Integral Cryptanalysis of WARP based on Monomial Prediction

Hosein Hadipour Maria Eichlseder

FSE 2023 - Kobe, Japan

> hossein.hadipour@iaik.tugraz.at

SCIENCE PASSION TECHNOLOGY

Motivation and Our Contributions

Motivation

- Integral analysis of WARP

Contributions

- Providing a generic SAT model for integral analysis based on monomial prediction
- Our model takes the key schedule into account
- We proposed a tool for key-recovery taking the FFT technique into account
- Thanks to our tools, we improved the integral attack of WARP by 11 rounds

Hosein Hadipour, Maria Eichlseder

Motivation and Our Contributions

Motivation

- Integral analysis of WARP

Contributions

- Providing a generic SAT model for integral analysis based on monomial prediction
- Our model takes the key schedule into account
- We proposed a tool for key-recovery taking the FFT technique into account
- Thanks to our tools, we improved the integral attack of WARP by **11** rounds

Outline

1 Boolean Functions and Integral Analysis

2 Monomial Prediction and Our SAT Model

3 Application of Our Modeling to Integral Analysis of WARP

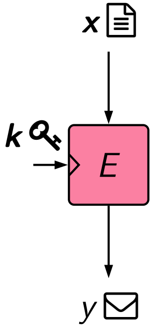
4 Key-Recovery

5 Conclusion

Boolean Functions and Integral Analysis

Integral Distinguisher and The Coefficients of ANF

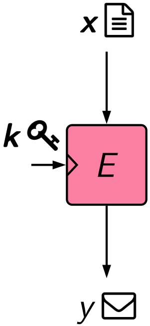
- $y = f(\boldsymbol{k}, \boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} \sum_{\boldsymbol{v} \in \mathbb{F}_2^k} a_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{k}^{\boldsymbol{v}} \boldsymbol{x}^{\boldsymbol{u}}$
- $\mathbb{C}_{\boldsymbol{u}} = \{x \in \mathbb{F}_2^n \,|\, \boldsymbol{x} \leq \boldsymbol{u}\}$
- $a_{\boldsymbol{u}}(\boldsymbol{k}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{k}, \boldsymbol{x})$
- Which monomial is key-independent in the ANF?
  - zero-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 0$
  - one-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 1$



Hosein Hadipour, Maria Eichlseder

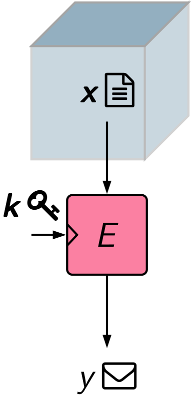
Integral Distinguisher and The Coefficients of ANF

- $y = f(\boldsymbol{k}, \boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$
- $\mathbb{C}_{\boldsymbol{u}} = \{x \in \mathbb{F}_2^n \,|\, \boldsymbol{x} \leq \boldsymbol{u}\}$
- $a_{\boldsymbol{u}}(\boldsymbol{k}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{k}, \boldsymbol{x})$
- Which monomial is key-independent in the ANF?
  - zero-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 0$
  - one-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 1$



Integral Distinguisher and The Coefficients of ANF

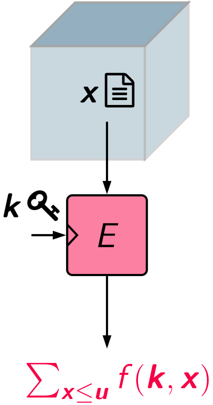
- $y = f(\boldsymbol{k}, \boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$
- $\mathbb{C}_{\boldsymbol{u}} = \{x \in \mathbb{F}_2^n \mid \boldsymbol{x} \leq \boldsymbol{u}\}$
- $a_{\boldsymbol{u}}(\boldsymbol{k}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{k}, \boldsymbol{x})$
- Which monomial is key-independent in the ANF?
  - zero-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 0$
  - one-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 1$



Hosein Hadipour, Maria Eichlseder

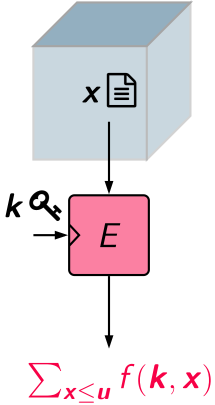
Integral Distinguisher and The Coefficients of ANF

- $y = f(\boldsymbol{k}, \boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$
- $\mathbb{C}_{\boldsymbol{u}} = \{x \in \mathbb{F}_2^n \,|\, \boldsymbol{x} \leq \boldsymbol{u}\}$
- $a_{\boldsymbol{u}}(\boldsymbol{k}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{k}, \boldsymbol{x})$
- Which monomial is key-independent in the ANF?
  - zero-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 0$
  - one-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 1$



Integral Distinguisher and The Coefficients of ANF

- $y = f(\boldsymbol{k}, \boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$
- $\mathbb{C}_{\boldsymbol{u}} = \{x \in \mathbb{F}_2^n \,|\, \boldsymbol{x} \leq \boldsymbol{u}\}$
- $a_{\boldsymbol{u}}(\boldsymbol{k}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{k}, \boldsymbol{x})$
- Which monomial is key-independent in the ANF?
  - zero-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 0$
  - one-sum: $\exists\, u,\ s.t. \forall \boldsymbol{k} :\ a_{\boldsymbol{u}}(\boldsymbol{k}) = 1$

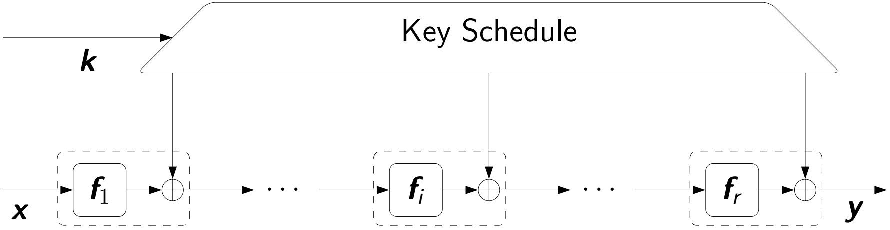


Hosein Hadipour, Maria Eichlseder

Monomial Prediction and Our SAT Model

Hosein Hadipour, Maria Eichlseder

Core Idea of Monomial Prediction [Hu+20]

