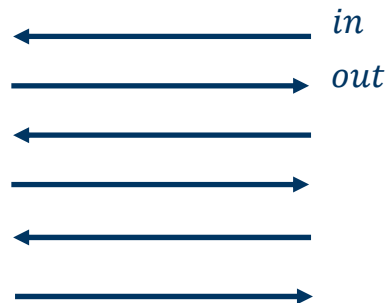
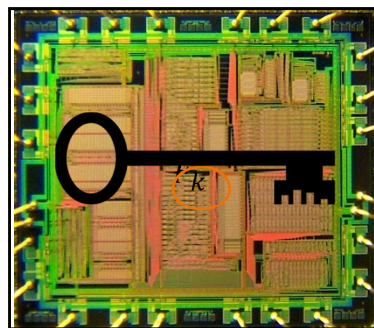


RUHR-UNIVERSITÄT BOCHUM

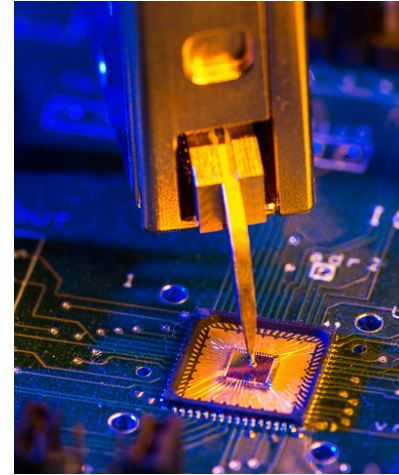
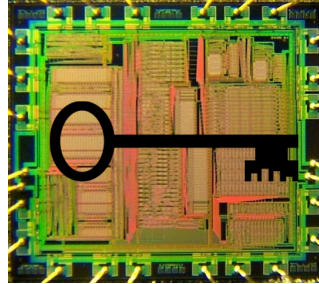
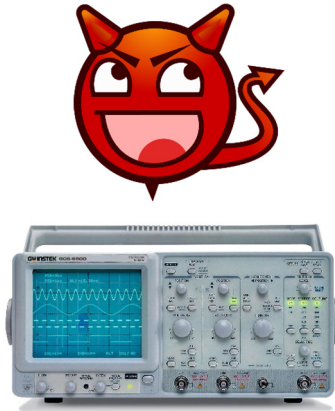
# Automated Generation of Masked Hardware (AGEMA)

David Knichel, [Amir Moradi](#), Nicolai Müller and Pascal Sasdrich

CHES, September 2022, Leuven, Belgium



# Passive Physical Attacks



**Physical characteristics can be exploited to extract secret information:**

- Timing
- Power Consumption
- Electromagnetic radiations
- ...

**Masking randomizes the intermediate values of a cryptographic computation to avoid dependencies between these values and the power consumption**

**It is usually applied on an algorithmic level**

- Does not rely on the power consumption characteristics of the device


**Each intermediate value is concealed by a random mask that is different for every execution**

**Basically, it corresponds to a secret sharing scheme:**

- Boolean secret sharing

# Boolean Secret Sharing

## First order Boolean secret sharing (two shares):

- Secret:  $x$
  - Random:  $m$
- 
 Shares:  $(x_1, x_2)$
- $$x_1 = x \oplus m$$
- $$x_2 = m$$
- $$x_1 \oplus x_2 = x$$
- One needs to know share  $x_1$  and  $x_2$  to compute secret  $x$ 
    - Neither of them alone provides enough information

## Linear Function $F$

- Definition  $F(x \oplus z) = F(x) \oplus F(z)$
- Boolean share before  $F$ :  $(x_1, x_2)$  with  $x_1 \oplus x_2 = x$
- Boolean share after  $F$ :  $(F(x_1), F(x_2))$  with  $F(x_1) \oplus F(x_2) = F(x_1 \oplus x_2) = F(x)$

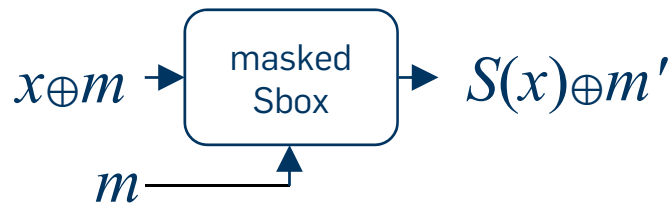
## Non-linear Function?

