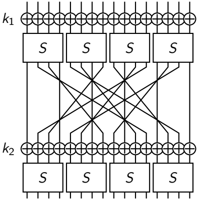
Applications of Mixed-Integer Linear Programming in Symmetric-key Cryptography



Aleksei Udovenko

SnT, University of Luxembourg

securityandtrust.lu

UNIVERSITÉ DU LUXEMBOURG

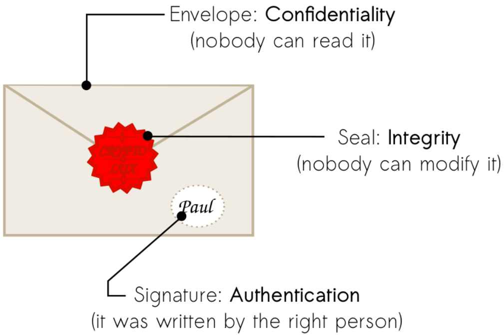
Separating Integer Points in Polyhedra: Theory and Applications,
1st July 2022

Luxembourg National Research Fund

Plan

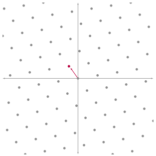
Cryptography

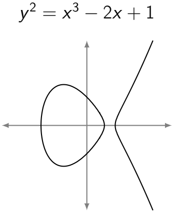
Providing secure communication in the presence of adversaries

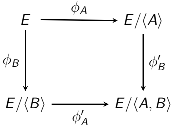


Public-key (Asymmetric) Cryptography

- different keys for encryption and decryption (public and private)
- basic use-cases: key exchange / key encapsulation, digital signatures
- advanced use-cases: FHE, ZK-proofs, MPC, iO, ...
- examples: RSA, (EC)DSA, Elliptic Curve Cryptography
- post-quantum: lattice-based, code-based, ...

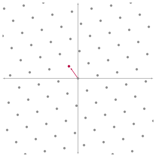


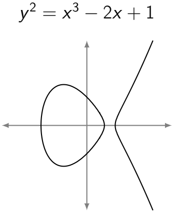


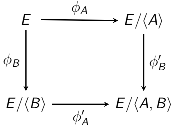


Public-key (Asymmetric) Cryptography

- different keys for encryption and decryption (public and private)
- basic use-cases: key exchange / key encapsulation, digital signatures
- advanced use-cases: FHE, ZK-proofs, MPC, iO, ...
- examples: RSA, (EC)DSA, Elliptic Curve Cryptography
- post-quantum: lattice-based, code-based, ...





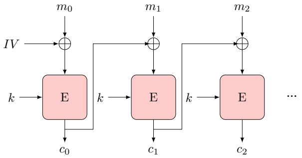


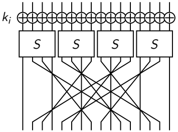
Security guarantees

- *provable* reductions to natural hard problems (usually)
- factoring, discrete logarithm, shortest lattice vector, ...

Secret-key (Symmetric-key) Cryptography

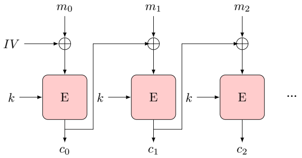
- requires pre-shared secret key (between both parties)
- main goal: authenticated encryption (confidentiality + integrity + authenticity)
- high-level constructions: *provably secure* based on low-level primitives
- low-level constructions: ad-hoc mixture of bitwise/arithmetic operations, lookup tables

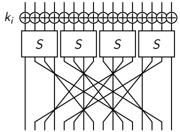




Secret-key (Symmetric-key) Cryptography

- requires pre-shared secret key (between both parties)
- main goal: authenticated encryption (confidentiality + integrity + authenticity)
- high-level constructions: *provably secure* based on low-level primitives
- low-level constructions: ad-hoc mixture of bitwise/arithmetic operations, lookup tables





Security guarantees (low-level)

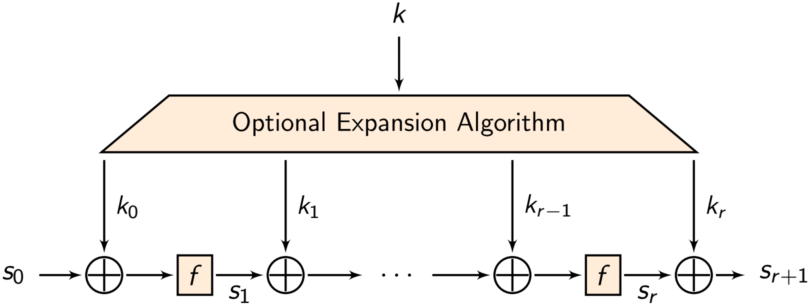
- cryptanalysis: researchers trying their best in breaking the security properties (faster than generic attacks)
- proving security against main attacks (linear/differential, integral)

Block Ciphers and Iterated Construction

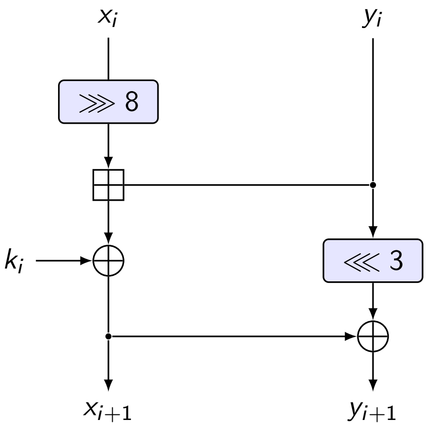
- block cipher : $\mathbb{F}_2^n \times \mathbb{F}_2^k \to \mathbb{F}_2^n$: (plaintext, key) $\mapsto$ ciphertext
- most cases: $n = 64, 128$ bits

Block Ciphers and Iterated Construction

- block cipher : $\mathbb{F}_2^n \times \mathbb{F}_2^k \to \mathbb{F}_2^n$: (plaintext, key) $\mapsto$ ciphertext
- most cases: $n = 64, 128$ bits
- (!) all designs are based on iterating a simple function f
- $\bigoplus$ stands for the XOR operation (addition in $\mathbb{F}_2^n$)
- **properties**: $a \oplus a = 0$ for all $a \in \mathbb{F}_2^n$, subtraction = addition

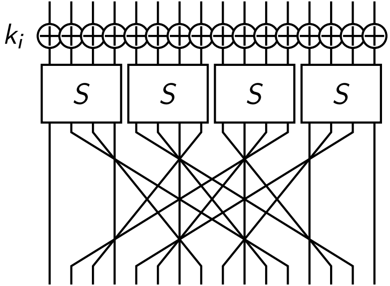


Example: ARX-based Block Cipher “Speck”



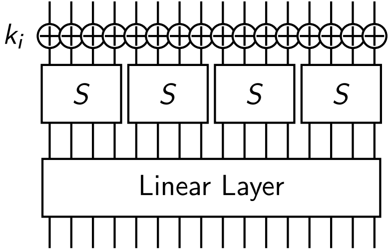
- $2w$-bit block size (32 to 128 bits)
- ARX design: Addition-Rotation-Xor
- addition modulo 2^w
- 22-34 rounds (depending on w and k)

Example: SPN Structure for Block Ciphers



- **S**ubstitution-**P**ermutation[1] **N**etwork
- 64/128-bit block size (block ciphers)
- 192-1600-bit state size (permutations)
- S-boxes : arbitrary small bijections, operating on 4-8 bits
- 6-60 rounds (depends on S-box, linear layer)

Example: SPN Structure for Block Ciphers



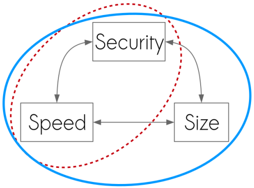
- **S**ubstitution-**P**ermutation[1] **N**etwork
- 64/128-bit block size (block ciphers)
- 192-1600-bit state size (permutations)
- S-boxes : arbitrary small bijections, operating on 4-8 bits
- 6-60 rounds (depends on S-box, linear layer)

[1] Historical term; "permutation" now is more generally a linear map

Security-Performance Trade-off

1. cryptographers **know** how to design secure block ciphers
2. **example**: AES designed in 1998, only 7/10 rounds broken[1] as of 2022
3. state-of-the-art: security-*performance* trade-off
4. **I**nternet **o**f **T**hings: need for *lightweight cryptography*

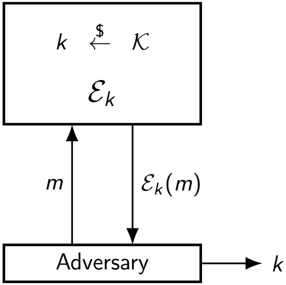
aggressive performance improvements require tedious cryptanalysis



[1] With a significant improvement over exhaustive search

What is Cryptanalysis?

