Cryptanalysis of SP Networks with Partial Non-Linear Layers

Eurocrypt 2015

Achiya Bar-On[1], Itai Dinur[2], Orr Dunkelman[3,5], Nathan Keller[1,5], Virginie Lallemand[4] and Boaz Tsaban[1]

[1] Department of Mathematics, Bar-Ilan University, Israel
[2] Département d'Informatique, École Normale Supérieure, Paris, France
[3] Computer Science Department, University of Haifa, Israel
[4] Inria, France
[5] Computer Science department, The Weizmann Institute, Rehovot, Israel

April 27, 2015

Introduction

Lightweight Symmetric Cryptography

- **Secure**
- Fast
- Compact
- Low memory
- Low power

Lightweight Symmetric Cryptography

- **Secure**
- Fast
- Compact
- Low memory
- Low power

plus...

Lightweight Symmetric Cryptography

- **Secure**
- Fast
- Compact
- Low memory
- Low power

plus...

- Easy to protect against **Side-Channel Attacks**

Lightweight Symmetric Cryptography

- **Secure**
- Fast
- Compact
- Low memory
- Low power

plus...

- Easy to protect against **Side-Channel Attacks**

Two design strategies:

- **Design** a cipher and **then** an optimized **protection** for this algorithm (e.g.: AES)
- **Design** a cipher **with that issue in mind** (e.g.: Picaro, Zorro, LS-designs)

Masking (Secret Sharing)

One of the most frequently considered **solutions to mitigate Side-Channel Attacks**

Order-d masking:

- **Randomizes** the data processed such that the observation of d intermediate values during encryption **does not provide information** about sensitive variables
- Induces **performance overhead** for **non-linear operations**

Masking (Secret Sharing)

One of the most frequently considered **solutions to mitigate Side-Channel Attacks**

Order-d masking:

- **Randomizes** the data processed such that the observation of d intermediate values during encryption **does not provide information** about sensitive variables
- Induces **performance overhead** for **non-linear operations**

To limit performance overhead:

- Use an easy-to-mask Sbox
- Limit the number of Sbox applications

Zorro

Benoît Gérard, Vincent Grosso, María Naya-Plasencia and François-Xavier Standaert
Block Ciphers that are Easier to Mask: How Far Can we Go?,
CHES 2013.

Zorro

Benoît Gérard, Vincent Grosso, María Naya-Plasencia and François-Xavier Standaert
Block Ciphers that are Easier to Mask: How Far Can we Go?,
CHES 2013.

Tweak the AES to allow **efficient masking**

Zorro

Benoît Gérard, Vincent Grosso, María Naya-Plasencia and François-Xavier Standaert
Block Ciphers that are Easier to Mask: How Far Can we Go?,
CHES 2013.

Tweak the AES to allow **efficient masking**

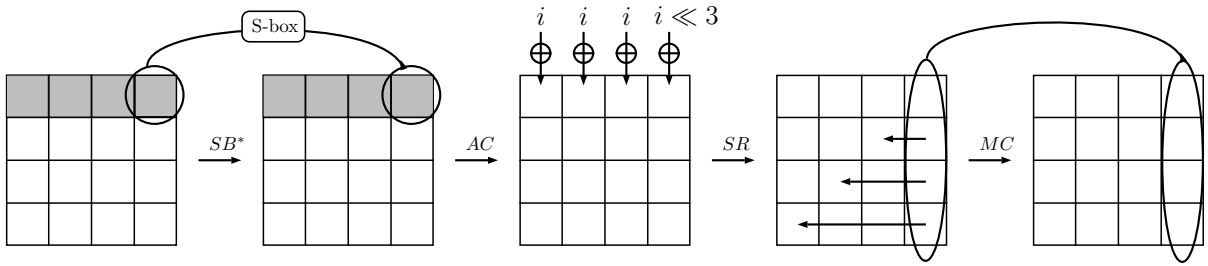
- same SR and MC

Zorro

Benoît Gérard, Vincent Grosso, María Naya-Plasencia and François-Xavier Standaert
Block Ciphers that are Easier to Mask: How Far Can we Go?,
CHES 2013.

Tweak the AES to allow **efficient masking**

- same SR and MC
- **Reduced number of Sbox applications per round:**
  $t = 4$ **Sboxes** (different from the AES Sbox)
  Only on the first row

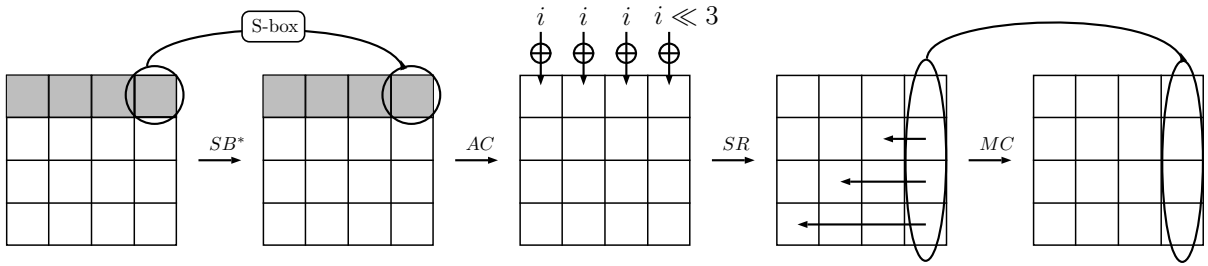


Zorro

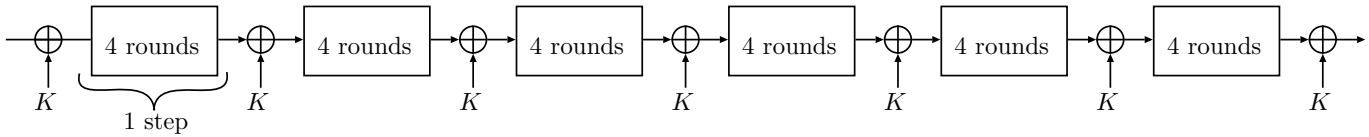
Benoît Gérard, Vincent Grosso, María Naya-Plasencia and François-Xavier Standaert
Block Ciphers that are Easier to Mask: How Far Can we Go?,
CHES 2013.

Tweak the AES to allow **efficient masking**

- same SR and MC
- **Reduced number of Sbox applications per round:**
  $t = 4$ **Sboxes** (different from the AES Sbox)
  Only on the first row



- **24 rounds**, XOR of the master key every 4 rounds



Analysis of Zorro

Given Zorro's unconventional design:

- Previous tools (e.g. the Wide Trail Strategy) **do not apply**
- The authors used **heuristic arguments**...

Analysis of Zorro

Given Zorro's unconventional design:

- Previous tools (e.g. the Wide Trail Strategy) **do not apply**
- The authors used **heuristic arguments**...

...that turned out to be **insufficient**

Analysis of Zorro

Given Zorro's unconventional design:

- Previous tools (e.g. the Wide Trail Strategy) **do not apply**
- The authors used **heuristic arguments**...

...that turned out to be **insufficient**

$\rightarrow$ 2 attacks on full Zorro were devised soon after the publication, using differential and linear cryptanalysis

Yanfeng Wang, Wenling Wu, Zhiyuan Guo and Xiaoli Yu
Differential Cryptanalysis and Linear Distinguisher of Full-Round Zorro,
ACNS 2014.

Shahram Rasoolzadeh, Zahra Ahmadian, Mahmood Salmasizadeh and Mohammad Reza Aref
Total Break of Zorro using Linear and Differential Attacks,
Cryptology ePrint Archive, Report 2014/220.

Analysis of Zorro

Given Zorro's unconventional design:

- Previous tools (e.g. the Wide Trail Strategy) **do not apply**
- The authors used **heuristic arguments**...

...that turned out to be **insufficient**

$\rightarrow$ 2 attacks on full Zorro were devised soon after the publication, using differential and linear cryptanalysis

However, those attacks exploit specific properties of the linear mappings of Zorro and do not apply if we slightly tweak it

Summary

- We propose **generic techniques for analysis of SPN with partial non-linear layers** (PSP networks) against differential and linear attacks:
  - Characteristic search tool
  - Key-recovery algorithm

Summary

- We propose **generic techniques for analysis of SPN with partial non-linear layers** (PSP networks) against differential and linear attacks:
  - Characteristic search tool
  - Key-recovery algorithm
- These techniques can be used to **break** such ciphers
- Or to **prove their security** against basic differential and linear cryptanalysis

High Probability Characteristic Search Algorithm

Existing Characteristic Search Tools

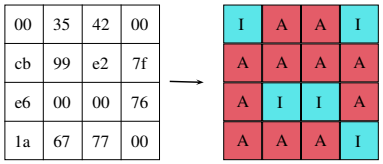
Search for characteristics with high probability or small number of active Sboxes (with non-zero difference)

- Use a compact representation of the state to perform efficient search: define a **pattern** which is a function that specifies whether each byte is active or non-active

Existing Characteristic Search Tools

Search for characteristics with high probability or small number of active Sboxes (with non-zero difference)

- Use a compact representation of the state to perform efficient search: define a **pattern** which is a function that specifies whether each byte is active or non-active



Existing Characteristic Search Tools

Search for characteristics with high probability or small number of active Sboxes (with non-zero difference)

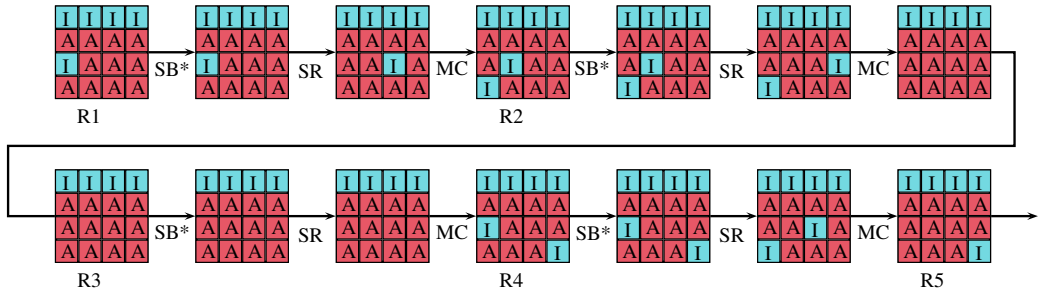
- Use a compact representation of the state to perform efficient search: define a **pattern** which is a function that specifies whether each byte is active or non-active
- Only check local consistency of patterns (verifying that each transition is possible)

Existing Characteristic Search Tools

Search for characteristics with high probability or small number of active Sboxes (with non-zero difference)

- Use a compact representation of the state to perform efficient search: define a **pattern** which is a function that specifies whether each byte is active or non-active
- Only check local consistency of patterns (verifying that each transition is possible)

For 5-round Zorro, they output a huge number of patterns which appear to have probability 1



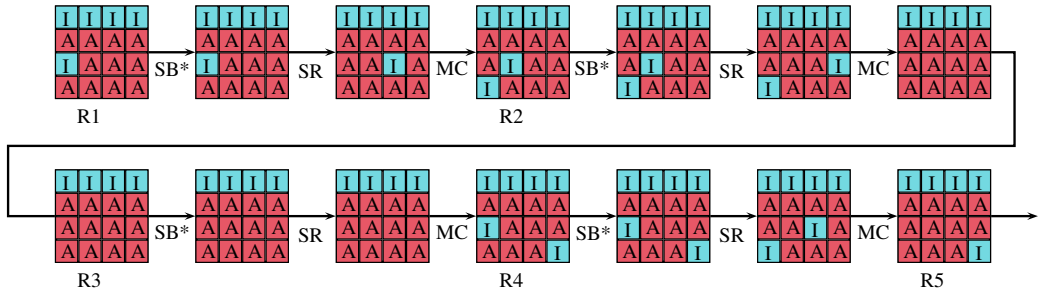
However, non of these patterns is possible for 5-round Zorro, since no characteristic follows them!

