Cryptanalysis of symmetric cipher using generic solvers

Arthur GONTIER

March 30, 2026

Université de Rennes

CAPSULE IRISA

Cryptography

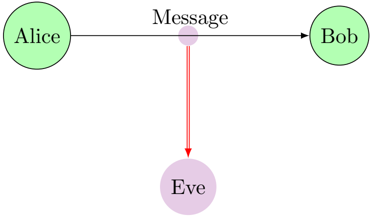
Communicate a secret:

- Confidentiality
- Integrity
- Authentication
- ...

Cryptography

Communicate a secret:

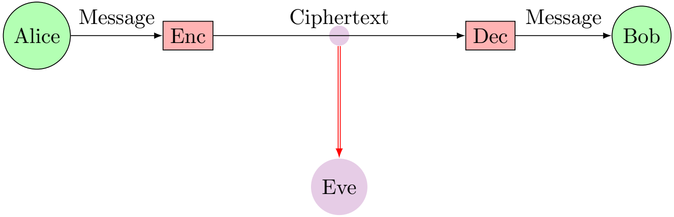
- Confidentiality
- Integrity
- Authentication
- ...



Cryptography

Communicate a secret:

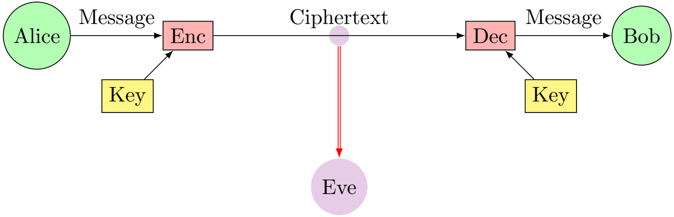
- Confidentiality
- Integrity
- Authentication
- ...



Cryptography

Communicate a secret:

- Confidentiality
- Integrity
- Authentication
- ...



Symmetric ciphers

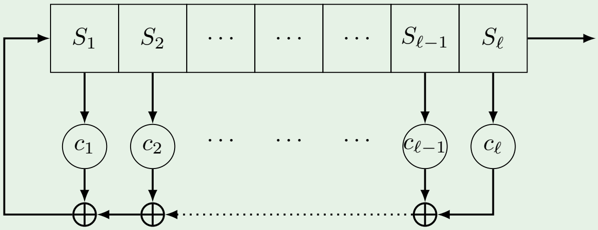
An invertible function E:

$$E : key \times message \to ciphertext$$

An iterated round function f:

$$E = f(f(\dots f(f(key, message))\dots))$$

Stream cipher example: Linear Feedback Shift Register (LFSR)



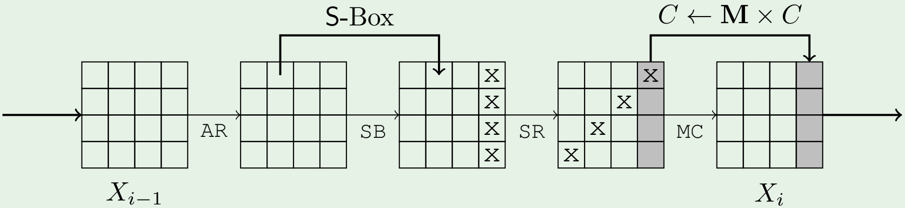
Some stream ciphers: A5/1 (GSM 1987), RC4 (Wifi 1987), E0 (Bluetooth 1999), **Trivium** (2004),...

Symmetric ciphers

Properties of a resistant cipher:

- Diffusion (permutation, XOR,. . .)
- Confusion (Substitution Box)

Block cipher example: Advanced Encryption Standard (AES)



Some block ciphers families: **Feistel networks**, Substitution permutation networks, ARX,. . .

Distinguishers

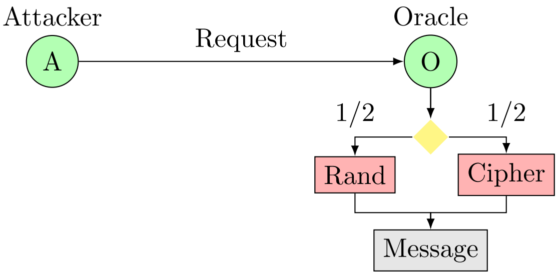
Distinguish a cipher from a random message

- Various types of distinguishers
- An analysis of each distinguisher on each cipher

Distinguishers

Distinguish a cipher from a random message

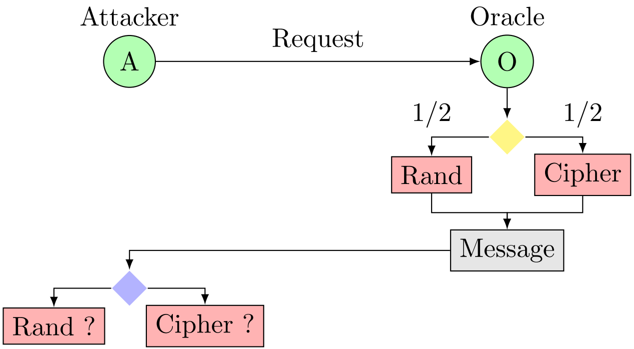
- Various types of distinguishers
- An analysis of each distinguisher on each cipher



Distinguishers

Distinguish a cipher from a random message

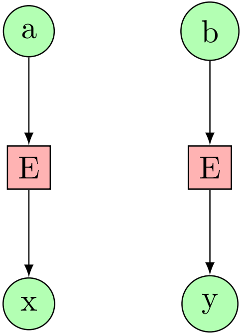
- Various types of distinguishers
- An analysis of each distinguisher on each cipher



Differentials [BS93, BS90]

Differential characteristics

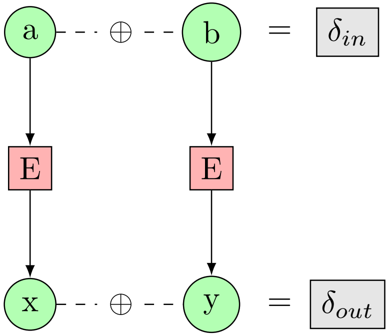
- Difference: $a \oplus b = \delta$
- Probability: $P(E(x) = E(x \oplus \delta_{in}) \oplus \delta_{out})$