- searching for flaws/weaknesses
- adversary can query the encryption/decryption oracles
- goal:
  - distinguish from a random permutation, or
  - recover the secret key



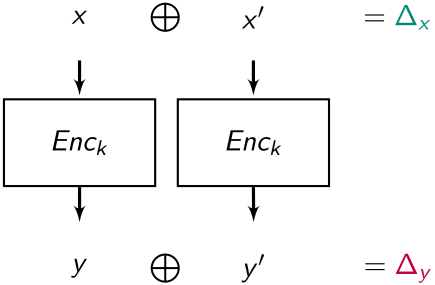
- attack as many rounds as possible
- the remaining rounds are called the *security margin*

Example: AES has 7/10 rounds attacked $\Rightarrow$ 30% security margin

Plan

1. Introduction - Cryptography
2. Differential Cryptanalysis
3. MILP for Differential/Linear Cryptanalysis
4. Division Property (Excerpts)
5. Discussion and Open Problems

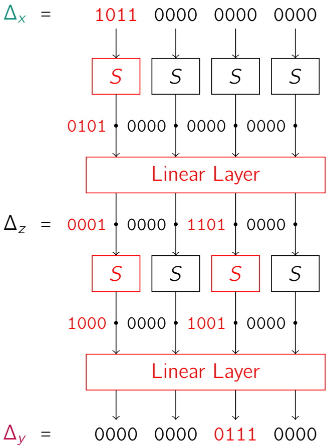
Differential Cryptanalysis - Idea (Biham and Shamir 1991)



- Biham and Shamir (1980s), IBM/NSA (1970s)
- **idea:** study propagation of differences (in $\mathbb{F}_2^n$)
- statistical bias = distinguisher, e.g.

$$\Pr_x[\Delta_x \xrightarrow{\mathrm{Enc}_k} \Delta_y] \gg 2^{-n}$$

Differential Cryptanalysis - Trails



How to find such biases?

- *propagate* differences through the structure of the cipher
- *deterministic* propagation through $\oplus k$:

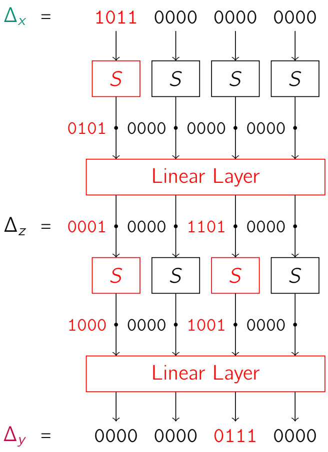
$$(x \oplus k) \oplus (x' \oplus k) = x \oplus x'$$

- *deterministic* propagation through linear maps:

$$L(x) \oplus L(x \oplus \Delta_x) = L(\Delta_x)$$

- active S-boxes are the only source of *nondeterminism*

Differential Cryptanalysis - Trails



How to find such biases?

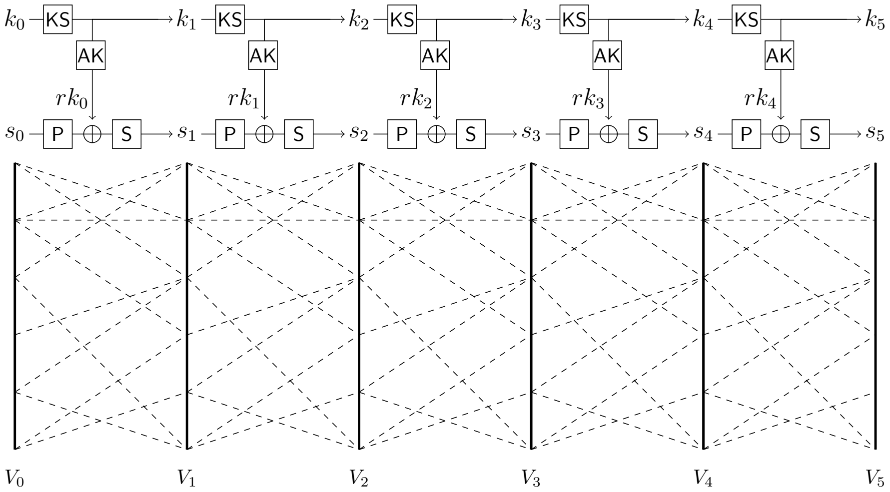
- differential trail: a sequence of state differences, e.g.

$$\Delta_x \to \Delta_z \to \Delta_y$$

- trail probability:

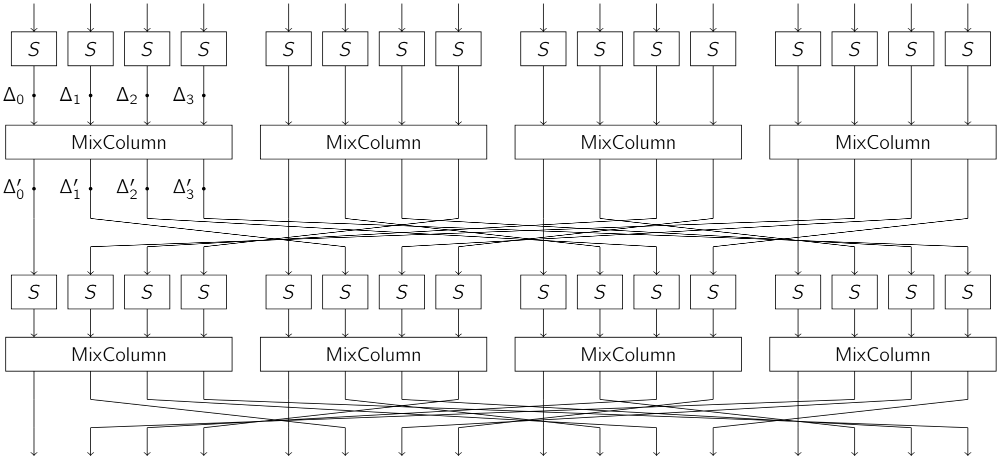
$$\begin{aligned} &\Pr[\Delta_x \xrightarrow{\text{1 round}} \Delta_z \xrightarrow{\text{1 round}} \Delta_y] \\ = &\Pr[\Delta_x \xrightarrow{\text{1 round}} \Delta_z] \cdot \Pr[\Delta_z \xrightarrow{\text{1 round}} \Delta_y] \\ = &\Pr[1011 \xrightarrow{S} 0101] \cdot \Pr[0001 \xrightarrow{S} 1000] \cdot \Pr[1101 \xrightarrow{S} 0111] \\ \leq &\Pr[\Delta_x \xrightarrow{\text{Enc}_k} \Delta_y] \end{aligned}$$