# Masking in Hardware

## Pre-computing the masked tables in software

- Sequential operations, time consuming, low efficiency
- High efficiency is desired in hardware

## Ad-hoc/heuristic schemes



# Masking in Hardware

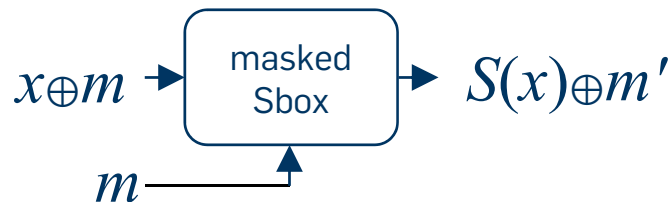
## Pre-computing the masked tables in software

- Sequential operations, time consuming, low efficiency
- High efficiency is desired in hardware

## Ad-hoc/heuristic schemes

## Processing the mask ( $m$ ) and masked data ( $x \oplus m$ ) simultaneously

- Joint distribution of leakages
  - It is called to be due to **glitches** [actually not always true]
  - Possible attacks



# Masking in Hardware

## Pre-computing the masked tables in software

- Sequential operations, time consuming, low efficiency
- High efficiency is desired in hardware

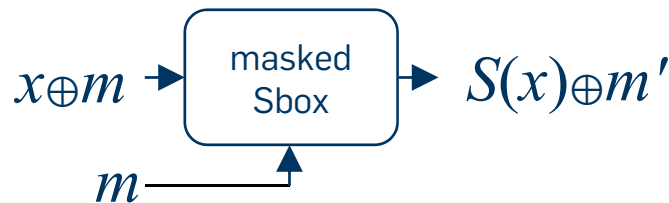
## Ad-hoc/heuristic schemes

## Processing the mask ( $m$ ) and masked data ( $x \oplus m$ ) simultaneously

- Joint distribution of leakages
  - It is called to be due to **glitches** [actually not always true]
  - Possible attacks

## Systematic schemes

- Threshold Implementation, provable security

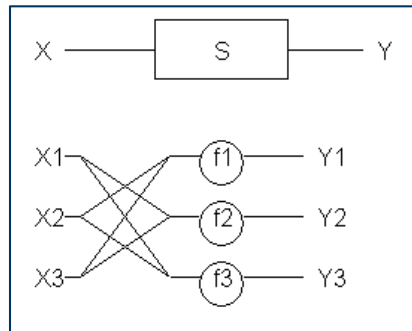




# Threshold Implementation

Let's consider an Sbox:

$$x_1 \oplus x_2 \oplus x_3 = x$$

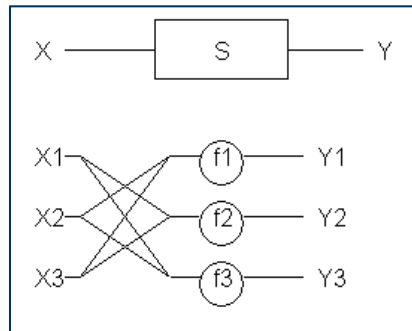


$$y_1 \oplus y_2 \oplus y_3 = y$$

# Threshold Implementation

Let's consider an Sbox:

$$x_1 \oplus x_2 \oplus x_3 = x$$



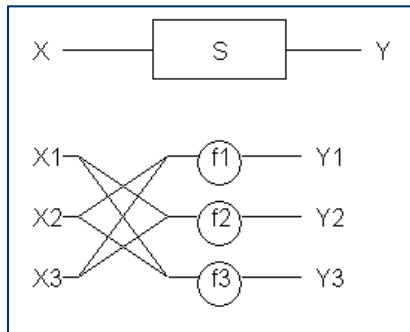
$$y_1 \oplus y_2 \oplus y_3 = y$$

Each  $f$  should be independent of one share

# Threshold Implementation

Let's consider an Sbox:

$$x_1 \oplus x_2 \oplus x_3 = x$$



$$y_1 \oplus y_2 \oplus y_3 = y$$

**Example:**

$$x = (a, b, c, d) \quad y = (e, f, g, h)$$

Each  $f$  should be independent of one share

$$S_1(a, b, c, d) = e$$

$$e = a \oplus bc \oplus d$$

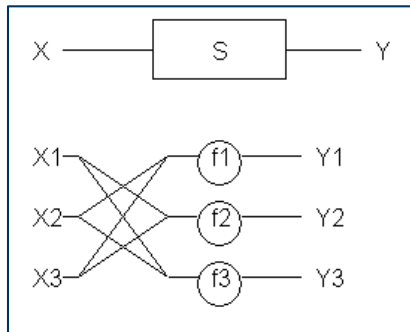
$$e = a_1 \oplus a_2 \oplus a_3 \oplus b_1c_1 \oplus b_1c_2 \oplus b_1c_3 \oplus b_2c_1 \oplus$$

$$b_2c_2 \oplus b_2c_3 \oplus b_3c_1 \oplus b_3c_2 \oplus b_3c_3 \oplus d_1 \oplus d_2 \oplus d_3$$

