RUHR-UNIVERSITÄT BOCHUM

BISON
Instantiating the Withened Swap-Or-Not Construction

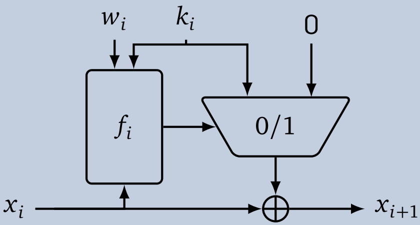
September 6th, 2018

Horst Görtz Institute for IT Security
Ruhr-Universität Bochum
Virginie Lallemand, Gregor Leander, Patrick Neumann, and *Friedrich Wiemer*

The WSN construction

Published by Tessaro at AsiaCrypt 2015 [`ia.cr/2015/868`].

Overview



Whitened Swap-Or-Not round function

$$x_i \mapsto x_i + f_{b(i)}(w_i + \max\{x_i, x_i + k_i\}) \cdot k_i$$

Security Proposition (informal)

The WSN construction with $\mathcal{O}(n)$ rounds is

$(2^{n-\mathcal{O}(\log n)}, 2^{n-\mathcal{O}(1)})$-secure.

(p,q)-secure: Attackers querying the encryption at most p and the underlying f_i's q times have only negl. advantage.

An Implementation

An Implementation

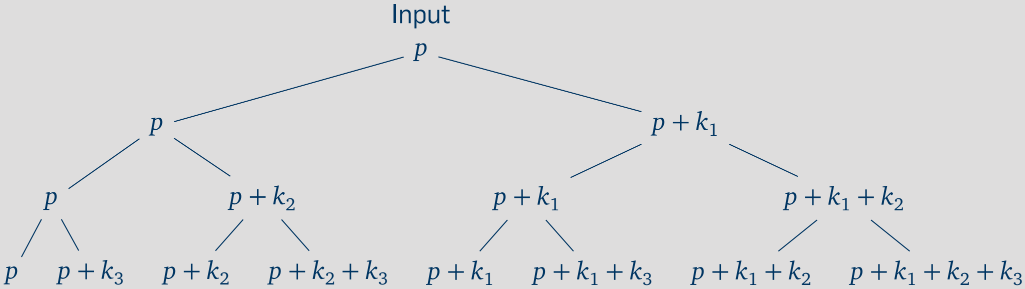
Construction

- $f(x) := ?$
- Key schedule?
- $\mathcal{O}(n)$ rounds?

Theoretical vs. practical constructions

Generic Analysis

On the number of rounds



Input

p

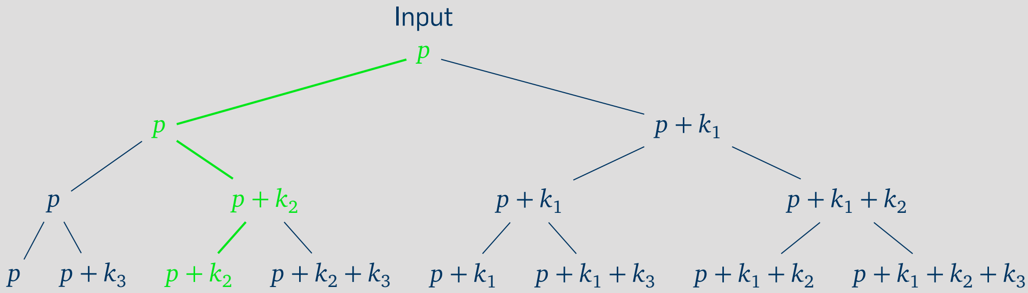
p $p + k_1$

p $p + k_2$ $p + k_1$ $p + k_1 + k_2$

p $p + k_3$ $p + k_2$ $p + k_2 + k_3$ $p + k_1$ $p + k_1 + k_3$ $p + k_1 + k_2$ $p + k_1 + k_2 + k_3$

