Algebraic Cryptanalysis

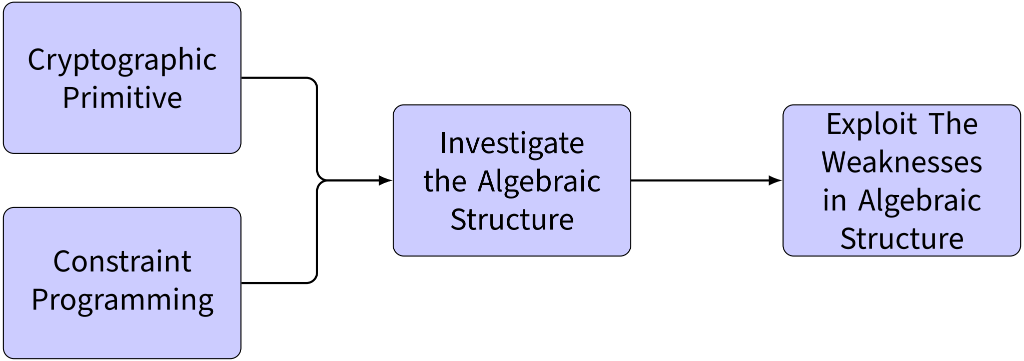
Hosein Hadipour

Cryptanalysis – ST 2024

Outside-in Approach in Algebraic Cryptanalysis

$\int$

Algebraic Cryptanalysis - Outside-in Approach



- Integral attack
- Cube attack on stream ciphers and sponge functions

Symbolic Representation of Boolean Functions

x^2

Notations

- $\mathbb{F}_2$: Field with two elements
- $\mathbb{F}_2^n$: Vector space of dimension n over $\mathbb{F}_2$
- $\boldsymbol{u} = (u_0, \cdots, u_{n-1})$: Bit vectors in $\mathbb{F}_2^2$
- $\boldsymbol{e}_i$: n-bit unit vector with 1 at position i and 0 elsewhere
- **A partial order over** $\mathbb{F}_2^n$: For any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}_2^n$: $\boldsymbol{u} \leq \boldsymbol{v}$ if $u_i \leq v_i$ for all i
- For all $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{F}_2^n$: if $\boldsymbol{x} < \boldsymbol{u}$, then $\boldsymbol{x}^{\boldsymbol{u}} = 0$
- For all $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{F}_2^n$: if $\boldsymbol{x} = \boldsymbol{u}$, then $\boldsymbol{x}^{\boldsymbol{u}} = 1$
- $\mathbb{F}_2[x_0, \cdots, x_{n-1}]$: Ring of polynomials with n variables over $\mathbb{F}_2$
- Monomial: $\boldsymbol{x}^{\boldsymbol{u}} = \pi_{\boldsymbol{u}}(\boldsymbol{x}) = \prod_{i=0}^{n-1} x_i^{u_i}$, e.g., $(x_2, x_1, x_0)^{(1,0,1)} = x_2 x_0$

Different Representations of Boolean Functions

- Truth Table (TT)
- Logical Expression
  - CNF:
    $(x_2 \vee x_1 \vee \neg x_0) \wedge (x_2 \vee \neg x_1 \vee x_0) \wedge$
    $(\neg x_2 \vee \neg x_1 \vee x_0) \wedge (\neg x_2 \vee \neg x_1 \vee \neg x_0)$
  - DNF:
    $(\neg x_2 \wedge \neg x_1 \wedge \neg x_0) \vee (\neg x_2 \wedge x_1 \wedge x_0) \vee$
    $(x_2 \wedge \neg x_1 \wedge \neg x_0) \vee (x_2 \wedge \neg x_1 \wedge x_0)$
- Algebraic Normal Form (ANF)

$$f(x_2, x_1, x_0) = x_0 \cdot x_2 + x_0 + x_1 + 1$$

x_2	x_1	x_0	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Algebraic Normal Form (ANF)

ANF

A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be uniquely represented by a polynomial in $\frac{\mathbb{F}_2[x_0,\ldots,x_{n-1}]}{\langle x_0^2+x_0,\ldots,x_{n-1}^2+x_{n-1}\rangle}$:

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}} \cdot \boldsymbol{x}^{\boldsymbol{u}}, \quad \text{where} \quad a_{\boldsymbol{u}} \in \mathbb{F}_2, \text{ and } \boldsymbol{x}^{\boldsymbol{u}} = x_0^{u_0} \cdots x_{n-1}^{u_{n-1}}.$$

