MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck

Kai Fu[1], Meiqin Wang[1], Yinghua Guo[1], Siwei Sun[2], Lei Hu[2]

Shandong University, China; The Academy of Sciences of China, China

FSE 2016 @ Bochum, Germany

Background

Automatic Search algorithm for Differential and Linear Trails Based on MILP

- Searching the minimum active S-boxes with MILP solver by [MWGP'11,WU-WANG'11]
- A general heuristic automatic search algorithm is given by [SHWQMS'14]
- Sun *et al.* @ eprint'14 transformed the heuristic method to non heuristic algorithm

Progress of Search Algorithm for differential and linear trails for ARX Ciphers

- The automatic search method for differential characteristics in ARX-based Hash functions is provided by [Leurent'12, MNS'11, De-Rechberger'06]
- The threshold search for differential characteristics in ARX block ciphers is given by [Biryukov-Velichkov'14]
- Biryukov *et al.* @ FSE'16 given an automatic algorithm for best differential and linear trails in ARX ciphers

Motivations and Contributions

Motivations:

- The current MILP-based algorithm cannot be applied to ARX ciphers
- There is no general search method for linear trails for ARX ciphers before FSE'16

Contributions:

- Proposed the MILP model to automatically search for differential trails for ARX ciphers under Markov assumption
- Proposed the MILP model to automatically search for linear trails for ARX ciphers under Markov assumption
- Applied to Speck and improved attack on Speck48/64/96/128

The differential is possible or not?

Theorem (see Lipmaa and Moriai @ FSE'02])

The differential $(\alpha, \beta \rightarrow \gamma)$ *is possible iff* $(\alpha[0] \oplus \beta[0] \oplus \gamma[0])$ *= 0 and* $\alpha[i-1] = \beta[i-1] = \gamma[i-1] = \alpha[i] \oplus \beta[i] \oplus \gamma[i]$ *for* $\alpha[i-1] = \beta[i-1] = \gamma[i-1]$, $i \in [1, n-1]$.

Example:
the differential $(\alpha, \beta \rightarrow \gamma) = (11100, 11100 \rightarrow 11110)$ is impossible as $\alpha[0] = \beta[0] = \gamma[0] \neq \alpha[1] \oplus \beta[1] \oplus \gamma[1]$.

How to calculate the differential probability of addition modulo 2^n efficiently ?

Theorem (see Lipmaa and Moriai @ FSE'02])

Assume that $(\alpha, \beta \rightarrow \gamma)$ is a possible differential characteristic, then the differential probability $xdp^+ = 2^{-\sum_{i=0}^{n-2} \neg eq(\alpha[i],\beta[i],\gamma[i])}$, where

$$eq(\alpha[i],\beta[i],\gamma[i]) = \begin{cases} 1 & \alpha[i] = \beta[i] = \gamma[i] \\ 0 & others \end{cases}.$$

Example:
for the differential $(\alpha, \beta \rightarrow \gamma)$ = (11100, 00110 $\rightarrow$ 10110), the probability $xdp^+(\alpha, \beta \rightarrow \gamma) = 2^{-(\neg eq(0,0,0)+\neg eq(0,1,1)+\neg eq(1,1,1)+\neg eq(1,0,0))} = 2^{-2}$.

Construct the MILP Model for Differential Characteristics of Modular Addition:

- using five linear inequalities to satisfy the first condition $\alpha[0] \oplus \beta[0] \oplus \gamma[0] = 0$:

$$\begin{cases} d_{\oplus} \geq \alpha[0], d_{\oplus} \geq \beta[0], d_{\oplus} \geq \gamma[0] \\ \alpha[0] + \beta[0] + \gamma[0] - 2d_{\oplus} \geq 0 \\ \alpha[0] + \beta[0] + \gamma[0] \leq 2 \end{cases}$$

  where $d_{\oplus}$ is a dummy bit variable.
- Using the vector $(\alpha[i], \beta[i], \gamma[i], \alpha[i+1], \beta[i+1], \gamma[i+1])$ to denote the relation of the differential values for the i-th and the $(i+1)$-th bits.

