$ weak keys)

Are generalized invariants useful?

- Lead to an efficient distinguishing attack on iSCREAM under weak key assumption (identifying $2^{96} + 2^{80}$ weak keys)
- Weak keys are **different** from those found for standard nonlinear invariants of iSCREAM

Are generalized invariants useful?

- Lead to an efficient distinguishing attack on iSCREAM under weak key assumption (identifying $2^{96} + 2^{80}$ weak keys)
- Weak keys are **different** from those found for standard nonlinear invariants of iSCREAM

Generalized nonlinear invariants are translates of standard invariants.

Are generalized invariants useful?

- Lead to an efficient distinguishing attack on iSCREAM under weak key assumption (identifying $2^{96} + 2^{80}$ weak keys)
- Weak keys are **different** from those found for standard nonlinear invariants of iSCREAM

Generalized nonlinear invariants are translates of standard invariants.

Remark

If some nonlinear term of g involves a nonzero bit of the round constant $c*$, then the classical invariant attack becomes rather inefficient. A pair of constants (a_1, a_2), can be helpful for eliminating the impact of this !!

Summary of the talk

Is the BCLR criterion optimal?

A large dimension of $W_L(D)$ should prevent from invariant attacks (regardless of S layer)?! When $W_L(D) \geq n-1$ block cipher is provably resistant against these attacks.

Is the BCLR criterion optimal?

A large dimension of $W_L(D)$ should prevent from invariant attacks (regardless of S layer)?! When $W_L(D) \geq n-1$ block cipher is provably resistant against these attacks.

Definition

For any S-box define the closed-loop invariant $CLI(S)$ as the following set

$$\{(g_1, g_2) : g_1(x) \oplus g_2(S(x)) = c_1, g_2(x) \oplus g_1(S(x)) = c_2, c_i \in GF(2)\}$$

Is the BCLR criterion optimal?

A large dimension of $W_L(D)$ should prevent from invariant attacks (regardless of S layer)?! When $W_L(D) \geq n-1$ block cipher is provably resistant against these attacks.

Definition

For any S-box define the closed-loop invariant $CLI(S)$ as the following set

$$\{(g_1, g_2) : g_1(x) \oplus g_2(S(x)) = c_1, g_2(x) \oplus g_1(S(x)) = c_2, c_i \in GF(2)\}$$

- $CLI(S)$ is a linear subspace

Is the BCLR criterion optimal?

A large dimension of $W_L(D)$ should prevent from invariant attacks (regardless of S layer)?! When $W_L(D) \geq n-1$ block cipher is provably resistant against these attacks.

Definition

For any S-box define the closed-loop invariant $CLI(S)$ as the following set

$$\{(g_1, g_2) : g_1(x) \oplus g_2(S(x)) = c_1, g_2(x) \oplus g_1(S(x)) = c_2, c_i \in GF(2)\}$$

- $CLI(S)$ is a linear subspace
- For every $g \in U(S)$, $(g, g) \in CLI(S)$ and $(g, 1 \oplus g) \in CLI(S)$

Is the BCLR criterion optimal?

A large dimension of $W_L(D)$ should prevent from invariant attacks (regardless of S layer)?! When $W_L(D) \geq n-1$ block cipher is provably resistant against these attacks.

Definition

For any S-box define the closed-loop invariant $CLI(S)$ as the following set

$$\{(g_1, g_2) : g_1(x) \oplus g_2(S(x)) = c_1, g_2(x) \oplus g_1(S(x)) = c_2, c_i \in GF(2)\}$$

- $CLI(S)$ is a linear subspace
- For every $g \in U(S)$, $(g, g) \in CLI(S)$ and $(g, 1 \oplus g) \in CLI(S)$
- Usually there are more elements in $CLI(S)$ than those induced by standard invariants !!