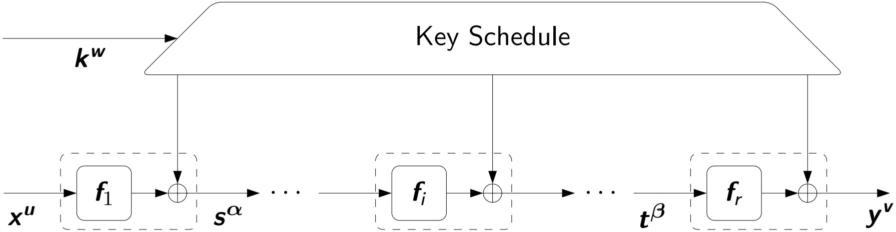


Core Idea

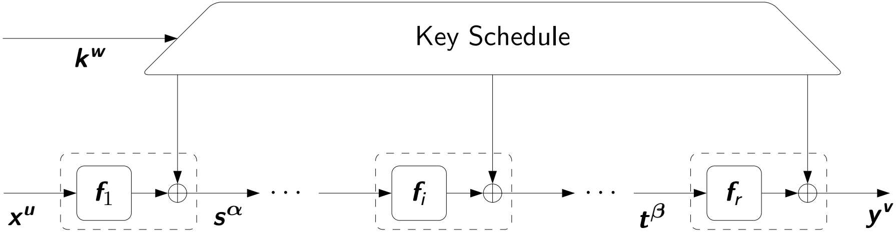
The absence (or presence) of a monomial in the ANF of a composite function can be checked by tracking the propagation of the given monomial through the building blocks of composite functions.

Hosein Hadipour, Maria Eichlseder

Monomial Trail and Integral Distinguisher

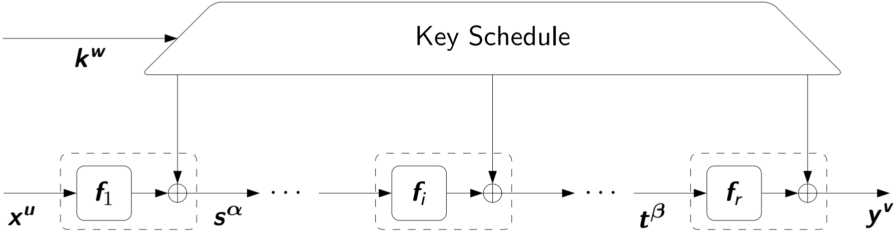


Monomial Trail and Integral Distinguisher



$$\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \xrightarrow{f_1} \boldsymbol{s}^{\boldsymbol{\alpha}} \xrightarrow{f_1} \cdots \xrightarrow{f_i} \cdots \xrightarrow{f_{r-1}} \boldsymbol{t}^{\boldsymbol{\beta}} \xrightarrow{f_r} \boldsymbol{y}^{\boldsymbol{v}}$$

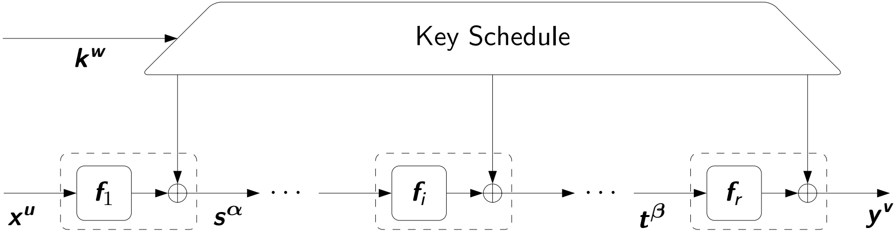
Monomial Trail and Integral Distinguisher



$$\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \xrightarrow{f_1} \boldsymbol{s}^{\boldsymbol{\alpha}} \xrightarrow{f_1} \cdots \xrightarrow{f_i} \cdots \xrightarrow{f_{r-1}} \boldsymbol{t}^{\boldsymbol{\beta}} \xrightarrow{f_r} \boldsymbol{y}^{\boldsymbol{v}}$$

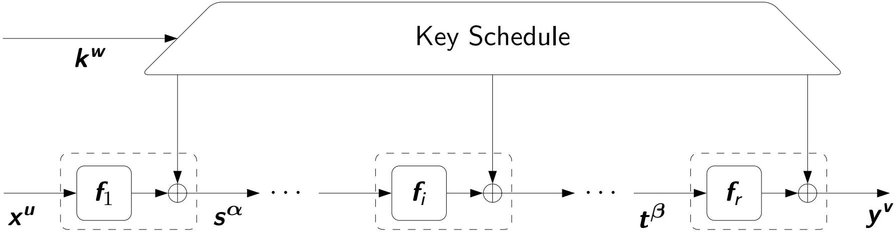
$$\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$$

Monomial Trail and Integral Distinguisher



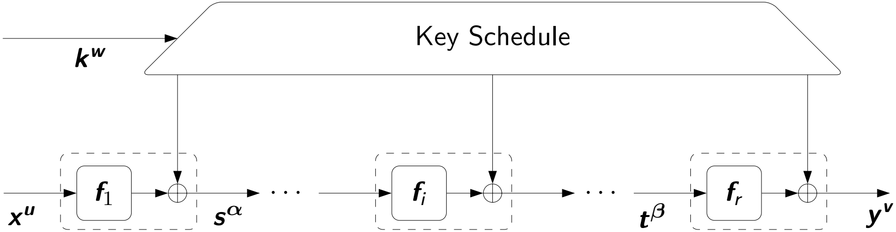
$$\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \rightarrow \boldsymbol{y}^{\boldsymbol{v}} \Rightarrow \boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$$

Monomial Trail and Integral Distinguisher



$$\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \not\rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}} \Rightarrow \boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \not\rightarrow \boldsymbol{y}^{\boldsymbol{v}}$$

Monomial Trail and Integral Distinguisher

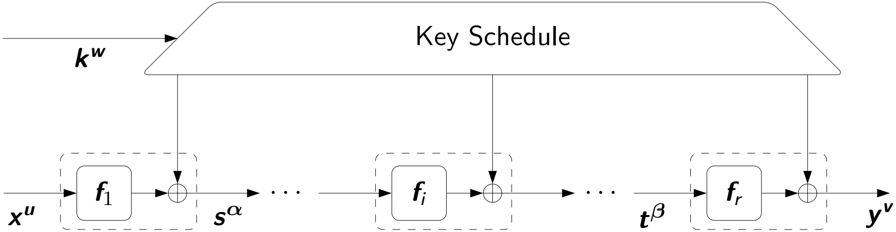


$$y^v = \sum_{u \in \mathbb{F}_2^n} \sum_{v \in \mathbb{F}_2^k} a_{u,v} k^v x^u = \sum_{u \in \mathbb{F}_2^n} a_u(k) \cdot x^u$$