- If $f(\boldsymbol{x}) = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}} \cdot \boldsymbol{x}^{\boldsymbol{u}}$, and $\boldsymbol{x} \leq \boldsymbol{u}$ iff $x_i \leq u_i$ for all i, then for all $\boldsymbol{u} \in \mathbb{F}_2^n$:
  - $a_{\boldsymbol{u}} = \sum_{\boldsymbol{x}\leq\boldsymbol{u}} f(\boldsymbol{x})$
  - $f(\boldsymbol{u}) = \sum_{\boldsymbol{x}\leq\boldsymbol{u}} a_{\boldsymbol{x}}$

Relations Between Coefficients and (Sum of the) Outputs

- If $f(\boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}} \cdot \boldsymbol{x}^{\boldsymbol{u}}$, then for all $\boldsymbol{u} \in \mathbb{F}_2^n$:
  - $a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{x})$
  - $f(\boldsymbol{u}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} a_{\boldsymbol{x}}$

x_1	x_0	$f(x_1, x_0)$
0	0	1
0	1	0
1	0	1
1	1	1

- $a_{(0,0)} = \sum_{u \leq (0,0)} f(x) = 1 \Rightarrow 1 \rightarrow f$
- $a_{(0,1)} = \sum_{u \leq (0,1)} f(x) = 1 \Rightarrow x_0 \rightarrow f$
- $a_{(1,0)} = \sum_{u \leq (1,0)} f(x) = 0 \Rightarrow x_1 \not\rightarrow f$
- $a_{(1,1)} = \sum_{u \leq (1,1)} f(x) = 1 \Rightarrow x_1 x_0 \rightarrow f$

$$f(x_1, x_0) =$$

Relations Between Coefficients and (Sum of the) Outputs

- If $f(\boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}} \cdot \boldsymbol{x}^{\boldsymbol{u}}$, then for all $\boldsymbol{u} \in \mathbb{F}_2^n$:
  - $a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{x})$
  - $f(\boldsymbol{u}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} a_{\boldsymbol{x}}$

x_1	x_0	$f(x_1, x_0)$
0	0	1
0	1	0
1	0	1
1	1	1

- $a_{(0,0)} = \sum_{u \leq (0,0)} f(x) = 1 \Rightarrow 1 \to f$
- $a_{(0,1)} = \sum_{u \leq (0,1)} f(x) = 1 \Rightarrow x_0 \to f$
- $a_{(1,0)} = \sum_{u \leq (1,0)} f(x) = 0 \Rightarrow x_1 \not\to f$
- $a_{(1,1)} = \sum_{u \leq (1,1)} f(x) = 1 \Rightarrow x_1 x_0 \to f$

$$f(x_1, x_0) =$$

Relations Between Coefficients and (Sum of the) Outputs

- If $f(\boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}} \cdot \boldsymbol{x}^{\boldsymbol{u}}$, then for all $\boldsymbol{u} \in \mathbb{F}_2^n$:
  - $a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{x})$
  - $f(\boldsymbol{u}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} a_{\boldsymbol{x}}$

x_1	x_0	$f(x_1, x_0)$
0	0	1
0	1	0
1	0	1
1	1	1

- $a_{(0,0)} = \sum_{u \leq (0,0)} f(x) = 1 \Rightarrow 1 \rightarrow f$
- $a_{(0,1)} = \sum_{u \leq (0,1)} f(x) = 1 \Rightarrow x_0 \rightarrow f$
- $a_{(1,0)} = \sum_{u \leq (1,0)} f(x) = 0 \Rightarrow x_1 \not\rightarrow f$
- $a_{(1,1)} = \sum_{u \leq (1,1)} f(x) = 1 \Rightarrow x_1 x_0 \rightarrow f$

$$f(x_1, x_0) = 1 +$$

Relations Between Coefficients and (Sum of the) Outputs

- If $f(\boldsymbol{x}) = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}} \cdot \boldsymbol{x}^{\boldsymbol{u}}$, then for all $\boldsymbol{u} \in \mathbb{F}_2^n$:
  - $a_{\boldsymbol{u}} = \sum_{\boldsymbol{x}\leq\boldsymbol{u}} f(\boldsymbol{x})$
  - $f(\boldsymbol{u}) = \sum_{\boldsymbol{x}\leq\boldsymbol{u}} a_{\boldsymbol{x}}$