# Threshold Implementation

Let's consider an Sbox:

$$x_1 \oplus x_2 \oplus x_3 = x$$



$$y_1 \oplus y_2 \oplus y_3 = y$$

Example:

$$x = (a, b, c, d) \quad y = (e, f, g, h)$$

$$S_1(a, b, c, d) = e$$

$$e = a \oplus bc \oplus d$$

$$e = a_1 \oplus a_2 \oplus a_3 \oplus b_1c_1 \oplus b_1c_2 \oplus b_1c_3 \oplus b_2c_1 \oplus$$

$$b_2c_2 \oplus b_2c_3 \oplus b_3c_1 \oplus b_3c_2 \oplus b_3c_3 \oplus d_1 \oplus d_2 \oplus d_3$$

Each  $f$  should be independent of one share

are clear where to go (to which  $f$ )

can be arbitrarily distributed among two component functions

$$f_1 = b_2c_3 \oplus b_3c_2 \oplus a_2 \oplus d_2 \oplus b_2c_2$$

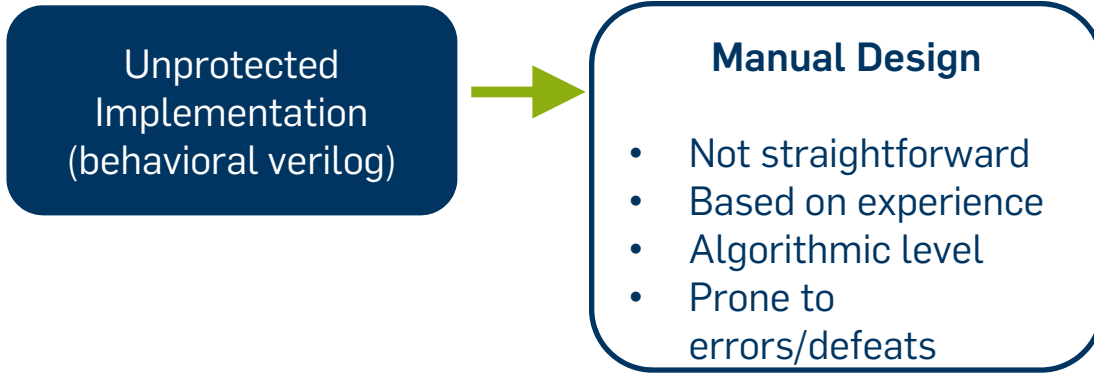
$$f_2 = b_3c_1 \oplus b_1c_3 \oplus a_3 \oplus d_3 \oplus b_3c_3$$

$$f_3 = b_1c_2 \oplus b_2c_1 \oplus a_1 \oplus d_1 \oplus b_1c_1$$

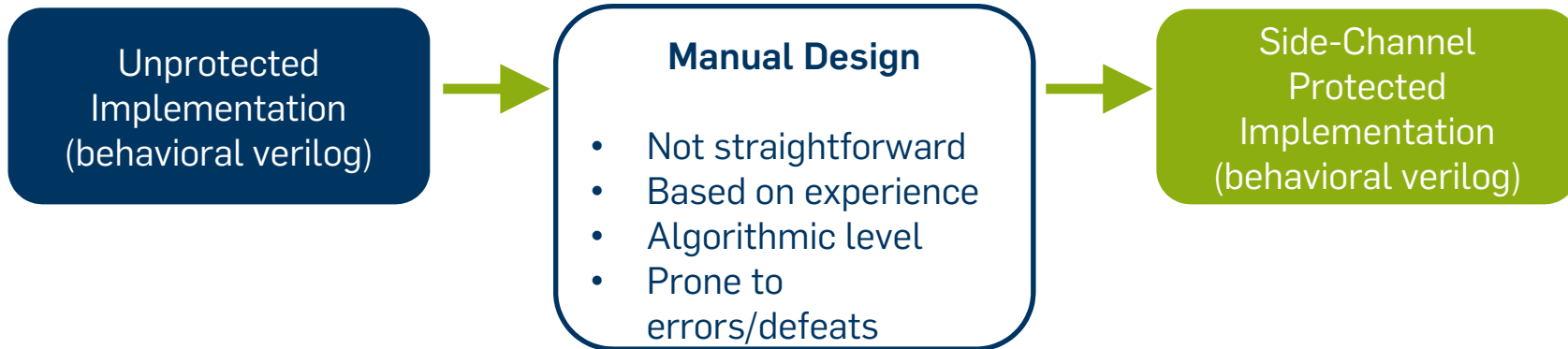
# How to Make a Masked Design?