Note: there are totally 56 (not 2^6) possible patterns for the vector.
For example:
the differential pattern $(0,0,0,1,1,1)$ is impossible as $\alpha_i = \beta_i = \gamma_i \neq \alpha_{i+1} \oplus \beta_{i+1} \oplus \gamma_{i+1}$.

- Adding the $\neg eq(\alpha[i], \beta[i], \gamma[i])$ to the vector to compute the differential probability efficiently
- Using **SAGE**(http://www.sagemath.org/) to generate the linear inequalities to construct the MILP model

How to use SAGE to generate linear inequalities?

fu@fu-Lenovo: ~

```
fu@fu-Lenovo:~$ sage

 Sage Version 6.1.1, Release Date: 2014-02-04
 Type "notebook()" for the browser-based notebook interface.
 Type "help()" for help.

sage: myPoints = [[0, 0, 0, 0],\
....: [0, 0, 1, 1],\
....: [0, 1, 0, 1],\
....: [0, 1, 1, 0],\
....: [1, 0, 0, 1],\
....: [1, 0, 1, 0],\
....: [1, 1, 0, 0],\
....: [1, 1, 1, 1]]
sage:
sage: poly_test = Polyhedron(vertices = myPoints)
sage: for v in poly_test.inequality_generator() :
....:           print(v)
....:
An inequality (0, 0, 1, 0) x + 0 >= 0
An inequality (1, 1, -1, 1) x + 0 >= 0
An inequality (1, 0, 0, 0) x + 0 >= 0
An inequality (1, 1, 1, -1) x + 0 >= 0
An inequality (0, 1, 0, 0) x + 0 >= 0
An inequality (-1, 1, 1, 1) x + 0 >= 0
An inequality (0, 0, 0, 1) x + 0 >= 0
An inequality (1, -1, 1, 1) x + 0 >= 0
An inequality (0, -1, 0, 0) x + 1 >= 0
An inequality (-1, -1, 1, -1) x + 2 >= 0
An inequality (0, 0, -1, 0) x + 1 >= 0
An inequality (1, -1, -1, -1) x + 2 >= 0
An inequality (0, 0, 0, -1) x + 1 >= 0
An inequality (-1, 1, -1, -1) x + 2 >= 0
An inequality (-1, 0, 0, 0) x + 1 >= 0
An inequality (-1, -1, -1, 1) x + 2 >= 0
sage:
```

possible pattern

linear inequalities

Using the Sun *et al.*'s **greedy algorithm**[@ eprint'14], we get 13 linear inequalities to describe the relation of the differential values for the i-th and the $(i+1)$-th bits.:

$$
\begin{aligned}
\beta[i]-\gamma[i]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq 0,\\
\alpha[i]-\beta[i]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq 0,\\
-\alpha[i]+\gamma[i]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq 0,\\
-\alpha[i]-\beta[i]-\gamma[i]-(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq -3,\\
\alpha[i]+\beta[i]+\gamma[i]-(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq 0,\\
-\beta[i]+\alpha[i+1]+\beta[i+1]+\gamma[i+1]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq 0,\\
\beta[i]+\alpha[i+1]-\beta[i+1]+\gamma[i+1]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq 0,\\
\beta[i]-\alpha[i+1]+\beta[i+1]+\gamma[i+1]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq 0,\\
\alpha[i]+\alpha[i+1]+\beta[i+1]-\gamma[i+1]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq 0,\\
\gamma[i]-\alpha[i+1]-\beta[i+1]-\gamma[i+1]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq -2,\\
-\beta[i]+\alpha[i+1]-\beta[i+1]-\gamma[i+1]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq -2,\\
-\beta[i]-\alpha[i+1]+\beta[i+1]-\gamma[i+1]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq -2,\\
-\beta[i]-\alpha[i+1]-\beta[i+1]+\gamma[i+1]+(\neg eq(\alpha[i],\beta[i],\gamma[i]))&\geq -2.
\end{aligned}
$$

MILP Model for Differential Characteristics of ARX Ciphers

- For every XOR operation with bit-level input differences a, b and bit-level output difference c, the constraints include

$$\begin{cases} d_{\oplus} \geq a, d_{\oplus} \geq b, d_{\oplus} \geq c \\ a+b+c \geq 2d_{\oplus} \\ a+b+c \leq 2 \end{cases}$$

  where $d_{\oplus}$ is a dummy bit variable.
- For each three-forked branch operation with input differences a, b and output difference c, the constraints should be

$$a = b = c.$$

- For the case of circular shift, we can also list some equations for the related bits.