x_1	x_0	$f(x_1, x_0)$
0	0	1
0	1	0
1	0	1
1	1	1

- $a_{(0,0)} = \sum_{u\leq(0,0)} f(x) = 1 \Rightarrow 1 \rightarrow f$
- $a_{(0,1)} = \sum_{u\leq(0,1)} f(x) = 1 \Rightarrow x_0 \rightarrow f$
- $a_{(1,0)} = \sum_{u\leq(1,0)} f(x) = 0 \Rightarrow x_1 \not\rightarrow f$
- $a_{(1,1)} = \sum_{u\leq(1,1)} f(x) = 1 \Rightarrow x_1x_0 \rightarrow f$

$$f(x_1, x_0) = 1 + x_0 +$$

Relations Between Coefficients and (Sum of the) Outputs

- If $f(\boldsymbol{x}) = \sum_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}} \cdot \boldsymbol{x}^{\boldsymbol{u}}$, then for all $\boldsymbol{u} \in \mathbb{F}_2^n$:
  - $a_{\boldsymbol{u}} = \sum_{\boldsymbol{x}\preceq\boldsymbol{u}} f(\boldsymbol{x})$
  - $f(\boldsymbol{u}) = \sum_{\boldsymbol{x}\preceq\boldsymbol{u}} a_{\boldsymbol{x}}$

x_1	x_0	$f(x_1, x_0)$
0	0	1
0	1	0
1	0	1
1	1	1

- $a_{(0,0)} = \sum_{u\preceq(0,0)} f(x) = 1 \Rightarrow 1 \to f$
- $a_{(0,1)} = \sum_{u\preceq(0,1)} f(x) = 1 \Rightarrow x_0 \to f$
- $a_{(1,0)} = \sum_{u\preceq(1,0)} f(x) = 0 \Rightarrow x_1 \not\to f$
- $a_{(1,1)} = \sum_{u\preceq(1,1)} f(x) = 1 \Rightarrow x_1x_0 \to f$

$$f(x_1, x_0) = 1 + x_0 + x_1 +$$

Relations Between Coefficients and (Sum of the) Outputs

- If $f(\boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}} \cdot \boldsymbol{x}^{\boldsymbol{u}}$, then for all $\boldsymbol{u} \in \mathbb{F}_2^n$:
  - $a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{x})$
  - $f(\boldsymbol{u}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} a_{\boldsymbol{x}}$

x_1	x_0	$f(x_1, x_0)$
0	0	1
0	1	0
1	0	1
1	1	1

- $a_{(0,0)} = \sum_{u \leq (0,0)} f(x) = 1 \Rightarrow 1 \rightarrow f$
- $a_{(0,1)} = \sum_{u \leq (0,1)} f(x) = 1 \Rightarrow x_0 \rightarrow f$
- $a_{(1,0)} = \sum_{u \leq (1,0)} f(x) = 0 \Rightarrow x_1 \not\rightarrow f$
- $a_{(1,1)} = \sum_{u \leq (1,1)} f(x) = 1 \Rightarrow x_1 x_0 \rightarrow f$

$$f(x_1, x_0) = 1 + x_0 + x_1 + x_1 x_0$$

Deriving ANF in SageMath

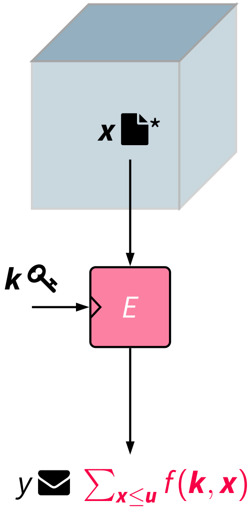
```
sage: from sage.crypto.boolean_function import BooleanFunction as BF
sage: f = BF([0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1])
sage: g = f.algebraic_normal_form(); g
x0*x2*x3 + x0*x2 + x0*x3 + x1*x2*x3 + x1*x2 + x1*x3 + x1
sage: g.variables()
(x0, x1, x2, x3)
sage: g.monomials()
[x0*x2*x3, x0*x2, x0*x3, x1*x2*x3, x1*x2, x1*x3, x1]
sage: g.degree()
3
```

Integral Distinguishers From the Algebraic Perspective

$\sum$

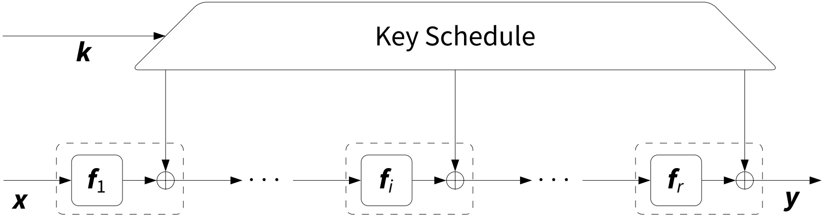
Integral Distinguisher and The Coefficients of ANF