Unprotected  
Implementation  
(behavioral verilog)

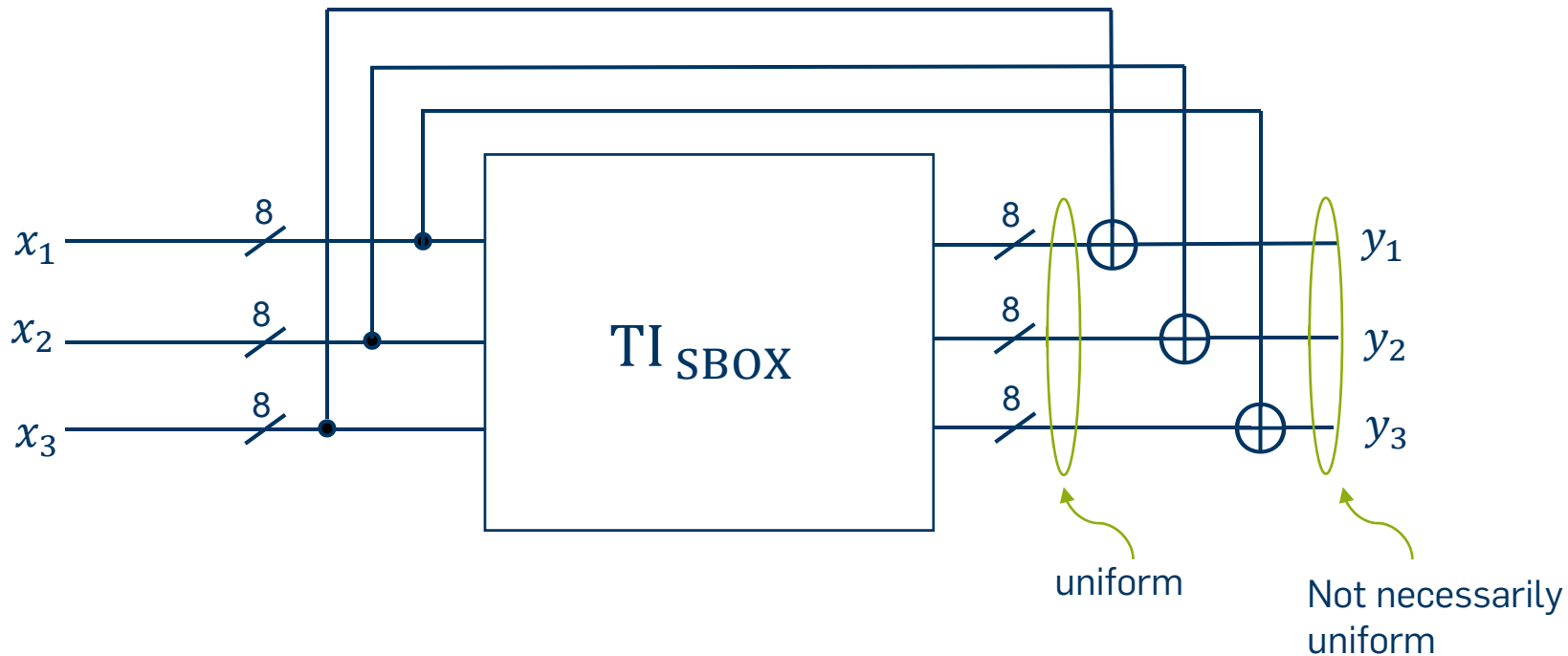
# How to Make a Masked Design?