From Monomial Trails to Integral Distinguisher

- If $\exists u$ s.t. $k^w x^u \not\rightsquigarrow y^v$ for all $w \in \mathbb{F}_2^k$ then $a_u(k) = 0$ (zero-sum)
- If $\exists u$ s.t. $k^w x^u \not\rightsquigarrow y^v$ for all $w \in \mathbb{F}_2^k \setminus \{0\}$ then $a_u(k) = \texttt{constant}$ (zero/one-sum)

Monomial Trail and Integral Distinguisher

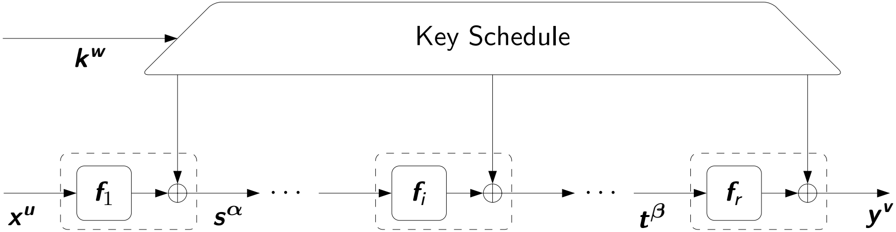


$$\boldsymbol{y}^{\boldsymbol{v}} = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} \sum_{\boldsymbol{v} \in \mathbb{F}_2^k} a_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{k}^{\boldsymbol{v}} \boldsymbol{x}^{\boldsymbol{u}} = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$$

From Monomial Trails to Integral Distinguisher

- If $\exists \boldsymbol{u}$ s.t. $\boldsymbol{k}^{\boldsymbol{w}} \boldsymbol{x}^{\boldsymbol{u}} \not\rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$ for all $\boldsymbol{w} \in \mathbb{F}_2^k$ then $a_{\boldsymbol{u}}(\boldsymbol{k}) = 0$ (zero-sum)
- If $\exists \boldsymbol{u}$ s.t. $\boldsymbol{k}^{\boldsymbol{w}} \boldsymbol{x}^{\boldsymbol{u}} \not\rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$ for all $\boldsymbol{w} \in \mathbb{F}_2^k \setminus \{0\}$ then $a_{\boldsymbol{u}}(\boldsymbol{k}) = \texttt{constant}$ (zero/one-sum)

Monomial Trail and Integral Distinguisher



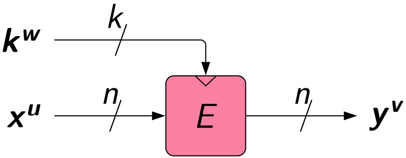
$$\boldsymbol{y}^{\boldsymbol{v}} = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} \sum_{\boldsymbol{v}\in\mathbb{F}_2^k} a_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{k}^{\boldsymbol{v}} \boldsymbol{x}^{\boldsymbol{u}} = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$$

From Monomial Trails to Integral Distinguisher

- If $\exists \boldsymbol{u}$ s.t. $\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \not\rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$ for all $\boldsymbol{w} \in \mathbb{F}_2^k$ then $a_{\boldsymbol{u}}(\boldsymbol{k}) = 0$ (zero-sum)
- If $\exists \boldsymbol{u}$ s.t. $\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \not\rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$ for all $\boldsymbol{w} \in \mathbb{F}_2^k \setminus \{0\}$ then $a_{\boldsymbol{u}}(\boldsymbol{k}) = \texttt{constant}$ (zero/one-sum)

Hosein Hadipour, Maria Eichlseder

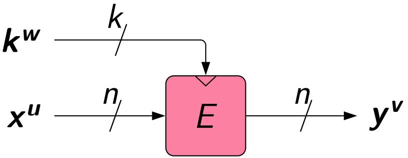
From Monomial Prediction to SAT Problem



$$y^v = \sum_{u \in \mathbb{F}_2^n} \sum_{v \in \mathbb{F}_2^k} a_{u,v} k^v x^u = \sum_{u \in \mathbb{F}_2^n} a_u(k) \cdot x^u$$

- Model the propagation of monomial trails through the building blocks by a CNF clause
- Main variables are the monomial exponents, i.e., $u, w, v, \ldots$ not $x, k, y, \ldots$
- Fix u to a certain vector and set v to e_i (w should be a free variable but non-zero)
- Any possible solution of the model is a monomial trail from $k^w x^u$ to y^v
- If the model is impossible, then $k^w x^u \not\rightsquigarrow y^v$ for all $w \in \mathbb{F}_2^k$, and $a_u(k) = \texttt{constant}$

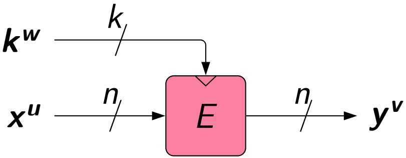
From Monomial Prediction to SAT Problem



$$y^v = \sum_{u \in \mathbb{F}_2^n} \sum_{v \in \mathbb{F}_2^k} a_{u,v} k^v x^u = \sum_{u \in \mathbb{F}_2^n} a_u(k) \cdot x^u$$

- Model the propagation of monomial trails through the building blocks by a `CNF` clause
- Main variables are the monomial exponents, i.e., $u, w, v, \ldots$ not $x, k, y, \ldots$
- Fix u to a certain vector and set v to e_i (w should be a free variable but non-zero)
- Any possible solution of the model is a monomial trail from $k^w x^u$ to y^v
- If the model is impossible, then $k^w x^u \not\rightsquigarrow y^v$ for all $w \in \mathbb{F}_2^k$, and $a_u(k) = \texttt{constant}$

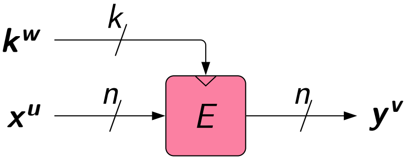
From Monomial Prediction to SAT Problem



$$y^v = \sum_{u \in \mathbb{F}_2^n} \sum_{v \in \mathbb{F}_2^k} a_{u,v} k^v x^u = \sum_{u \in \mathbb{F}_2^n} a_u(k) \cdot x^u$$

- Model the propagation of monomial trails through the building blocks by a CNF clause
- Main variables are the monomial exponents, i.e., $u, w, v, \ldots$ not $x, k, y, \ldots$
- Fix u to a certain vector and set v to e_i (w should be a free variable but non-zero)
- Any possible solution of the model is a monomial trail from $k^w x^u$ to y^v
- If the model is impossible, then $k^w x^u \not\rightsquigarrow y^v$ for all $w \in \mathbb{F}_2^k$, and $a_u(k) = \texttt{constant}$