Existing Characteristic Search Tools

Search for characteristics with high probability or small number of active Sboxes (with non-zero difference)

- Use a compact representation of the state to perform efficient search: define a **pattern** which is a function that specifies whether each byte is active or non-active
- Only check local consistency of patterns (verifying that each transition is possible)

For 5-round Zorro, they output a huge number of patterns which appear to have probability 1

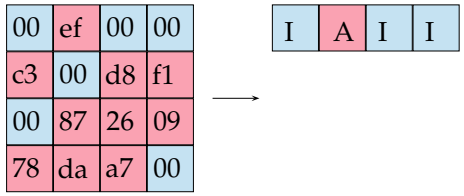


However, non of these patterns is possible for 5-round Zorro, since no characteristic follows them!

→ **not applicable to Zorro and SPN with partial non-linear layers in general**

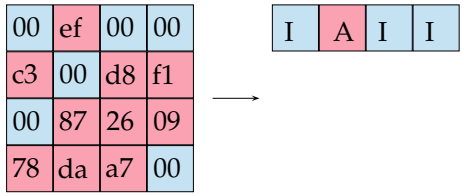
Considerations in Devising our Tool: Our Approach

1. We use patterns for the Sbox positions



Considerations in Devising our Tool: Our Approach

1. We use patterns for the Sbox positions



2. Since local consistency check for pattens in insufficient, we keep track of a **global** system

High Probability Differential Characteristic Search Algorithm

Given **r** rounds of a PSP Network with **t** Sboxes per round, Search for high probability characteristics with a maximum of **a** active Sboxes

high probability $\rightarrow$ small a

Algorithm Overview

- Iterate over **all the possible patterns**
- For each one, decide whether it is possible
- If it is, return the **possible characteristics**

High Probability Differential Characteristic Search Algorithm

Given **r** rounds of a PSP Network with **t** Sboxes per round, Search for high probability characteristics with a maximum of **a** active Sboxes

high probability $\rightarrow$ small a

Algorithm Overview

- Iterate over **all the possible patterns**
- For each one, decide whether it is possible
- If it is, return the **possible characteristics**

Exploited properties:

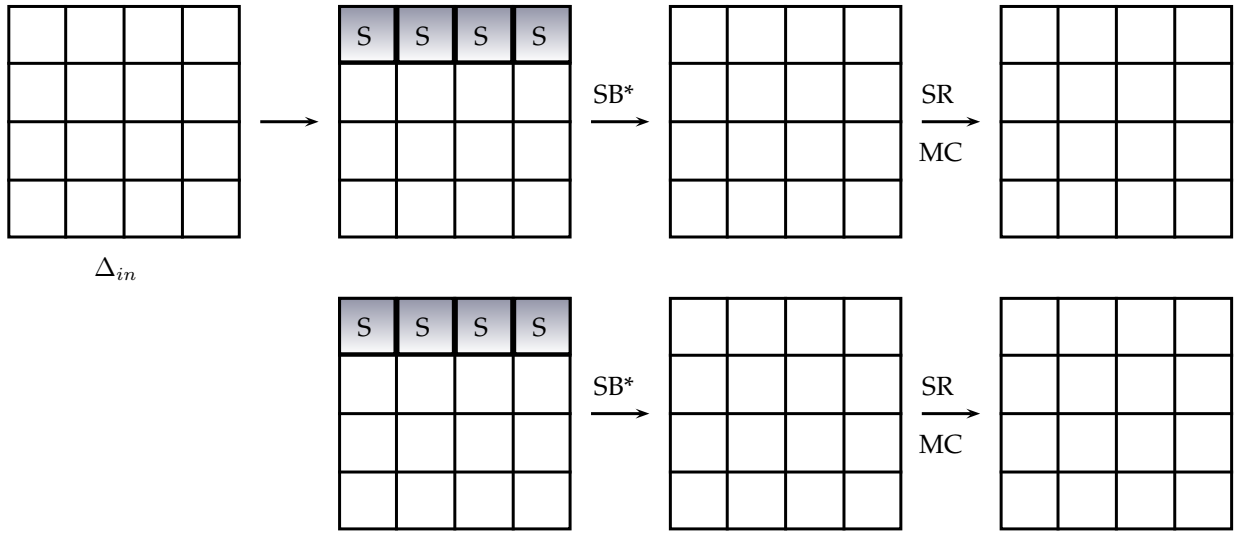
- The number of possible patterns for small a is **small**: $\binom{tr}{\leq a}$
  (equal to the number of possibilities for fixing the positions of the **a** active Sboxes)
- Each pattern can be analyzed **efficiently**
  All the characteristics described by a pattern reside in a restricted **linear subspace**, which we can compute by **linear algebra**

High Probability Differential Characteristic Search Algorithm

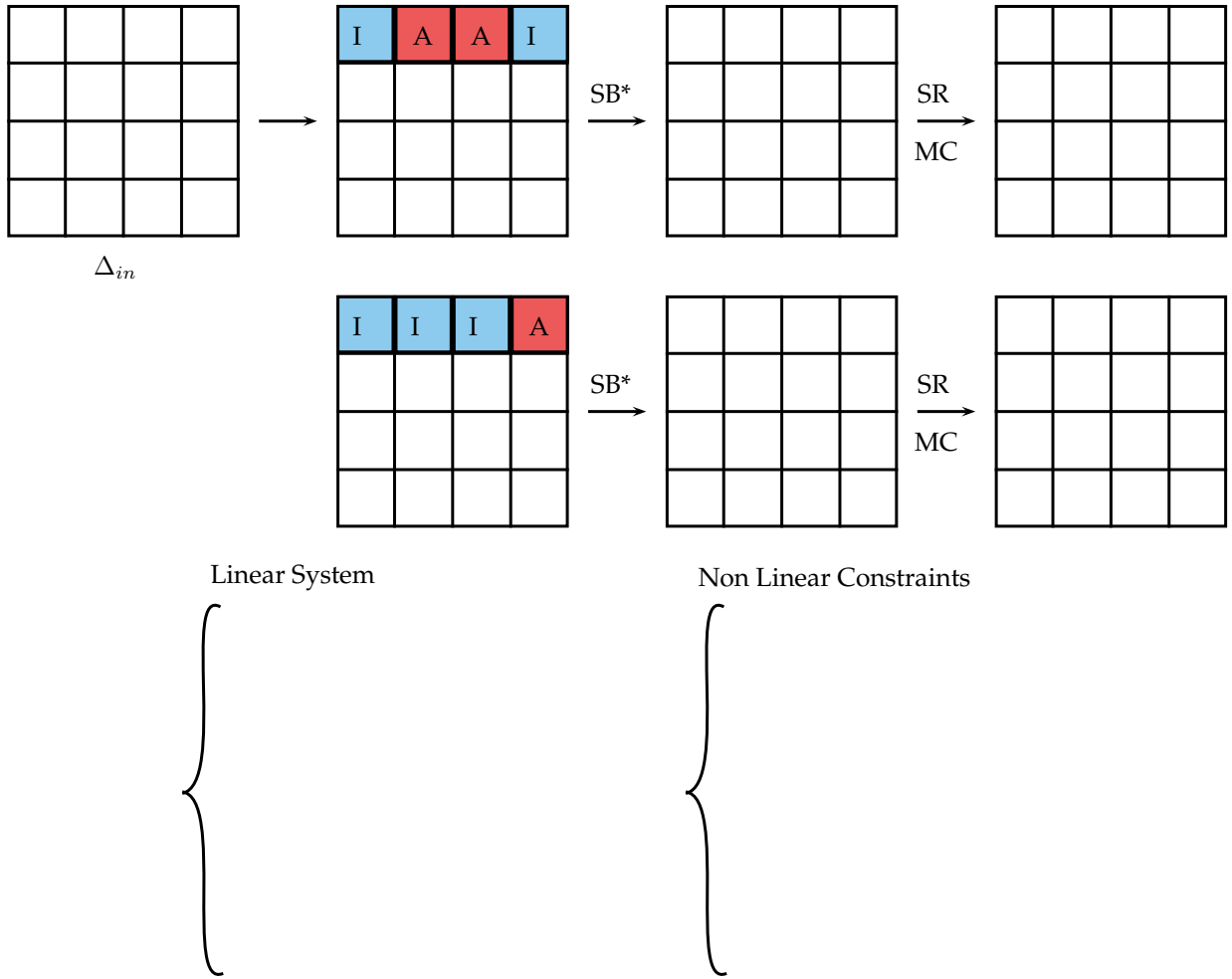
Given a fixed pattern, compute the linear subspace containing all the characteristics described by it:

- The **difference at the plaintext** is initially unknown and is represented using 128 binary variables
- Compute the **chain of intermediate differences** and maintain a system of **linear equations** describing the constraints due to the 2 following rules:
  - An **inactive** Sbox imposes **8 linear equations** on the variables
  - The output difference of an **active** Sbox is **linearized** by adding 8 new variables

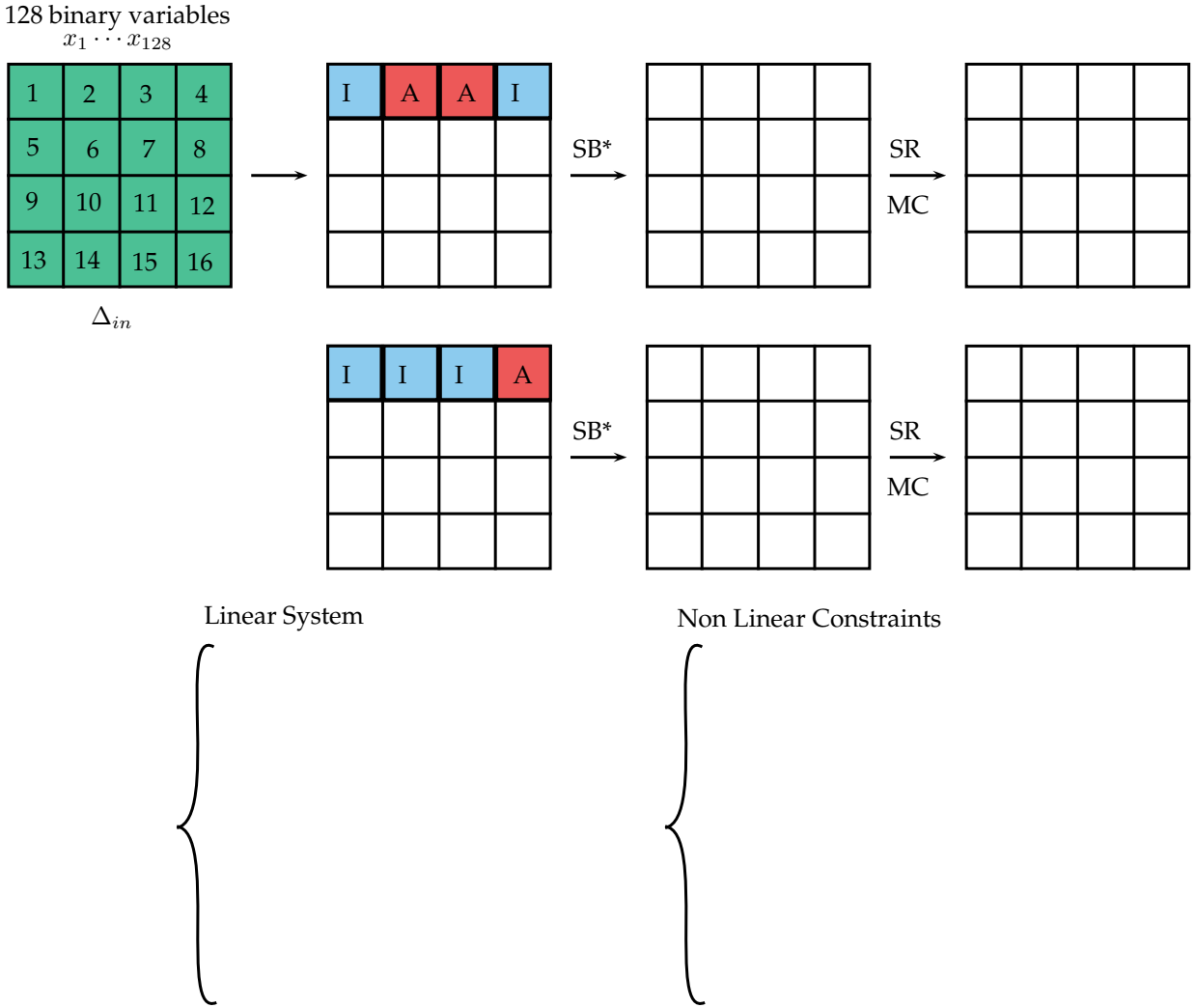
High Probability Differential Characteristic Search Algorithm