Differential Trails - Graph-based Viewpoint

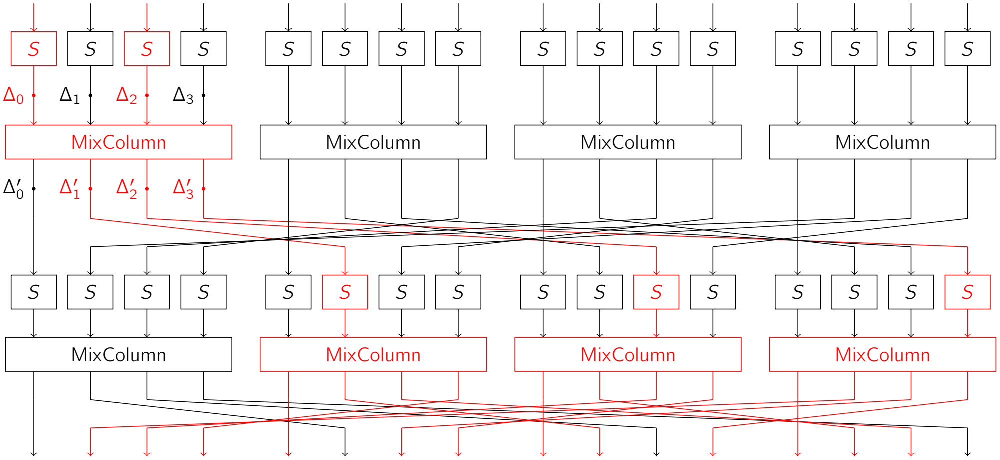


Plan

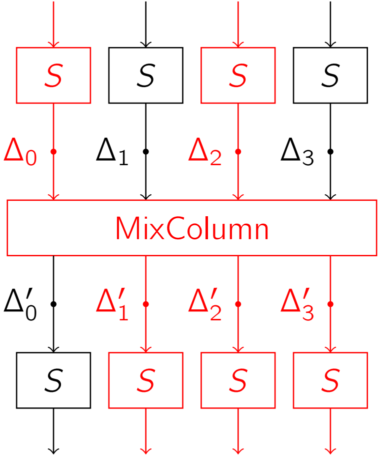
AES: Resistance to Differential Cryptanalysis (Daemen and Rijmen 2002)



AES: Resistance to Differential Cryptanalysis (Daemen and Rijmen 2002)



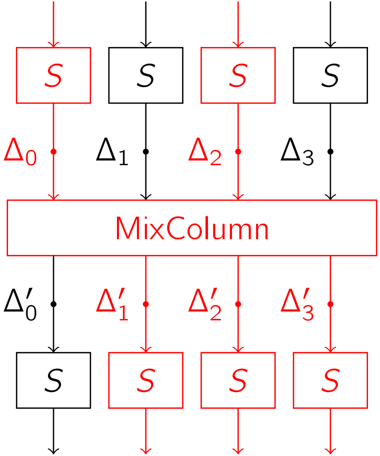
AES: Resistance to Differential Cryptanalysis



MDS property

- either all differences are 0 (inactive)
- or at least are 5 non-zero (out of 8)

AES: Resistance to Differential Cryptanalysis



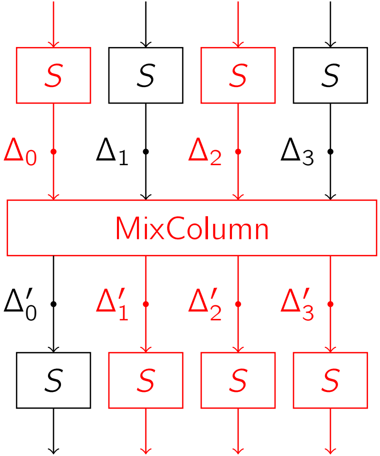
MDS property

- either all differences are 0 (inactive)
- or at least are 5 non-zero (out of 8)

Pen-and-paper resistance bound

- let $\delta_S = \max_{a \neq 0, b \neq 0} \Pr[a \xrightarrow{S} b] = 2^{-6}$
- every 2 rounds : ≥ 5 active S-boxes
- $\Rightarrow$ prob. of *any* 10-round trail $\boxed{\leq (\delta_S)^{5 \cdot 5} = 2^{-150}}$

AES: Resistance to Differential Cryptanalysis



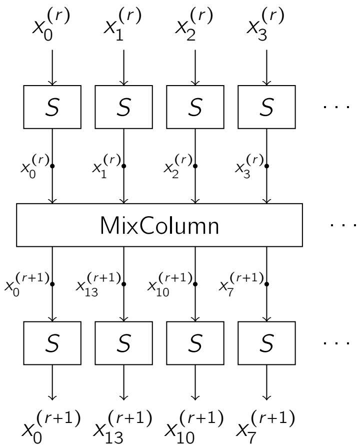
MDS property

- either all differences are 0 (inactive)
- or at least are 5 non-zero (out of 8)

Pen-and-paper resistance bound

- let $\delta_S = \max_{a \neq 0, b \neq 0} \Pr[a \xrightarrow{S} b] = 2^{-6}$
- every 2 rounds : ≥ 5 active S-boxes
- $\Rightarrow$ prob. of *any* 10-round trail $\boxed{\leq (\delta_S)^{5 \cdot 5} = 2^{-150}}$
- better: every 4 rounds : ≥ 25 active S-boxes
- $\Rightarrow$ prob. of *any* 4-round trail $\boxed{\leq 2^{-150}}$

MILP Model for Counting Active S-boxes (Mouha, Wang, Gu, and Preneel 2012)



variables:

$$x_i^{(r)} \in \{0,1\}, \quad 0 \leq i < 16, \quad 0 \leq r < 10 \quad \text{(S-box active?)}$$

$$L_j^{(r)} \in \{0,1\}, \quad 0 \leq j < 4, \quad 0 \leq r < 9 \quad \text{(MixColumn active?)}$$

MixColumn (ex. for the 1st MixColumn 1st round):

$$x_0^{(0)} + x_1^{(0)} + x_2^{(0)} + x_3^{(0)} + x_0^{(1)} + x_{13}^{(1)} + x_{10}^{(1)} + x_7^{(1)} \geq 5L_0^{(0)},$$

$$x_0^{(0)} \leq L_0^{(0)},$$

$$\ldots$$

$$x_7^{(1)} \leq L_0^{(0)}.$$

at least one active: $\sum_i x_i^{(0)} \geq 1$

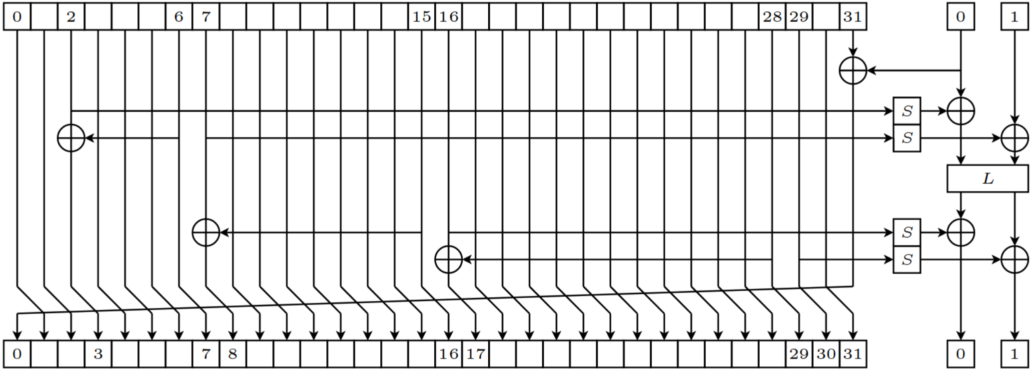
objective: minimize $\sum_{i,r} x_i^{(r)}$

MILP Model for Counting Active S-boxes (Mouha, Wang, Gu, and Preneel 2012)

Results (AES):

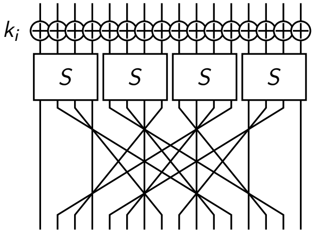
r	1	2	3	4	5	6	7	8	9	10	11	12	13	14
bound	1	5	9	25	26	30	34	50	51	55	59	75	76	80

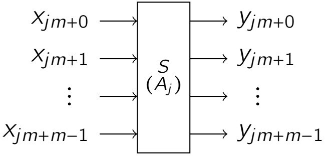
MILP Model for Counting Active S-boxes (Mouha, Wang, Gu, and Preneel 2012)