- $y = f(\boldsymbol{k}, \boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} \sum_{\boldsymbol{v} \in \mathbb{F}_2^k} a_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{k}^{\boldsymbol{v}} \boldsymbol{x}^{\boldsymbol{u}} = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}}$
- $\mathbb{C}_{\boldsymbol{u}} = \{x \in \mathbb{F}_2^n \mid \boldsymbol{x} \leq \boldsymbol{u}\}$
- $a_{\boldsymbol{u}}(\boldsymbol{k}) = \sum_{\boldsymbol{x} \leq \boldsymbol{u}} f(\boldsymbol{k}, \boldsymbol{x})$
- 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: $\exists\, u,\ s.t. \forall \boldsymbol{k} : \ a_{\boldsymbol{u}}(\boldsymbol{k}) = 1$



Monomial Prediction and Our SAT Model

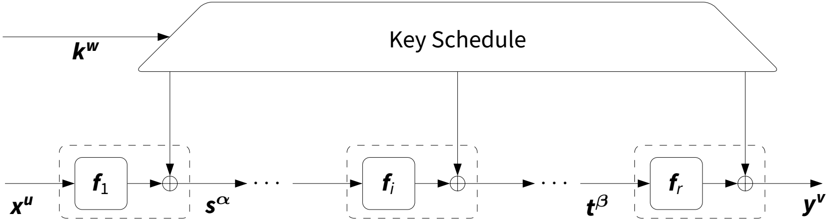
Core Idea of Monomial Prediction [HSWW20]



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.

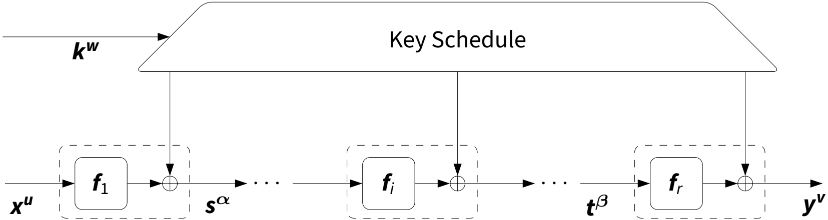
Monomial Trail and Integral Distinguisher



$$k^w x^u \xrightarrow{f_1} s^\alpha \xrightarrow{f_2} \cdots \xrightarrow{f_i} \cdots \xrightarrow{f_{r-1}} t^\beta \xrightarrow{f_r} y^v$$

$$k^w x^u \rightsquigarrow y^v$$

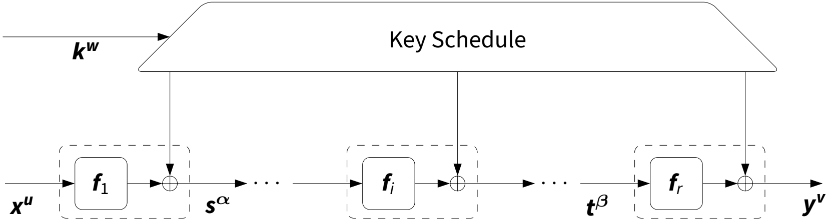
Monomial Trail and Integral Distinguisher



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

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

Monomial Trail and Integral Distinguisher

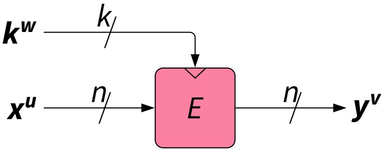


$$y^v = \sum_{u \in \mathbb{F}_2^n} \sum_{v \in \mathbb{F}_2^k} a_{u,v} k^v x^u = \sum_{u \in \mathbb{F}_2^n} a_u(k) \cdot x^u$$

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

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 $a_{\boldsymbol{u}}(\boldsymbol{k}) = \texttt{constant}$

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

$\boldsymbol{u} \backslash \boldsymbol{v}$	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 $\texttt{MPT}(\boldsymbol{u}, \boldsymbol{v}) = 1$ 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

