Automatic Search for Linear Trails of the SPECK Family

Yuan Yao[1,2] Bin Zhang[2] Wenling Wu[2]

[1]TCA Laboratory, Institute of Software, Chinese Academy of Sciences

[2]University of Chinese Academy of Sciences

Information Security Conference, 2015

Outline

SPECK

- By NSA in 2013.
- Lightweight.
- Feistel-like.
- ARX.
- For software applications.

Previous Work

- Differential Analysis by Alex Biryukov et. al. at CT-RSA 2014.
- Differential Analysis by Farzaneh Abed et. al. at FSE 2014.
- Differential Analysis by Alex Biryukov et. al. at FSE 2014.
- Differential Analysis by Itai Dinur at SAC 2014.
- Differential Fault Analysis by Harshal Tupsamudre et. al. at FDTC 2014.

Previous Work

- Differential Analysis by Alex Biryukov et. al. at CT-RSA 2014.
- Differential Analysis by Farzaneh Abed et. al. at FSE 2014.
- Differential Analysis by Alex Biryukov et. al. at FSE 2014.
- Differential Analysis by Itai Dinur at SAC 2014.
- Differential Fault Analysis by Harshal Tupsamudre et. al. at FDTC 2014.

Linear Cryptanalysis???

Our Contribution

- Linear cryptanalysis of SPECK.
- An implementation of Wallén's algorithm.

Basics

Definition (Correlation)

$c_X \triangleq 2\Pr(X = 0) - 1.$

Basics

Definition (Correlation)

$c_X \triangleq 2\Pr(X = 0) - 1.$

$$H_0 : c_X = 0 \longleftrightarrow H_1 : c_X \neq 0$$

Basics

Definition (Correlation)

$c_X \triangleq 2\Pr(X = 0) - 1.$

$$H_0 : c_X = 0 \longleftrightarrow H_1 : c_X \neq 0$$

Lemma (Piling-up Lemma)

$c_{X \oplus Y} = c_X c_Y.$

Basics

Definitions (Inner Product)

$$X \cdot Y = \bigoplus_{i=0}^{n-1} X_i \& Y_i \in \mathbb{F}_2.$$

Linear Approximation

$\vec{S}[0]$

r rounds encryption

$\vec{S}[r]$

Linear Approximation

$$\vec{S}[0] \cdot \vec{\Gamma}[0]$$

r rounds encryption

$$\vec{S}[r] \cdot \vec{\Gamma}[r]$$

$$\vec{S}[0] \cdot \vec{\Gamma}[0] \oplus \vec{S}[r] \cdot \vec{\Gamma}[r] \in \mathbb{F}_2$$

Linear Trail

$$\vec{S}[0]\cdot\vec{\Gamma}[0]$$

$$\vec{S}[1]\cdot\vec{\Gamma}[1]$$

$$\vec{S}[2]\cdot\vec{\Gamma}[2]$$

$$\vdots$$

$$\vec{S}[r-1]\cdot\vec{\Gamma}[r-1]$$

$$\vec{S}[r]\cdot\vec{\Gamma}[r]$$

$$\vec{S}[0]\cdot\vec{\Gamma}[0]\oplus\vec{S}[r]\cdot\vec{\Gamma}[r]$$

$$\|$$

$$\bigoplus_{i=0}^{r-1}\left(\vec{S}[i]\cdot\vec{\Gamma}[i]\oplus\vec{S}[i+1]\cdot\vec{\Gamma}[i+1]\right)$$

Linear Trail

$$\vec{S}[0] \cdot \vec{\Gamma}[0]$$

$$\vec{S}[1] \cdot \vec{\Gamma}[1]$$

$$\vec{S}[2] \cdot \vec{\Gamma}[2]$$

$$\vdots$$

$$\vec{S}[r-1] \cdot \vec{\Gamma}[r-1]$$

$$\vec{S}[r] \cdot \vec{\Gamma}[r]$$

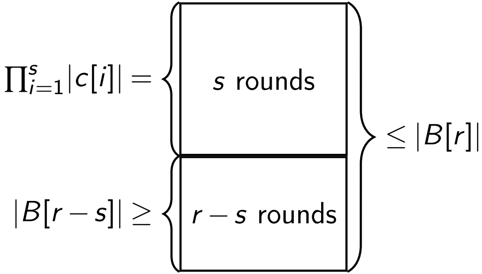
$$\vec{S}[0] \cdot \vec{\Gamma}[0] \oplus \vec{S}[r] \cdot \vec{\Gamma}[r]$$

$$\|$$

$$\bigoplus_{i=0}^{r-1} \left(\vec{S}[i] \cdot \vec{\Gamma}[i] \oplus \vec{S}[i+1] \cdot \vec{\Gamma}[i+1] \right)$$