Differentials [BS93, BS90]

Differential characteristics

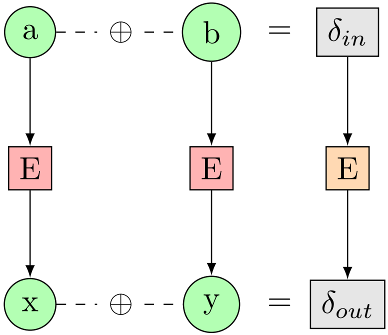
- Difference: $a \oplus b = \delta$
- Probability: $P(E(x) = E(x \oplus \delta_{in}) \oplus \delta_{out})$



Differentials [BS93, BS90]

Differential characteristics

- Difference: $a \oplus b = \delta$
- Probability: $P(E(x) = E(x \oplus \delta_{in}) \oplus \delta_{out})$



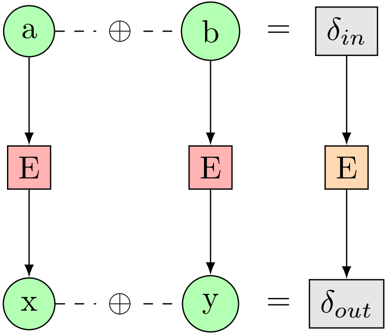
Differentials [BS93, BS90]

Differential characteristics

- Difference: $a \oplus b = \delta$
- Probability: $P(E(x) = E(x \oplus \delta_{in}) \oplus \delta_{out})$

Truncated characteristics [Knu94]

$$\Delta x_i = \begin{cases} 0 & \text{if} \quad \delta x_i = 0 \\ 1 & \text{if} \quad \delta x_i \in [\![1, 2^n - 1]\!] \end{cases}$$



Differentials [BS93, BS90]

Differential characteristics

- Difference: $a \oplus b = \delta$
- Probability: $P(E(x) = E(x \oplus \delta_{in}) \oplus \delta_{out})$

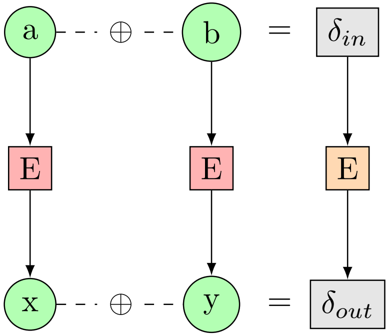
Truncated characteristics [Knu94]

$$\Delta x_i = \begin{cases} 0 & \text{if} \quad \delta x_i = 0 \\ 1 & \text{if} \quad \delta x_i \in [\![1, 2^n - 1]\!] \end{cases}$$

Two steps method [BN10, FJP13, GLMS20]

Solving methods

Branch and bound, dynamic programming, generic solvers, ...



Generic solvers

1. Model the problem

Generic solvers

1. Model the problem

Linear Programming MILP

- **Constraints**: Linear
- **Variables**: Integer or/and real
- **Optimize** a linear objective

Generic solvers

1 Model the problem

Linear Programming MILP

- **Constraints**: Linear
- **Variables**: Integer or/and real
- **Optimize** a linear objective

Boolean satisfaction SAT

- **Constraints**: CNF
- **Variables**: Boolean
- **Find** a satisfiable assignment

Generic solvers

1 Model the problem

Linear Programming MILP

- **Constraints**: Linear
- **Variables**: Integer or/and real
- **Optimize** a linear objective

Boolean satisfaction SAT

- **Constraints**: CNF
- **Variables**: Boolean
- **Find** a satisfiable assignment

Constraint Programming CP

- **Constraints**: Various
- **Variables**: Integer, set,...
- **Find** a satisfiable assignment or **optimize** an objective

Generic solvers

1 Model the problem

Linear Programming MILP

- **Constraints**: Linear
- **Variables**: Integer or/and real
- **Optimize** a linear objective

Boolean satisfaction SAT

- **Constraints**: CNF
- **Variables**: Boolean
- **Find** a satisfiable assignment

Constraint Programming CP

- **Constraints**: Various
- **Variables**: Integer, set,. . .
- **Find** a satisfiable assignment or **optimize** an objective

2 Solve the model

Generic solvers

1 Model the problem

Linear Programming MILP

- **Constraints**: Linear
- **Variables**: Integer or/and real
- **Optimize** a linear objective

Boolean satisfaction SAT

- **Constraints**: CNF
- **Variables**: Boolean
- **Find** a satisfiable assignment

Constraint Programming CP

- **Constraints**: Various
- **Variables**: Integer, set,...
- **Find** a satisfiable assignment or **optimize** an objective

2 Solve the model

Solving MILP model

- Branch and bound
- Simplex/Barrier method

↪ Gurobi

Generic solvers

1. Model the problem

Linear Programming MILP

- **Constraints**: Linear
- **Variables**: Integer or/and real
- **Optimize** a linear objective

Boolean satisfaction SAT

- **Constraints**: CNF
- **Variables**: Boolean
- **Find** a satisfiable assignment

Constraint Programming CP

- **Constraints**: Various
- **Variables**: Integer, set,...
- **Find** a satisfiable assignment or **optimize** an objective

2. Solve the model

Solving MILP model

- Branch and bound
- Simplex/Barrier method

↪ Gurobi

Solving SAT model

- Conflict Driven Clause Learning

↪ PicatSAT

Generic solvers

1 Model the problem

Linear Programming MILP

- **Constraints**: Linear
- **Variables**: Integer or/and real
- **Optimize** a linear objective

Boolean satisfaction SAT

- **Constraints**: CNF
- **Variables**: Boolean
- **Find** a satisfiable assignment

Constraint Programming CP

- **Constraints**: Various
- **Variables**: Integer, set,...
- **Find** a satisfiable assignment or **optimize** an objective