# How to Make a Masked Design?



# Non-Composability in the TI context



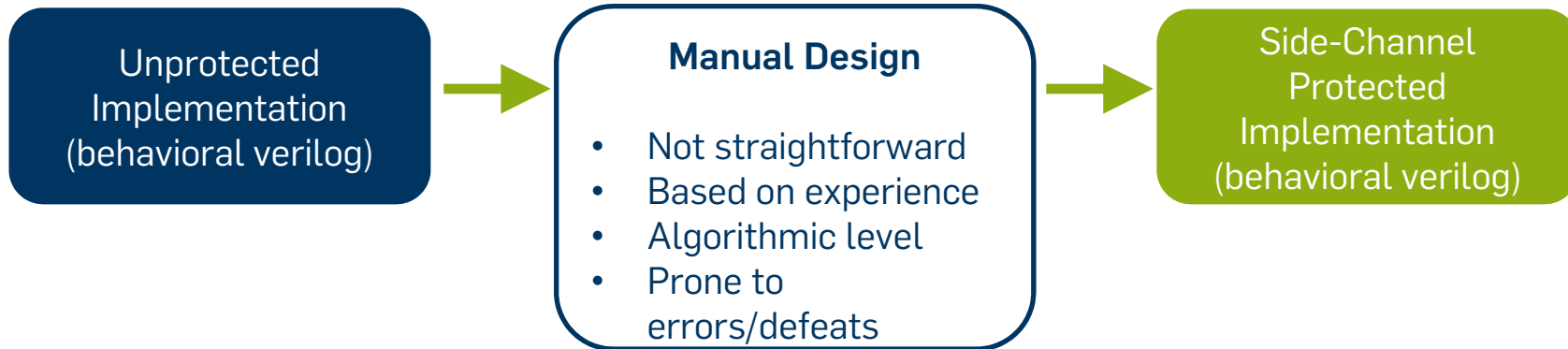


**Masking large and complex circuits is a hard task especially for high security orders**

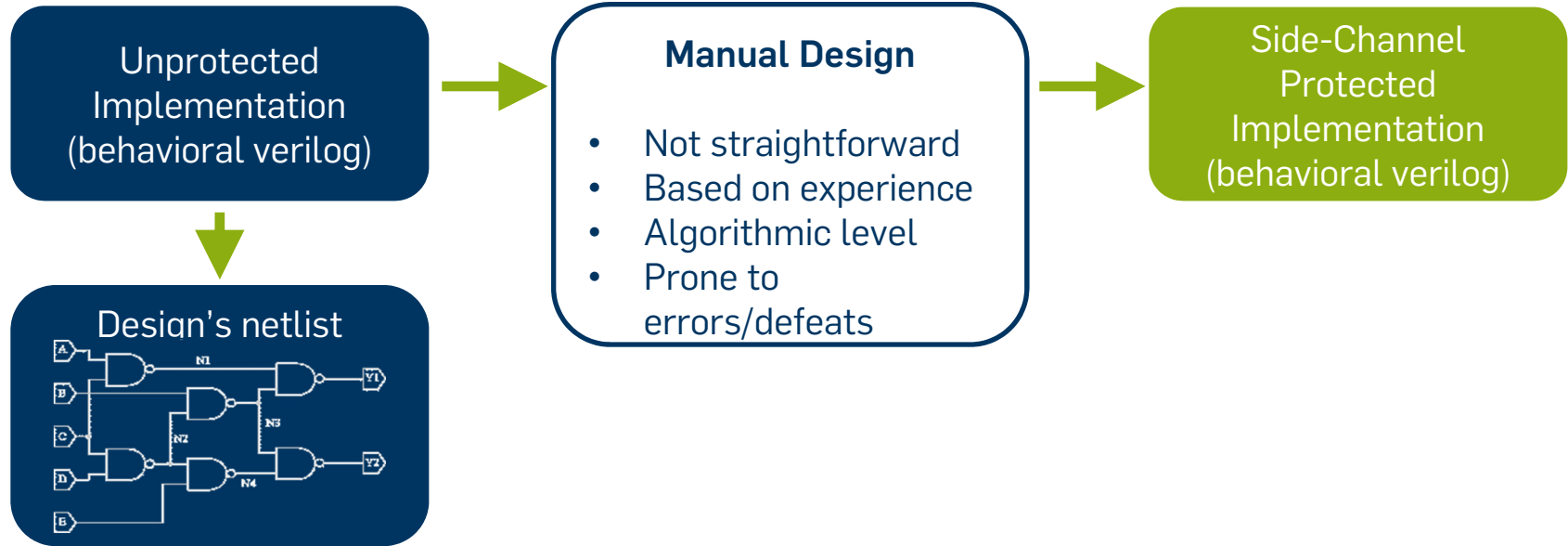
**Composable hardware gadgets offer a systematic way to generate provable secure designs**

- Arbitrary security orders possible
- Based on formal security notions
- Following divide-and-conquer approach based on fundamental building blocks
- Simply replacing unprotected gates (or larger modules) with its masked and composable counterpart

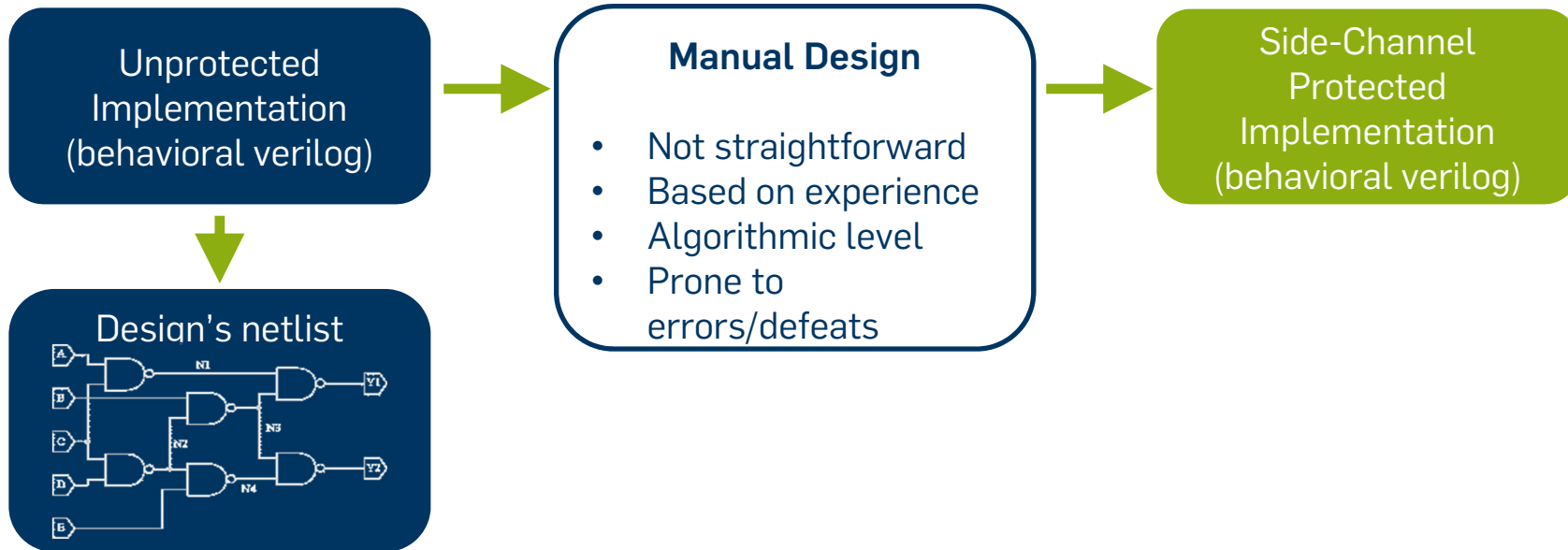
# How to Make a Secure Design?