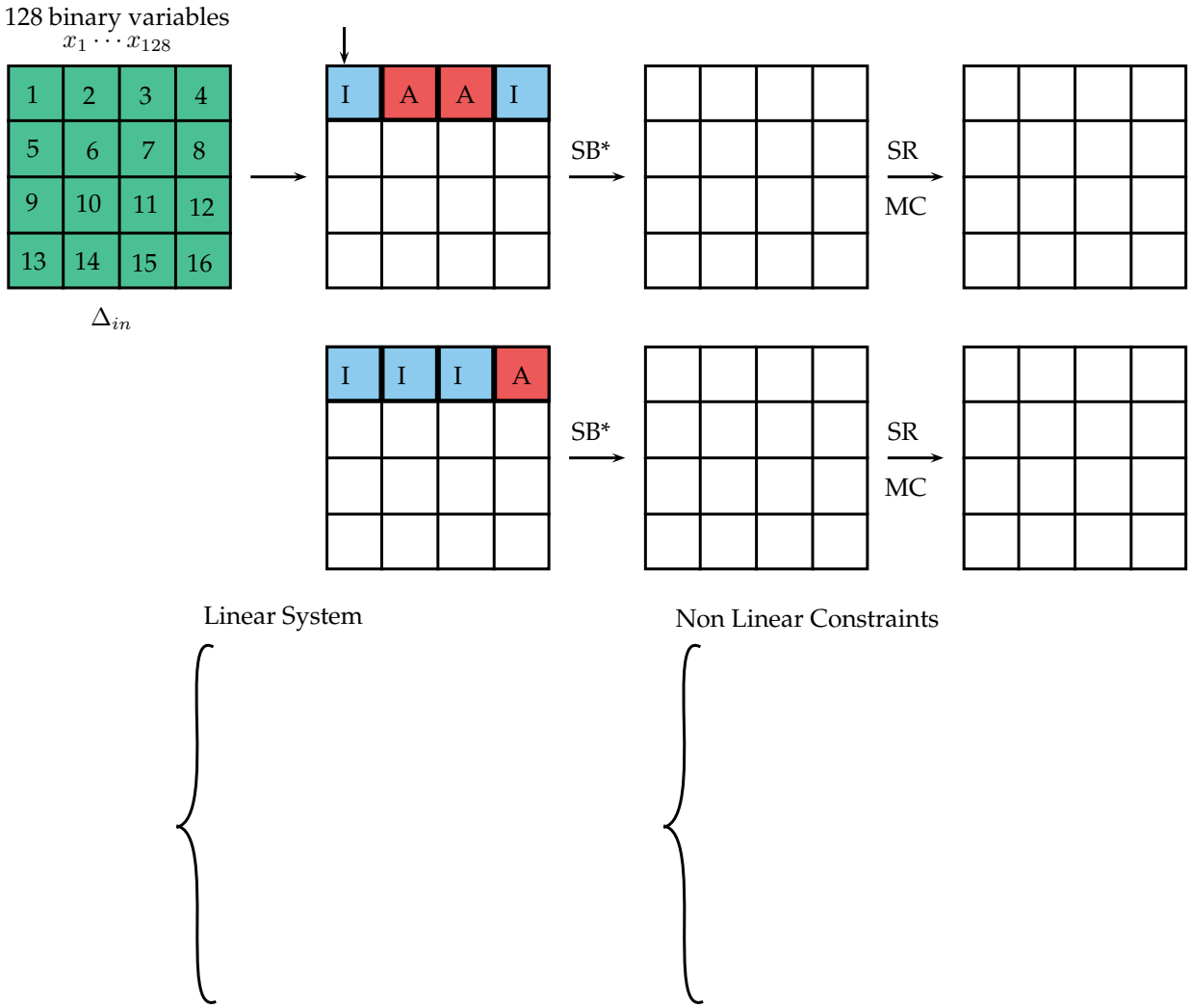
High Probability Differential Characteristic Search Algorithm



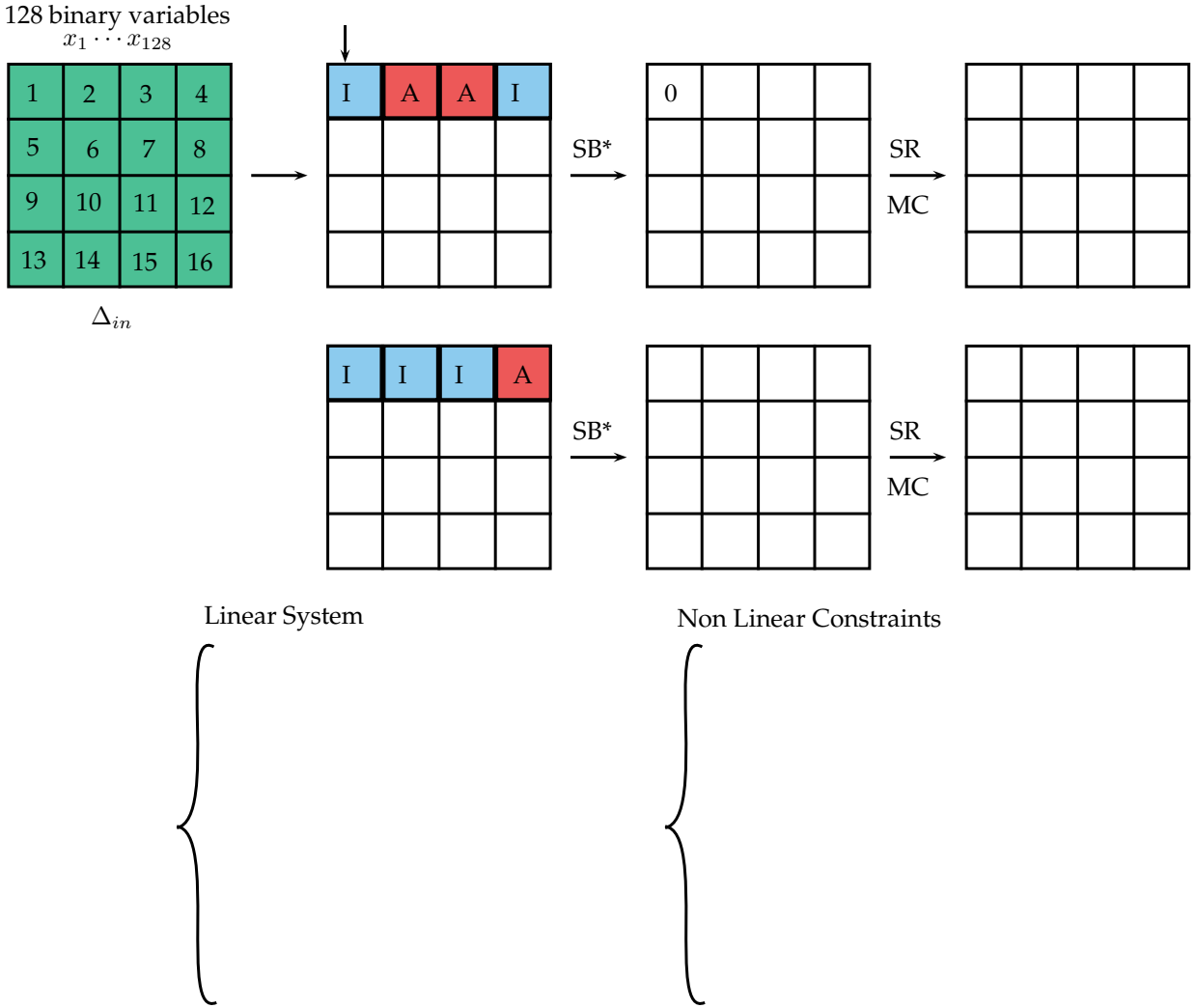
High Probability Differential Characteristic Search Algorithm



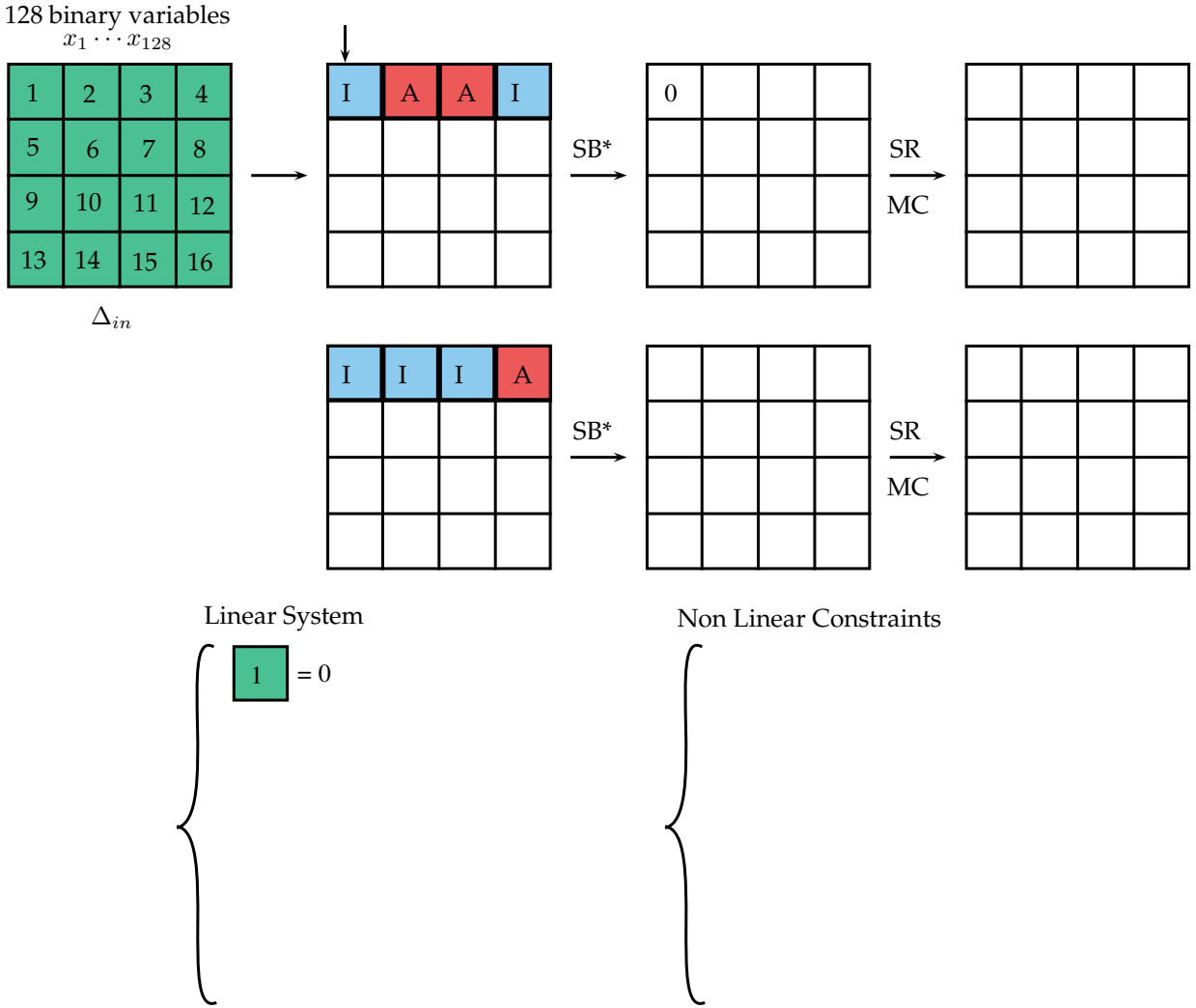
High Probability Differential Characteristic Search Algorithm