Midori64

Midori64 uses an SPN structure and a very simple key schedule. The initial state of Midori64 can be seen as a 4×4-nibble array. A 64-bit plaintext is first input into the initial state and the key pre-whitening operation is performed. Then the state is iteratively operated 16 times with the round function. At last, the state is XORed with the post whitening key.

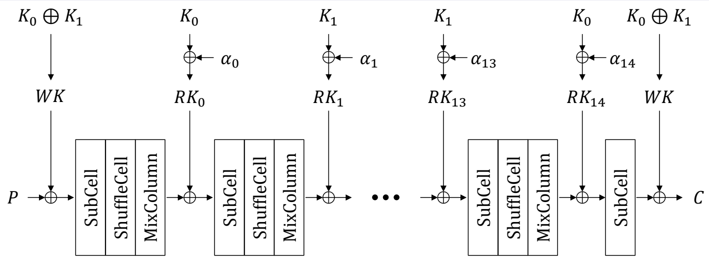


Figure: The structure of Midori64

The variant of Midori64

Construction

The Midori64 variant shares the same round function and key schedule scheme as the original Midori64. However, the only different place is that the round constants are selected from the following parameters:

Let $\alpha_i^* = ({\alpha^*}_i^1 || ... || {\alpha^*}_i^{16}), {\alpha^*}_i^j \in GF(2)^4, (i = 0, 1, ..., 14), (j = 1, ..., 16).$

(1)If $i \bmod 2 = 1$, *the 1st and 3rd bits of* ${\alpha^*}_i^j$ *are always 0.*

(2)If $i \bmod 2 = 0$, α_i^* *can choose random round constants.*

The Closed-loop Invariant for Midori64

1. For the S-box of Midori64, we can find the Closed-loop invariant below.

$$\begin{cases} g_1'(x[4],\ldots,x[1]) \oplus g_2'(y[4],\ldots,y[1]) = 1 \\ g_2'(x[4],\ldots,x[1]) \oplus g_1'(y[4],\ldots,y[1]) = 1 \\ g_1' = x[1] \oplus x[2] \oplus x[4] \oplus x[1]x[3] \\ g_2' = y[1] \oplus y[3] \end{cases}$$

The Closed-loop Invariant for Midori64

1. For the S-box of Midori64, we can find the Closed-loop invariant below.

$$\begin{cases} g_1'(x[4], ..., x[1]) \oplus g_2'(y[4], ..., y[1]) = 1 \\ g_2'(x[4], ..., x[1]) \oplus g_1'(y[4], ..., y[1]) = 1 \\ g_1' = x[1] \oplus x[2] \oplus x[4] \oplus x[1]x[3] \\ g_2' = y[1] \oplus y[3] \end{cases}$$

2. The linear layer of Midori64 is selected as an orthogonal matrix operation. Therefore,

$$g_1(X) = \sum_{j=1}^{16} g_1'(x_j), g_2(X) = \sum_{t=1}^{16} g_2'(x_t)$$

are the closed-loop invariants of the round function.

Attacking variant of Midori64

Attacking variant of Midori64

- There are $CLI(S)$ for our variant of Midori64, with $\deg(g_1) = 2$ and $\deg(g_2) = 1$.

Attacking variant of Midori64

- There are $CLI(S)$ for our variant of Midori64, with $\deg(g_1) = 2$ and $\deg(g_2) = 1$.
- One can specify $CLI(S)$ for the whole round of this variant of Midori-64

Attacking variant of Midori64

- There are $CLI(S)$ for our variant of Midori64, with $\deg(g_1) = 2$ and $\deg(g_2) = 1$.
- One can specify $CLI(S)$ for the whole round of this variant of Midori-64
- Efficient distinguishing attack !!

Attacking variant of Midori64

- There are $CLI(S)$ for our variant of Midori64, with $\deg(g_1) = 2$ and $\deg(g_2) = 1$.
- One can specify $CLI(S)$ for the whole round of this variant of Midori-64
- Efficient distinguishing attack !!