# How to Make a Secure Design?



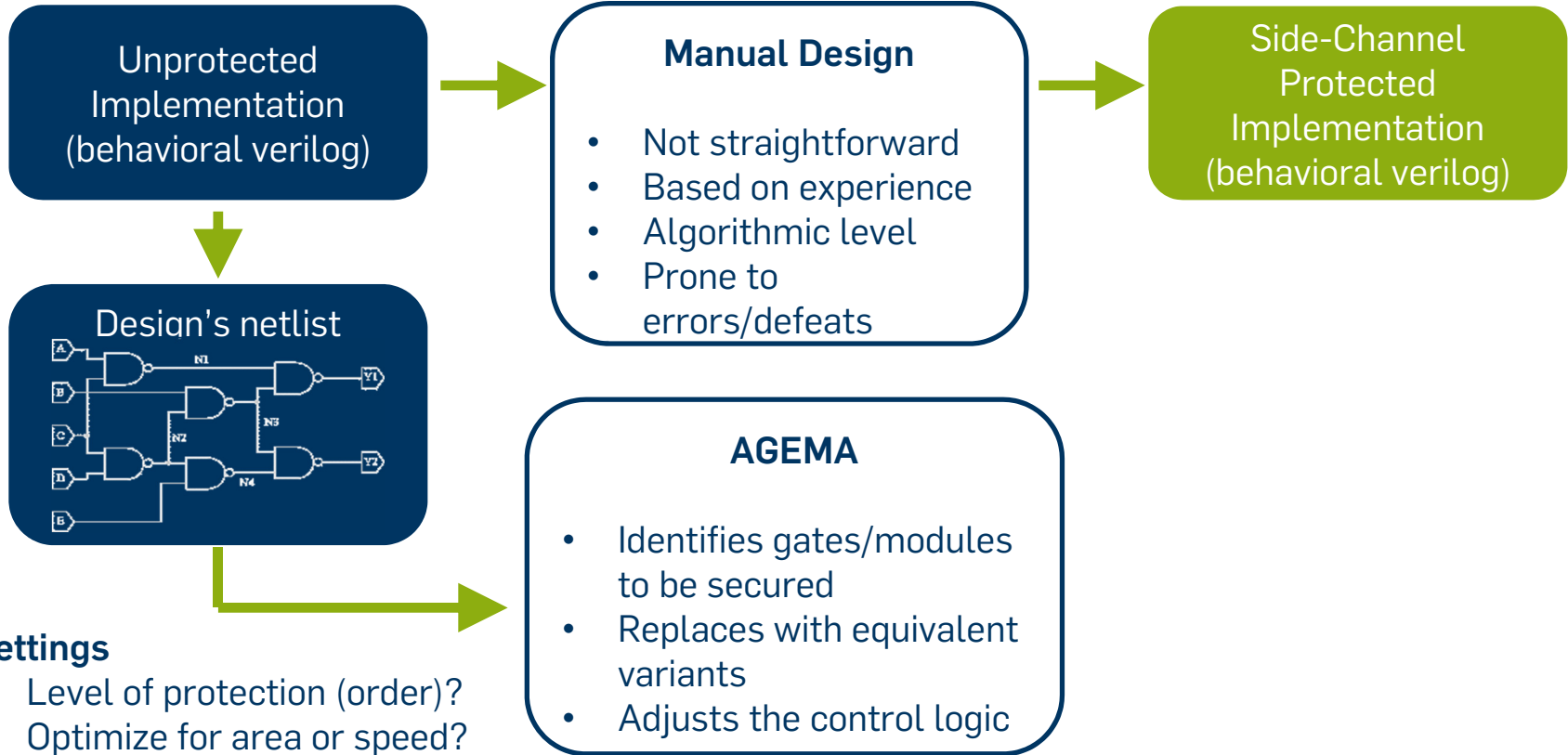
# How to Make a Secure Design?