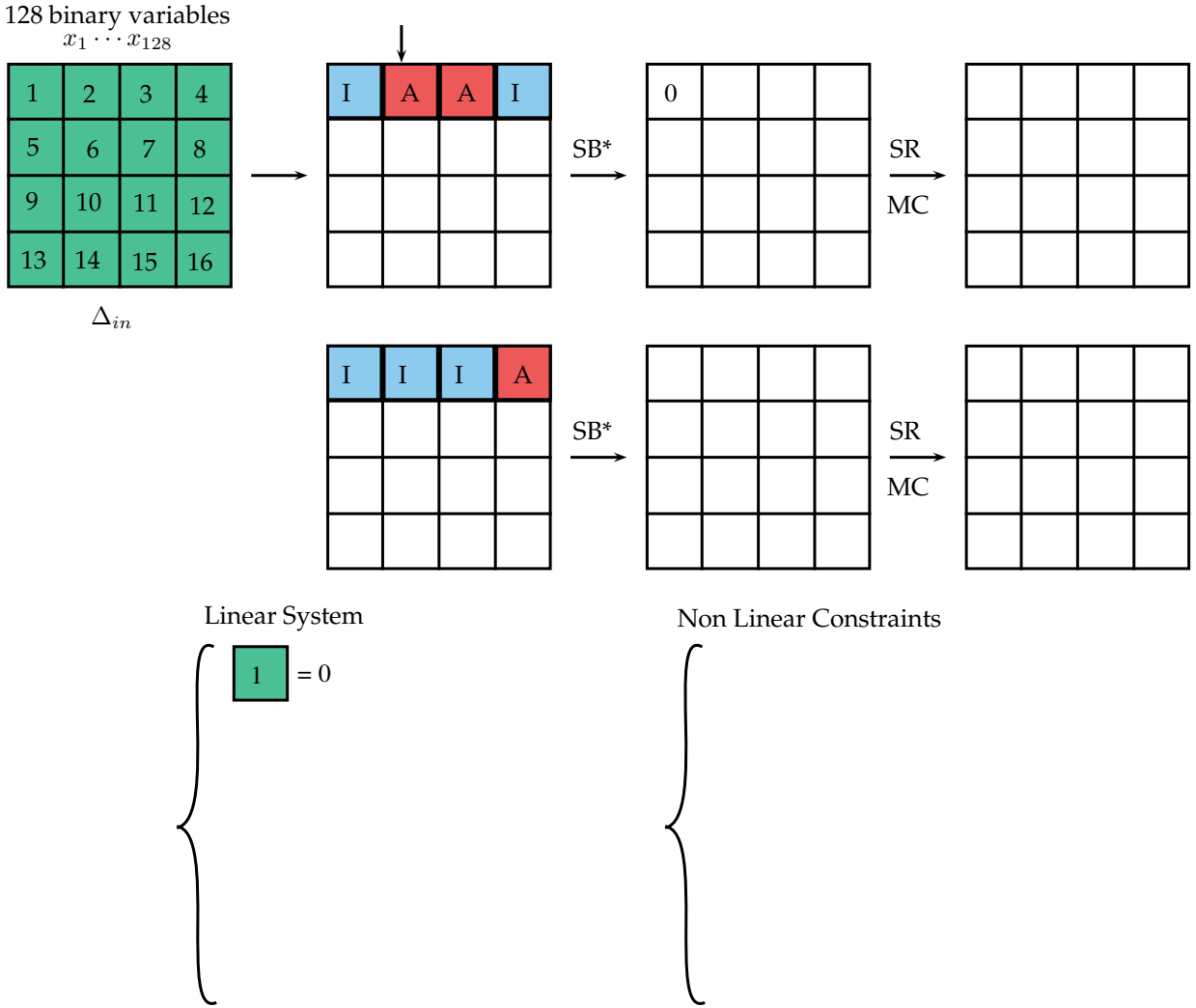
High Probability Differential Characteristic Search Algorithm



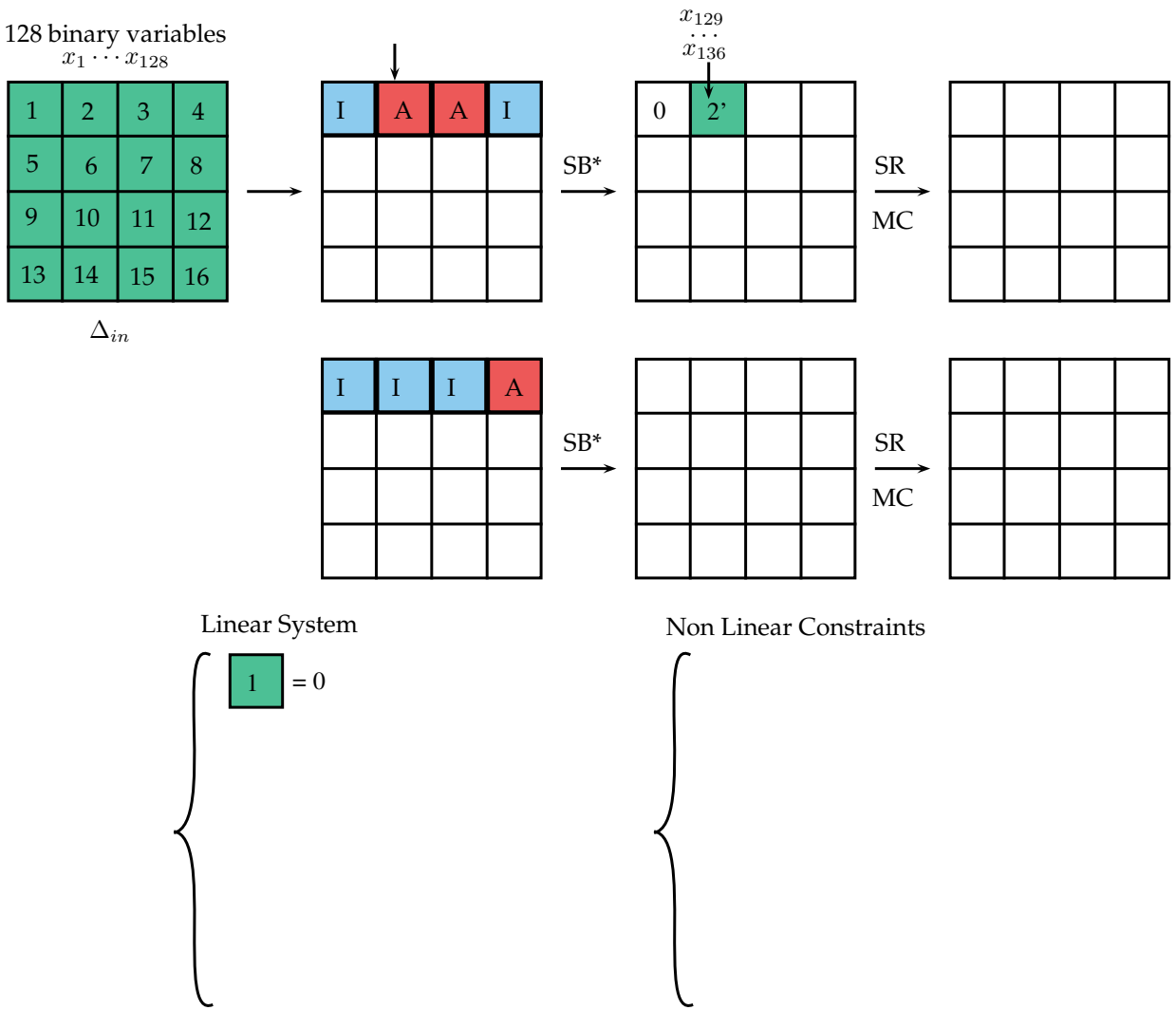
High Probability Differential Characteristic Search Algorithm



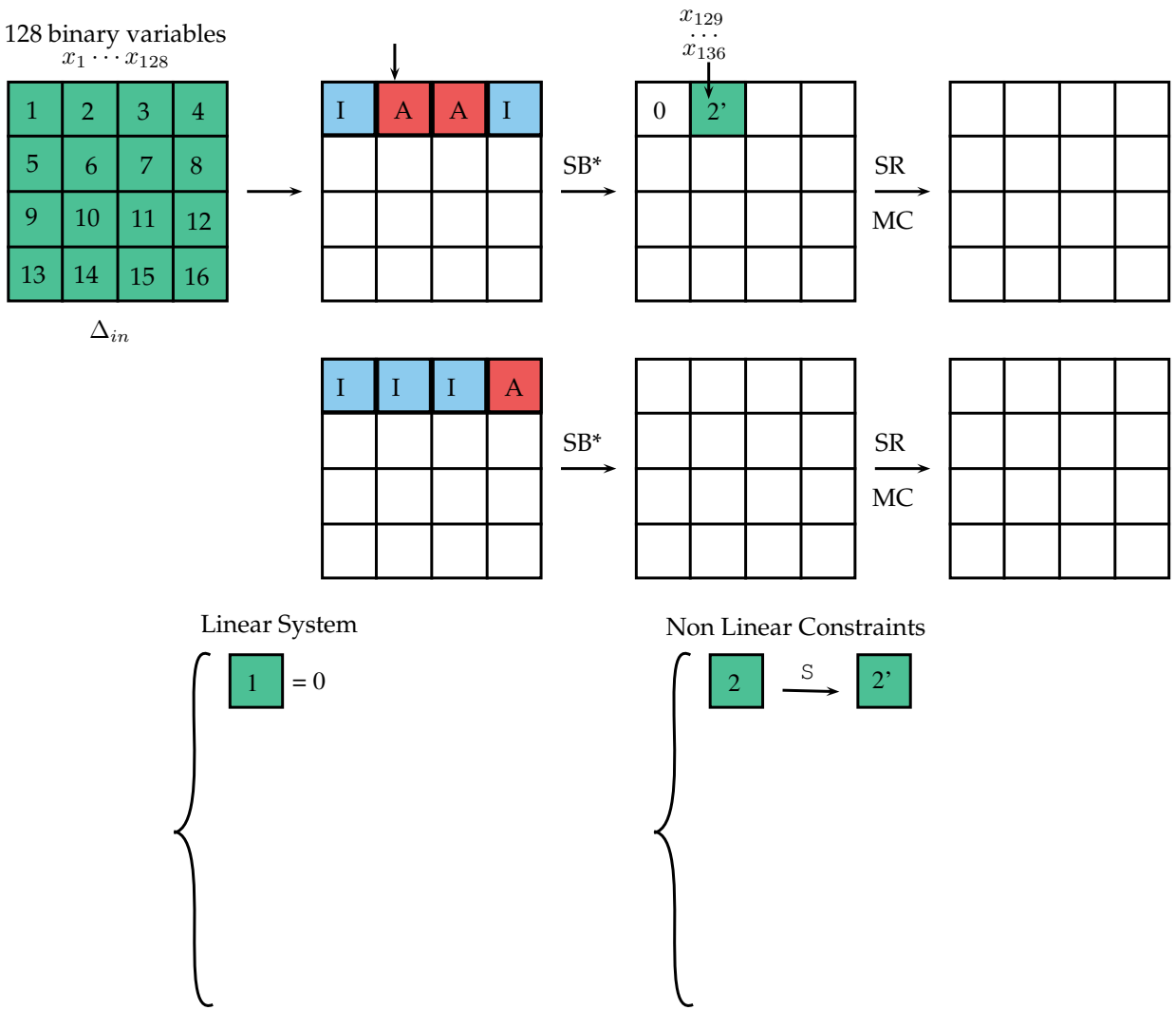
High Probability Differential Characteristic Search Algorithm