MILP Model for Differential Characteristics of ARX Ciphers

- Only consider the modular addition with two inputs
- Assume that the two inputs to modular addition are independent.
- Assume that consecutive rounds are independent.
- Set objective function as $\sum_{j=1}^{r}\sum_{i=0}^{n-2}\neg eq(\alpha_j[i],\beta_j[i],\gamma_j[i])$ for r-round differential characteristic.

Note: the practical probability of our identified differential characteristics for some fixed key may vary significantly from that derived from our model due to the dependency issues.

How to calculate the linear correlation of addition modulo 2^n?

Theorem (see Wallén @ FSE'03, Nyberg and Wallén@ FSE'06)

For the linear approximation of addition modulo 2^n of two inputs with input masks $\Lambda_\alpha, \Lambda_\beta$ and output mask Γ, $\Lambda_\alpha, \Lambda_\beta, \Gamma \in \mathbb{F}_2^n$ and $\Lambda_\alpha = (\Lambda_\alpha[n-1], \ldots, \Lambda_\alpha[0])$, $\Lambda_\beta = (\Lambda_\beta[n-1], \ldots, \Lambda_\beta[0])$, $\Gamma = (\Gamma[n-1], \ldots, \Gamma[0])$, we define the vector $u = (u[n-1], \ldots, u[0])$ where $u[i] = 4\Gamma[i] + 2\Lambda_\alpha[i] + \Lambda_\beta[i], 0 \le u[i] < 8, 0 \le i < n$. The correlation can be computed with the following linear representation,

$$cor_{\boxplus}(\Gamma, \Lambda_\alpha, \Lambda_\beta) = LA_{u[n-1]}A_{u[n-2]} \cdots A_{u[1]}A_{u[0]}C,$$

where $A_r, r = 0, \ldots, 7$, is 2×2 matrice,

$$A_0 = \frac{1}{2}\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, A_1 = A_2 = -A_4 = \frac{1}{2}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

$$A_7 = \frac{1}{2}\begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}, -A_3 = A_5 = A_6 = \frac{1}{2}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

L is a row vector $L = (1\ 0)$,and C is a column vector $C = (1\ 1)^T$.

Example:

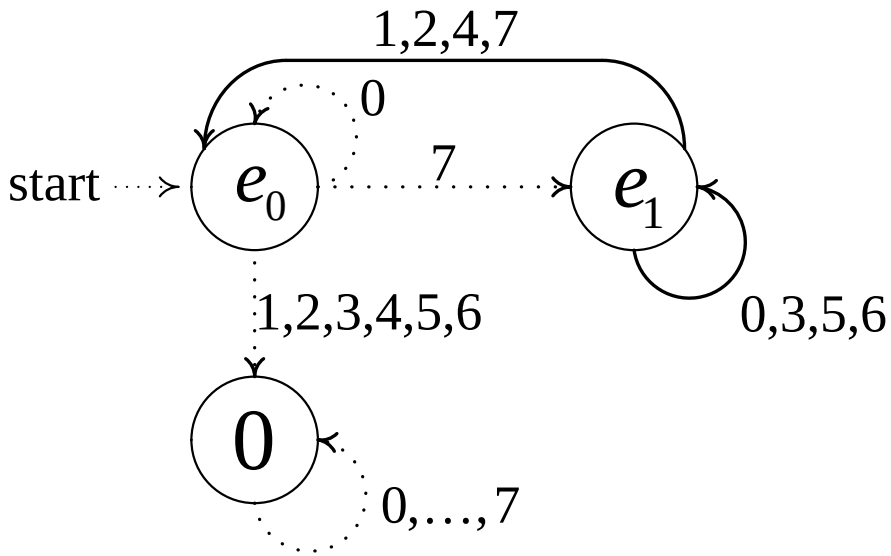
For the linear approximation with binary vector masks
$(\Gamma = 10100, \Lambda_\alpha = 11110, \Lambda_\beta = 11000)$, $u = 73620_8$, the $cor_{\boxplus}$=$LA_7A_3A_6A_2A_0C$ = -2^{-3}.