From Monomial Prediction to SAT Problem

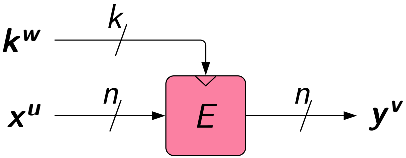


$$\boldsymbol{y}^{\boldsymbol{v}} = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} \sum_{\boldsymbol{v}\in\mathbb{F}_2^k} a_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{k}^{\boldsymbol{v}} \boldsymbol{x}^{\boldsymbol{u}} = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$$

- Model the propagation of monomial trails through the building blocks by a `CNF` clause
- Main variables are the monomial exponents, i.e., $\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}, \ldots$ not $\boldsymbol{x}, \boldsymbol{k}, \boldsymbol{y}, \ldots$
- Fix $\boldsymbol{u}$ to a certain vector and set $\boldsymbol{v}$ to $\boldsymbol{e}_i$ ($\boldsymbol{w}$ should be a free variable but non-zero)
- Any possible solution of the model is a monomial trail from $\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}}$ to $\boldsymbol{y}^{\boldsymbol{v}}$
- If the model is impossible, then $\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \not\rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$ for all $\boldsymbol{w} \in \mathbb{F}_2^k$, and $a_{\boldsymbol{u}}(\boldsymbol{k}) = \texttt{constant}$

Hosein Hadipour, Maria Eichlseder

From Monomial Prediction to SAT Problem

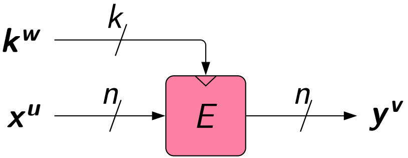


$$\boldsymbol{y}^{\boldsymbol{v}} = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} \sum_{\boldsymbol{v}\in\mathbb{F}_2^k} a_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{k}^{\boldsymbol{v}} \boldsymbol{x}^{\boldsymbol{u}} = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$$

- Model the propagation of monomial trails through the building blocks by a `CNF` clause
- Main variables are the monomial exponents, i.e., $\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}, \ldots$ not $\boldsymbol{x}, \boldsymbol{k}, \boldsymbol{y}, \ldots$
- Fix $\boldsymbol{u}$ to a certain vector and set $\boldsymbol{v}$ to $\boldsymbol{e}_i$ ($\boldsymbol{w}$ should be a free variable but non-zero)
- Any possible solution of the model is a monomial trail from $\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}}$ to $\boldsymbol{y}^{\boldsymbol{v}}$
- If the model is impossible, then $\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \not\rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$ for all $\boldsymbol{w} \in \mathbb{F}_2^k$, and $a_{\boldsymbol{u}}(\boldsymbol{k}) = \texttt{constant}$

Hosein Hadipour, Maria Eichlseder

From Monomial Prediction to SAT Problem



$$\boldsymbol{y}^{\boldsymbol{v}} = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n}\sum_{\boldsymbol{v}\in\mathbb{F}_2^k} a_{\boldsymbol{u},\boldsymbol{v}}\boldsymbol{k}^{\boldsymbol{v}}\boldsymbol{x}^{\boldsymbol{u}} = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k})\cdot\boldsymbol{x}^{\boldsymbol{u}}$$

- Model the propagation of monomial trails through the building blocks by a CNF clause
- Main variables are the monomial exponents, i.e., $\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}, \ldots$ not $\boldsymbol{x}, \boldsymbol{k}, \boldsymbol{y}, \ldots$
- Fix $\boldsymbol{u}$ to a certain vector and set $\boldsymbol{v}$ to $\boldsymbol{e}_i$ ($\boldsymbol{w}$ should be a free variable but non-zero)
- Any possible solution of the model is a monomial trail from $\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}}$ to $\boldsymbol{y}^{\boldsymbol{v}}$
- If the model is impossible, then $\boldsymbol{k}^{\boldsymbol{w}}\boldsymbol{x}^{\boldsymbol{u}} \not\rightsquigarrow \boldsymbol{y}^{\boldsymbol{v}}$ for all $\boldsymbol{w} \in \mathbb{F}_2^k$, and $a_{\boldsymbol{u}}(\boldsymbol{k}) = \texttt{constant}$

Hosein Hadipour, Maria Eichlseder

Monomial Prediction Table ($\texttt{MPT}$)

- Let $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ be an m-bit to n-bit vectorial Boolean function. Then $\texttt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 1$ if $\boldsymbol{x}^{\boldsymbol{u}} \xrightarrow{f} \boldsymbol{y}^{\boldsymbol{v}}$, and $\texttt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 0$ otherwise.

Monomial Prediction Table (MPT)

- Let $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ be an m-bit to n-bit vectorial Boolean function. Then $\mathtt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 1$ if $\boldsymbol{x}^{\boldsymbol{u}} \xrightarrow{f} \boldsymbol{y}^{\boldsymbol{v}}$, and $\mathtt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 0$ otherwise.

x	$S(x)$
0	c
1	a
2	d
3	3
4	e
5	b
6	f
7	7
8	8
9	9
a	1
b	5
c	0
d	2
e	4
f	6

Monomial Prediction Table (MPT)

- Let $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ be an m-bit to n-bit vectorial Boolean function. Then $\texttt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 1$ if $\boldsymbol{x}^{\boldsymbol{u}} \xrightarrow{f} \boldsymbol{y}^{\boldsymbol{v}}$, and $\texttt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 0$ otherwise.

x	$S(x)$
0	c
1	a
2	d
3	3
4	e
5	b
6	f
7	7
8	8
9	9
a	1
b	5
c	0
d	2
e	4
f	6

$\boldsymbol{u} \backslash \boldsymbol{v}$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	1	·	·	·	1	·	·	·	1	·	·	·	1	·	·	·
1	·	·	1	·	1	·	·	·	·	·	1	·	1	·	·	·
2	·	1	·	·	·	1	·	·	·	1	·	·	·	1	·	·
3	·	·	·	1	·	1	·	·	1	1	1	·	·	1	·	·
4	·	·	1	·	·	·	1	·	·	·	1	·	·	·	1	·
5	·	1	1	1	·	·	1	·	·	1	1	1	·	·	1	·
6	·	·	·	1	·	·	·	1	·	·	·	1	·	·	·	1
7	·	1	·	·	1	1	1	·	·	1	·	·	·	·	·	1
8	·	·	·	·	1	·	·	·	·	·	·	·	1	·	·	·
9	·	1	1	·	1	·	·	·	·	1	1	·	1	·	·	·
a	·	·	·	·	·	1	·	·	1	1	·	·	·	1	·	·
b	·	1	·	1	1	·	·	·	1	·	1	·	·	1	·	·
c	·	·	1	·	·	·	1	·	1	·	1	·	·	·	1	·
d	·	·	·	1	·	·	1	·	·	·	1	1	·	·	1	·
e	·	1	·	1	1	·	·	1	1	·	·	1	·	·	·	1
f	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1