Analysis up to all 96 rounds

MILP Models for Bit-Permutation Ciphers (Sun, Hu, Song, Xie, and Wang 2014b)





variables:

$$x_i^{(r)} \in \{0,1\}, \quad 0 \le i < n, \quad 0 \le r < R \quad \text{(bit active?)}$$
$$A_j^{(r)} \in \{0,1\}, \quad 0 \le j < n/m, \quad 0 \le r < R \quad \text{(S-box active?)}$$

S-box activity:

$$x_{mj+0} + \ldots + x_{mj+m-1} + y_{mj+0} + \ldots + y_{mj+m-1} \ge \mathrm{Br}_S \cdot A_j$$
$$x_{mj+0} \le A_j, \quad \ldots, \quad y_{mj+m-1} \le A_j$$

S-box input-output activity:

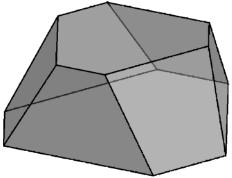
$$x_{mj+0} + \ldots + x_{mj+m-1} \le m \cdot (y_{mj+0} + \ldots + y_{mj+m-1})$$
$$y_{mj+0} + \ldots + y_{mj+m-1} \le m \cdot (x_{mj+0} + \ldots + x_{mj+m-1})$$

objective: minimize $\sum_{j,r} A_j^{(r)}$

Searching for Correct Trails (Sun, Hu, Wang, Qiao, Ma, and Song 2014a)

- goal: model valid differential transitions through an S-box precisely:

$$D_S = \{(\Delta_x, \Delta_y) : \Delta_x \xrightarrow{S} \Delta_y\} \subseteq \{0,1\}^{2n}$$

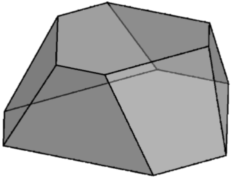


- idea: compute the convex hull of D (over $\mathbb{R}$, H-repr.)
- indeed, any convex combination of a subset of $\{0,1\}^{2n}$ can not equal to a new integral point

Searching for Correct Trails (Sun, Hu, Wang, Qiao, Ma, and Song 2014a)

- goal: model valid differential transitions through an S-box precisely:

$$D_S = \{(\Delta_x, \Delta_y) : \Delta_x \xrightarrow{S} \Delta_y\} \subseteq \{0,1\}^{2n}$$

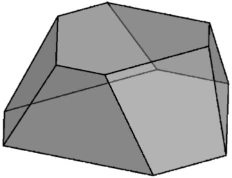


- idea: compute the convex hull of D (over $\mathbb{R}$, H-repr.)
- indeed, any convex combination of a subset of $\{0,1\}^{2n}$ can not equal to a new integral point
- simplification: for binary variables, many inequalities are redundant
  $\Rightarrow$ select t inequalities **greedily**, maximizing the number of remaining "bad" points each next inequality removes (hundreds $\rightarrow$ tens inequalities)

Searching for Correct Trails (Sun, Hu, Wang, Qiao, Ma, and Song 2014a)

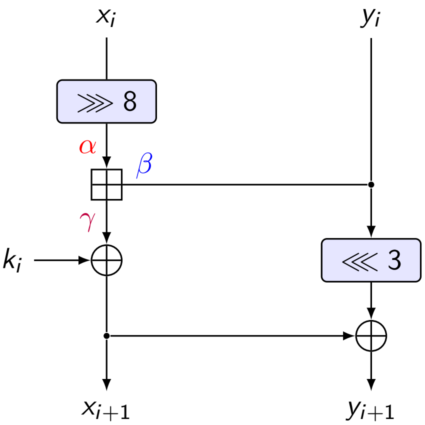
- goal: model valid differential transitions through an S-box precisely:

$$D_S = \{(\Delta_x, \Delta_y) : \Delta_x \xrightarrow{S} \Delta_y\} \subseteq \{0,1\}^{2n}$$



- idea: compute the convex hull of D (over $\mathbb{R}$, H-repr.)
- indeed, any convex combination of a subset of $\{0,1\}^{2n}$ can not equal to a new integral point
- simplification: for binary variables, many inequalities are redundant
  $\Rightarrow$ select t inequalities **greedily**, maximizing the number of remaining "bad" points each next inequality removes (hundreds $\rightarrow$ tens inequalities)
- objective is still to minimize the number of active S-boxes

Optimizing Differential Probability - ARX (Fu, Wang, Guo, Sun, and Hu 2016)



- diff. transition $(\alpha, \beta) \xrightarrow{\boxplus} \gamma$
- condition (Lipmaa and Moriai 2002):

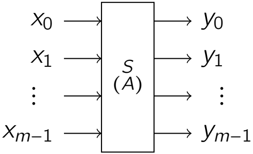
$$\alpha_0 \oplus \beta_0 \oplus \gamma_0 = 0$$
$$(\alpha_{i-1} = \beta_{i-1} = \gamma_{i-1})$$
$$\Rightarrow \gamma_{i-1} = \alpha_i \oplus \beta_i \oplus \gamma_i \quad (1 \le i \le n-1)$$

- ConvexHull+Greedy : 13 inequalities / bit
- diff. prob. is $2^{-\ell}$, where

$$\ell = |\{i \in [0, n-2] : \overbrace{\alpha_i = \beta_i = \gamma_i}^{d_i}\}| = \sum_{i=0}^{n-2} d_i$$

- **objective**: minimize $\sum_r \ell^{(r)} = \sum_r \sum_{i=0}^{n-2} d_i^{(r)}$

Optimizing Differential Probability - SPN (Sun, Hu, Wang, Wang, Qiao, Ma, Shi, Song, and Fu 2014c; Abdelkhalek, Sasaki, Todo, Tolba, and Youssef 2017)

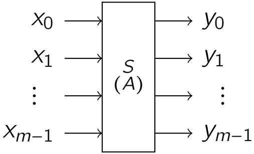


- often (e.g., small S-boxes), few distinct prob. values $p_1, \ldots, p_k$
- **example**: 8-bit SKINNY128 S-box S:

$$\Pr[\Delta_x \xrightarrow{S} \Delta_y] \in \{0, 2^{-2}, 2^{-2.4}, 2^{-2.7}, 2^{-3}, 2^{-3.2}, 2^{-3.4}, 2^{-3.7}, 2^{-4}, 2^{-4.4}, 2^{-5}, 2^{-5.4}, 2^{-6}, 2^{-7}\}$$

(14 distinct values)

Optimizing Differential Probability - SPN (Sun, Hu, Wang, Wang, Qiao, Ma, Shi, Song, and Fu 2014c; Abdelkhalek, Sasaki, Todo, Tolba, and Youssef 2017)



- often (e.g., small S-boxes), few distinct prob. values $p_1, \ldots, p_k$
- introduce variables $v_1, \ldots, v_k \in \{0,1\}$, $\sum_i v_i = A$
- model each set

$$V_j = \{(\Delta_x, \Delta_y) : \Pr[\Delta_x \xrightarrow{S} \Delta_y] = p_j\} \subseteq \mathbb{F}_2^{2n}$$

  on variables $(x_0, \ldots, x_{m-1}, y_0, \ldots, y_{m-1})$

- separate sets by using big-M constraints (M suff. large): (set V_j)

$$\sum_{i=0}^{m-1} c_i x_i + \sum_{i=0}^{m-1} c_i' y_i \geq c \quad \rightsquigarrow \quad \sum_{i=0}^{m-1} c_i x_i + \sum_{i=0}^{m-1} c_i' y_i + M \cdot (1 - v_j) \geq c$$

- objective: maximize $v_1 \log p_1 + \ldots + v_k \log p_k$

Logical Condition Modeling - SPN (Abdelkhalek, Sasaki, Todo, Tolba, and Youssef 2017)

Modeling a subset of $\{0,1\}^n$ in MILP (binary variables)