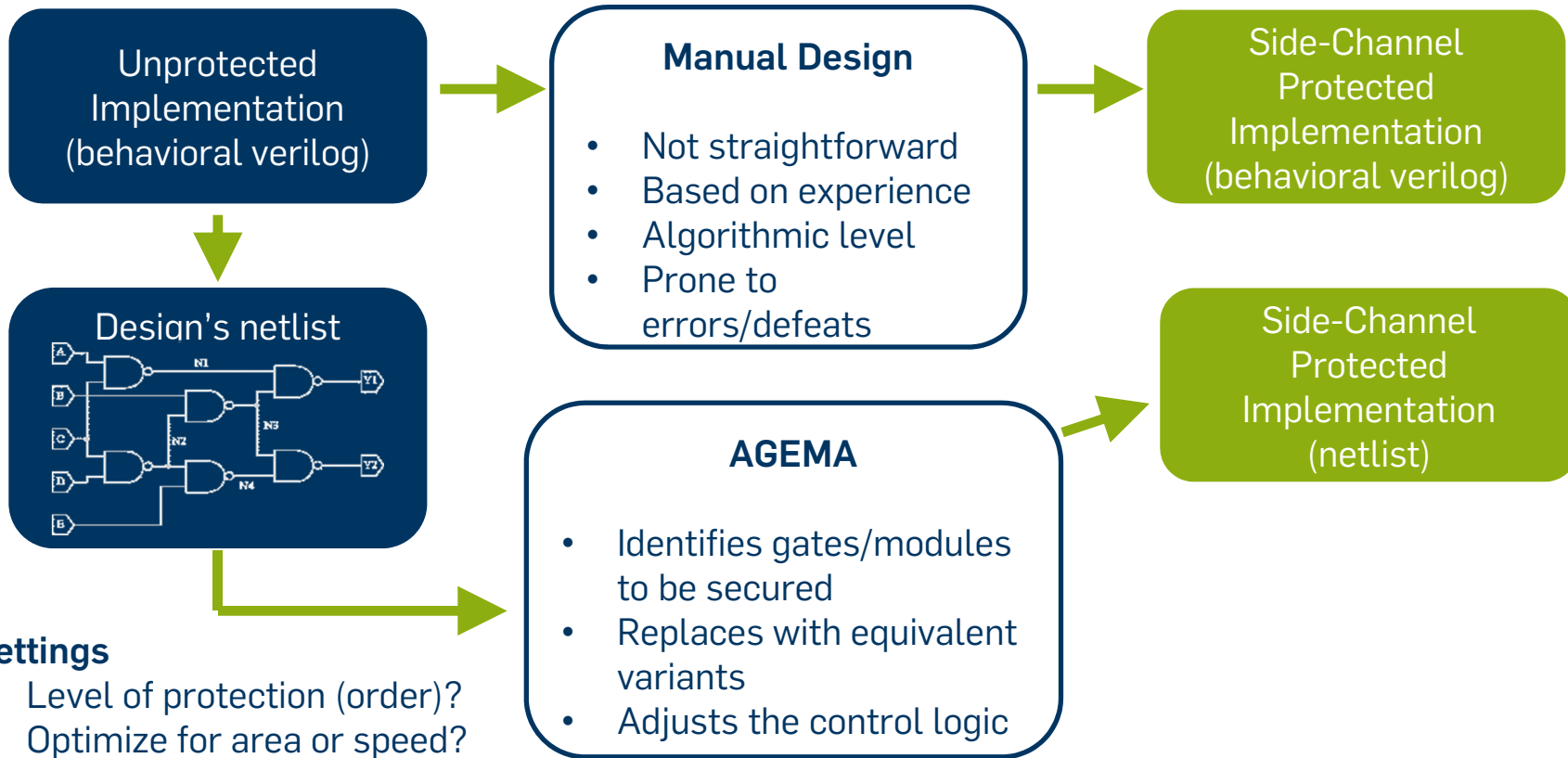
## Settings

- Level of protection (order)?
- Optimize for area or speed?