Matsui Search

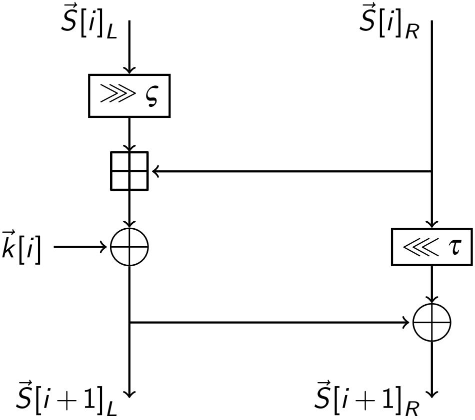
- Proposed at EUROCRYPT 1994.
- Branch-and-bound: $|B[r-s]\prod_{i=1}^{s} c[i]| \leq |B[r]|$



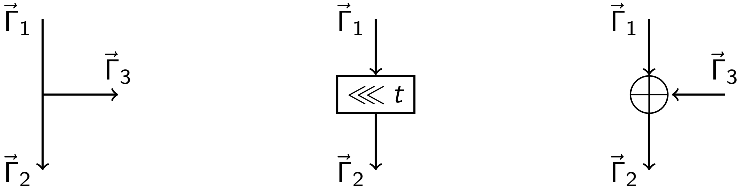
Matsui Search Algorithm

function Search$(B,\ T = \{\})$
  $r \leftarrow \text{Sizeof}(B) - 1, s \leftarrow \text{Sizeof}(T)$
  if $s = r$ **then**
    $\hat{B}[r] \leftarrow \prod_{i=1}^{r} c[i]$
  else
    for T' **in** Extend(T) **do**
      if $|B[r-(s+1)]\prod_{i=1}^{s+1} c'[i]| > |\hat{B}[r]|$ **then**
        Search$(B,\ T')$
      else
        return
      end if
    end for
  end if
end function

Round Function of SPECK



Approximations of Primitives

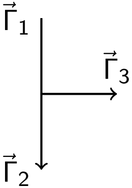


$\vec{\Gamma}_1 \oplus \vec{\Gamma}_2 \oplus \vec{\Gamma}_3 = \vec{0}$

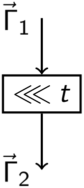
$\vec{\Gamma}_2 = \vec{\Gamma}_1 \lll t$

$\vec{\Gamma}_1 = \vec{\Gamma}_2 = \vec{\Gamma}_3$

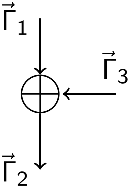
Approximations of Primitives



$\vec{\Gamma}_1 \oplus \vec{\Gamma}_2 \oplus \vec{\Gamma}_3 = \vec{0}$



$\vec{\Gamma}_2 = \vec{\Gamma}_1 \lll t$



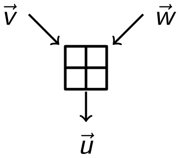
$\vec{\Gamma}_1 = \vec{\Gamma}_2 = \vec{\Gamma}_3$

Modulo Addition???

Approximations of Modulo Addition

Definition

$$c\left(\vec{u},\vec{v},\vec{w}\right) \triangleq c_{\vec{u}\cdot\left(\vec{Z}_1 \boxplus \vec{Z}_2\right)\oplus\vec{v}\cdot\vec{Z}_1\oplus\vec{w}\cdot\vec{Z}_2}.$$



Linear Approximation Table

- Enumerate $\vec{u}, \vec{v}, \vec{w}$, calculate $c\left(\vec{u}, \vec{v}, \vec{w}\right)$, and sort.
- Time: $O\left(2^{3n}\right)$, Memory: $O\left(2^{3n}\right)$.