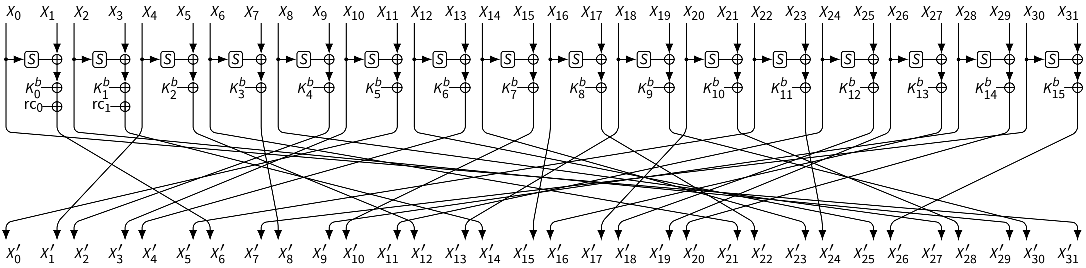
$$
\begin{array}{lll}
(u_2 \vee \neg v_1 \vee \neg v_3) & \wedge (\neg u_1 \vee \neg v_0 \vee \neg v_1 \vee v_2) & \wedge (\neg u_0 \vee \neg u_1 \vee \neg u_2 \vee \neg v_2 \vee v_3) \\
\wedge (u_2 \vee u_3 \vee \neg v_3) & \wedge (\neg u_0 \vee \neg u_1 \vee \neg u_3 \vee v_2) & \wedge (\neg u_0 \vee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

Sbox Analyzer: https://github.com/hadipourh/sboxanalyzer

Application to Integral Analysis of WARP

WARP[BBI+20]

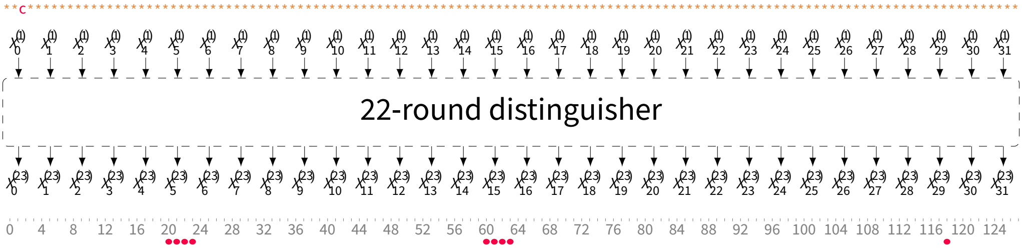
- Proposed in SAC 2020 [BBI+20] as the lightweight alternative of AES-128
- 128-bit block/key size, and 41 rounds (40.5 rounds)
- Splits 128-bit K into two halves $K^{(0)}||K^{(1)}$ and uses $K^{(r-1 \mod 2)}$ in the rth round



22-round Integral Distinguisher for WARP

- The best previous integral distinguisher: 20 rounds [BBI+20]

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



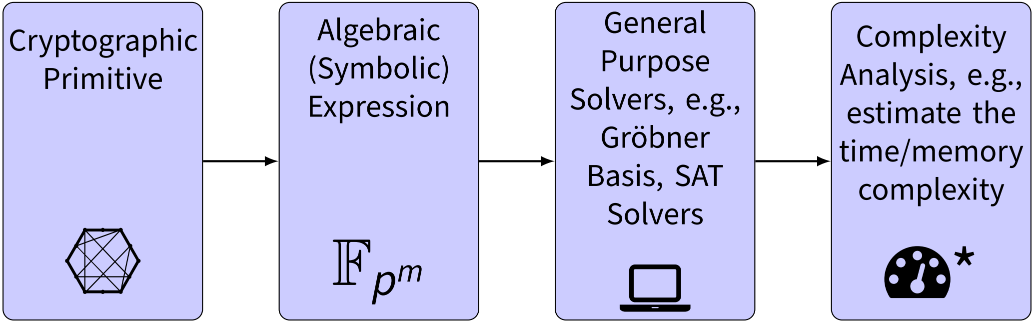
Our tool: https://github.com/hadipourh/mpt

Inside-out Approach for Algebraic Cryptanalysis

$$\iint$$

not covered 2024

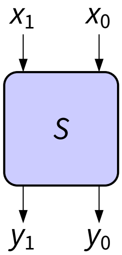
Algebraic Cryptanalysis - Inside-out Approach