Monomial Prediction Table (MPT)

- Let $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ be an m-bit to n-bit vectorial Boolean function. Then $\mathtt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 1$ if $\boldsymbol{x}^{\boldsymbol{u}} \xrightarrow{f} \boldsymbol{y}^{\boldsymbol{v}}$, and $\mathtt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 0$ otherwise.

x	$S(x)$
0	c
1	a
2	d
3	3
4	e
5	b
6	f
7	7
8	8
9	9
a	1
b	5
c	0
d	2
e	4
f	6

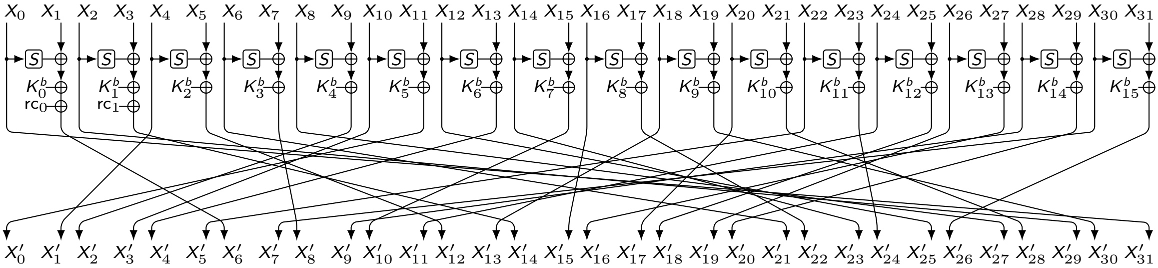
$$
\begin{array}{lll}
(u_2 \vee \neg v_1 \vee \neg v_3) & \wedge (\neg u_1 \vee \neg v_0 \vee \neg v_1 \vee v_2) & \wedge (\neg u_0 \vee \neg u_1 \vee \neg u_2 \vee \neg v_2 \vee v_3) \\
\wedge (u_2 \vee u_3 \vee \neg v_3) & \wedge (\neg u_0 \vee \neg u_1 \vee \neg u_3 \vee v_2) & \wedge (\neg u_0 \vee \neg u_3 \vee v_0 \vee \neg v_1 \vee \neg v_3) \\
\wedge (u_1 \vee \neg v_1 \vee \neg v_2) & \wedge (\neg u_1 \vee u_2 \vee v_0 \vee v_2 \vee v_3) & \wedge (\neg u_0 \vee \neg u_1 \vee \neg u_3 \vee v_0 \vee v_1 \vee v_3) \\
\wedge (u_1 \vee u_3 \vee \neg v_2) & \wedge (u_2 \vee \neg u_3 \vee v_1 \vee v_2 \vee v_3) & \wedge (\neg u_0 \vee \neg u_2 \vee \neg u_3 \vee \neg v_0 \vee v_1 \vee \neg v_3) \\
\wedge (u_0 \vee \neg u_2 \vee u_3 \vee v_3) & \wedge (u_1 \vee \neg v_0 \vee \neg v_2 \vee \neg v_3) & \wedge (\neg u_1 \vee \neg u_2 \vee \neg u_3 \vee v_1 \vee \neg v_2) \\
\wedge (u_0 \vee \neg u_1 \vee u_3 \vee v_2) & \wedge (\neg u_0 \vee u_1 \vee u_3 \vee v_0 \vee v_1) & \wedge (\neg u_1 \vee \neg u_2 \vee \neg u_3 \vee v_1 \vee v_3) \\
\wedge (\neg u_2 \vee v_0 \vee v_1 \vee v_3) & \wedge (\neg u_1 \vee u_3 \vee \neg v_0 \vee v_2 \vee \neg v_3) & \wedge (u_0 \vee u_1 \vee \neg u_3 \vee v_0 \vee v_1 \vee v_2) \\
\wedge (u_0 \vee u_1 \vee u_2 \vee \neg v_3) & \wedge (u_0 \vee u_1 \vee \neg u_2 \vee \neg v_1 \vee v_3) & \wedge (\neg u_3 \vee v_0 \vee \neg v_1 \vee \neg v_2 \vee \neg v_3) \\
\wedge (u_1 \vee u_2 \vee \neg v_2 \vee \neg v_3) & \wedge (u_1 \vee \neg u_2 \vee u_3 \vee \neg v_1 \vee v_3) & \wedge (\neg u_0 \vee u_1 \vee u_2 \vee v_1 \vee v_2 \vee v_3) \\
\wedge (\neg u_2 \vee \neg v_0 \vee \neg v_1 \vee v_3) & \wedge (\neg u_1 \vee u_3 \vee \neg v_1 \vee v_2 \vee \neg v_3). &
\end{array}
$$

Hosein Hadipour, Maria Eichlseder

Application of Our Modeling to Integral Analysis of WARP

WARP[Ban+20]

- Proposed in SAC 2020 [Ban+20] as the lightweight alternative of `AES-128`
- 128-bit block/key size, and 41 rounds (40.5 rounds)
- Splits 128-bit K into two halves $K^{(0)}||K^{(1)}$ and uses $K^{(r-1 \mod 2)}$ in the rth round



Hosein Hadipour, Maria Eichlseder

22-round Integral Distinguisher for WARP

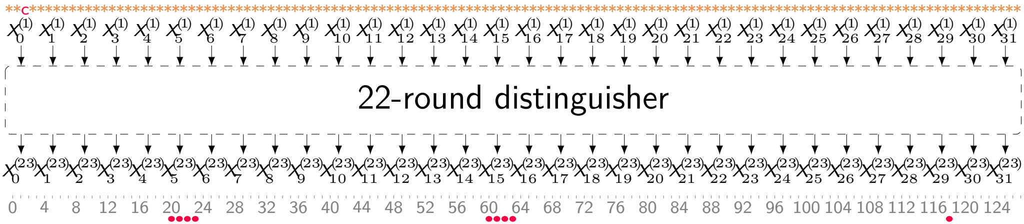
The best previous integral distinguisher: 20 rounds [Ban+20]

$$(2) \xrightarrow{\text{22 rounds}} (\underline{20, 21, 22, 23}, \ 118, \ \underline{60, 61, 62, 63}),$$