Procedure

1. use Quine-McCluskey or Espresso algorithms to find minimal/small CNF formula for the modeled set $V \subseteq \{0,1\}^n$, e.g.:

$$x \in V \quad \Leftrightarrow \quad (x_0 \vee \neg x_3 \vee \neg x_7) \wedge \ldots \wedge (\neg x_1 \vee x_3 \vee x_4 \vee x_5)$$

2. convert each *clause*

$$(x_{i_0} \vee \ldots \vee x_{i_k} \vee \neg x_{j_0} \vee \ldots \vee \neg x_{j_\ell})$$

into the equivalent *inequality*

$$x_{i_0} + \ldots + x_{i_k} + (1 - x_{j_0}) + \ldots + (1 - x_{j_\ell}) \geq 1$$

Modeling a subset of $\{0,1\}^n$ in MILP (binary variables)

- (Sun, Hu, Wang, Qiao, Ma, and Song 2014a)
  Convex hull + greedy reduction
- (Abdelkhalek, Sasaki, Todo, Tolba, and Youssef 2017)
  Logical condition modeling (CNF minimization)
- (Sasaki and Todo 2017)
  Replace "greedy reduction" by SetCover Minimization (MILP-based)
- (Boura and Coggia 2020)
  "Distorted-Ball" inequalities + SetCover Minimization
- (Udovenko 2021b)
  Monotone Learning + SetCover Minimization
- (Sun 2021)
  MIP-based method + SetCover Minimization
- (Derbez and Lambin 2022)
  CNF to MILP (lossless compression)
- (Averkov, Hojny, and Schymura 2021; Averkov, Hojny, and Schymura 2022)
  Efficient MIP Techniques (see later a talk by Christopher Hojny)

Plan

Division Trails - Specifics

- No "probability" to optimize (UNSAT)

Conventional Division Property

- trail found $\Rightarrow$??? (imprecision)
- trail can not exist $\Rightarrow$ attack
- monotonicity (active bits number is non-increasing) $\Rightarrow$ less uncertainty
- more often feasible

Perfect Division Property

- need to compute *parity* of the number of trails
- less often feasible

Modeling Linear Maps - Criteria (Zhang and Rijmen 2019)

Proposition

Consider the linear map defined by a matrix $L \in \mathbb{F}_2^{n\times n}$. Then,

$$u \xrightarrow{L} v$$

if and only if the submatrix defined by 1s from u and v is invertible.

Example:

$$\begin{array}{c} \\ v_1{=}0 \\ v_2{=}1 \\ v_3{=}1 \end{array} \begin{array}{c} \begin{array}{ccc} u_1{=}0 & u_2{=}1 & u_3{=}1 \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{array}\right] \end{array} \quad \Leftrightarrow \quad 011 \xrightarrow{L} 011$$

Modeling Linear Maps - Imprecise (Sun, Wang, and Wang 2016)

If a matrix $L \in \mathbb{F}_2^{n \times n}$ is invertible, then there exists a permutation σ such that

$$\prod_{i=1}^{n} L_{i,\sigma(i)} = 1$$

$u \xrightarrow{L} v$ implies a matching between 1s in u and 1s in v, with the edges given by 1s in L

$$\begin{aligned}
\text{variables:} \quad & t_{i,j} (\text{edges for } L_{i,j} = 1), u_i, v_i \in \{0, 1\} && i, j \in [1, n] \\
\text{equations:} \quad & u_i = \sum_{j=1}^{n} L_{i,j} t_{i,j} && i \in [1, n], L_{i,j} = 1 \\
& v_j = \sum_{i=1}^{n} L_{i,j} t_{i,j} && j \in [1, n], L_{i,j} = 1
\end{aligned}$$

Modeling Linear Maps - Precise Methods

- (SMT-based) (Hu, Wang, and Wang 2020):
- introduce variables for the inverse matrix $L_{u,v}^{-1}$
- add matrix multiplication constraint $L_{u,v} \times L_{u,v}^{-1} = Id$

Modeling Linear Maps - Precise Methods

- (SMT-based) (Hu, Wang, and Wang 2020):
- introduce variables for the inverse matrix $L_{u,v}^{-1}$
- add matrix multiplication constraint $L_{u,v} \times L_{u,v}^{-1} = Id$

- (MILP-based) (Hong, Zhang, Chen, Lin, and Xiang 2021):
- combine different implementations of the linear layer
- correct transition must satisfy all of them

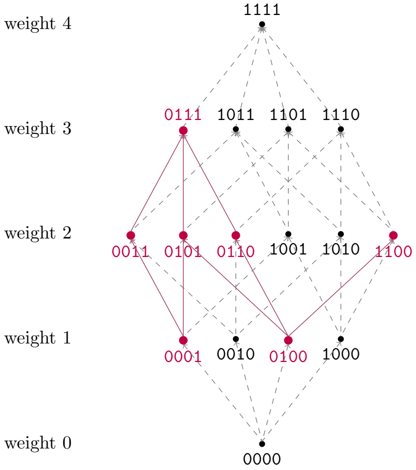
Modeling Linear Maps - Precise Methods

- (SMT-based) (Hu, Wang, and Wang 2020):
- introduce variables for the inverse matrix $L_{u,v}^{-1}$
- add matrix multiplication constraint $L_{u,v} \times L_{u,v}^{-1} = Id$

- (MILP-based) (Hong, Zhang, Chen, Lin, and Xiang 2021):
- combine different implementations of the linear layer
- correct transition must satisfy all of them

- (MILP-based) (ElSheikh and Youssef 2021; Derbez and Lambin 2022):
- implement basic (lossy) constraints
- use **callbacks** with **lazy constraints** to discard wrong solutions on-the-fly

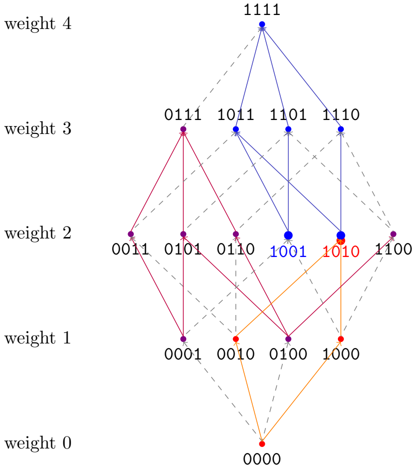
Modeling Convex Sets



modeling an convex set $X \subseteq \{0,1\}^n$:

- **partial order** on $\{0,1\}^n$:
  $u \preceq v \;\text{ iff }\; \forall i \;\; u_i \leq v_i$

Modeling Convex Sets



modeling an convex set $X \subseteq \{0,1\}^n$:

- **partial order** on $\{0,1\}^n$:
  $u \preceq v$ iff $\forall i \quad u_i \leq v_i$
- combined CNF
  of the upper/lower bounds:

$$\underbrace{(\neg x_0 \vee \neg x_3)}_{1001} \wedge \underbrace{(\neg x_0 \vee \neg x_2)}_{1010} \wedge \underbrace{(x_1 \vee x_3)}_{1010}$$

Plan

Optimizations

- drop integrality constraints on some variables (implied by integrality of other variables)
- divide-and-conquer / multiple-stage approaches:
  - bounds on every $k < r$ rounds e.g. on the number of active S-boxes / bits
  - lazy evaluation of complex constraints (via callbacks)
  - finding approximate solutions and specifying them afterwards
- using some optimizer features:
  - piecewise-linear functions
  - callbacks / lazy constraints

Performance/Feasibility

- experimental method (trial and error)
- many impressive results but no clear performance understanding
- trend: minimizing *number* of inequalities
- (Sasaki and Todo 2017) noticed that $\neq$ performance improving

Open Problems

1. performance *criteria* for models
2. good models for *concrete* Boolean sets $\subseteq \{0,1\}^n$ (up to $n = 32$)
3. good models for square submatrix invertability (rows/columns identified by binary variables)
4. fine-tuning solvers and their features
5. more applications (more cryptanalysis techniques)

affine.group/slides/2022_MILPinSymCrypto.pdf

Acknowledgments

- The author thanks Gennadiy Averkov, Christopher Hojny, and Matthias Schymura for the invitation to the workshop.
- The author thanks Qingju Wang and Baptiste Lambin for helpful discussions and references.
- Most figures are taken from the "TikZ for Cryptographers" collection (Jean 2016).
- The envelope figure is due to Léo Perrin.
- Thanks to CryptoBib for the relevant BibTeX database.