- Attacks based on Gröbner Basis (GB)
- Attacks based on SAT solvers

Deriving the IO Relations for a Vectorial Boolean Function

- Consider this 2-bit S-box: $[1, 0, 3, 2]$



$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & x_1 \\ 0 & 1 & 0 & 1 & x_0 \\ 0 & 0 & 1 & 1 & y_1 \\ 1 & 0 & 1 & 0 & y_0 \\ 0 & 0 & 0 & 1 & x_1x_0 \\ 0 & 0 & 1 & 1 & x_1y_1 \\ 0 & 0 & 1 & 0 & x_1y_0 \\ 0 & 0 & 0 & 1 & x_0y_1 \\ 0 & 0 & 0 & 0 & x_0y_0 \\ 0 & 0 & 1 & 0 & y_1y_0 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 & x_1x_0 + x_1 + x_0 + 1 \\ 0 & 1 & 0 & 0 & x_1x_0 + x_0 \\ 0 & 0 & 1 & 0 & x_1x_0 + x_1 \\ 0 & 0 & 0 & 1 & x_1x_0 \\ 0 & 0 & 0 & 0 & x_0 + y_0 + 1 \\ 0 & 0 & 0 & 0 & x_1 + y_1 \\ 0 & 0 & 0 & 0 & x_1y_1 + x_1 \\ 0 & 0 & 0 & 0 & x_1x_0 + x_1y_0 + x_1 \\ 0 & 0 & 0 & 0 & x_1x_0 + x_0y_1 \\ 0 & 0 & 0 & 0 & x_0y_0 \\ 0 & 0 & 0 & 0 & x_1x_0 + x_1 + y_1y_0 \end{pmatrix}$$

- `sage.crypto.sboxes.SBox.polynomials()`

Gröbner Basis

not covered 2024

Ring of Multivariate Polynomials over a Finite Field

- $\mathbb{F}_q$ a finite field of order q
- Ring of multivariate polynomials over $\mathbb{F}_q$: $P = \mathbb{F}_q[x_1, \ldots, x_n]$

$$P = \{\sum_u a_u x^u \mid n \in \mathbb{N}, a_u \in \mathbb{F}_q, u \in \mathbb{Z}^n_{\geq 0}\}$$

- Monomial ordering: decides how we compare monomials

```
sage: P.<x, y, z> = PolynomialRing(GF(127), order='lex')
sage: f = P.random_element(); f
62*x*y - 29*y*z + 35*z^2 - 54*z - 55
sage: x*y > y^3 # variables then degree
True
sage: P.<x, y, z> = PolynomialRing(GF(127), order='deglex')
sage: x*y > y^3 # degree then variables
False
sage: f.monomials(), f.lm()
([x*y, y*z, z^2, z, 1], x*y)
```

Ideals and Varieties

- Ideal: $\mathcal{I} \subset P$, such that for all $f, g \in \mathcal{I}$, and all $h \in P$: $f + gh \in \mathcal{I}$
- $\langle f_1, \dots, f_m \rangle$ is the ideal spanned by $F = \{f_1, \dots, f_m\}$

$$\mathrm{Id}(F) = \left\{ \sum_{i=1}^{n} r_i \cdot f_i \;\middle|\; n \in \mathbb{N}, h_i \in R, f_i \in F \right\}.$$

- Variety of an ideal $\mathcal{I}$: $\mathcal{V}(\mathcal{I}) = \{x \in \mathbb{F}_q^n \mid f(x) = 0 \text{ for all } f \in \mathcal{I}\}$
- Zero-dimensional variety: Contains only finitely many points ($\dim(V) = 0$)

```
sage: P.<x, y, z> = PolynomialRing(GF(127), order='lex')
Defining x, y, z
sage: I = ideal(x*y + z, y^3 + 1, z^2 - 5*x - 1)
sage: (x*y + z) + P.random_element()*(y^3 + 1) in I
True
sage: I.variety()
[{z: 21, y: 108, x: 88}, {z: 6, y: 108, x: 7}]
```

Ideal Membership in $\mathbb{F}_q[x]$

- Assume that $P = \mathbb{F}_q[x]$, and $I = \langle f \rangle$, where $f \in P$. Check if $f(x) \in I$.
- $\forall\, f(x), g(x) \in P\ \exists\, q(x), r(x) \in P$:

$$g(x) = q(x)f(x) + r(x), \text{ where } r = 0, \text{ or } \deg(r(x)) < \deg(f(x))$$

- Divide $g(x)$ by $f(x)$. If $r(x) = 0$, then $f \in I$.

```
sage: P.<x> = PolynomialRing(GF(7))
sage: f, g = P.random_element(), P.random_element(degree=6); f, g
(4*x^2 + 4*x + 5
,3*x^6 + 6*x^5 + 4*x^4 + 4*x^3 + 2*x^2 + 6*x)
sage: I = Ideal([f])
sage: g // f, g%f, g in I
(2*x^3 + 5*x^2 + 2, 0, True)
```

- What if we want to check if $g \in I = \langle f_1, \ldots, f_m \rangle$, where $g, f_1, \ldots, f_m \in P = \mathbb{F}_q[x_1, \ldots, x_n]$?