A Fast Implementation:

Nyberg and Wallén utilized the automaton to calculate $LA_{u[n-1]}A_{u[n-2]}\cdots A_{u[1]}A_{u[0]}C$ by multiplying from left to right. Let $e_0 = L = (1\ 0)$ and $e_1 = (0\ 1)$, then

$$e_0A_0 = e_0, \quad e_0A_7 = e_1, \quad e_0A_r = 0, for\ r \in [1,6], \quad e_1A_3 = -\frac{1}{2}e_1,$$

$$e_1A_0 = e_1A_5 = e_1A_6 = \frac{1}{2}e_1, e_1A_1 = e_1A_2 = e_1A_7 = \frac{1}{2}e_0, e_1A_4 = -\frac{1}{2}e_0.$$



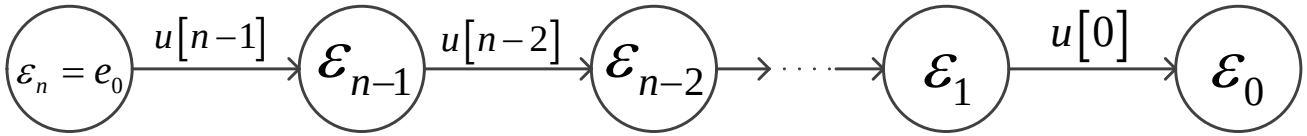
For example, as $u = 73620_8$, $LA_7A_3A_6A_2A_0C = -2^{-3}$.

Based on observation, we get

Proposition

If the correlation for the linear approximation is non-zero, the absolute value of the correlation can be computed as follows,

$$\left|cor_{\boxplus}(\Gamma,\Lambda_\alpha,\Lambda_\beta)x\right| = 2^{-\#\{0<i<n|\varepsilon_i=e_1\}}.$$



where $u = (u[n-1],\ldots,u[0])$, $u[i] = 4\Gamma[i] + 2\Lambda_\alpha[i] + \Lambda_\beta[i], 0 \le u[i] < 8, 0 \le i < n$.

For the state transition from ε_{i+1} to ε_i under $u[i]$, $0 \le i < n$, we define the bit variable s_i as follows, $s_i = 0$ if $\varepsilon_i = e_0$, and $s_i = 1$ if $\varepsilon_i = e_1$.

Construct the MILP Model for Linear Approximations of Modular Addition:

- Using the vector $(s_{i+1}, \Gamma[i], \Lambda_\alpha[i], \Lambda_\beta[i], s_i)$ to denote the state transition, so $e_{s_{i+1}} A_{u[i]} = e_{s_i}$.
- Adding the additional constraint $\varepsilon_n = e_0$.
- Using the **SAGE** and **greedy algorithm** to get linear inequalities.

Note: There are only 10(not 2^5) possible transitions for the vector and the 8 linear inequalities is listed below:

$$
\begin{aligned}
&s_{i+1} - \Gamma[i] - \Lambda_\alpha[i] + \Lambda_\beta[i] + s_i \geq 0, \quad && s_{i+1} + \Gamma[i] + \Lambda_\alpha[i] - \Lambda_\beta[i] - s_i \geq 0,\\
&s_{i+1} + \Gamma[i] - \Lambda_\alpha[i] - \Lambda_\beta[i] + s_i \geq 0, \quad && s_{i+1} - \Gamma[i] + \Lambda_\alpha[i] - \Lambda_\beta[i] + s_i \geq 0,\\
&s_{i+1} + \Gamma[i] - \Lambda_\alpha[i] + \Lambda_\beta[i] - s_i \geq 0, \quad && s_{i+1} - \Gamma[i] + \Lambda_\alpha[i] + \Lambda_\beta[i] - s_i \geq 0,\\
&-s_{i+1} + \Gamma[i] + \Lambda_\alpha[i] + \Lambda_\beta[i] + s_i \geq 0, \quad && s_{i+1} + \Gamma[i] + \Lambda_\alpha[i] + \Lambda_\beta[i] + s_i \leq 4.
\end{aligned}
$$