Abdelkhalek, Ahmed, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef (2017). "MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics". In: *IACR Trans. Symm. Cryptol.* 2017.4, pp. 99–129. ISSN: 2519-173X. DOI: 10.13154/tosc.v2017.i4.99-129.

Averkov, Gennadiy, Christopher Hojny, and Matthias Schymura (Dec. 2021). "Computational aspects of relaxation complexity: possibilities and limitations". English. In: *Mathematical Programming*. ISSN: 0025-5610. DOI: 10.1007/s10107-021-01754-8.

– (2022). *Efficient MIP Techniques for Computing the Relaxation Complexity*. DOI: 10.48550/ARXIV.2203.05224. URL: https://arxiv.org/abs/2203.05224.

Biham, Eli and Adi Shamir (Jan. 1991). "Differential Cryptanalysis of DES-like Cryptosystems". In: *Journal of Cryptology* 4.1, pp. 3–72. DOI: 10.1007/BF00630563.

Boura, Christina and Daniel Coggia (2020). "Efficient MILP Modelings for Sboxes and Linear Layers of SPN ciphers". In: *IACR Trans. Symm. Cryptol.* 2020.3, pp. 327–361. ISSN: 2519-173X. DOI: 10.13154/tosc.v2020.i3.327-361.

Daemen, Joan and Vincent Rijmen (2002). *The design of Rijndael: AES-the advanced encryption standard*. Springer.

Derbez, Patrick and Baptiste Lambin (2022). "Fast MILP Models for Division Property". In: *IACR Transactions on Symmetric Cryptology* 2022.2, 289–321. DOI: 10.46586/tosc.v2022.i2.289-321. URL: https://tosc.iacr.org/index.php/ToSC/article/view/9722.

ElSheikh, Muhammad and Amr M. Youssef (Dec. 2021). "On MILP-Based Automatic Search for Bit-Based Division Property for Ciphers with (Large) Linear Layers". In: *ACISP 21*. Ed. by Joonsang Baek and Sushmita Ruj. Vol. 13083. LNCS. Springer, Heidelberg, pp. 111–131. DOI: 10.1007/978-3-030-90567-5_6.

Fu, Kai, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu (Mar. 2016). "MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck". In: *FSE 2016*. Ed. by Thomas Peyrin. Vol. 9783. LNCS. Springer, Heidelberg, pp. 268–288. DOI: 10.1007/978-3-662-52993-5_14.

Hong, Chunlei, Shasha Zhang, Siwei Chen, Da Lin, and Zejun Xiang (2021). "More Accurate Division Property Propagations Based on Optimized Implementations of Linear Layers". In: *Information Security and Cryptology*. Ed. by Yu Yu and Moti Yung. Cham: Springer International Publishing, pp. 212–232. ISBN: 978-3-030-88323-2.

Hu, Kai, Qingju Wang, and Meiqin Wang (2020). "IACR Transactions class documentation". In: *IACR Trans. Symm. Cryptol.* 2020.1, pp. 396–424. ISSN: 2519-173X. DOI: 10.13154/tosc.v2020.i1.396-424.

Jean, Jérémy (2016). *TikZ for Cryptographers*. https://www.iacr.org/authors/tikz/.

Lipmaa, Helger and Shiho Moriai (Apr. 2002). “Efficient Algorithms for Computing Differential Properties of Addition”. In: *FSE 2001*. Ed. by Mitsuru Matsui. Vol. 2355. LNCS. Springer, Heidelberg, pp. 336–350. DOI: 10.1007/3-540-45473-X_28.

Mouha, Nicky, Qingju Wang, Dawu Gu, and Bart Preneel (2012). “Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming”. In: *Information Security and Cryptology*. Ed. by Chuan-Kun Wu, Moti Yung, and Dongdai Lin. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 57–76. ISBN: 978-3-642-34704-7.

Sasaki, Yu and Yosuke Todo (2017). “New Algorithm for Modeling S-box in MILP Based Differential and Division Trail Search”. In: *Innovative Security Solutions for Information Technology and Communications*. Ed. by Pooya Farshim and Emil Simion. Cham: Springer International Publishing, pp. 150–165. ISBN: 978-3-319-69284-5.

Sun, Ling, Wei Wang, and Meiqin Q. Wang (2016). "MILP-aided Bit-Based Division Property for Primitives with Non-Bit-Permutation Linear Layers". In: *IET Information Security* 14.1, pp. 12–20. ISSN: 1751-8717.

Sun, Siwei, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song (Dec. 2014a). "Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers". In: *ASIACRYPT 2014, Part I*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8873. LNCS. Springer, Heidelberg, pp. 158–178. DOI: 10.1007/978-3-662-45611-8_9.

Sun, Siwei, Lei Hu, Ling Song, Yonghong Xie, and Peng Wang (2014b). "Automatic Security Evaluation of Block Ciphers with S-bP Structures Against Related-Key Differential Attacks". In: *Information Security and Cryptology*. Ed. by Dongdai Lin, Shouhuai Xu, and Moti Yung. Cham: Springer International Publishing, pp. 39–51. ISBN: 978-3-319-12087-4.

Sun, Siwei, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu (2014c). *Towards Finding the Best Characteristics of Some Bit-oriented Block Ciphers and Automatic Enumeration of (Related-key) Differential and Linear Characteristics with Predefined Properties*. Cryptology ePrint Archive, Report 2014/747. https://eprint.iacr.org/2014/747.

Sun, Yao (2021). *Towards the Least Inequalities for Describing a Subset in Z_2^n*. Cryptology ePrint Archive, Paper 2021/1084. https://eprint.iacr.org/2021/1084. URL: https://eprint.iacr.org/2021/1084.

Todo, Yosuke (Apr. 2015). "Structural Evaluation by Generalized Integral Property". In: *EUROCRYPT 2015, Part I*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. LNCS. Springer, Heidelberg, pp. 287–314. DOI: 10.1007/978-3-662-46800-5_12.

Udovenko, Aleksei (Dec. 2021a). "Convexity of Division Property Transitions: Theory, Algorithms and Compact Models". In: *ASIACRYPT 2021, Part I*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13090. LNCS. Springer, Heidelberg, pp. 332–361. DOI: 10.1007/978-3-030-92062-3_12.

– (2021b). *MILP modeling of Boolean functions by minimum number of inequalities*. Cryptology ePrint Archive, Report 2021/1099. https://eprint.iacr.org/2021/1099.

Xiang, Zejun, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin (Dec. 2016). "Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers". In: *ASIACRYPT 2016, Part I*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. Springer, Heidelberg, pp. 648–678. DOI: 10.1007/978-3-662-53887-6_24.

Zhang, Wenying and Vincent Rijmen (2019). "Division cryptanalysis of block ciphers with a binary diffusion layer". In: *IET Inf. Secur.* 13.2, pp. 87–95.