22-round Integral Distinguisher for WARP

The best previous integral distinguisher: 20 rounds [Ban+20]

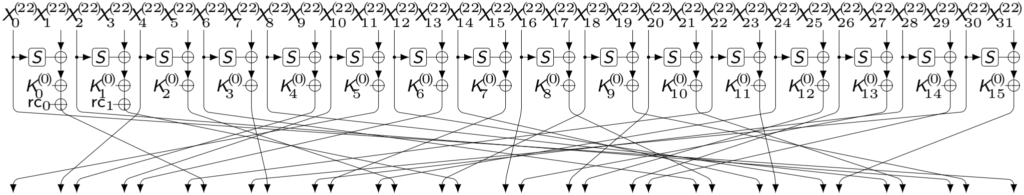
$$(2) \xrightarrow{\text{22 rounds}} (\underline{20, 21, 22, 23},\ \ 118,\ \ \underline{60, 61, 62, 63}),$$



Hosein Hadipour, Maria Eichlseder
FSE 2023 - Kobe, Japan

23-round Integral Distinguisher for WARP

- Any r-round integral distinguisher of WARP can be extended by 1 round

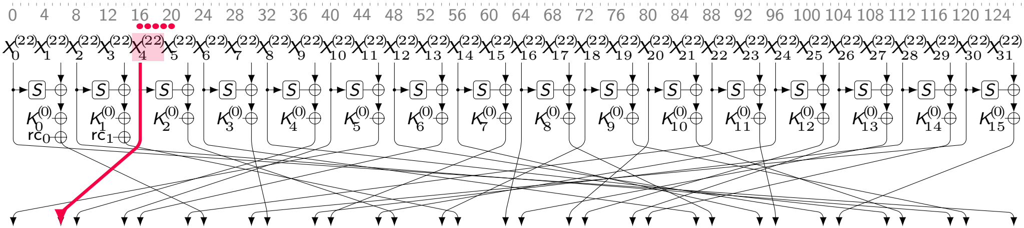


- $\sum_{\mathbb{C}} X_4^{(22)} = \sum_{\mathbb{C}} X_1^{(23)}$

- $\sum_{\mathbb{C}} X_{11}^{(22)} = \sum_{\mathbb{C}} \left(S(X_4^{(23)}) \oplus X_0^{(23)}\right) \oplus \sum_{\mathbb{C}} K_i^{(b)}$

23-round Integral Distinguisher for WARP

Any r-round integral distinguisher of WARP can be extended by 1 round

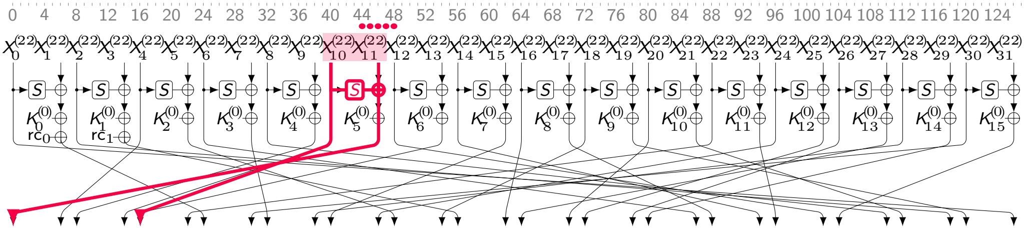


$\sum_{\mathbb{C}} X_4^{(22)} = \sum_{\mathbb{C}} X_1^{(23)}$

$\sum_{\mathbb{C}} X_{11}^{(22)} = \sum_{\mathbb{C}} \left(S(X_4^{(23)}) \oplus X_0^{(23)} \right) \oplus \sum_{\mathbb{C}} K_i^{(b)}$

23-round Integral Distinguisher for WARP

Any r-round integral distinguisher of WARP can be extended by 1 round

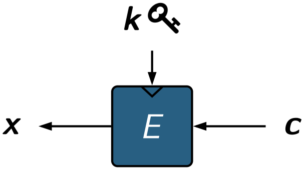


$\sum_{\mathbb{C}} X_4^{(22)} = \sum_{\mathbb{C}} X_1^{(23)}$

$\sum_{\mathbb{C}} X_{11}^{(22)} = \sum_{\mathbb{C}} \left(S(X_4^{(23)}) \oplus X_0^{(23)} \right) \oplus \sum_{\mathbb{C}} K_i^{(b)}$

Key-Recovery

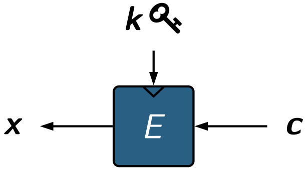
Naive Approach v.s. FFT Technique [TA14]



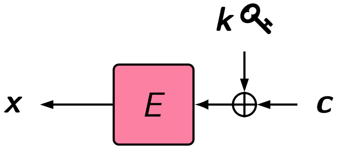
- Naive approach:
  - $\sum \boldsymbol{x} = \sum_{c \in \mathbb{C}} f(\boldsymbol{k}, \boldsymbol{c})$
  - $T_{tot} = 2^{|\mathbf{k}|}|\mathbb{C}|$, where $\mathbb{C} = \mathbf{2}^{|\mathbf{k}|}$
  - $T_{tot} = 2^{2|\boldsymbol{k}|}$
- FFT technique:
  - $\sum \boldsymbol{x} = \sum_{c \in \mathbb{C}} F(\boldsymbol{k} \oplus \boldsymbol{c})$
  - $T_{tot} = 4 \cdot |\boldsymbol{k}| \cdot 2^{|\boldsymbol{k}|}$

Naive Approach v.s. FFT Technique [TA14]

- Naive approach:
  - $\sum \boldsymbol{x} = \sum_{c \in \mathbb{C}} f(\boldsymbol{k}, \boldsymbol{c})$
  - $T_{tot} = 2^{|\mathbf{k}|} |\mathbb{C}|$, where $\mathbb{C} = \mathbf{2}^{|\mathbf{k}|}$
  - $T_{tot} = 2^{2|\boldsymbol{k}|}$



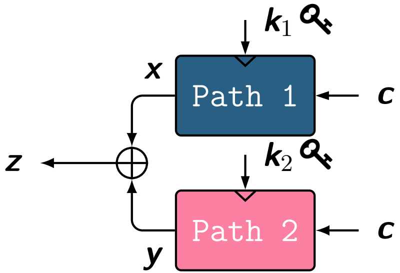
- FFT technique:
  - $\sum \boldsymbol{x} = \sum_{c \in \mathbb{C}} F(\boldsymbol{k} \oplus \boldsymbol{c})$
  - $T_{tot} = 4 \cdot |\boldsymbol{k}| \cdot 2^{|\boldsymbol{k}|}$