Generic Analysis

On the number of rounds



Input

p

p $\quad$ $p + k_1$

p $\quad$ $p + k_2$ $\quad$ $p + k_1$ $\quad$ $p + k_1 + k_2$

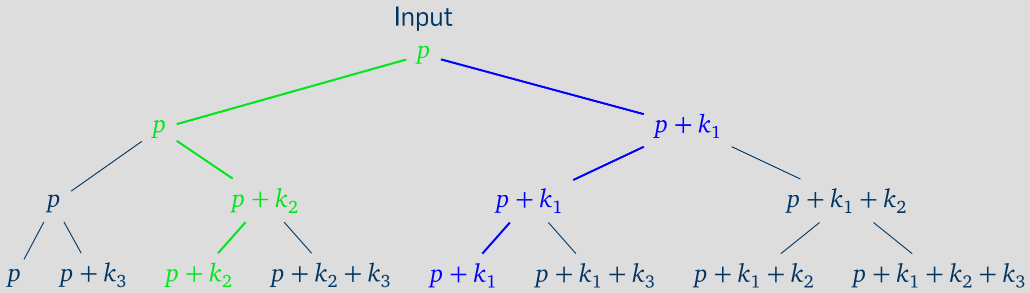
p $\quad$ $p + k_3$ $\quad$ $p + k_2$ $\quad$ $p + k_2 + k_3$ $\quad$ $p + k_1$ $\quad$ $p + k_1 + k_3$ $\quad$ $p + k_1 + k_2$ $\quad$ $p + k_1 + k_2 + k_3$

Encryption

$$E_{k,w}(p) := p + \sum_{i=1}^{r} \lambda_i k_i = c$$

Generic Analysis

On the number of rounds



Encryption

$$E_{k,w}(p) := p + \sum_{i=1}^{r} \lambda_i k_i = c$$

Generic Analysis

On the number of rounds

Observation

- The ciphertext is the plaintext plus a random subset of the round keys:

$$c = p + \sum_{i=1}^{r} \lambda_i k_i$$

- For pairs p_i, c_i: $\operatorname{span}\{p_i + c_i\} \subseteq \operatorname{span}\left\{k_j\right\}$.

Generic Analysis

On the number of rounds

Observation

- The ciphertext is the plaintext plus a random subset of the round keys:

$$c = p + \sum_{i=1}^{r} \lambda_i k_i$$

- For pairs p_i, c_i: $\operatorname{span}\{p_i + c_i\} \subseteq \operatorname{span}\left\{k_j\right\}$.

Distinguishing Attack for $r < n$ rounds

There is an $u \in \mathbb{F}_2^n \setminus \{0\}$, s. t. $\langle u, p \rangle = \langle u, c \rangle$ holds always:

$$\begin{aligned} \langle u, c \rangle &= \left\langle u, p + \sum \lambda_i k_i \right\rangle \\ &= \langle u, p \rangle + \left\langle u, \sum \lambda_i k_i \right\rangle = \langle u, p \rangle + 0 \end{aligned}$$

for all $u \in \operatorname{span}\{k_1, \ldots, k_r\}^\perp \neq \{0\}$

Generic Analysis

On the number of rounds

Observation

- The ciphertext is the plaintext plus a random subset of the round keys:

$$c = p + \sum_{i=1}^{r} \lambda_i k_i$$

- For pairs p_i, c_i: $\mathrm{span}\,\{p_i + c_i\} \subseteq \mathrm{span}\,\{k_j\}$.

Distinguishing Attack for $r < n$ rounds

There is an $u \in \mathbb{F}_2^n \setminus \{0\}$, s. t. $\langle u, p\rangle = \langle u, c\rangle$ holds always:

$$\begin{aligned} \langle u, c\rangle &= \left\langle u, p + \sum \lambda_i k_i \right\rangle \\ &= \langle u, p\rangle + \left\langle u, \sum \lambda_i k_i \right\rangle = \langle u, p\rangle + 0 \end{aligned}$$

for all $u \in \mathrm{span}\,\{k_1, \ldots, k_r\}^{\perp} \neq \{0\}$

Rationale 1

Any instance must iterate at least n rounds; any set of n consecutive keys should be linear indp.

Generic Analysis

On the Boolean functions f_i

Observation

If the f_i do not depend on the MSB, i. e.

$$f_i(x) = f_i(x + e_n)$$

then this propagates through r rounds w. h. p.:

$$\Pr\left[E_{k,w}(x) + E_{k,w}(x + e_n) = e_n\right] \geqslant \left(1 - 2^{-1}\right)^r$$

Why could this happen?