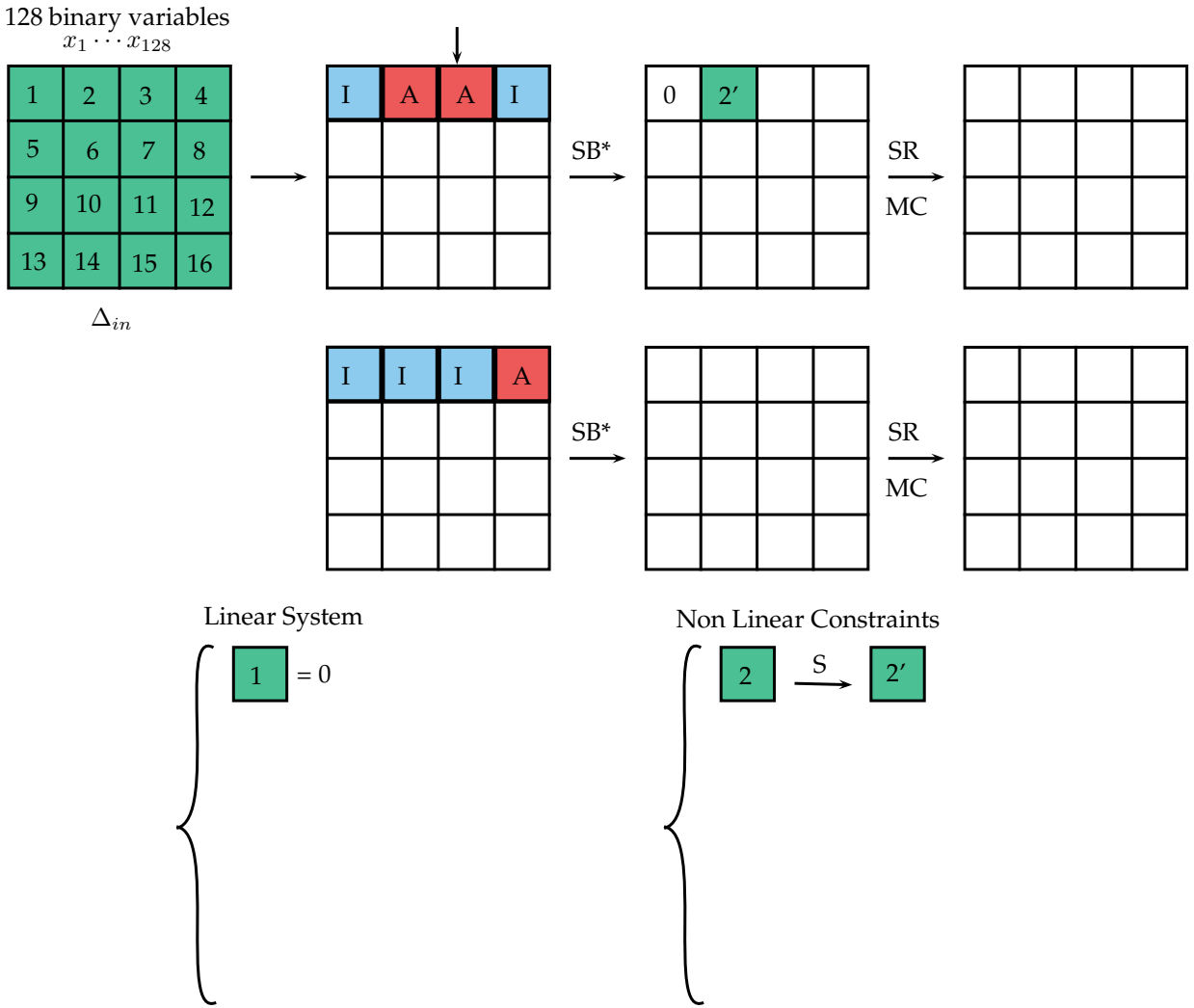
High Probability Differential Characteristic Search Algorithm



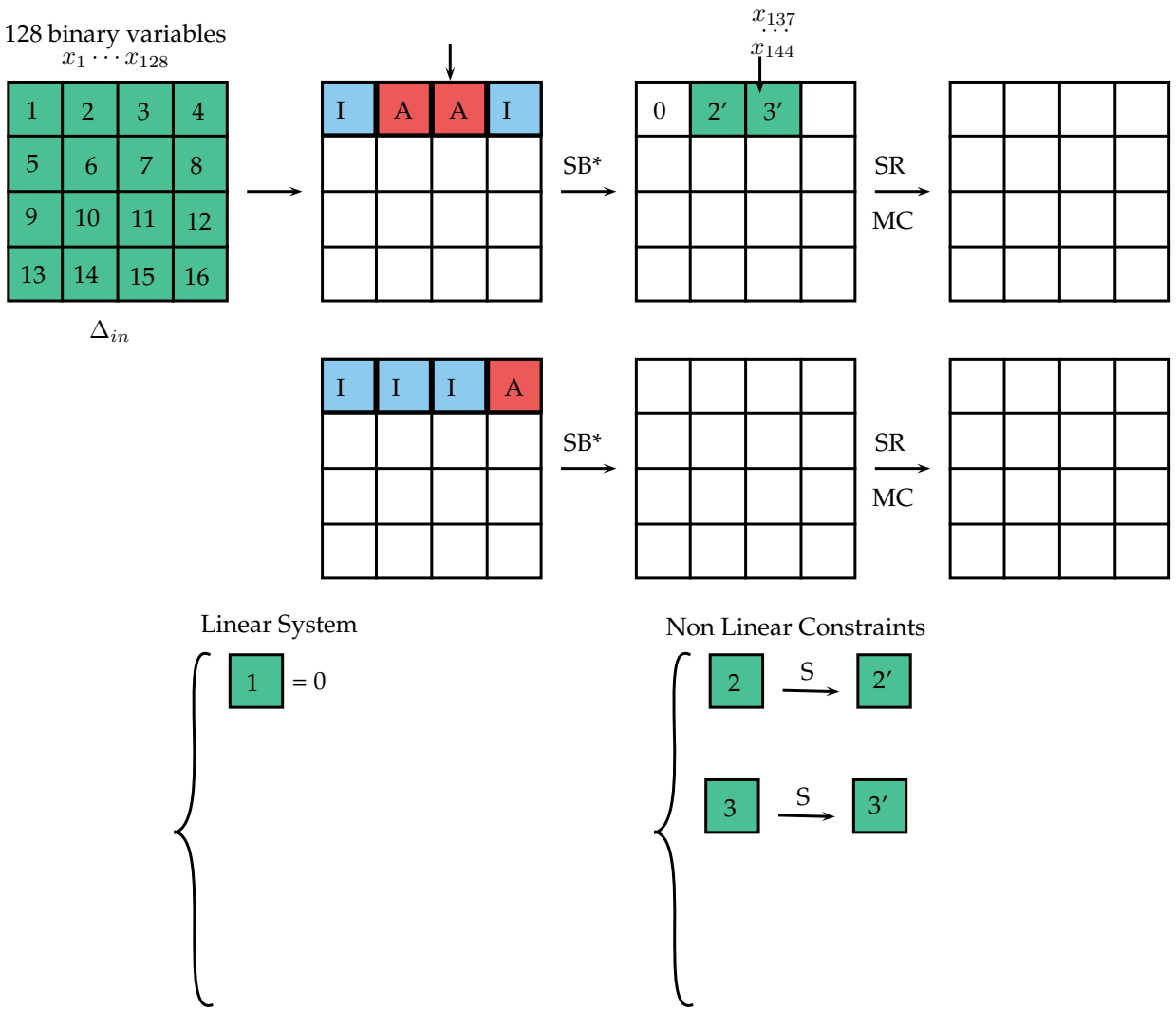
High Probability Differential Characteristic Search Algorithm



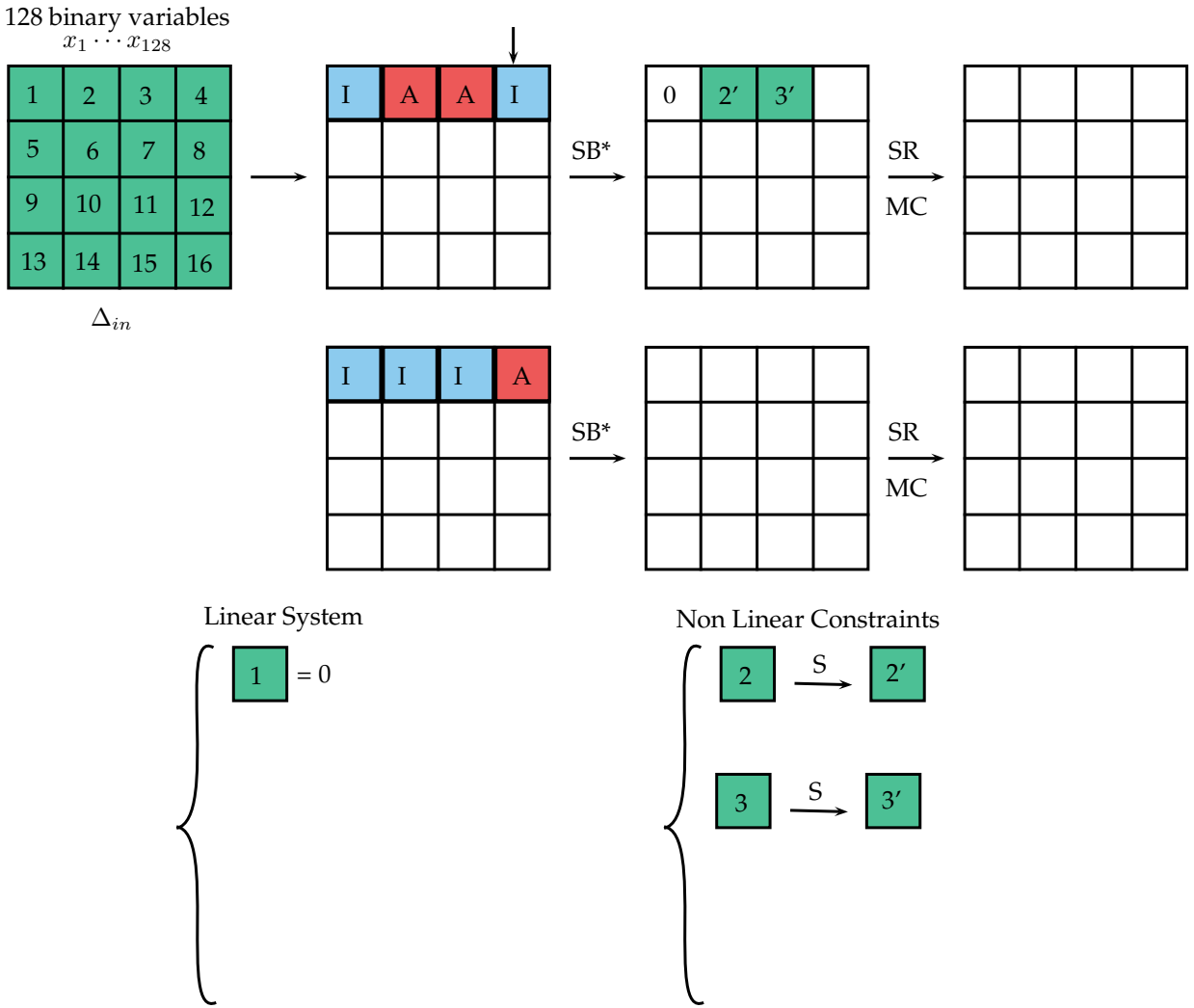
High Probability Differential Characteristic Search Algorithm