Linear Approximation Table

- Enumerate $\vec{u}, \vec{v}, \vec{w}$, calculate $c\left(\vec{u}, \vec{v}, \vec{w}\right)$, and sort.
- Time: $O\left(2^{3n}\right)$, Memory: $O\left(2^{3n}\right)$.

Generate Online!!!

Wallén's Theorem

Theorem

Let $S^0(0,0) \triangleq \{null\}$, $S^0(n,k) = S^1(n,k) \triangleq \emptyset$ *when* $k < 0$ *or* $k \geq n > 0$, *and*

$$S^0(n,k) \triangleq (S^0(n-1,k) \parallel \{0\}) \cup (S^1(n-1,k-1) \parallel \{1,2,4,7\})$$
$$S^1(n,k) \triangleq (S^0(n-1,k) \parallel \{7\}) \cup (S^1(n-1,k-1) \parallel \{0,3,5,6\})$$

otherwise, where $S^\star \parallel \Omega \triangleq \{\vec{a} \parallel \vec{b} \mid \vec{a} \in S^\star, \vec{b} \in \Omega\}$. *Then*

$$S(n,k) \triangleq S^0(n,k) \cup S^1(n,k)$$

is the set of all masks such that $c(\vec{u}, \vec{v}, \vec{w}) = \pm 2^{-k}$.

Wallén's Theorem

Example

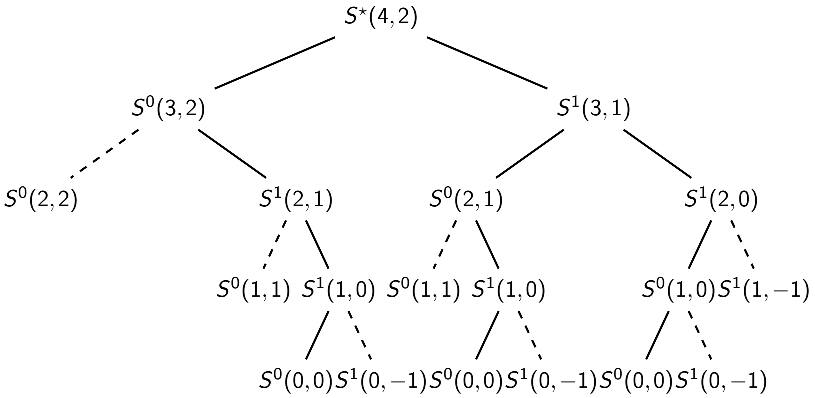
$$S^0(n,0) = \{(0\cdots 00)\},$$
$$S^1(n,0) = \{(0\cdots 07)\},$$

thus

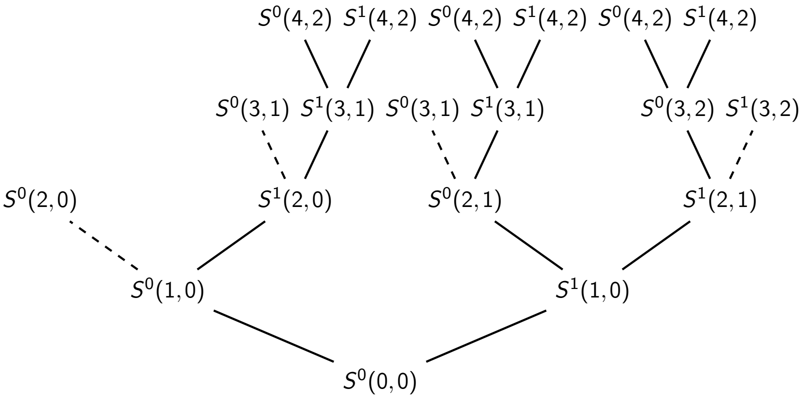
$$S(n,0) = \{ ((0\cdots 00),(0\cdots 00),(0\cdots 00)), ((0\cdots 01),(0\cdots 01),(0\cdots 01)) \}$$

is the set of all masks such that $c(\vec{u},\vec{v},\vec{w}) = \pm 1$.