Hosein Hadipour, Maria Eichlseder

MitM [SW12]

- Naive approach:
  - $\boldsymbol{x} = F(\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{c})$
  - $T = |\mathbb{C}| \cdot 2^{|\boldsymbol{k}_1 \cup \boldsymbol{k}_2|}$
- MitM:
  - $\boldsymbol{x} = g(\boldsymbol{k}_1, \boldsymbol{c}),\ \boldsymbol{y} = h(\boldsymbol{k}_2, \boldsymbol{c})$
  - $T = |\mathbb{C}| \cdot 2^{|\boldsymbol{k}_1|} + |\mathbb{C}| \cdot 2^{|\boldsymbol{k}_2|}$



$$\sum \boldsymbol{z} = 0$$

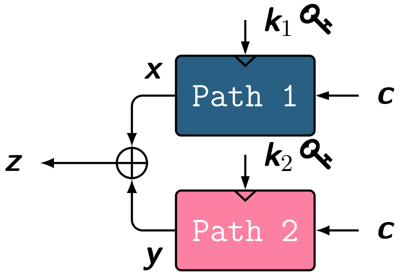
MitM [SW12]

Naive approach:

- $\boldsymbol{x} = F(\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{c})$
- $T = |\mathbb{C}| \cdot 2^{|\boldsymbol{k}_1 \cup \boldsymbol{k}_2|}$

MitM:

- $\boldsymbol{x} = g(\boldsymbol{k}_1, \boldsymbol{c}),\ \boldsymbol{y} = h(\boldsymbol{k}_2, \boldsymbol{c})$
- $T = |\mathbb{C}| \cdot 2^{|\boldsymbol{k}_1|} + |\mathbb{C}| \cdot 2^{|\boldsymbol{k}_2|}$



$$\sum \boldsymbol{z} = 0 \iff \sum \boldsymbol{x} = \sum \boldsymbol{y}$$

Overall View of Our Key-Recovery Tool

1- Assume that $\boldsymbol{x} = \boldsymbol{y} \oplus \boldsymbol{z}$ and $\sum \boldsymbol{x} = 0$

2- For each path, i.e., $\boldsymbol{y}$, and $\boldsymbol{z}$:

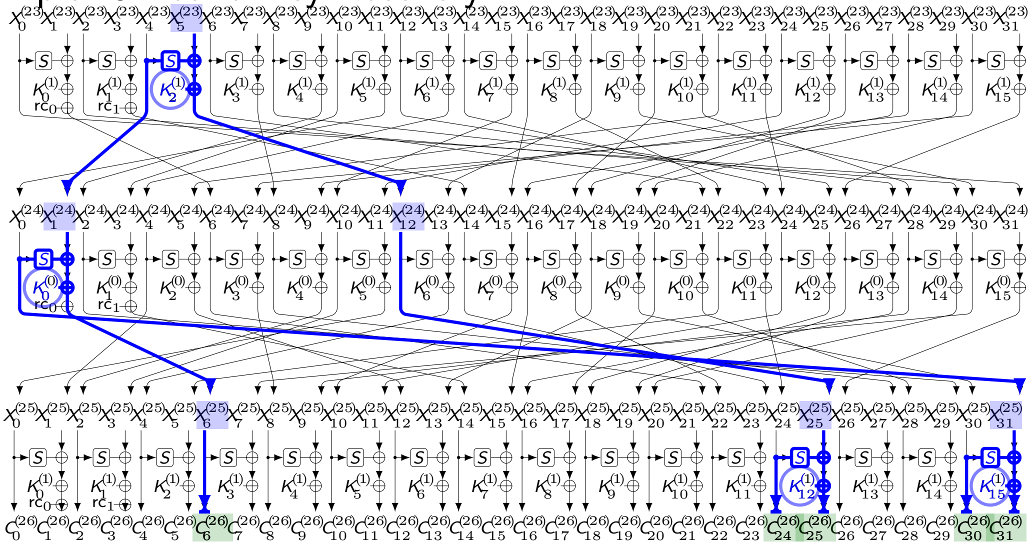
- Build the graph of dependencies: $\boldsymbol{y} = f(\boldsymbol{k}, \boldsymbol{c})$
- Simplify the dependency graph: reform $f(\boldsymbol{k}, \boldsymbol{c})$ to $F(\tilde{\boldsymbol{k}} \oplus \tilde{\boldsymbol{c}})$
- Use FFT to compute the list $[\sum \boldsymbol{y} \mid \tilde{\boldsymbol{k}} = 0, \ldots, 2^{|\boldsymbol{k}|-1}]$

3- Compare the two lists to find candidates for the involved key bits

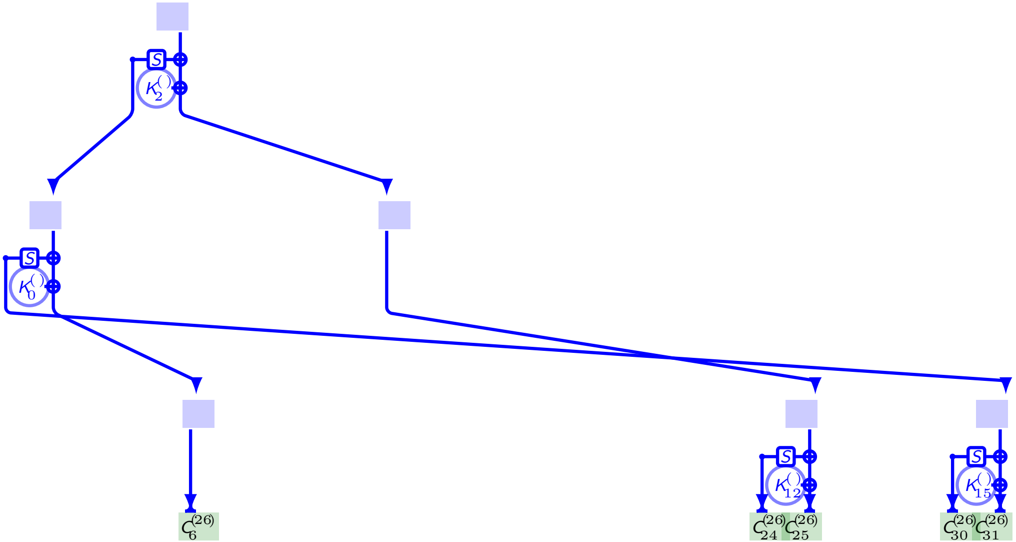
4- Brute force the remaining keys to find the correct key