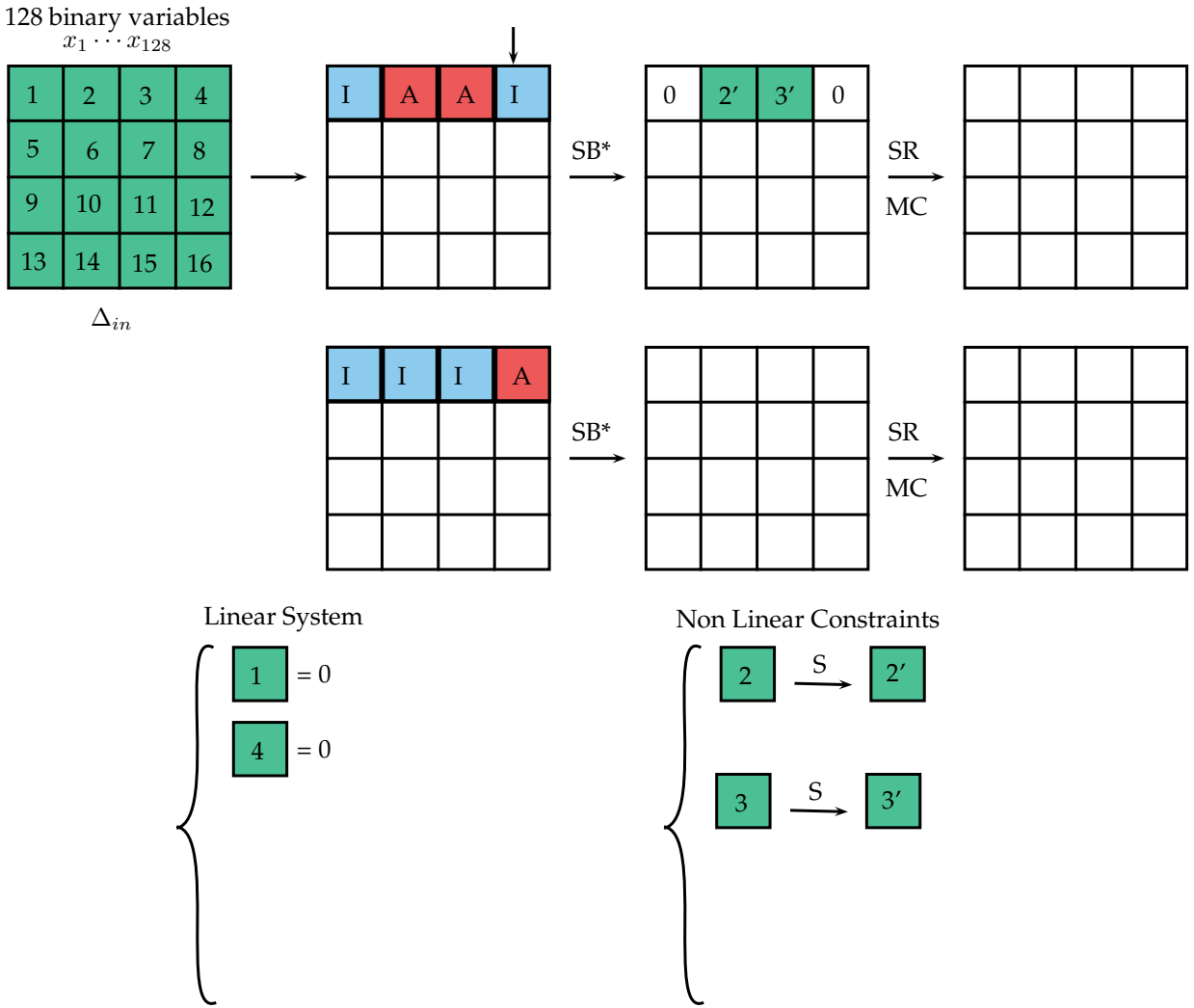
High Probability Differential Characteristic Search Algorithm



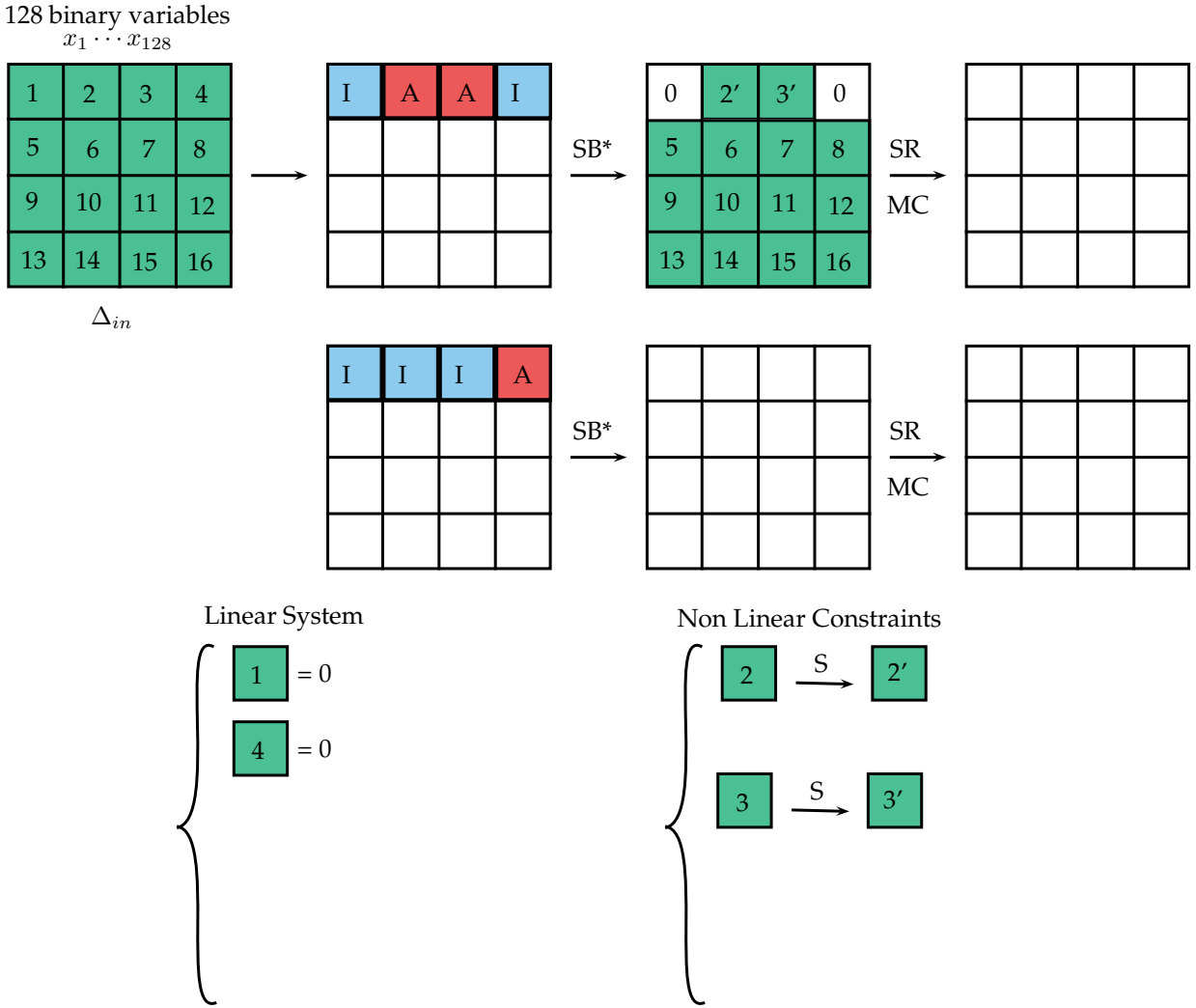
High Probability Differential Characteristic Search Algorithm



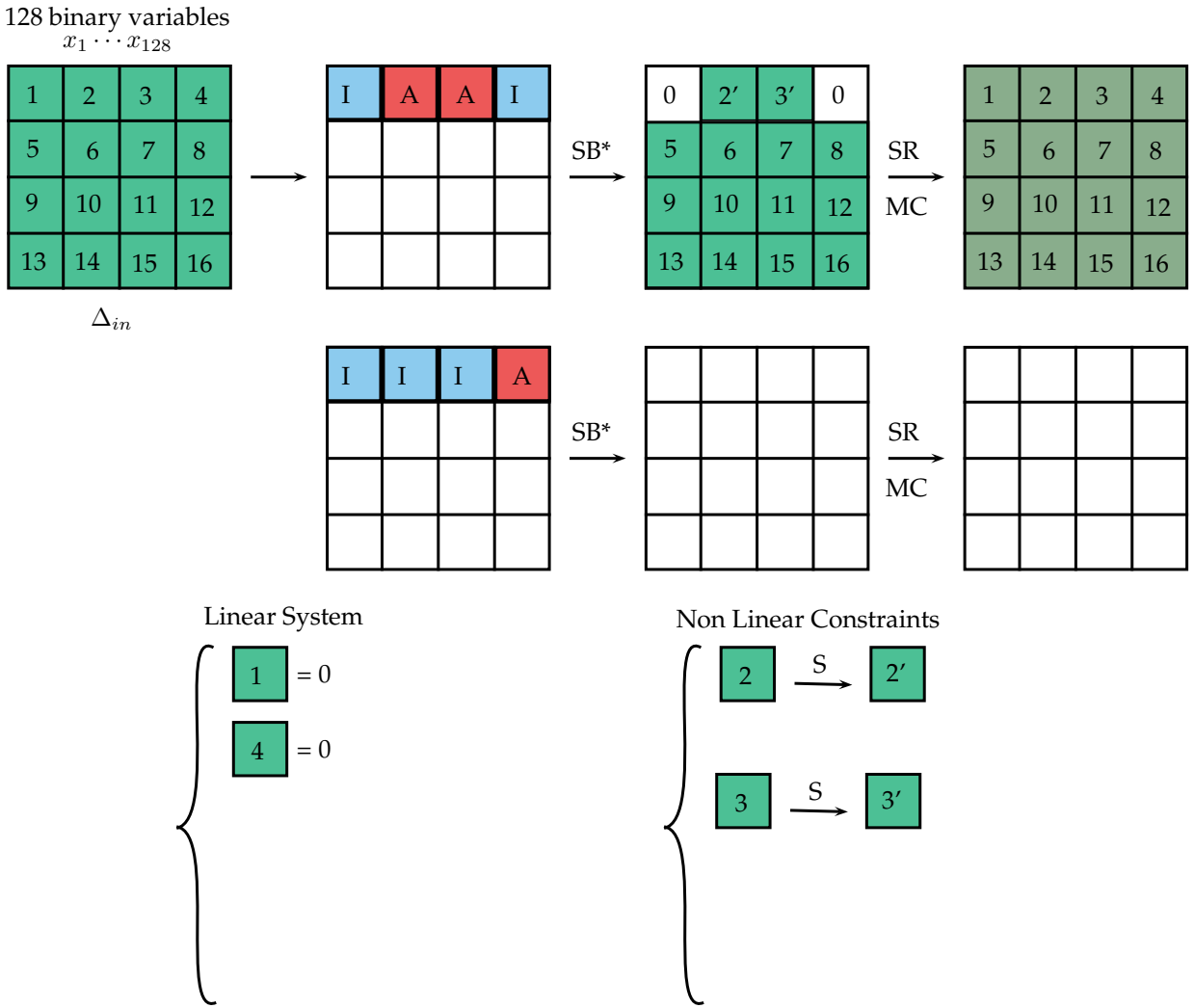
High Probability Differential Characteristic Search Algorithm



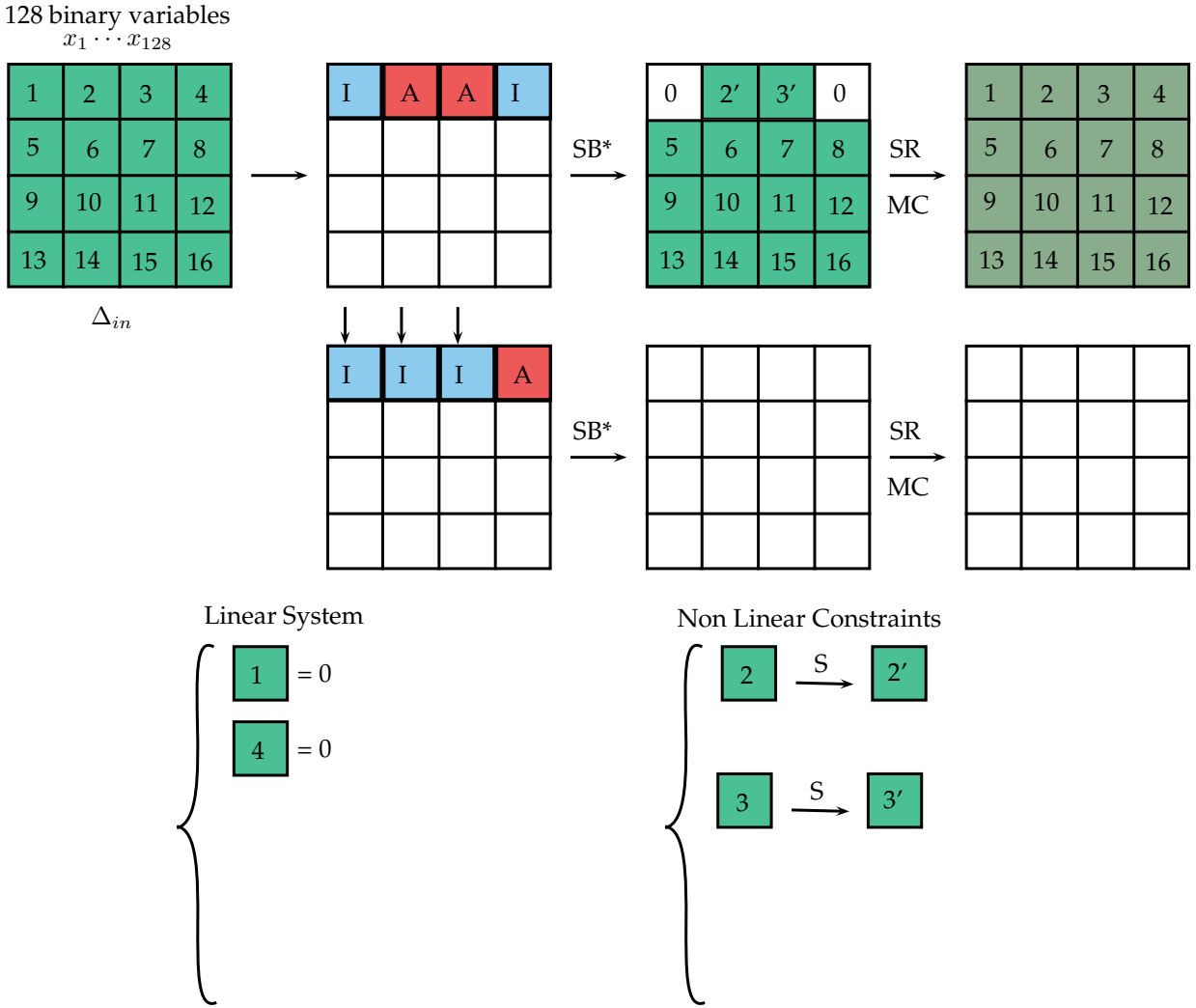
High Probability Differential Characteristic Search Algorithm