Ideal Membership in $\mathbb{F}_q[x_1, \ldots, x_n]$

- To check if $f \in \langle f_1, \ldots, f_m \rangle$ we follow the same idea.
- **Division Algorithm**: Represent f in the form:
  - $$f = q_1 f_1 + \cdots + q_m f_m + r, \text{ where } r = 0$$
  - $r = 0$, or no terms of r is divisible by any of $\mathtt{LT}(f_1), \ldots, \mathtt{LT}(f_m)$

```
sage: P.<x, y> = PolynomialRing(GF(7), order='deglex')
sage: f1, f2 = x*y - 1, y^2 - 1
sage: f = x*y^2 - x
sage: f.reduce([f1]).reduce([f2])
-x + y
sage: f.reduce([f2]).reduce([f1])
0
```

- Remainder is not unique!
- We can not decide if $f \in \langle f_1, \ldots, f_m \rangle$ based on any basis.

Gröbner Basis

- Let $I \subseteq P = \mathbb{F}_q[x_1, \ldots, x_n]$ be an ideal
- A Gröbner basis $G = \{g_1, \ldots, g_t\}$ for I is a special basis:
  - $f \,\%\, G$ is unique regardless of the order of elements in G
  - It can solve the ideal membership problem: $f \in I \iff f \,\%\, G = 0$

```
sage: P.<x, y> = PolynomialRing(GF(7), order='deglex')
sage: f1, f2 = x*y - 1, y^2 - 1
sage: I = ideal([f1, f2])
sage: f = x*y^2 - x
sage: g1, g2 = I.groebner_basis()
sage: f.reduce([g1]).reduce([g2]) == f.reduce([g2]).reduce([g1]) == 0
True
sage: f in I
True
```

Gröbner Basis and Solving System of Polynomial Equations

- $F = \{f_1, \ldots, f_n\}$ be a system of polynomial equations in n variables
- F is linear: Gaussian elimination

```
sage: P.<x, y, z> = PolynomialRing(GF(7), order='deglex')
sage: F = Sequence([-3*y + x, -2*x - y - 3*z + 2, x + y + 2*z - 1])
sage: A, v = F.coefficient_matrix()
sage: A.echelonize() # Echelon form
sage: (A*v).T
[x + 2, y + 3, z - 3]
```

- Let us compute the Gröbner basis of F:

```
sage: F.groebner_basis()
[x + 2, y + 3, z - 3]
```

- It is not by accident! Gröbner basis generalizes row echelon form over $\mathbb{F}_q^n$

Gröbner Basis - Generalizing Row Echelon Form I

Reduced Gröbner Basis

The reduced Gröbner basis $G = \{g_1, g_2, \ldots, g_n\}$ (in a specific term order) generating the zero-dimensional ideal I is of the form

$$G = \begin{cases} g_1 & = x_1^d + h_1(x_1), \\ g_2 & = x_2 + h_2(x_1), \\ & \vdots \\ g_n & = x_n + h_n(x_1), \end{cases}$$

where h_i is a polynomial in x_1 of degree at most $d - 1$.

- Note that g_1 is now a univariate equation and we can solve it by factorization!
- Use the result to solve for the other variables

Gröbner Basis - Example

- Find the zeros (variety) of $I = \langle x + y + z, xy + xz + yz, xyz - 1 \rangle \subseteq \mathbb{F}_{127}[x, y, z]$

```
sage: P.<x, y, z> = PolynomialRing(GF(127), order='lex')
sage: I = ideal([x + y + z, x*y + x*z + y*z, x*y*z - 1])
sage: G = I.groebner_basis()
sage: for f in G: print(f)
x + y + z
y^2 + y*z + z^2
z^3 - 1
sage: V = I.variety(); V
[{z: 19, y: 1, x: 107},
  {z: 19, y: 107, x: 1},
  {z: 1, y: 19, x: 107},
  {z: 1, y: 107, x: 19},
  {z: 107, y: 19, x: 1},
  {z: 107, y: 1, x: 19}]
```

Gröbner Basis - Systems with 0 or 1 Solution

- If the system has 1 solution:

$$G = \begin{cases} g_1 & = x_1 - a_1, \\ g_2 & = x_2 - a_2, \\ & \vdots \\ g_n & = x_n - a_n, \end{cases}$$