- For example, when the difference does not influence the lexicographic ordering of x and $x + k_i$.
- Gets worse when depending on less bits.
- Compare to AES! Its round function depends on only 32 out of 128 bits.

Generic Analysis

On the Boolean functions f_i

Observation

If the f_i do not depend on the MSB, i. e.

$$f_i(x) = f_i(x + e_n)$$

then this propagates through r rounds w. h. p.:

$$\Pr\left[E_{k,w}(x) + E_{k,w}(x + e_n) = e_n\right] \geqslant \left(1 - 2^{-1}\right)^r$$

Why could this happen?

- For example, when the difference does not influence the lexicographic ordering of x and $x + k_i$.
- Gets worse when depending on less bits.
- Compare to AES! Its round function depends on only 32 out of 128 bits.

Rationale 2

For any instance, the f_i should depend on all bits, and for any $\delta \in \mathbb{F}_2^n :\ \Pr[f_i(x) = f_i(x + \delta)] \approx \frac{1}{2}$.

A genus of the WSN construction: BISON

Generic properties of **B**ent wh**I**tened **S**wap **O**r **N**ot

- Consecutive round keys linearly independent
- At least n iterations of the round function
- The round function depends on all bits
- All derivatives are balanced (*bent*)

Rational 1 & 2: WSN is *slow* in practice!

But what about

Differential Cryptanalysis?

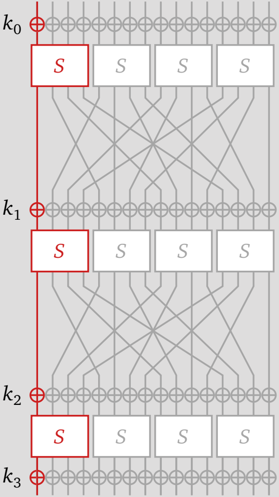
Differential Cryptanalysis

Primer

For block cipher $E_k(x)$ compute

$$\Pr\left[E_k(x) + E_k(x+\alpha) = \beta\right] = p_{E_k}(\alpha, \beta).$$

Notation: $\Pr[\alpha \rightarrow \beta]$.



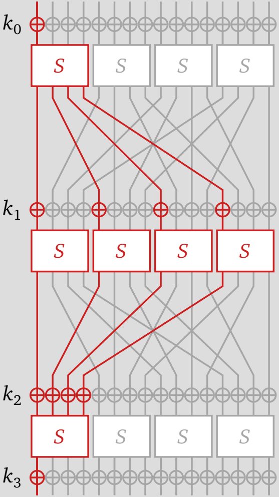
Differential Cryptanalysis

Primer

For block cipher $E_k(x)$ compute

$$\Pr\left[E_k(x)+E_k(x+\alpha)=\beta\right]=p_{E_k}(\alpha,\beta).$$

Notation: $\Pr[\alpha\rightarrow\beta]$.



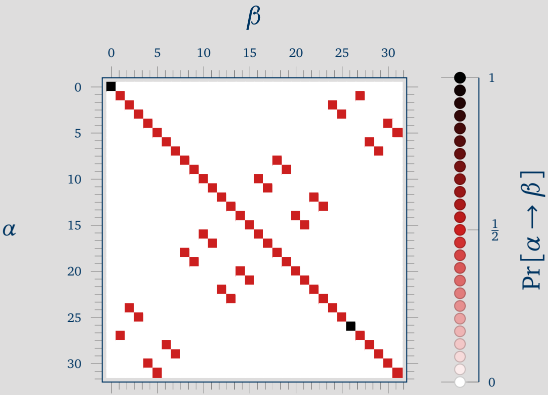
Differential Cryptanalysis

One round

Proposition

For one round of BISON, the probabilities are:

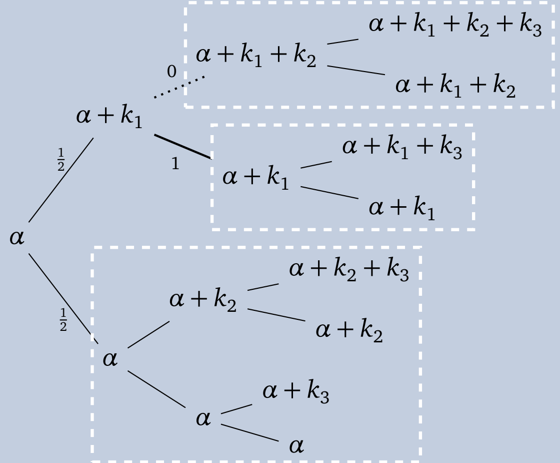
$$\Pr[\alpha \rightarrow \beta] = \begin{cases} 1 & \text{if } \alpha = \beta = k \text{ or } \alpha = \beta = 0 \\ \frac{1}{2} & \text{else if } \beta \in \{\alpha, \alpha + k\} \\ 0 & \text{else} \end{cases}$$



Differential Cryptanalysis

More rounds

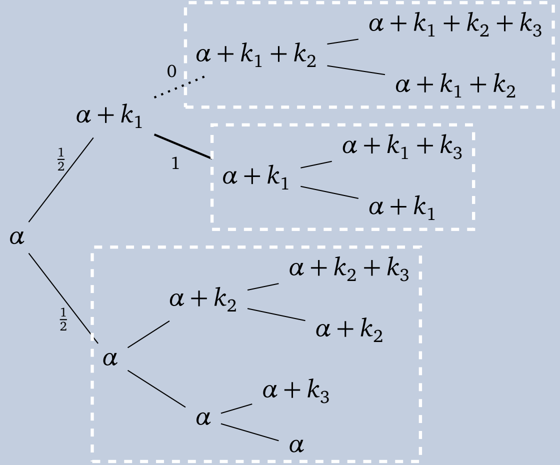
Example differences over $r = 3$ rounds



Differential Cryptanalysis

More rounds

Example differences over $r = 3$ rounds



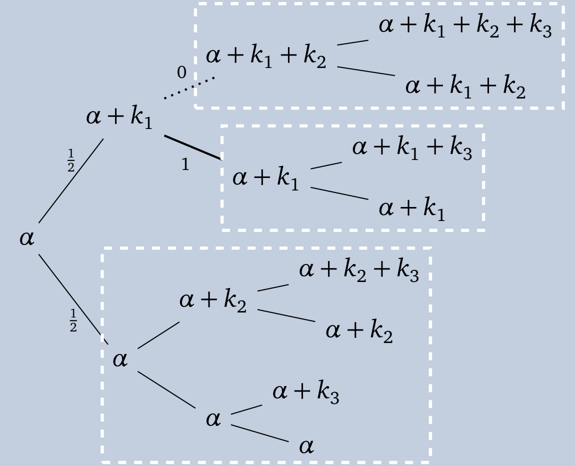
Probabilities of output differences

$$\Pr[\alpha \to \beta] = \begin{cases} 2^{-r} & \text{if } \beta \text{ in normal branch} \\ 2^{-r+1} & \text{if } \beta \text{ in collapsed branch} \\ 0 & \text{if } \beta \text{ in impossible branch} \end{cases}$$

Differential Cryptanalysis

More rounds

Example differences over $r = 3$ rounds



Probabilities of output differences

$$\Pr[\alpha \to \beta] = \begin{cases} 2^{-r} & \text{if } \beta \text{ in normal branch} \\ 2^{-r+1} & \text{if } \beta \text{ in collapsed branch} \\ 0 & \text{if } \beta \text{ in impossible branch} \end{cases}$$

Collapsing

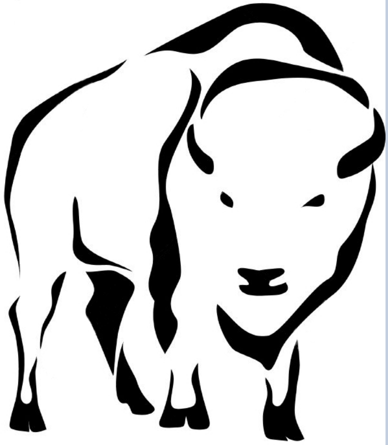
How many branches can collapse?

Only two

For $r \leqslant n$ rounds and linearly indp. round keys this happens only once.

BISON

A concrete species



Addressing Rationale 1

The Key Schedule

Rationale 1

Any instance must iterate at least n rounds; any set of n consecutive keys should be linear indp.