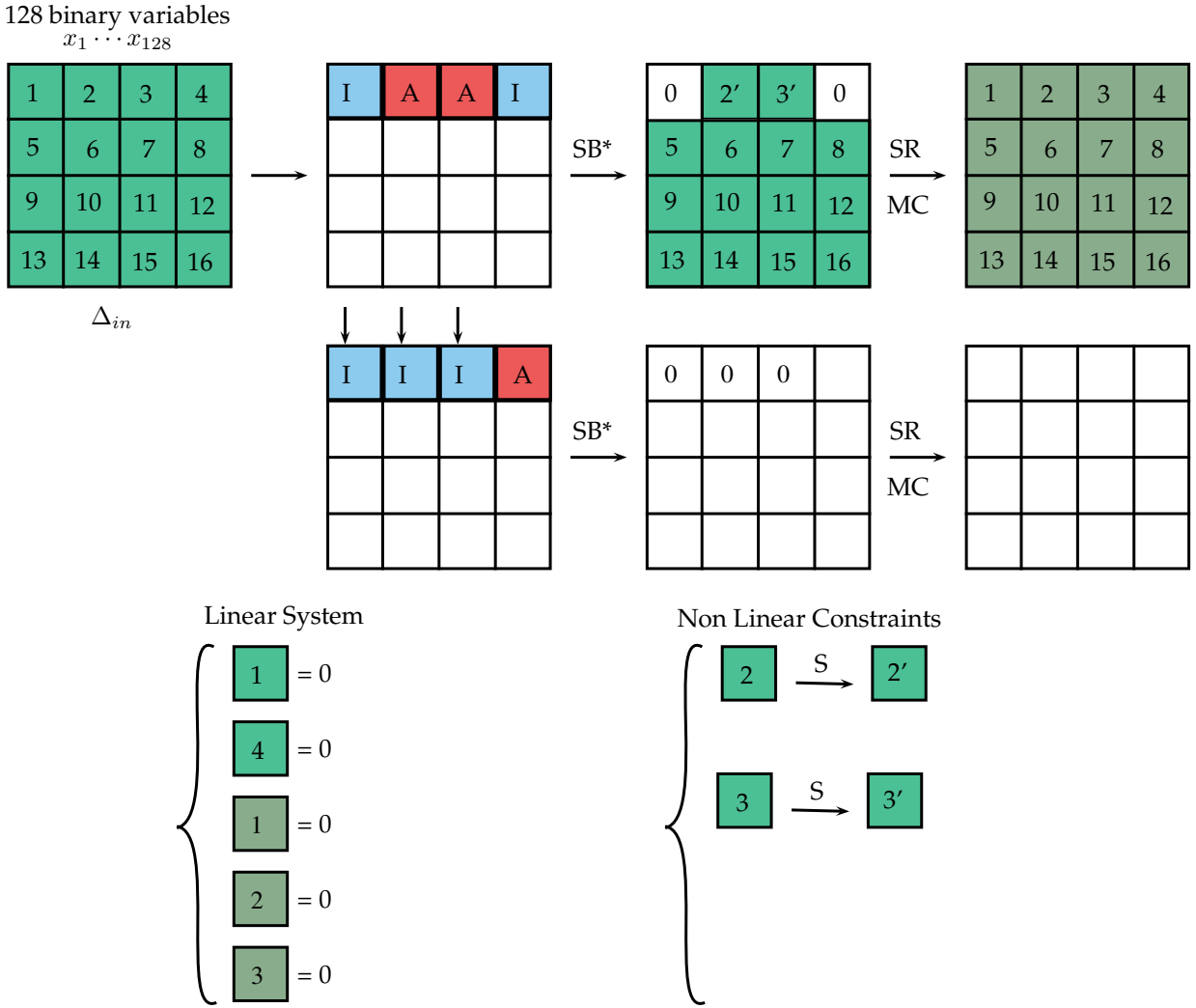
High Probability Differential Characteristic Search Algorithm



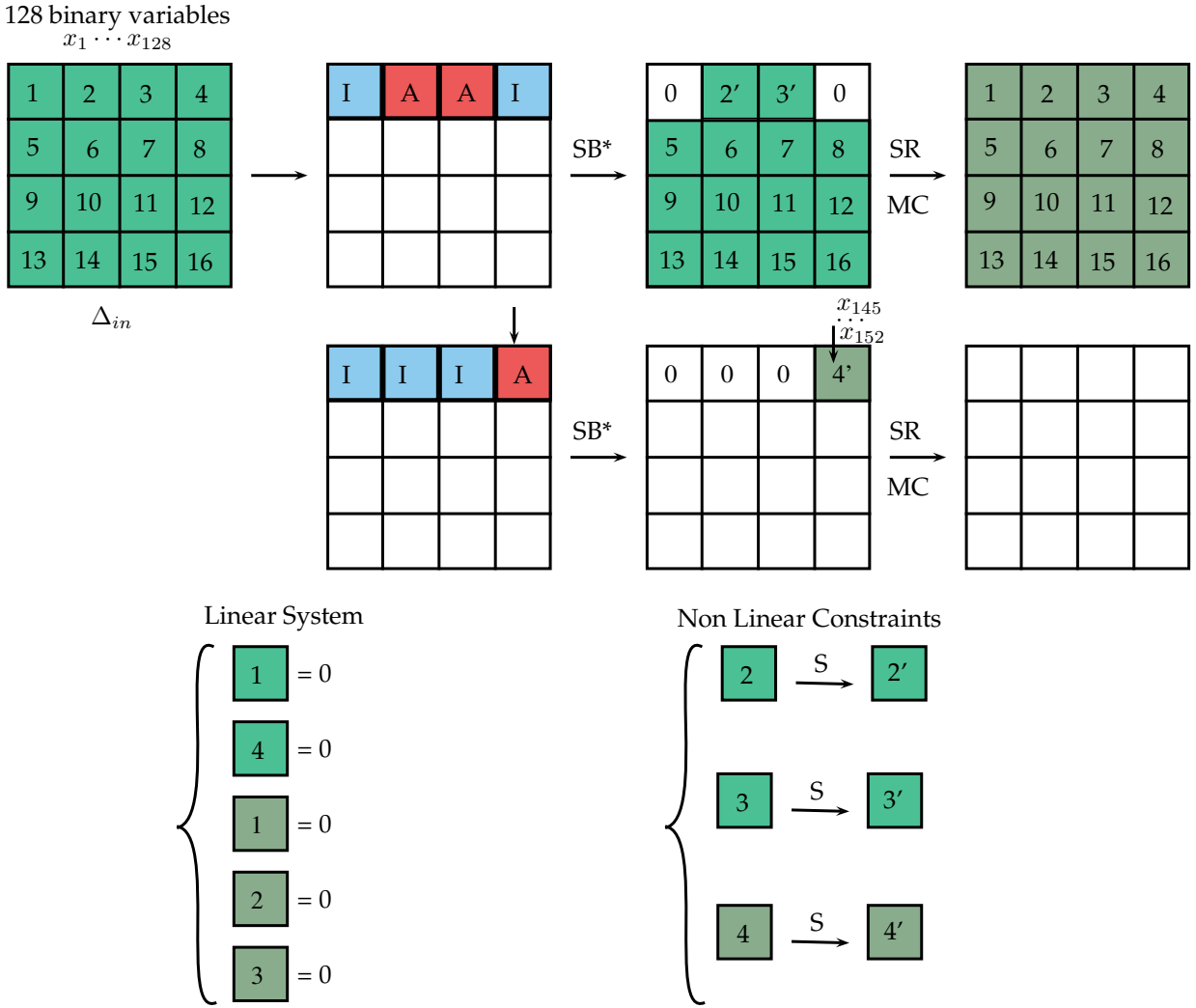
High Probability Differential Characteristic Search Algorithm



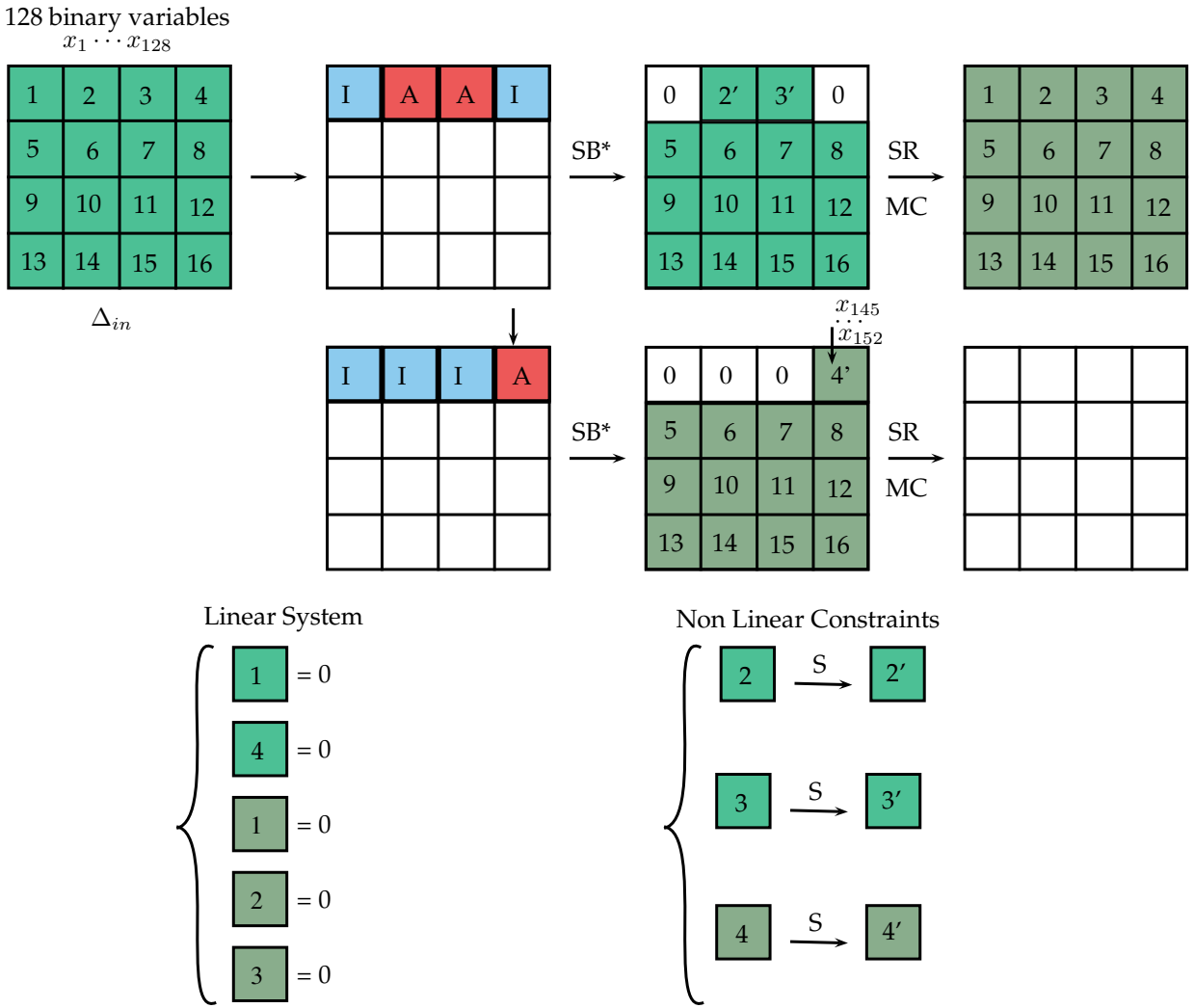
High Probability Differential Characteristic Search Algorithm