2 Solve the model

Solving MILP model

- Branch and bound
- Simplex/Barrier method

↪ Gurobi

Solving SAT model

- Conflict Driven Clause Learning

↪ PicatSAT

Solving CP model

- Branch and bound
- Filtering algorithms

↪ Choco

Subject of the thesis

Cryptography problems

- **Design** with good properties
- **Analysis** with all distinguishers

Goal: Find new CP models for the design and analysis of symmetric ciphers

Contributions

Superpoly recovery on Trivium

- **Problem**: Cube attack, algebraic attack
- **Target**: Trivium
- **Method**: Graph representation, MILP model

Contributions

Superpoly recovery on Trivium

- **Problem**: Cube attack, algebraic attack
- **Target**: Trivium
- **Method**: Graph representation, MILP model

Explanation generation for CP solvers

- **Problem**: Deduce new constraint explanations
- **Target**: Improve CDCL-CP solvers
- **Method**: Rule system, rewriting algorithm

Contributions

Superpoly recovery on Trivium

- **Problem**: Cube attack, algebraic attack
- **Target**: Trivium
- **Method**: Graph representation, MILP model

Optimal permutation for diffusion in GFN

- **Problem**: Optimal full diffusion
- **Target**: Generalized Feistel Networks
- **Method**: Graph representation, ad hoc algorithm

Explanation generation for CP solvers

- **Problem**: Deduce new constraint explanations
- **Target**: Improve CDCL-CP solvers
- **Method**: Rule system, rewriting algorithm

Automated tool for differentials

- **Problem**: Differential characteristics
- **Target**: All word-oriented cipher
- **Method**: Graph representation, CP model

Table of Contents

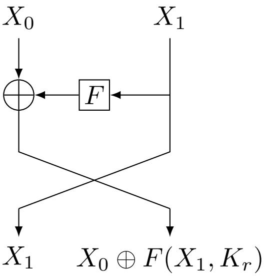
Feistel

Feistel Networks

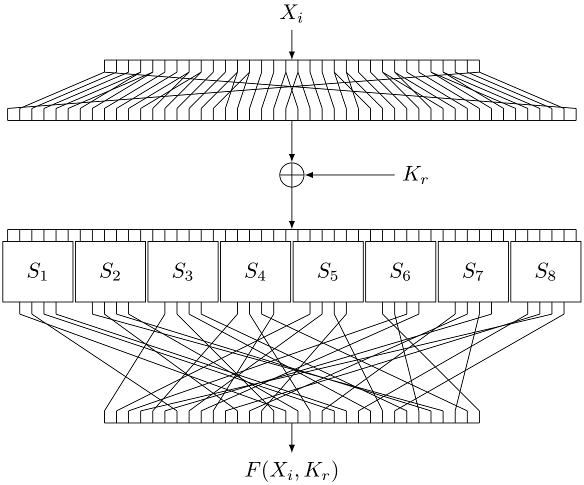
- Horst Feistel [Smi71, Fei73]
- Data Encryption Standard (DES) 1977-1999 (-2017)

Properties

- Two blocks X_0, X_1
- If F is a "pseudorandom" function, the cipher is perfect after 4 rounds