Hosein Hadipour, Maria Eichlseder

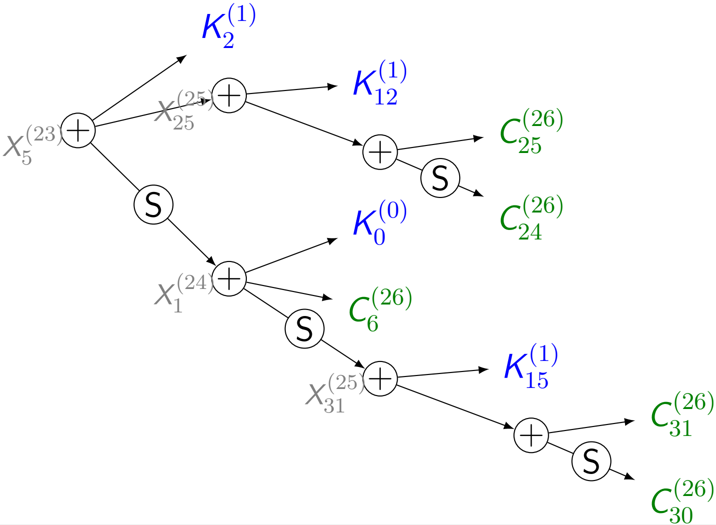
Example: 3-Round Key Recovery



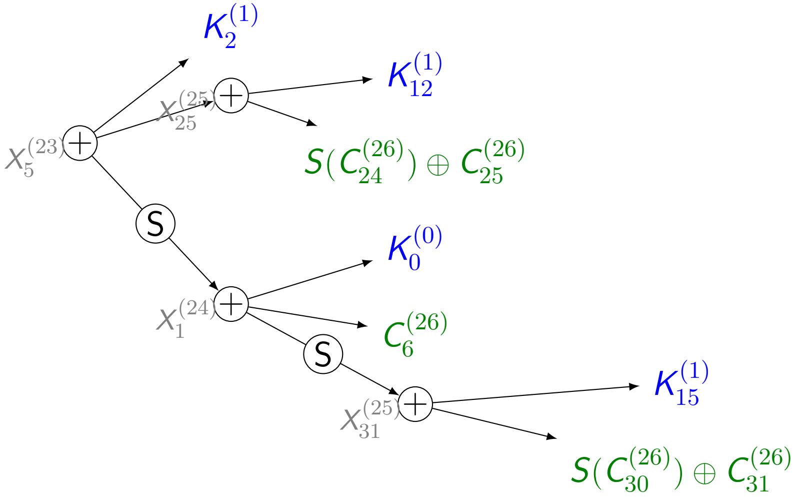
Example: 3-Round Key Recovery



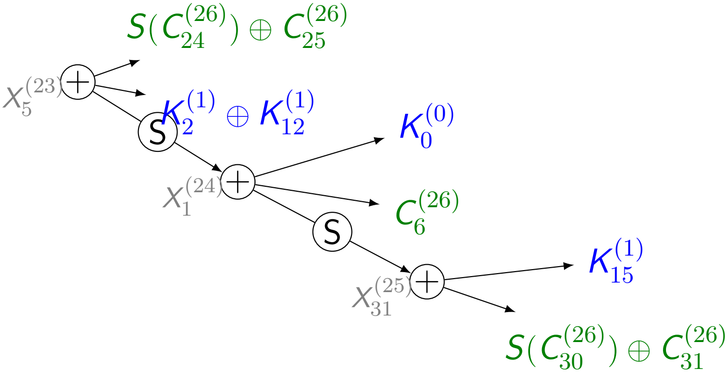
Example: Dependency Graph



Example: Dependency Graph



Example: Dependency Graph



Summary of Our Result

#R	Data	Time	Memory	Attack	Reference
32	2^{127}	2^{127}	2^{108}	Integral	This paper
21	2^{124}	-	-	Integral	[Ban+20]
18	$2^{104.62}$	-	-	Differential	[TB22]
21	-	-	-	Impossible diff.	[Ban+20]
21	2^{113}	2^{113}	2^{72}	Differential	[KY21]
23	$2^{106.62}$	$2^{106.62}$	$2^{106.62}$	Differential	[TB22]
24	$2^{126.06}$	$2^{125.18}$	$2^{127.06}$	Rectangle	[TB22]

Conclusion

Contributions

- We provided a SAT model for integral analysis based on Monomial prediction
- Our modeling is generic and can be applied to other (binary field) block ciphers
- We proposed a tool for key-recovery taking the FFT technique into account
- Overall, we improved the integral attack of `WARP` by **11** rounds

Thanks for your attention!

https://github.com/hadipourh/mpt

Bibliography I

[Ban+20] Subhadeep Banik et al. **WARP: Revisiting GFN for Lightweight 128-Bit Block Cipher**. SAC 2020. Vol. 12804. LNCS. Springer, 2020, pp. 535–564. DOI: 10.1007/978-3-030-81652-0_21.

[Hu+20] Kai Hu et al. **An Algebraic Formulation of the Division Property: Revisiting Degree Evaluations, Cube Attacks, and Key-Independent Sums**. ASIACRYPT 2020. Vol. 12491. LNCS. Springer, 2020, pp. 446–476. DOI: 10.1007/978-3-030-64837-4_15.

[KY21] Manoj Kumar and Tarun Yadav. **MILP Based Differential Attack on Round Reduced WARP**. SPACE 2021. Vol. 13162. LNCS. Springer, 2021, pp. 42–59. DOI: 10.1007/978-3-030-95085-9_3.

[SW12] Yu Sasaki and Lei Wang. **Meet-in-the-Middle Technique for Integral Attacks against Feistel Ciphers**. SAC 2012. Vol. 7707. LNCS. Springer, 2012, pp. 234–251. DOI: 10.1007/978-3-642-35999-6_16.

[TA14] Yosuke Todo and Kazumaro Aoki. **FFT Key Recovery for Integral Attack**. CANS 2014. Vol. 8813. LNCS. Springer, 2014, pp. 64–81. DOI: 10.1007/978-3-319-12280-9_5.

Bibliography II

[TB22] Je Sen Teh and Alex Biryukov. **Differential cryptanalysis of WARP**. J. Inf. Secur. Appl. 70 (2022), p. 103316. DOI: 10.1016/j.jisa.2022.103316. URL: https://doi.org/10.1016/j.jisa.2022.103316.