Division property (Todo 2015)

Algebraic Normal Form (ANF)

Every function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ admits a unique expression of the form

$$f(x) = \bigoplus_{u \in \mathbb{F}_2^n} x^u = \bigoplus_{u \in \mathbb{F}_2^n} \prod_{i=1}^{n} x_i^{u_i}$$

Example: $f(x) = x_1 \oplus x_1 x_4 \oplus x_2 x_4 x_5$

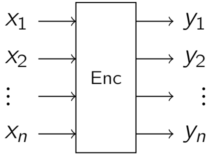
Division property (Todo 2015)

Algebraic Normal Form (ANF)

Every function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ admits a unique expression of the form

$$f(x) = \bigoplus_{u \in \mathbb{F}_2^n} x^u = \bigoplus_{u \in \mathbb{F}_2^n} \prod_{i=1}^{n} x_i^{u_i}$$

Example: $f(x) = x_1 \oplus x_1 x_4 \oplus x_2 x_4 x_5$

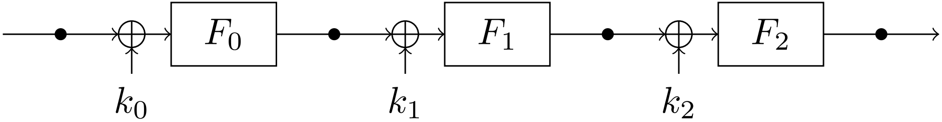


Integral cryptanalysis: does the ANF of

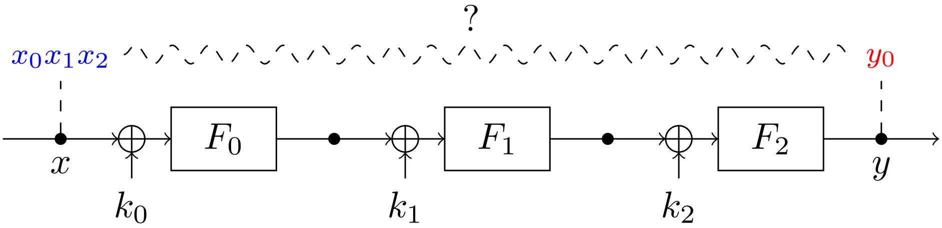
$$y_i = y_i(x_1, \ldots, x_n)$$

a) contain the monomial $x_1 x_2 \ldots x_{n-1}$? (Perfect Div. Prop.)
b) contain a monomial *multiple* of $x_1 x_2 \ldots x_{n-1}$? (Conv. Div. Prop.)

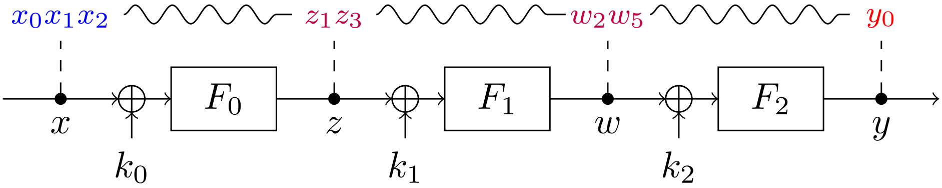
Division Trails (Monomial Trails)



Division Trails (Monomial Trails)



Division Trails (Monomial Trails)



Division Trails - Specifics

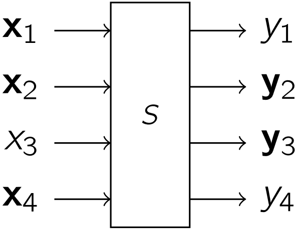
Conventional Division Property

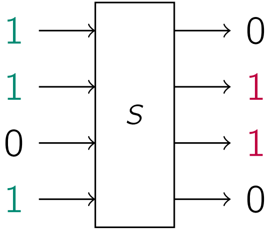
- trail found $\Rightarrow$??? (imprecision)
- trail can not exist $\Rightarrow$ attack
- monotonicity (active bits number is non-increasing) $\Rightarrow$ less uncertainty
- more often feasible

Perfect Division Property

- need to compute *parity* of the number of trails
- less often feasible

Modeling Example (S-boxes) (Xiang, Zhang, Bao, and Lin 2016)

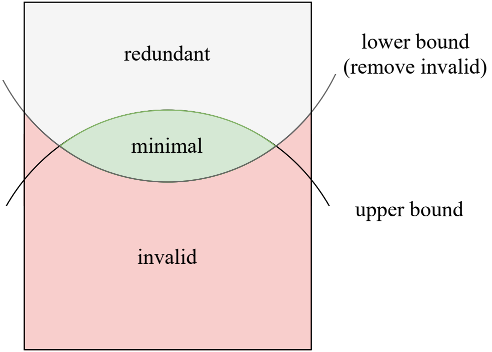




- does the ANF of $(y_2 y_3)(x)$ contain the monomial $x_1 x_2 x_4$? (or its multiple)
- encode as a bit-vector transition $1101 \xrightarrow{S} 0110$
- MILP model of a subset of $\{0,1\}^{2n}$ (e.g. convex hull + greedy/SetCover, etc.)

venko 2021a) conv. div. prop.: **convex** set, with respect to a partial order, very compact models

Conventional Division Property - Redundancy



Modeling Linear Maps - Criteria (Zhang and Rijmen 2019)

Proposition

Consider the linear map defined by a matrix $L \in \mathbb{F}_2^{n \times n}$. Then,

$$u \xrightarrow{L} v$$

if and only if the submatrix defined by 1s from u and v is invertible.

Example:

$$\begin{array}{c} \\ v_1{=}0 \\ v_2{=}1 \\ v_3{=}1 \end{array} \begin{array}{c} \begin{array}{ccc} u_1{=}0 & u_2{=}1 & u_3{=}1 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \Leftrightarrow \quad 011 \xrightarrow{L} 011$$

Modeling Linear Maps - Imprecise (Sun, Wang, and Wang 2016)

If a matrix $L \in \mathbb{F}_2^{n \times n}$ is invertible, then there exists a permutation σ such that

$$\prod_{i=1}^{n} L_{i,\sigma(i)} = 1$$

$u \xrightarrow{L} v$ implies a matching between 1s in u and 1s in v, with the edges given by 1s in L

$$\begin{aligned}
\text{variables:} \quad & t_{i,j} (\text{edges for } L_{i,j} = 1), u_i, v_i \in \{0,1\} && i,j \in [1,n] \\
\text{equations:} \quad & u_i = \sum_{j=1}^{n} L_{i,j} t_{i,j} && i \in [1,n], L_{i,j} = 1 \\
& v_j = \sum_{i=1}^{n} L_{i,j} t_{i,j} && j \in [1,n], L_{i,j} = 1
\end{aligned}$$

Modeling Linear Maps - Precise Methods

Wang, and Wang 2020) (SMT-based):

- introduce variables for the inverse matrix $L_{u,v}^{-1}$
- add matrix multiplication constraint $L_{u,v} \times L_{u,v}^{-1} = Id$

Modeling Linear Maps - Precise Methods

Wang, and Wang 2020) (SMT-based):

- introduce variables for the inverse matrix $L_{u,v}^{-1}$
- add matrix multiplication constraint $L_{u,v} \times L_{u,v}^{-1} = Id$

n, Lin, and Xiang 2021) (MILP-based):

- combine different implementations of the linear layer
- correct transition must satisfy all of them

Modeling Linear Maps - Precise Methods

Wang, and Wang 2020) (SMT-based):

- introduce variables for the inverse matrix $L_{u,v}^{-1}$
- add matrix multiplication constraint $L_{u,v} \times L_{u,v}^{-1} = Id$

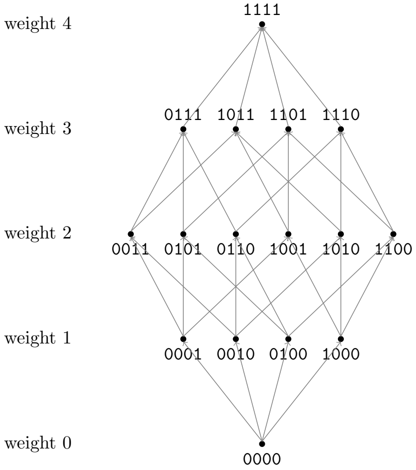
n, Lin, and Xiang 2021) (MILP-based):

- combine different implementations of the linear layer
- correct transition must satisfy all of them

erbez and Lambin 2022) (MILP-based):