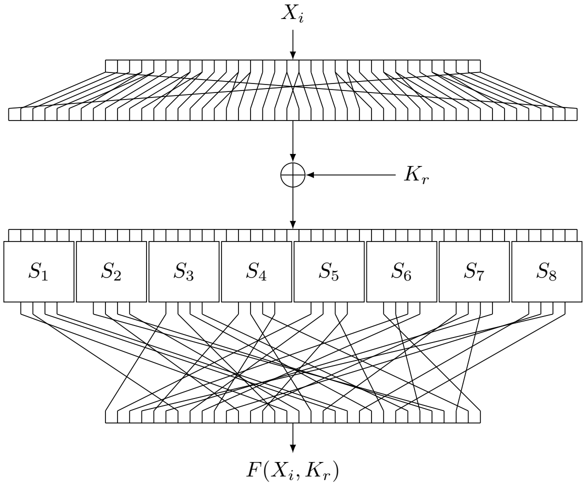


Examples of F functions

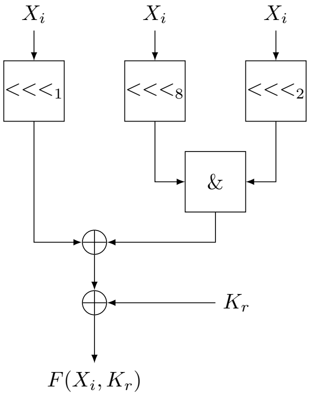


DES

Examples of F functions

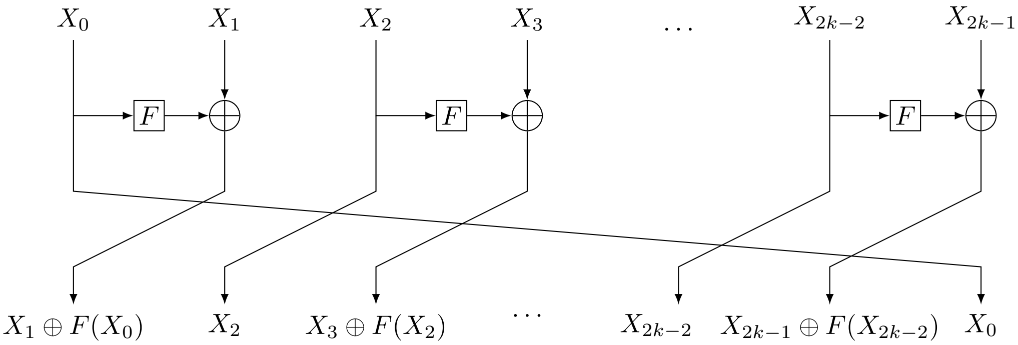


DES

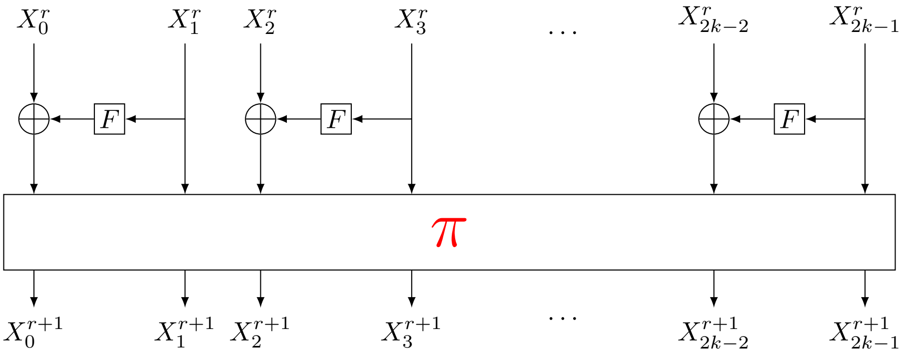


SIMON

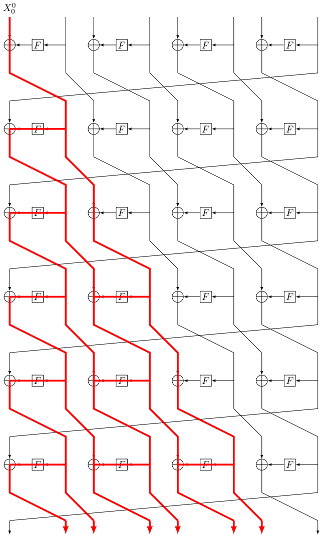
Generalizations of Feistel Networks [ZMI89]



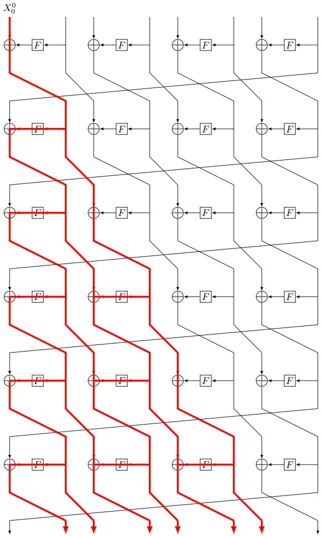
Generalizations of Feistel Networks [ZMI89]

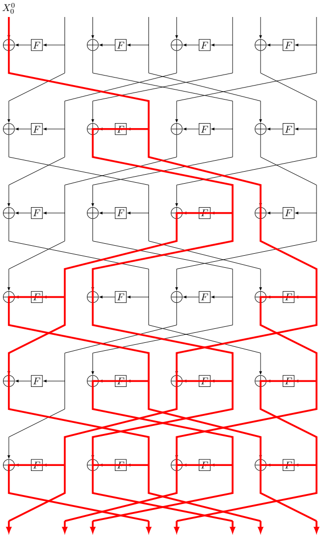


Diffusion in GFN



Diffusion in GFN





Best permutations

Definition (Diffusion Round)

$DR_i(\pi)$ is the minimum number of rounds r such that X_i^0 are diffused to all X_i^r.

Best permutations

Definition (Diffusion Round)

$DR_i(\pi)$ is the minimum number of rounds r such that X_i^0 are diffused to all X_i^r.

Previous methods

- [SM10] Enumerate all π and test diffusion
- [CGT19] Use equivalence classes
- [DFLM19] Focus on even-odd π

GOAL: Find the best π in the general case

$2k$	even-odd			non-even-odd	
	LB	DR	Ref	DR	Ref
6	5	5	[SM10]	6	[SM10]
8	6	6		6	
10	6	7		7	
12	7	8		8	
14	7	8		8	
16	7	8		8	
18	8	8	[CGT19]	9	[CGT19]
20	8	9		9	
22	8	8		?	
24	8	9			
26	8	9	[DFLM19]		
28	9	9			
30	9	9			
32	9	9			
34	9	10			
36	9	9			

CP model with Set variables

Integer variables P_i (with $i \in [\![0, 2k-1]\!]$) represents π:

$$\mathit{AllDifferent}(P)$$

CP model with Set variables

Integer variables P_i (with $i \in [\![0, 2k-1]\!]$) represents π:

$$AllDifferent(P)$$

Set variables $S_{i,r}$ (with $i \in [\![0, 2k-1]\!], r \in [\![1, R]\!]$) represents the diffusion from block i after r rounds:

$$\begin{cases} S_{2i,1} = \{P_{2i}\} \\ S_{2i+1,1} = \{P_{2i}, P_{2i+1}\} \end{cases} \quad \forall i \in [\![0, k-1]\!]$$

CP model with Set variables

Integer variables P_i (with $i \in [\![0, 2k-1]\!]$) represents π:

$$AllDifferent(P)$$

Set variables $S_{i,r}$ (with $i \in [\![0, 2k-1]\!], r \in [\![1, R]\!]$) represents the diffusion from block i after r rounds:

$$\begin{cases} S_{2i,1} = \{P_{2i}\} \\ S_{2i+1,1} = \{P_{2i}, P_{2i+1}\} \end{cases} \quad \forall i \in [\![0, k-1]\!]$$

$$S_{i,r} = \bigcup_{j \in S_{i,r-1}} S_{j,1} \quad \forall i \in [\![0, 2k-1]\!]\ \forall r \in [\![2, R]\!]$$

Optimizations:

- Symmetry constraints

Early models conclusions

Other models:

- Adjacency matrices
- Graph variable
- Table constraints

Early models conclusions

Other models:

- Adjacency matrices
- Graph variable
- Table constraints

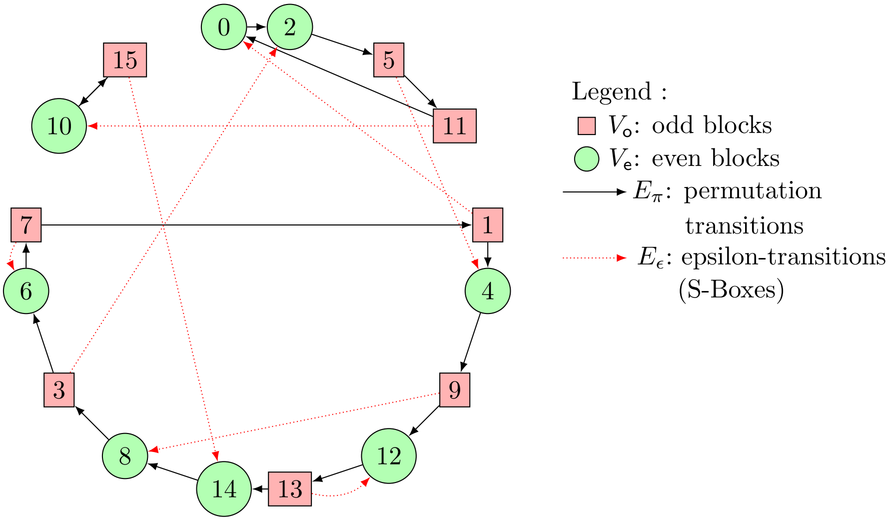
Conclusion:

- **2 New lines:** the set model can find the best π for up to 24 blocks.
- **Not efficient enough:** still a lot of symmetric solutions.