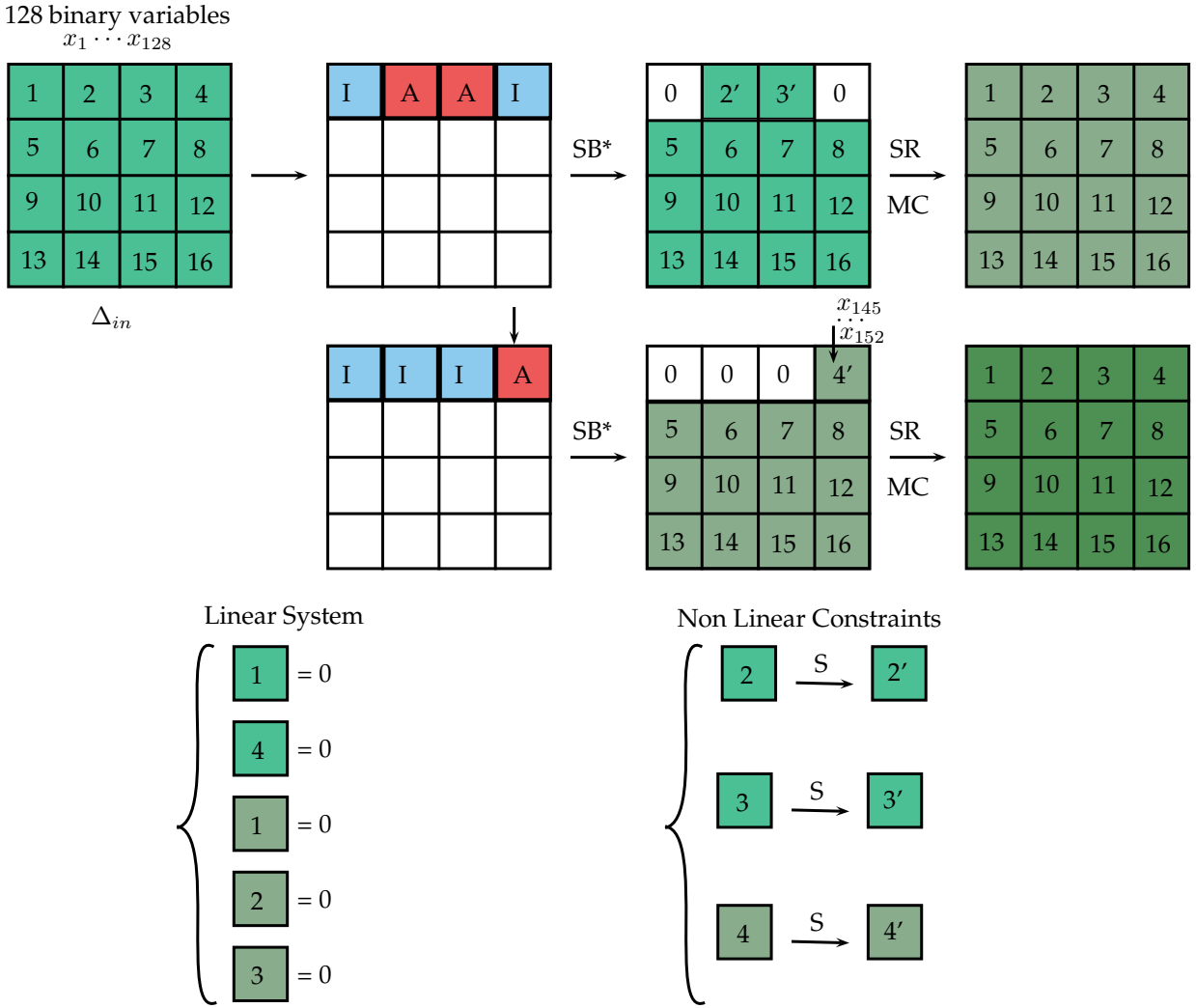
High Probability Differential Characteristic Search Algorithm



High Probability Differential Characteristic Search Algorithm



High Probability Differential Characteristic Search Algorithm



High Probability Differential Characteristic Search Algorithm

To analyze **r** rounds of a PSP network with **t** Sboxes by round:

- At most a active Sboxes so at most **128+8a** variables
- At least $rt - a$ inactive Sboxes so at least **8(rt-a)** linear equations

e.g.: when a=0 and t=4 there are 128 variables and $32r$ constraints, so we expect no solution for $r > 4$ rounds

High Probability Differential Characteristic Search Algorithm

To analyze **r** rounds of a PSP network with **t** Sboxes by round:

- At most a active Sboxes so at most **128+8a** variables
- At least $rt - a$ inactive Sboxes so at least **8(rt-a)** linear equations

e.g.: when a=0 and t=4 there are 128 variables and $32r$ constraints, so we expect no solution for $r > 4$ rounds

- Solve the **system of linear equations over GF(2)** to deduce the values of **some of the variables**
- Iterate over the computed subspace and **post filter** the characteristics according to the DDT
- Return the solution characteristics

Pattern-Prefix Search

- Iterates over the **tree** of **possible prefixes of patterns** (DFS algorithm)
- More efficient: **avoid work duplication**
- For example, discards **all patterns at once** if their prefix is impossible, instead of analyzing and discarding each one separately

Key Recovery Algorithm

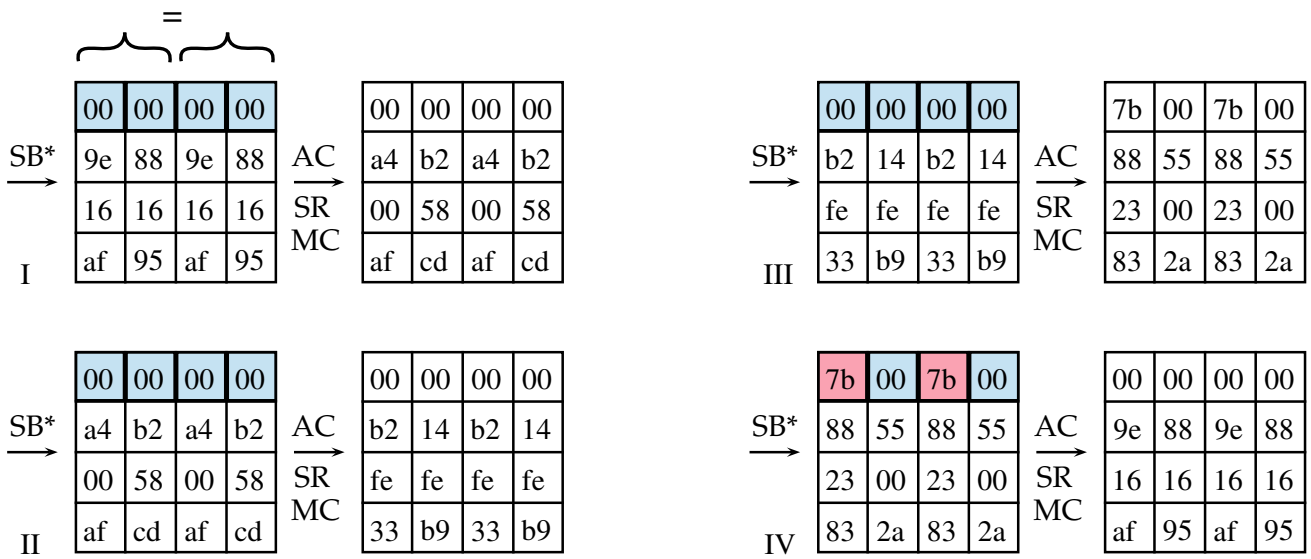
Key Recovery Algorithm for Differential Attacks

- Exploit the partial Sbox layer to analyze many rounds at the end of the cipher with **no increase in the attack complexity**
- Given an r-round characteristic, in the paper we show how to efficiently recover the key for $r + \lfloor 16/t \rfloor$ rounds

Application to Zorro

Application to Zorro

- The characteristic search algorithm returns the following characteristic (**iterative on 4 rounds**) as the **best differential characteristic on 9 rounds:**



Probability on 4 rounds: $(6/256)^2 = 2^{-10.83}$

Previous Analysis and New Results

- We cover **19** rounds with the iterative characteristic (8 active Sboxes, $p = 2^{-43.32}$) + **5** rounds with the key recovery part

Previous Analysis and New Results

- We cover **19** rounds with the iterative characteristic (8 active Sboxes, $p = 2^{-43.32}$) + **5** rounds with the key recovery part

Source	Time	Data	Memory	Technique
[Wang et al. 13]	2^{112}	2^{112} CP	negligible	Differential
[Rasoolzadeth et al. 14]	2^{55}	$2^{55.12}$ CP	2^{17}	Differential
[Rasoolzadeth et al. 14]	$2^{57.85}$	$2^{45.44}$ KP	2^{17}	Linear
New	2^{45}	$2^{41.5}$ CP	2^{10}	Differential
New	2^{45}	2^{45} KP	2^{17}	Linear

KP - Known Plaintext, CP - Chosen Plaintext

- We fully simulated this attack on a single desktop PC

Structural Flaw or Bad Luck?

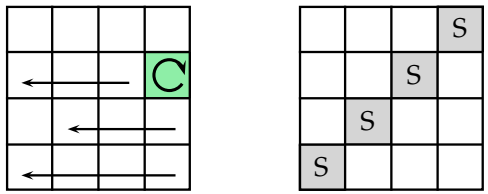
As we show in the paper, the structural weakness in Zorro is due to a **subtle inter-relation between the AES' SR and MC operations**.

Structural Flaw or Bad Luck?

As we show in the paper, the structural weakness in Zorro is due to a **subtle inter-relation between the AES' SR and MC operations**.

We analyzed Partial Substitution Permutation Networks that slightly deviate from the AES design strategy:

- Lightly modified ShiftRows
- Different Sbox positions



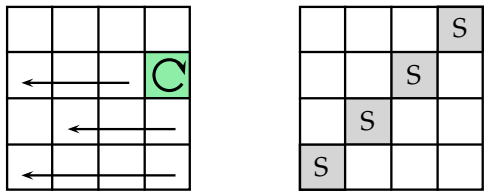
We used our tools to show that some variants provide **significantly better resistance** to basic differential/linear cryptanalysis

Structural Flaw or Bad Luck?

As we show in the paper, the structural weakness in Zorro is due to a **subtle inter-relation between the AES' SR and MC operations**.

We analyzed Partial Substitution Permutation Networks that slightly deviate from the AES design strategy:

- Lightly modified ShiftRows
- Different Sbox positions



We used our tools to show that some variants provide **significantly better resistance** to basic differential/linear cryptanalysis

$\rightarrow$ The generic design is **not inherently flawed**, and can be reused (**with caution**) to build secure block ciphers.

Conclusion

- We introduced **new generic algorithms for differential and linear cryptanalysis of SPN with partial non-linear layers**
- We used these techniques to mount a **practical attack on full-round Zorro**
- Our tools will be useful to design secure PSP networks in the future

Thank you for your attention