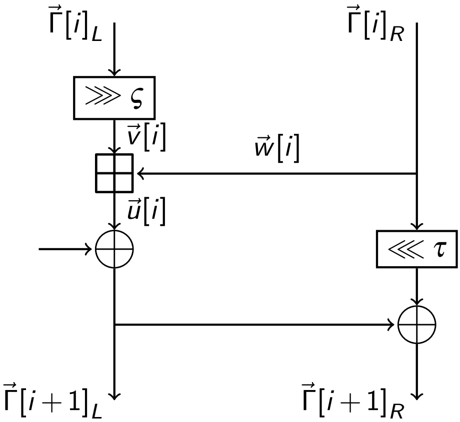
Top-down Method



Bottom-up Method



Extend()



$$\vec{u}[i] = \vec{\Gamma}[i+1]_L \oplus \vec{\Gamma}[i+1]_R$$
$$\vec{v}[i] = \vec{\Gamma}[i]_L \ggg \varsigma$$
$$\vec{w}[i] = \vec{\Gamma}[i]_R \oplus \left(\vec{\Gamma}[i+1]_R \ggg \tau\right)$$

Extend()

$$\vec{u}[r] = \vec{X}[r+1] \oplus \vec{Y}[r+1]$$

$$\vec{u}[r-1] = (\vec{v}[r] \lll \varsigma) \oplus \vec{w}[r] \oplus \left(\vec{Y}[r+1] \ggg \tau\right)$$

$$\vec{u}[i] = (\vec{v}[i+1] \lll \varsigma) \oplus \vec{w}[i+1] \oplus ((\vec{u}[i+1] \oplus (\vec{v}[i+2] \lll \varsigma)) \ggg \tau)$$

Search Results

- SPECK-32

Rounds(r)	1	2	3	4	5	6	7	8
$\lvert B[r]\rvert$	1	1	2^{-1}	2^{-3}	2^{-5}	2^{-7}	2^{-9}	2^{-12}
Rounds(r)	9	10	11	12	13	14	15	16
$\lvert B[r]\rvert$	2^{-14}	2^{-17}	2^{-19}	2^{-20}	2^{-22}	2^{-24}	2^{-26}	2^{-28}
Rounds(r)	17	18	19	20	21	22		
$\lvert B[r]\rvert$	2^{-30}	2^{-34}	2^{-36}	2^{-38}	2^{-40}	2^{-42}		

- SPECK-48/ 64/ 96/ 128: Omitted.

Search Results

- SPECK-32

Rounds(r)	1	2	3	4	5	6	7	8
$\lvert B[r]\rvert$	1	1	2^{-1}	2^{-3}	2^{-5}	2^{-7}	2^{-9}	2^{-12}
Rounds(r)	9	10	11	12	13	14	15	16
$\lvert B[r]\rvert$	2^{-14}	2^{-17}	2^{-19}	2^{-20}	2^{-22}	2^{-24}	2^{-26}	2^{-28}
Rounds(r)	17	18	19	20	21	22		
$\lvert B[r]\rvert$	2^{-30}	2^{-34}	2^{-36}	2^{-38}	2^{-40}	2^{-42}		

- SPECK-48/ 64/ 96/ 128: Omitted.

Linear Distinguishers

Block Length	Trail Length	Correlation	Rounds	Data
32	9	2^{-14}	10	2^{28}
48	9	2^{-20}	10	2^{40}
64	11	2^{-25}	12	2^{50}
64	12	2^{-31}	13	2^{62}
96	6	2^{-11}	7	2^{22}
128	6	2^{-11}	7	2^{22}

Key Recovery Attacks

Block/ Key Length	Rounds (this paper/ Dinur/ Total)	Data (this paper/ Dinur)	Average Time (this paper/ Dinur)
32/ 64	12/ 14/ 22	$2^{30.8668}/2^{31}$	$2^{61.2164}/2^{63}$
48/ 72	11/ 14/ 22	$2^{43.727}/2^{41}$	$2^{68.345}/2^{65}$
48/ 96	12/ 15/ 23	$2^{43.727}/2^{41}$	$2^{92.345}/2^{89}$
64/ 96	13/ 18/ 26	$2^{54.6279}/2^{61}$	$2^{86.1551}/2^{93}$
64/ 96	14/ 18/ 26	$2^{62.7302}/2^{61}$	$2^{95.8714}/2^{93}$
64/ 128	14/ 19/ 27	$2^{54.8029}/2^{61}$	$2^{118.155}/2^{125}$
64/ 128	15/ 19/ 27	$2^{62.7302}/2^{61}$	$2^{127.871}/2^{125}$
96/ 96	8/ 16/ 28	$2^{27.6463}/2^{85}$	$2^{74.8954}/2^{85}$
96/ 144	9/ 17/ 29	$2^{27.6463}/2^{85}$	$2^{122.895}/2^{133}$
128/ 128	8/ 17/ 32	$2^{28.2959}/2^{113}$	$2^{92.7363}/2^{113}$
128/ 192	9/ 18/ 33	$2^{28.2959}/2^{113}$	$2^{156.736}/2^{177}$
128/ 256	7/ 19/ 34	$2^{28.2959}/2^{113}$	$2^{220.736}/2^{241}$