$2k$	even-odd			non-even-odd	
	bound	DR	Ref	DR	Ref
6	5	5	[SM10]	6	[SM10]
8	6	6		6	
10	6	7		7	
12	7	8		8	
14	7	8		8	
16	7	8		8	
18	8	8	[CGT19]	9	[CGT19]
20	8	9		9	
22	8	8		9	**New**
24	8	9		≥ 9	
26	8	9	[DFLM19]	**?**	
28	9	9			
30	9	9			
32	9	9			
34	9	10			
36	9	9			

Graph representation of GFN permutations

i	0	1	2	...
$\pi(i)$	2	4	5	...



Properties

Diffusion round $DR(\pi) = R$

All pairs of nodes have a path of R edges in E_π

Properties

Diffusion round $DR(\pi) = R$

All pairs of nodes have a path of R edges in E_π

Diffusion round $DR(\pi) = R$

All pair of nodes in (V_e, V_o) have a path of $R - 1$ edges in E_π

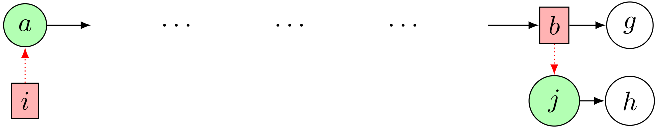
Properties

Diffusion round $DR(\pi) = R$

All pairs of nodes have a path of R edges in E_π

Diffusion round $DR(\pi) = R$

All pair of nodes in (V_e, V_o) have a path of $R - 1$ edges in E_π



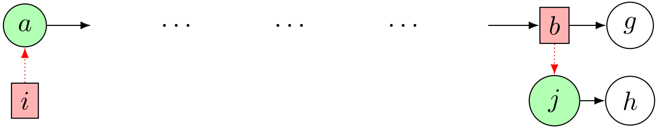
Properties

Diffusion round $DR(\pi) = R$

All pairs of nodes have a path of R edges in E_π

Diffusion round $DR(\pi) = R$

All pair of nodes in (V_e, V_o) have a path of $R - 1$ edges in E_π



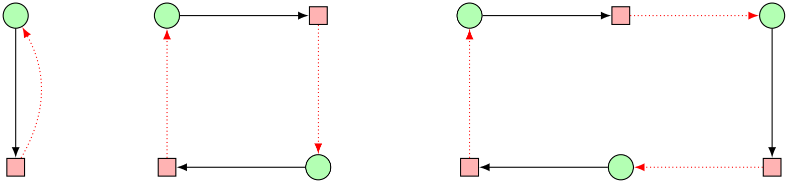
Diffusion round (even-odd) $DR(\pi) = R$

All pair of nodes in (V_o, V_e) have a path of $R - 3$ edges in E_π

Properties for symmetries

Definition (ϵ-cycle)

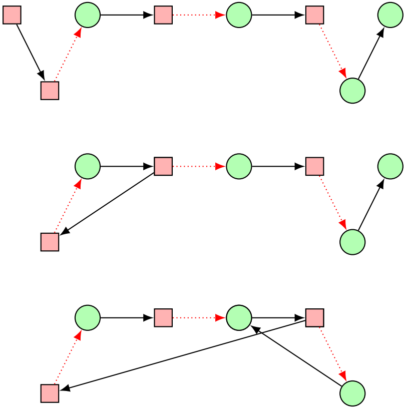
An ϵ-cycle is a path $c = (e_1, \ldots, e_{2l})$ in which the first and last nodes are equal, and edges alternate between E_π and E_ϵ one by one.



Properties for symmetries

Definition (ϵ-chain)

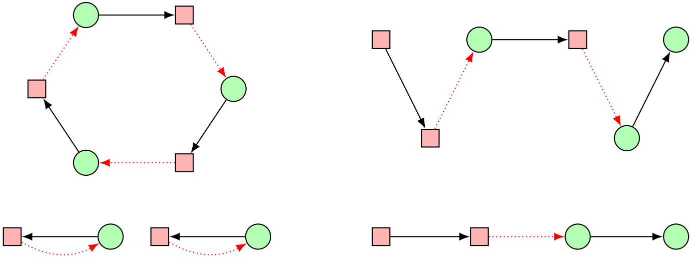
An ϵ-chain is a path $c = (e_1, \ldots, e_{2l+1})$ in which the two first nodes are in V_o, and the two last nodes are in V_e. The edges alternate between E_π and E_ϵ one by one.



Skeleton

Definition (Skeleton)

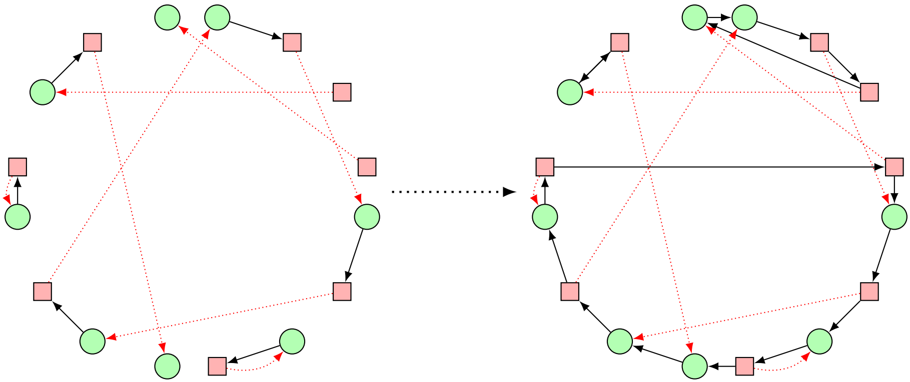
A *skeleton* of size k is a set of ϵ-cycles and ϵ-chains whose sum of sizes is k.