Remark

The attack works despite the fact that in this version of Midori64 $\dim W_L(D) = 64 = n$ - **standard invariant attack does not apply !!**

Building $W_L(D)$ of large dimension

For the Midori64 variant, the round keys repeat each second round. 64-bit round constants α_i^*, for $i = 0, \ldots, 14$ may be defined so that

$$D := \{\alpha_0^* \oplus \alpha_2^*, \alpha_0^* \oplus \alpha_4^*, \ldots, \alpha_0^* \oplus \alpha_{14}^*, \alpha_1^* \oplus \alpha_3^*, \alpha_1^* \oplus \alpha_5^*, \ldots, \alpha_1^* \oplus \alpha_{13}^*\},$$

has the maximum dimension $n = 64$.

Building $W_L(D)$ of large dimension

For the Midori64 variant, the round keys repeat each second round. 64-bit round constants α_i^*, for $i = 0, \ldots, 14$ may be defined so that

$$D := \{\alpha_0^* \oplus \alpha_2^*, \alpha_0^* \oplus \alpha_4^*, \ldots, \alpha_0^* \oplus \alpha_{14}^*, \alpha_1^* \oplus \alpha_3^*, \alpha_1^* \oplus \alpha_5^*, \ldots, \alpha_1^* \oplus \alpha_{13}^*\},$$

has the maximum dimension $n = 64$.

Conclusion

No obvious weaknesses for the choice of round constants, $\dim W_L(D) = n$ protects against standard invariant attacks, BUT attack based on *CLI* still applies !!

Additional criteria

Ensuring that $W_L(D)$ is large appears to be necessary BUT NOT sufficient criterion !!

Additional criteria

Ensuring that $W_L(D)$ is large appears to be necessary BUT NOT sufficient criterion !!

Design/security criterion

One must make sure that every round constant lies outside $LS(g_i)$ for every $(g_1, g_2) \in CLI(S)$.

Additional criteria

Ensuring that $W_L(D)$ is large appears to be necessary BUT NOT sufficient criterion !!

Design/security criterion

One must make sure that every round constant lies outside $LS(g_i)$ for every $(g_1, g_2) \in CLI(S)$.

Using computer simulations one can verify that PRESENT, PRINCE and Lblock are **resistant** against (CLI) generalized invariant attacks.

Summary of the talk

1. Overview
2. Generalized Nonlinear Invariants
3. Closed Loop Invariants
4. Conclusions

There are more generalizations and attempts to unify the work on invariant attacks. For instance, Beyne [2018] proposed a unified study within the framework of correlation matrices giving more insight towards a general structure.

There are more generalizations and attempts to unify the work on invariant attacks. For instance, Beyne [2018] proposed a unified study within the framework of correlation matrices giving more insight towards a general structure.

Work of Beierle, Canteaut and Leander [2018] shows a nice proposal to study the actual mathematical nature of these invariants in the framework of linear approximations thus reducing this kind of cryptanalysis to linear cryptanalysis.

There are more generalizations and attempts to unify the work on invariant attacks. For instance, Beyne [2018] proposed a unified study within the framework of correlation matrices giving more insight towards a general structure.

Work of Beierle, Canteaut and Leander [2018] shows a nice proposal to study the actual mathematical nature of these invariants in the framework of linear approximations thus reducing this kind of cryptanalysis to linear cryptanalysis.

There are many open questions regarding invariant attacks including: how to employ the generalized nonlinear invariants into these frameworks?

There are more generalizations and attempts to unify the work on invariant attacks. For instance, Beyne [2018] proposed a unified study within the framework of correlation matrices giving more insight towards a general structure.

Work of Beierle, Canteaut and Leander [2018] shows a nice proposal to study the actual mathematical nature of these invariants in the framework of linear approximations thus reducing this kind of cryptanalysis to linear cryptanalysis.

There are many open questions regarding invariant attacks including: how to employ the generalized nonlinear invariants into these frameworks?

Current work: Further generalization of the concept and deeper theoretical analysis !!

Merci beaucoup!