Masks of Carry

Example

$\vec{u} = (1100), \vec{v} = \vec{w} = (1000)$, then

$$\vec{\phi} = \vec{v} \oplus \vec{u} = (0100),$$
$$\vec{\varphi} = \vec{w} \oplus \vec{u} = (0100).$$

Common Prefix Mask & Correlation

Lemma

Let $\vec{\delta}$ *be the CPM of* $\vec{u}, \vec{v}, \vec{w}$. *Then*

$$c(\vec{u}, \vec{v}, \vec{w}) = \begin{cases} (-1)^{wt\left(\vec{\delta}\vec{\phi}\vec{\varphi}\right)} 2^{-wt\left(\vec{\delta}\right)}, & \text{if } \vec{\phi} = \vec{\phi}\vec{\delta} \text{ and } \vec{\varphi} = \vec{\varphi}\vec{\delta} \\ 0, & \text{otherwise} \end{cases}$$

More Explicit Formula

Theorem

$\vec{\delta}$ is the CPM of $\vec{u},\vec{v},\vec{w}$, and $c(\vec{u},\vec{v},\vec{w}) \neq 0$ if and only if

$$\vec{\phi} = \vec{\phi}\vec{\delta}$$

$$\vec{\varphi} = \vec{\varphi}\vec{\delta}$$

$$\vec{\gamma} \gg 1 = \left(\left(\vec{u} \oplus \vec{\delta}\right) \gg 1\right) \oplus \vec{\delta}$$

$$\vec{0} = \left((\vec{u} \gg 1) \oplus \vec{\delta}\right)\left(\left(\vec{\delta} \oplus \vec{1}\right) \gg 1\right)$$

$$\vec{0} = \left((\vec{v} \gg 1) \oplus \vec{\delta}\right)\left(\left(\vec{\delta} \oplus \vec{1}\right) \gg 1\right)$$

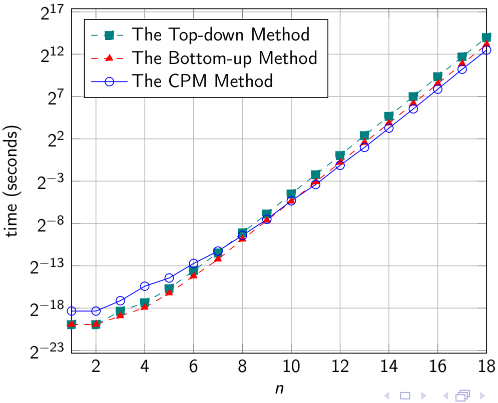
$$\vec{0} = \left((\vec{w} \gg 1) \oplus \vec{\delta}\right)\left(\left(\vec{\delta} \oplus \vec{1}\right) \gg 1\right)$$

CPM Method

1. Generate $\vec{\delta}$ in increasing order of Hamming weight.
2. Generate unknowns in $\vec{u}, \vec{v}, \vec{w}$.

Performance Comparison

- Task: Generating $\bigcup_{k=0}^{n-1} S(n,k)$.
- Platform: 32-bit Win7 with Visual C++ 2015 CTP optimized by /Ox.



Conclusions

- It is hard to find linear trails for large blocks.
- SPECK-32 is immune to the 1-dimensional linear cryptanalysis.
- Linear cryptanalysis seems less efficient than differential cryptanalysis to SPECK.

Further Work

- Threshold search.
- Vectorial linear cryptanalysis.
- Apply the search to other ARX ciphers.

Q & A

Q & A

yaoyuan@tca.iscas.ac.cn

Acknowledgment

- Thanks to my family, my supervisors, and my friends.
- Thanks to ISC, and anonymous reviewers.
- Thanks to all of you.