Number of skeletons: $\sum_{i=0}^{k} \mathcal{N}_i \times \mathcal{N}_{k-i}$

- 16 blocks: 163 skeletons (22 even-odd)
- 32 blocks: 5591 skeletons (231 even-odd)

Complete the skeleton

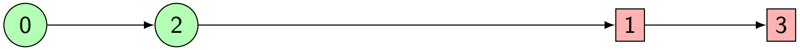


Build the paths: an example with $DR(\pi) = 5$

Paths of length 4:

$0 \rightarrow 3$?

Build the paths: an example with $DR(\pi) = 5$



Paths of length 4:

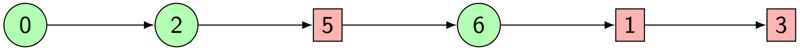
$0 \rightarrow 3$ ✗

Build the paths: an example with $DR(\pi) = 5$

Paths of length 4:

$0 \rightarrow 3$?

Build the paths: an example with $DR(\pi) = 5$



Paths of length 4:

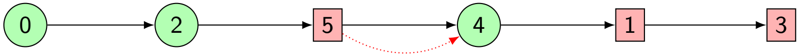
$0 \rightarrow 3$ ✗

Build the paths: an example with $DR(\pi) = 5$

Paths of length 4:

$0 \rightarrow 3$ ✓

Build the paths: an example with $DR(\pi) = 5$

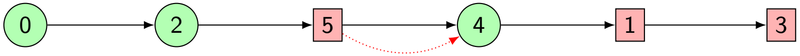


Paths of length 4:

$0 \rightarrow 3$ ✓

$0 \rightarrow 1$ ✓

Build the paths: an example with $DR(\pi) = 5$



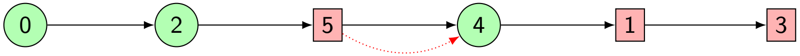
Paths of length 4:

$0 \to 3$ ✓

$0 \to 1$ ✓

$2 \to 3$ ✓

Build the paths: an example with $DR(\pi) = 5$



Paths of length 4:

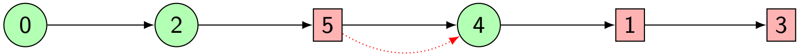
$0 \rightarrow 3$ ✓

$0 \rightarrow 1$ ✓

$2 \rightarrow 3$ ✓

$2 \rightarrow 1$ ✗

Build the paths: an example with $DR(\pi) = 5$



Paths of length 4:

$0 \to 3$ ✓

$0 \to 1$ ✓

$2 \to 3$ ✓

$2 \to 1$ ✗

No π with $DR = 5$

Contribution

Implementation:

- Parallel path builder algorithm on each skeleton
- Strategy: start the paths by the smallest ϵ-chains

Contribution

Implementation:

- Parallel path builder algorithm on each skeleton
- Strategy: start the paths by the smallest ϵ-chains

Results:

- Non-even-odd permutations are not strictly better for up to 32 blocks and 36 blocks.

$2k$	even-odd			non-even-odd	
	LB	DR	Ref	DR	Ref
6	5	5	[SM10]	6	[SM10]
8	6	6		6	
10	6	7		7	
12	7	8		8	
14	7	8		8	
16	7	8		8	
18	8	8	[CGT19]	9	[CGT19]
20	8	9		9	
22	8	8		9	**New**
24	8	9		≥ 9	
26	8	9	[DFLM19]	≥ 9	
28	9	9		≥ 9	
30	9	9		≥ 9	
32	9	9		≥ 9	
34	9	10		≥ 9	
36	9	9		≥ 9	

Table of Contents

Automatic differential cryptanalysis

Problem

- Each cipher must be resistant to differentials

Automatic differential cryptanalysis

Problem

- Each cipher must be resistant to differentials

Automated tools for differentials ?

- YAARX [Leu12] dedicated algorithms (ARX)
- CryptoSMT [Köl] SMT models
- TAGADA [LDLS21] DAG to MiniZinc models
- CASCADA [RR22] SMT models
- CLAASP [BGG+23] DAG to MiniZinc models

Automatic differential cryptanalysis

Problem

- Each cipher must be resistant to differentials

Automated tools for differentials ?

- YAARX [Leu12] dedicated algorithms (ARX)
- CryptoSMT [Köl] SMT models
- TAGADA [LDLS21] DAG to MiniZinc models
- CASCADA [RR22] SMT models
- CLAASP [BGG+23] DAG to MiniZinc models

A tool for truncated differentials: Tagada

- **Input:** cipher DAG and optional bound
- **Output:** truncated solutions

Generate a MiniZinc model and solve it (Solver: PicatSAT)

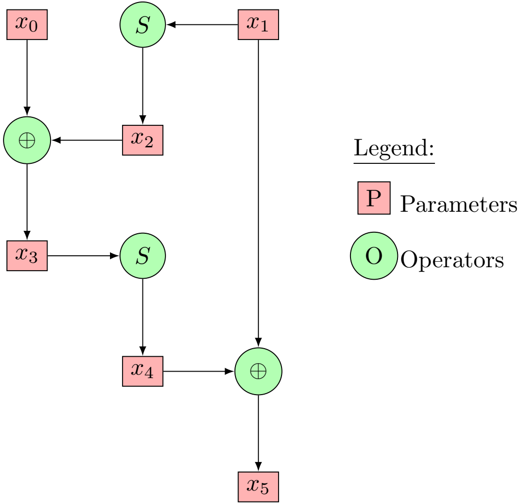
Unified description of ciphers

Directed Acyclic Graph (DAG)

- **Parameters:** variables or constants
- **Operators:** cipher operators

Example (S operator)

- Domain: $[\![0, 256]\!]$
- Co-Domain: $[\![0, 256]\!]$
- Function:
  - type: S-Box
  - lookup table $[170, 22, 3, \ldots]$



Second step with CP (Choco)

Contribution

- **Input:** cipher DAG, truncated solutions and optional bound
- **Output:** differentials of best probability

Generate a CP model and solve it (Solver: Choco)

↪ **How to model all the operators ?** (S-Boxes, XORs, LFSRs, Galois Fields operations,. . .)