Design Decisions

- Choose number of rounds as $2 \cdot n$
- Round keys derived from the state of LFSRs
- Add round constants c_i to w_i round keys

Implications

- Clocking an LFSR is cheap
- For an LFSR with feedback polynomial of degree n, every n consecutive states are linearly independent
- Round constants avoid structural weaknesses

Addressing Rationale 2

The Round Function

Rationale 2

For any instance, the f_i should depend on all bits, and for any $\delta \in \mathbb{F}_2^n :\ \Pr[f_i(x) = f_i(x+\delta)] \approx \frac{1}{2}$.

Design Decisions

- Choose $f : \mathbb{F}_2^n \to \mathbb{F}_2$ to be bent
- Choose the simplest bent function known:

$$f(x, y) := \langle x, y \rangle$$

Implications

- Bent functions only exists for even n
- Instance not possible for every block length n

Further Cryptanalysis

Linear Cryptanalysis

For $r \geqslant n$ rounds, the correlation of any non-trivial linear trail for BISON is upper bounded by $2^{-\frac{n+1}{2}}$.

Invariant Attacks

For $r \geqslant n$ rounds, neither invariant subspaces nor nonlinear invariant attacks do exist for BISON.

Zero Correlation

For $r > 2n - 2$ rounds, BISON does not exhibit any zero correlation linear hulls.

Impossible Differentials

For $r > n$ rounds, there are no impossible differentials for BISON.

Conclusion/Questions

Thank you for your attention!

BISON

- A first instance of the WSN construction
- Good results for differential cryptanalysis

Open Problems

- Construction for linear cryptanalysis
- Further analysis: division properties

Thank you!

Questions?

Details

Addressing Rationale 2

The detailed answer

Rationale 2

For any instance, the f_i should depend on all bits, and for any $\delta \in \mathbb{F}_2^n : \ \Pr[f_i(x) = f_i(x+\delta)] \approx \frac{1}{2}$.

Design Decisions

- Choose $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$ to be bent
- Replace $x \mapsto \max\{x, x+k\}$ by

$$\Phi_k(x) : \mathbb{F}_2^n \to \mathbb{F}_2^{n-1}$$

$$\Phi_k(x) := (x + x[i(k)] \cdot k)[j]_{\substack{1 \leqslant j \leqslant n \\ j \neq i(k)}}$$

Implications

- With Φ_k we preserve the bent properties
- Bent functions only exists for even n
- Encryption now only possible for odd block lengths

BISON's round function

For round keys $k_i \in \mathbb{F}_2^n$ and $w_i \in \mathbb{F}_2^{n-1}$ the round function computes

$$R_{k_i,w_i}(x) := x + f_{b(i)}\big(w_i + \Phi_{k_i}(x)\big) \cdot k_i.$$

where

- Φ_{k_i} and $f_{b(i)}$ are defined as

$$\Phi_k(x) : \mathbb{F}_2^n \to \mathbb{F}_2^{n-1}$$
$$\Phi_k(x) := (x + x[i(k)] \cdot k)[j]_{\substack{1 \leqslant j \leqslant n \\ j \neq i(k)}}$$

$$f_{b(i)} : \mathbb{F}_2^{\frac{n-1}{2}} \times \mathbb{F}_2^{\frac{n-1}{2}} \to \mathbb{F}_2$$
$$f_{b(i)}(x,y) := \langle x, y \rangle + b(i),$$

- and $b(i)$ is 0 if $i \leqslant \frac{r}{2}$ and 1 else.

RUB

BISON's key schedule

Given

- primitive $p_k, p_w \in \mathbb{F}_2[x]$ with degrees n, $n-1$ and companion matrices C_k, C_w.
- master key $K = (k, w) \in \left(\mathbb{F}_2^n \times \mathbb{F}_2^{n-1}\right) \setminus \{0, 0\}$

The ith round keys are computed by

$$\begin{aligned} \mathrm{KS}_i : \mathbb{F}_2^n \times \mathbb{F}_2^{n-1} &\to \mathbb{F}_2^n \times \mathbb{F}_2^{n-1} \\ \mathrm{KS}_i(k, w) &:= (k_i, c_i + w_i) \end{aligned}$$

where

$$k_i = (C_k)^i k, \qquad c_i = (C_w)^{-i} e_1, \qquad w_i = (C_w)^i w.$$