  where $(a_1, \ldots, a_n) \in \mathbb{F}_q^n$
- If the system has no solution

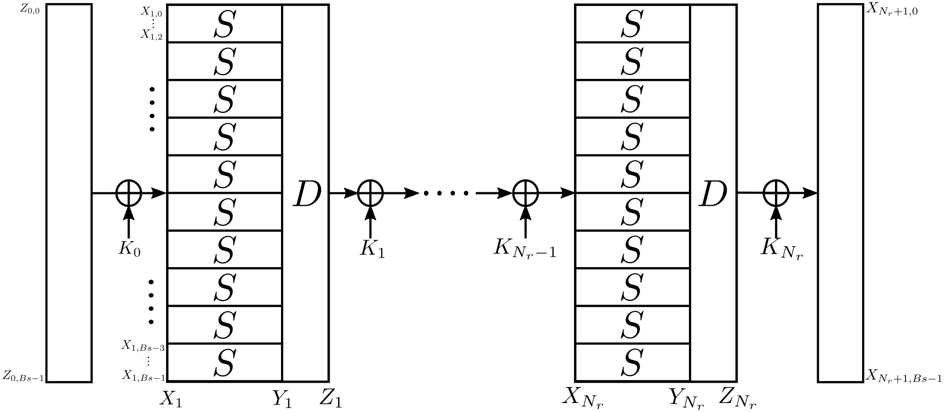
$$G = \langle 1 \rangle$$

Gröbner Basis - Complexity

- A fundamental tool in computational algebraic geometry (1965) [Buc65]
- It solves the ideal membership problem and multivariate polynomial equations, and . . .
- General algorithms, for any input system:
  - Buchberger, F4, F5: They always terminate and give the Gröbner basis
  - But, the time is hard to predict for any instances
- A hard problem:
  - Ideal membership problem is an EXPSPACE-Complete problem
  - Existence of solution to a system of polynomial equations over a finite field is NP-Complete [FY79]

Algebraic Cryptanalysis of CTC Block Cipher

- CTC Block Cipher [Cou06; Cou07]



https://github.com/hadipourh/CTC2-Fast-Algebraic-Attack

Algebraic Cryptanalysis – Summary

- It is originally a deterministic Attack
- Exploit the algebraic representation of cryptographic primitives
- There are many approaches, we discussed only two:
  - Investigating the algebraic representation to find some weaknesses
  - Expressing the cipher as a system of polynomial equations and solving it
- We can take advantage of general-purpose solvers, e.g., Gröbner basis, SAT, etc.
- Work in many different fields ($\mathbb{F}_2, \mathbb{F}_p, \mathbb{F}_{2^n}$)

Questions

1. What are some differences between statistical and algebraic attacks?
2. What is the core idea of the monomial prediction technique?
3. How does the monomial prediction technique help us find an integral distinguisher?
4. If certain variables corresponding to the secret key are absent in the ANF (Algebraic Normal Form) of the ciphertext bit, how can you exploit this to find a zero-sum distinguisher or use a cube attack?
5. Why can Gröbner basis be used to solve systems of polynomial equations?

Bibliography I

[BBI+20] Subhadeep Banik, Zhenzhen Bao, Takanori Isobe, Hiroyasu Kubo, Fukang Liu, Kazuhiko Minematsu, Kosei Sakamoto, Nao Shibata, and Maki Shigeri. **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.

[Buc65] Bruno Buchberger. **Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal**. PhD thesis. University of Innsbruck, 1965.

[Cou06] Nicolas T. Courtois. **How Fast can be Algebraic Attacks on Block Ciphers ?** Cryptology ePrint Archive, Paper 2006/168. 2006. URL: https://eprint.iacr.org/2006/168.

[Cou07] Nicolas T. Courtois. **CTC2 and Fast Algebraic Attacks on Block Ciphers Revisited**. Cryptology ePrint Archive, Paper 2007/152. 2007. URL: https://eprint.iacr.org/2007/152.

Bibliography II

[FY79] Aviezri S Fraenkel and Yaacov Yesha. **Complexity of problems in games, graphs and algebraic equations.** Discrete Applied Mathematics 1.1-2 (1979), pp. 15–30.

[HSWW20] Kai Hu, Siwei Sun, Meiqin Wang, and Qingju Wang. **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.

Convert ANF to CNF

- By ANF-to-CNF conversion, we can use SAT solvers to solve (Boolean) polynomial equations:

```
sage: B = BooleanPolynomialRing(names=["a", "b", "c"]); B.inject_variables()
Defining a, b, c
sage: from sage.sat.converters.polybori import CNFEncoder
sage: from sage.sat.solvers.dimacs import DIMACS
sage: fn = tmp_filename(); solver = DIMACS(filename=fn)
sage: e = CNFEncoder(solver, B)
sage: e([a*b + a + 1, a*c + b])
[None, a, b, c]
sage: _ = solver.write()
sage: print(open(fn).read())
p cnf 3 5
-2 0
1 0
-3 -1 2 0
-2 1 0
-2 3 0
```

- A dedicated open-source tool: Bosphorus