S-Box and Differential Distribution Table (DDT)

DDT	0	1	2	3	4	...
0	**64**	0	0	0	0	
1	0	0	0	6	0	
2	0	0	0	8	0	...
3	14	4	2	2	10	
4	0	0	0	6	0	
⋮			⋮			⋱

Computing DDT

$$DDT(\delta_{in}, \delta_{out}) = \#\{X | S(X) \oplus S(X \oplus \delta_{in}) = \delta_{out}\}$$

Modeling DDT with table constraint

- List of tuples: $tuple(\delta_{in}, \delta_{out}, Prob)$
- **Filtering:** efficient data structure to retain always one valid tuple

S-Box and Differential Distribution Table (DDT)

Computing DDT

$$DDT(\delta_{in}, \delta_{out}) = \#\{X | S(X) \oplus S(X \oplus \delta_{in}) = \delta_{out}\}$$

Modeling DDT with table constraint

- List of tuples: $tuple(\delta_{in}, \delta_{out}, Prob)$
- **Filtering:** efficient data structure to retain always one valid tuple

DDT	0	1	2	3	4	...
0	**64**	0	0	0	0	
1	0	0	0	6	0	
2	0	0	0	8	0	...
3	14	4	2	2	10	
4	0	0	0	6	0	
⋮			⋮			⋱

$$tuple(0, 0, 1)$$
$$tuple(3, 0, \frac{14}{64})$$
$$tuple(3, 1, \frac{4}{64})$$
$$\dots$$

XOR filtering algorithm

Previous works

- Table constraint
- Dedicated algorithm

Filtering quality

- The set is unlikely to filter values.

↪ Filtering condition: $\#D_a \times \#D_b \leq \#D_c$

Algorithm 1: 3-variable XOR filtering

Input: IntVar a, IntVar b, IntVar c (target)

```
set ← ∅;
// Loop through possible values
for all v1 ∈ D_a do
    for all v2 ∈ D_b do
        set ← set ∪ {v1 ⊕ v2};
D_c ← D_c ∩ set;
```

Other filtering algorithms

Non-linear Operators		
Operator	Name	Constraint
DDT	Differential Distribution Table	Table

Linear Operators		
$\oplus$	N-ary Bitwise XOR	Custom
$\otimes_K$	Galois Field Multiplication with Constant	
LFSR	Linear Feedback Shift Register	
$\ll$ or $\gg$	Left (Right) Shift	
$\lll$ or $\ggg$	Left (Right) Circular Shift	
$\odot_K$	Galois Field Matrix Multiplication with Constant Matrix	Decomposition to $\otimes_K$ and $\oplus$
$=$	Equal	Native
$\&_K$	Bitwise AND with Constant	Table
$\Vert_K$	Bitwise OR with Constant	
$AB \to (A, B)$	Split	
$(A, B) \to AB$	Concat	
T	Linear Lookup Table	

Optimisations

Optimizations

- **Simplification:** Remove inactive S-Boxes
- **Heuristic:** Start search near S-Boxes
- **Solving:** Parallel competitive models
- **Solving:** Two steps together

Algorithm 2: TWOSTEP

$List1 \leftarrow$ STEP1-ENUM(LB) ;
$List2 \leftarrow$ STEP2-PARALLEL$(List1)$;

Optimisations

Optimizations

- **Simplification:** Remove inactive S-Boxes
- **Heuristic:** Start search near S-Boxes
- **Solving:** Parallel competitive models
- **Solving:** Two steps together

Algorithm 5: TwoStep

$S1, UB \leftarrow$ Step1-opt() ;
while $LB < UB$ **do**
  $S2, LB \leftarrow$ Step2($S1, LB$) ;
  if $LB < UB$ **then**
    $S1 \leftarrow$ Step1-next(UB) ;
    if $S1$ *is* `null` **then**
      $S1, UB \leftarrow$ Step1-opt(UB)

Contribution

Tagada two steps results:

- Reproduce all these results within a day

Cipher	Max R	Proba	Ref
Midori-64	16	2^{-16}	[Gér18]
Midori-128	20	2^{-40}	
Warp	41	2^{-40}	[TB22]
Twine-80	18	2^{-64}	[SMS+20]
Twine-128	16	2^{-52}	
Skinny-64-TK1	11	2^{-64}	[DDH+21]
Skinny-128-TK1	11	2^{-74}	
AES-128	5	2^{-105}	[GLMS20]
AES-192	9	2^{-146}	
AES-256	14	2^{-146}	
Rijndael-128-160	7	2^{-120}	[RGMS22]
Rijndael-128-224	12	2^{-212}	
Rijndael-160-128	4	2^{-112}	
Rijndael-160-160	6	2^{-138}	

Cipher	Max R	Proba	Ref
Rijndael-160-192	8	2^{-141}	[RGMS22]
Rijndael-160-224	9	2^{-190}	
Rijndael-160-256	11	2^{-204}	
Rijndael-192-128	3	2^{-54}	
Rijndael-192-160	5	2^{-118}	
Rijndael-192-192	7	2^{-153}	
Rijndael-192-224	8	2^{-205}	
Rijndael-192-256	9	2^{-179}	
Rijndael-224-128	3	2^{-54}	
Rijndael-224-160	4	2^{-122}	
Rijndael-224-192	5	2^{-124}	
Rijndael-224-224	7	2^{-196}	
Rijndael-224-256	8	2^{-182}	
Rijndael-256-128	3	2^{-54}	
Rijndael-256-160	4	2^{-130}	
Rijndael-256-192	5	2^{-148}	
Rijndael-256-224	4	2^{-115}	
Rijndael-256-256	6	2^{-128}	

Table of Contents

Conclusion and future work

Contributions

- Algebraic analysis of Trivium
- Design of GFN
- Model generation for differentials
- Explanations for CP solvers

Conclusion and future work

Contributions

- Algebraic analysis of Trivium
- Design of GFN
- Model generation for differentials
- Explanations for CP solvers

Future work

- Graph representations to help modeling.
  - ↪ New properties and strategies
- Model generation to help cryptanalysis.
  - ↪ Faster cryptanalysis
- Explanations to improve solvers
  - ↪ User feedback
  - ↪ Solution proof

Conclusion and future work

Contributions

- Algebraic analysis of Trivium
- Design of GFN
- Model generation for differentials
- Explanations for CP solvers

Future work

- Graph representations to help modeling.
  - ↪ New properties and strategies
- Model generation to help cryptanalysis.
  - ↪ Faster cryptanalysis
- Explanations to improve solvers
  - ↪ User feedback
  - ↪ Solution proof

Constraint Programming for cryptography

✗ not a blind safe option.