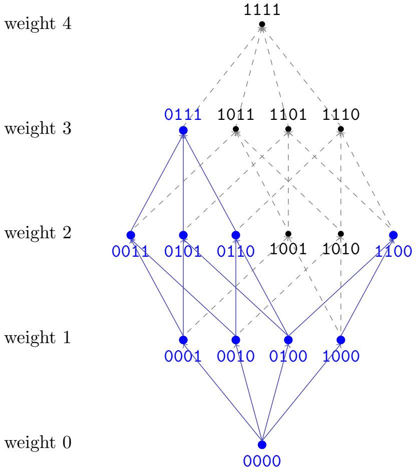
- implement basic (lossy) constraints
- use **callbacks** with **lazy constraints** to discard wrong solutions on-the-fly

Monotonicity and Convexity on $\{0,1\}^n$ - Definitions



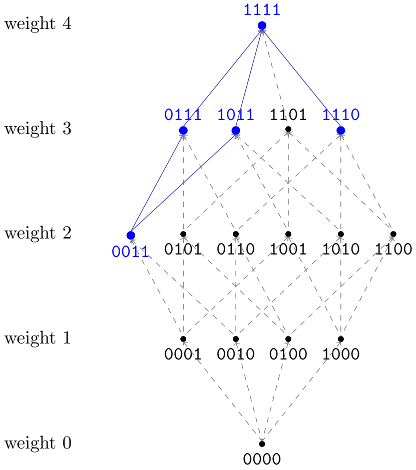
- **partial order** on $\{0,1\}^n$:
  $u \preceq v$ iff $\forall i \ \ u_i \leq v_i$

Monotonicity and Convexity on $\{0,1\}^n$ - Definitions



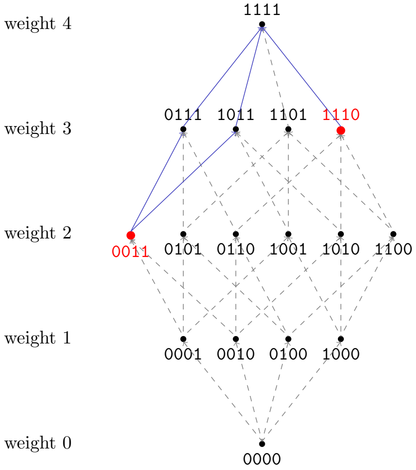
- **partial order** on $\{0,1\}^n$:
  $u \preceq v$ iff $\forall i \;\; u_i \leq v_i$
- lower set: $u \notin X \not\preceq v \in X$

Monotonicity and Convexity on $\{0,1\}^n$ - Definitions



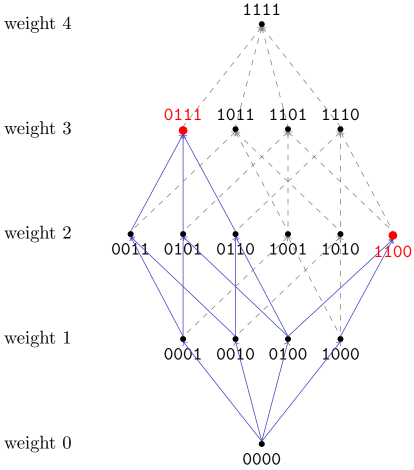
- **partial order** on $\{0,1\}^n$:
  $u \preceq v$ iff $\forall i \;\; u_i \leq v_i$
- lower set: $u \notin X \not\preceq v \in X$
- upper set: $u \in X \not\preceq v \notin X$

Monotonicity and Convexity on $\{0,1\}^n$ - Definitions



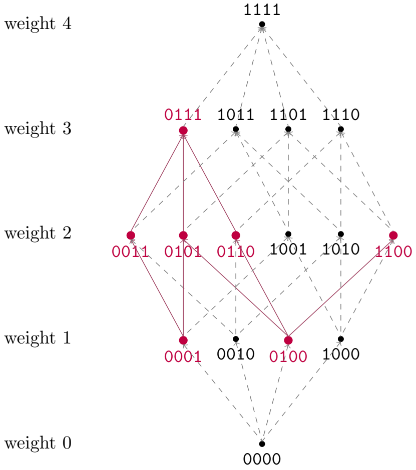
- **partial order** on $\{0,1\}^n$:
  $u \preceq v$ iff $\forall i \quad u_i \leq v_i$
- lower set: $u \notin X \not\preceq v \in X$
- upper set: $u \in X \not\preceq v \notin X$
- extreme elements
  (resp. maximal/minimal)
  form a compact representation:
  {**11, 111*}

Monotonicity and Convexity on $\{0,1\}^n$ - Definitions



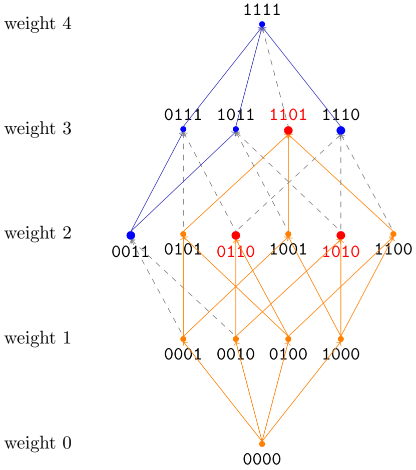
- **partial order** on $\{0,1\}^n$:
  $u \preceq v$ iff $\forall i \quad u_i \leq v_i$
- lower set: $u \notin X \not\preceq v \in X$
- upper set: $u \in X \not\preceq v \notin X$
- extreme elements
  (resp. maximal/minimal)
  form a compact representation:
  {**11, 111*}
  {0***, **00}

Monotonicity and Convexity on $\{0,1\}^n$ - Definitions



- **partial order** on $\{0,1\}^n$:
  $u \preceq v$ iff $\forall i \;\; u_i \leq v_i$
- lower set: $u \notin X \not\preceq v \in X$
- upper set: $u \in X \not\preceq v \notin X$
- extreme elements
  (resp. maximal/minimal)
  form a compact representation:
  {**11, 111*}
  {0***, **00}
- convex set: lower set $\cap$ upper set
  (two-sided bound)

Modeling Upper/Lower Sets

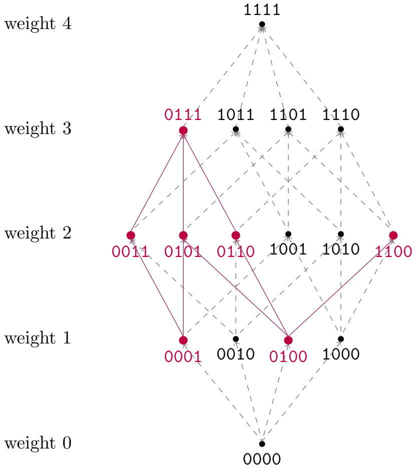


modeling an upper set $X \subseteq \{0,1\}^n$:

- monotone CNF (from the max-set of the complement):

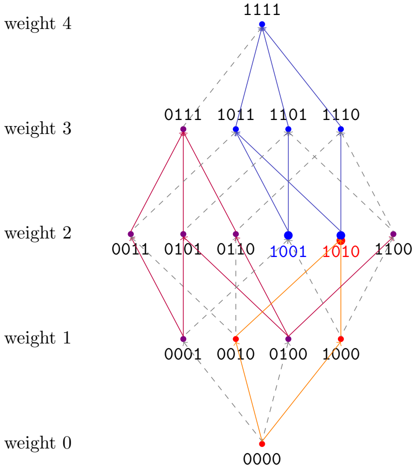
$$\underbrace{(x_0 \vee x_3)}_{0110} \wedge \underbrace{(x_2)}_{1101} \wedge \underbrace{(x_1 \vee x_3)}_{1010}$$

Modeling Convex Sets



modeling an convex set $X \subseteq \{0,1\}^n$:

Modeling Convex Sets



modeling an convex set $X \subseteq \{0,1\}^n$:

- combined CNF
  of the upper/lower bounds:

$$\underbrace{(\neg x_0 \vee \neg x_3)}_{1001} \wedge \underbrace{(\neg x_0 \vee \neg x_2)}_{1010} \wedge \underbrace{(x_1 \vee x_3)}_{1010}$$

Conventional Division Property - Convexity and Redundancy (Udovenko 2021a)