MILP Model for Linear Approximations for ARX Ciphers

- For each three-forked branch operation with input masks a, b and output mask c, the constraints should be

$$\begin{cases} d_{\oplus} \geq a, d_{\oplus} \geq b, d_{\oplus} \geq c \\ a+b+c \geq 2d_{\oplus} \\ a+b+c \leq 2 \end{cases}$$

  where $d_{\oplus}$ is a dummy bit variable.
- For every XOR operation with bit-level input differences a, b and bit-level output difference c, the constraints include

$$a = b = c.$$

- For the case of circular shift, we can also list some equations for the related bits.

MILP Model for Linear Approximations for ARX Ciphers

- Only consider the modular addition with two inputs
- Assume that the two inputs to modular addition are independent.
- Assume that consecutive rounds are independent.
- Set objective function as $\sum_{j=1}^{r}\sum_{i=1}^{n-1} s_i$ for r-round linear approximation.

Note: The practical correlation of our identified linear approximations for some fixed key may vary significantly from that derived from our model due to the dependency issues.

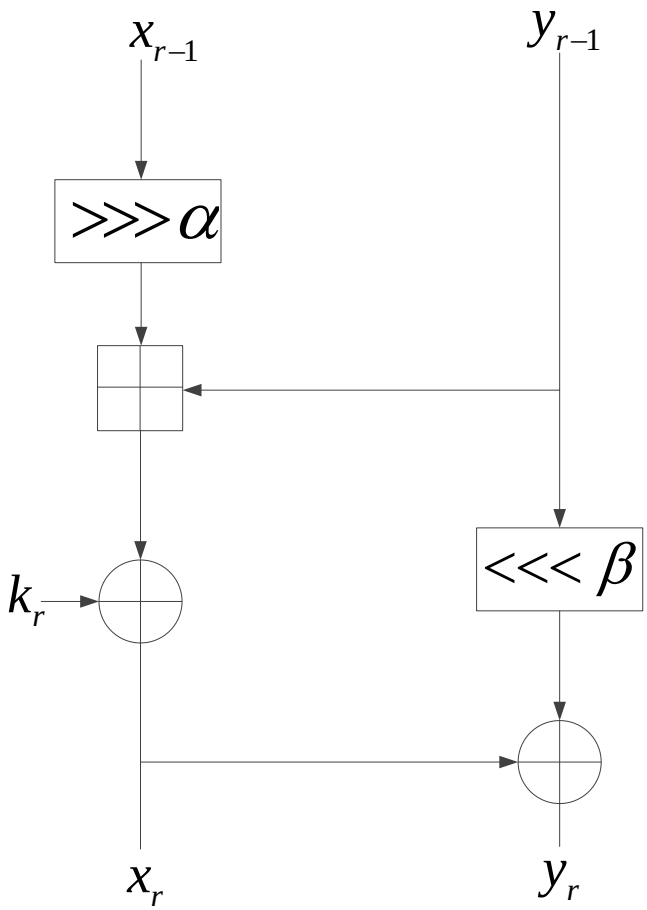
Speck:

- Proposed in 2003 by the researchers from the NSA
- Simplicity and Efficiency
- Based on ARX construction-Modular Addition, Bit Rotation and XOR
- Flexible and Suitable for a variety of platform

Parameters for Speck Family of Block Ciphers

Variant $2n/mn$	Block Size $2n$	Word Size n	Key Size mn	Key Words m	Rounds r	α	β
32/64	32	16	64	4	22	7	2
48/72	48	24	72	3	22	8	3
48/96	48	24	96	4	23	8	3
64/96	64	32	96	3	26	8	3
64/128	64	32	128	4	27	8	3
96/96	96	48	96	2	28	8	3
96/144	96	48	144	3	29	8	3
128/128	128	64	128	2	32	8	3
128/192	128	64	192	3	33	8	3
128/256	128	64	256	4	34	8	3

Round Function of Speck



Search Procedure

- *Step* 1 Convert the system of inequalities describing N rounds of Speck into a format that is readable by **Gurobi**.
- *Step* 2 Use **Gurobi** to search for trails with the input from *Step* 1.