✓ adapted when a constraint with a powerful filtering algorithm can be used

Conclusion and future work

Contributions

Algebraic analysis of Trivium

Design of GFN

Model generation for differentials

Explanations for CP solvers

Future work

Graph representations to help modeling.
↪ New properties and strategies

Model generation to help cryptanalysis.
↪ Faster cryptanalysis

Explanations to improve solvers
↪ User feedback
↪ Solution proof

Constraint Programming for cryptography

✗ not a blind safe option.

✓ adapted when a constraint with a powerful filtering algorithm can be used

"Unlike Theseus's boat, half of this thesis is still CP in the end"

Bibliography I

Emanuele Bellini, David Gérault, Juan Grados, Yun Ju Huang, Mohamed Rachidi, Sharwan K. Tiwari, and Rusydi H. Makarim.
CLAASP: a cryptographic library for the automated analysis of symmetric primitives.
IACR Cryptol. ePrint Arch., page 622, 2023.

Alex Biryukov and Ivica Nikolic.
Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, camellia, khazad and others.
In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 322–344. Springer, 2010.

Eli Biham and Adi Shamir.
Differential cryptanalysis of DES-like cryptosystems.
In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.

Eli Biham and Adi Shamir.
Differential Cryptanalysis of the Data Encryption Standard.
Springer, 1993.

Bibliography II

Victor Cauchois, Clément Gomez, and Gaël Thomas.
General diffusion analysis: How to find optimal permutations for Generalized Type-II Feistel Schemes.
IACR Trans. Symmetric Cryptol., 2019(1):264–301, 2019.

Stéphanie Delaune, Patrick Derbez, Paul Huynh, Marine Minier, Victor Mollimard, and Charles Prud'homme.
Efficient Methods to Search for Best Differential Characteristics on SKINNY.
In Kazue Sako and Nils Ole Tippenhauer, editors, *19th International Conference on Applied Cryptography and Network Security, (ACNS'21)*, volume 12727 of *Lecture Notes in Computer Science*, pages 184–207. Springer, 2021.

Patrick Derbez, Pierre-Alain Fouque, Baptiste Lambin, and Victor Mollimard.
Efficient search for optimal diffusion layers of Generalized Feistel Networks.
IACR Trans. Symmetric Cryptol., 2019(2):218–240, 2019.

Horst Feistel.
Cryptography and computer privacy.
Scientific american, 228(5):15–23, 1973.

Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin.
Structural evaluation of AES and chosen-key distinguisher of 9-round AES-128.
In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 183–203. Springer, 2013.

Bibliography III

David Gérault.
Security analysis of contactless communication protocols. (Analyse de sécurité des protocoles de communication sans contact).
PhD thesis, University of Clermont Auvergne, Clermont-Ferrand, France, 2018.

David Gérault, Pascal Lafourcade, Marine Minier, and Christine Solnon.
Computing AES related-key differential characteristics with constraint programming.
Artif. Intell., 278, 2020.

Lars R. Knudsen.
Truncated and higher order differentials.
In Bart Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 1994.

Stefan Kölbl.
CryptoSMT: An easy to use tool for cryptanalysis of symmetric primitives (2015).
URL: https://github. com/kste/cryptosmt.

Luc Libralesso, François Delobel, Pascal Lafourcade, and Christine Solnon.
Automatic generation of declarative models for differential cryptanalysis.
In Laurent D. Michel, editor, *27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021*, volume 210 of *LIPIcs*, pages 40:1–40:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

Bibliography IV

Gaëtan Leurent.

Analysis of differential attacks in ARX constructions.

In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 226–243. Springer, 2012.

Loïc Rouquette, David Gérault, Marine Minier, and Christine Solnon.

And Rijndael?: Automatic related-key differential analysis of Rijndael.

In Lejla Batina and Joan Daemen, editors, *Progress in Cryptology - AFRICACRYPT 2022: 13th International Conference on Cryptology in Africa, AFRICACRYPT 2022, Fes, Morocco, July 18-20, 2022, Proceedings*, Lecture Notes in Computer Science, pages 150–175. Springer Nature Switzerland, 2022.

Adrián Ranea and Vincent Rijmen.

Characteristic automated search of cryptographic algorithms for distinguishing attacks (CASCADA).

IET Inf. Secur., 16(6):470–481, 2022.

Tomoyasu Suzaki and Kazuhiko Minematsu.

Improving the Generalized Feistel.

In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, volume 6147 of *Lecture Notes in Computer Science*, pages 19–39. Springer, 2010.

Bibliography V

JL Smith.
The design of lucifer, a cryptographic device for data communication.
Technical report, IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 1971.

Kosei Sakamoto, Kazuhiko Minematsu, Nao Shibata, Maki Shigeri, Hiroyasu Kubo, Yuki Funabiki, and Takanori Isobe.
Security of related-key differential attacks on TWINE, revisited.
IEICE Trans. Fundam. Electron. Commun. Comput. Sci., 103-A(1):212–214, 2020.

Je Sen Teh and Alex Biryukov.
Differential cryptanalysis of WARP.
J. Inf. Secur. Appl., 70:103316, 2022.

Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai.
On the construction of block ciphers provably secure and not relying on any unproved hypotheses.
In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 461–480. Springer, 1989.