Note

- The source code is published in **https://github.com/fukai6/milp_speck.git**.
- The code is used to convert the system of inequalities describing N rounds of Speck into LP format which is readable by **Gurobi**.
- Other MILP optimizers, such as **CPLEX**, also can be used.

http://www.gurobi.com

Table : Summary of Differential Characteristics and Linear Approximations for Speck

Cipher	Differential Characteristic			Linear Approximation		
	# Rounds	$\log_2 p$	Ref.	# Rounds	$\log_2 c$	Ref.
Speck32	9	-31	Abed *et al.* @ FSE'14	9	-14	Yao *et al.* @ ISC'15
	9	-30	Biyukov *et al.* @ FSE'14	9	-14	This paper
	9	-30	This paper			
Speck48	10	-41	Abed *et al.* @ FSE'14	9	-20	Yao *et al.* @ ISC'15
	11	-47	Biyukov *et al.* @ FSE'14	10	-22	This paper
	11	-45	This paper			
Speck64	13	-59	Abed *et al.* @ FSE'14	11	-25	Yao *et al.* @ ISC'15
	13	-51	This paper	12	-31	Yao *et al.* @ ISC'15
	14	-60	Biyukov *et al.* @ FSE'14	13	-30	This paper
	14	-56	This paper			
	15	-62	This paper			
Speck96	13	-84	Abed *et al.* @ FSE'14	6	-11	Yao *et al.* @ ISC'15
	13	-67	This paper	15	-45	This paper
	16	-87	This paper			
Speck128	14	-112	Abed *et al.* @ FSE'14	6	-11	Yao *et al.* @ ISC'15
	14	-90	This paper	16	-58	This paper
	19	-119	This paper			

Key Recovery Attack on Speck:

We use the generic key recovery framework for Speck given by Dinur @ SAC'14.

Table : Summary of Attacks on Speck

Variant $2n/mn$	Rounds Attacked/ Total Rounds	Time	Data	Memory	Method	Ref.
48/72	14/22	2^{65}	2^{41}	2^{22}	Differential	Dinur @ SAC'14
	15/22	$\mathbf{2^{70}}$	$\mathbf{2^{46}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**
48/96	15/23	2^{89}	2^{41}	2^{22}	Differential	Dinur @ SAC'14
	16/23	$\mathbf{2^{94}}$	$\mathbf{2^{46}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**
64/96	18/26	2^{93}	2^{61}	2^{22}	Differential	Dinur @ SAC'14
	19/26	$\mathbf{2^{95}}$	$\mathbf{2^{63}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**
64/128	19/27	2^{125}	2^{61}	2^{22}	Differential	Dinur @ SAC'14
	20/27	$\mathbf{2^{127}}$	$\mathbf{2^{63}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**
96/96	16/28	2^{85}	2^{85}	2^{22}	Differential	Dinur @ SAC'14
	19/28	$\mathbf{2^{88}}$	$\mathbf{2^{88}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**
96/144	17/29	2^{133}	2^{85}	2^{22}	Differential	Dinur @ SAC'14
	20/29	$\mathbf{2^{136}}$	$\mathbf{2^{88}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**
128/128	17/32	2^{113}	2^{113}	2^{22}	Differential	Dinur @ SAC'14
	22/32	$\mathbf{2^{120}}$	$\mathbf{2^{120}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**
128/192	18/33	2^{177}	2^{113}	2^{22}	Differential	Dinur @ SAC'14
	23/33	$\mathbf{2^{184}}$	$\mathbf{2^{120}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**
128/256	19/34	2^{241}	2^{113}	2^{22}	Differential	Dinur @ SAC'14
	24/34	$\mathbf{2^{248}}$	$\mathbf{2^{120}}$	$\mathbf{2^{22}}$	**Differential**	**This Paper**

Summarize

- We propose the MILP model to automatically search for differential and linear approximations for ARX ciphers
- Our identified differential characteristics for Speck64, Speck96 and Speck128 are extended for one, three and five rounds
- Our identified differential characteristic for Speck48 has higher probability
- We improve previous linear approximations for Speck variants with block size greater than 32
- We improve the currently best public attacks for Speck48, Speck64, Speck96 and Speck128

Thanks for Your Attention!