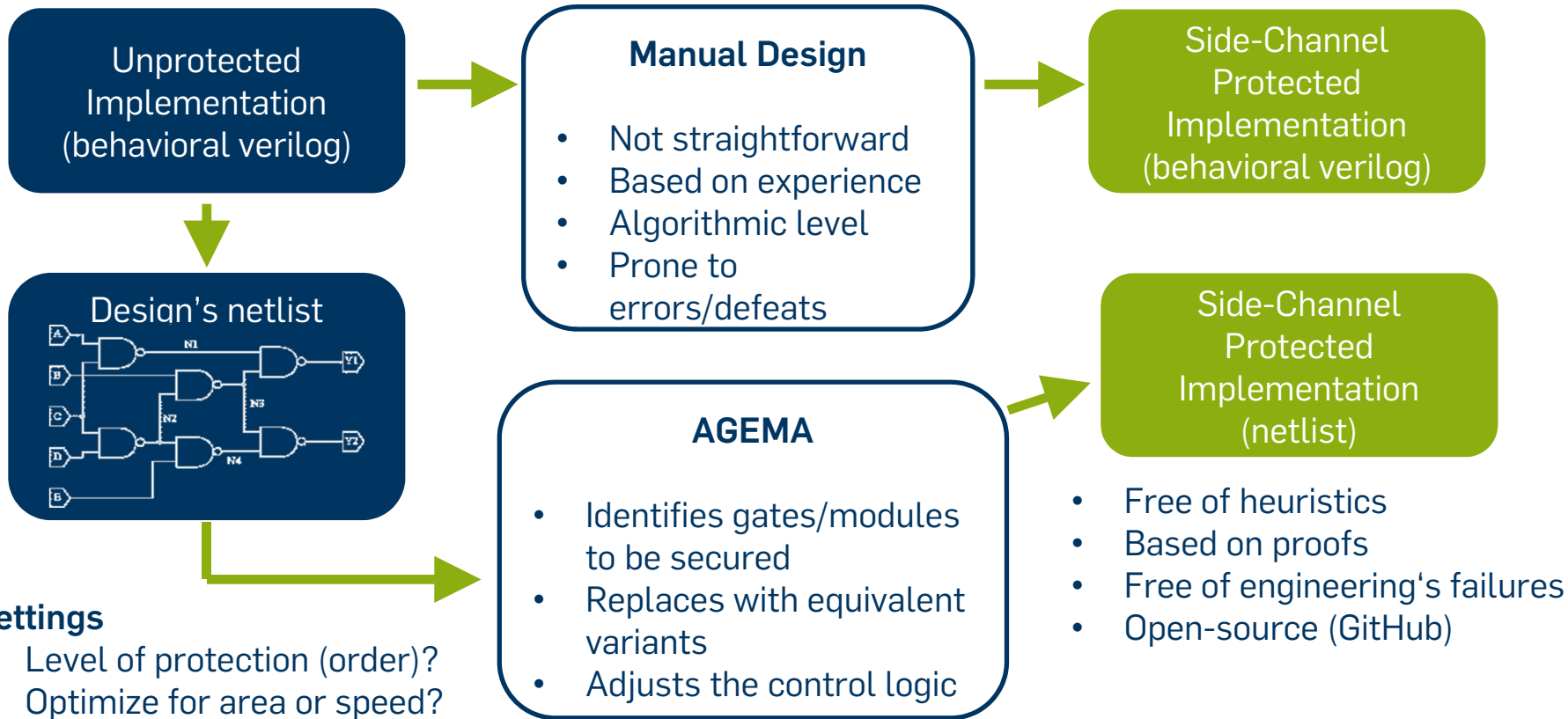
# How to Make a Secure Design?

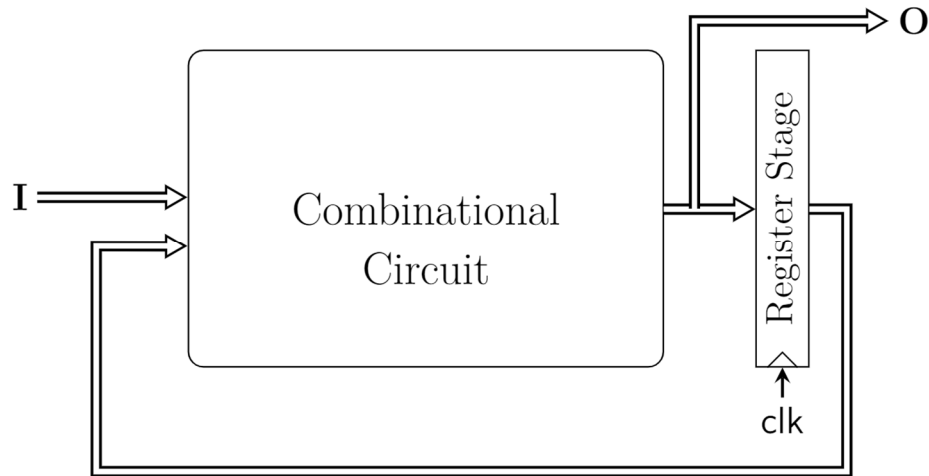


# How to Make a Secure Design?

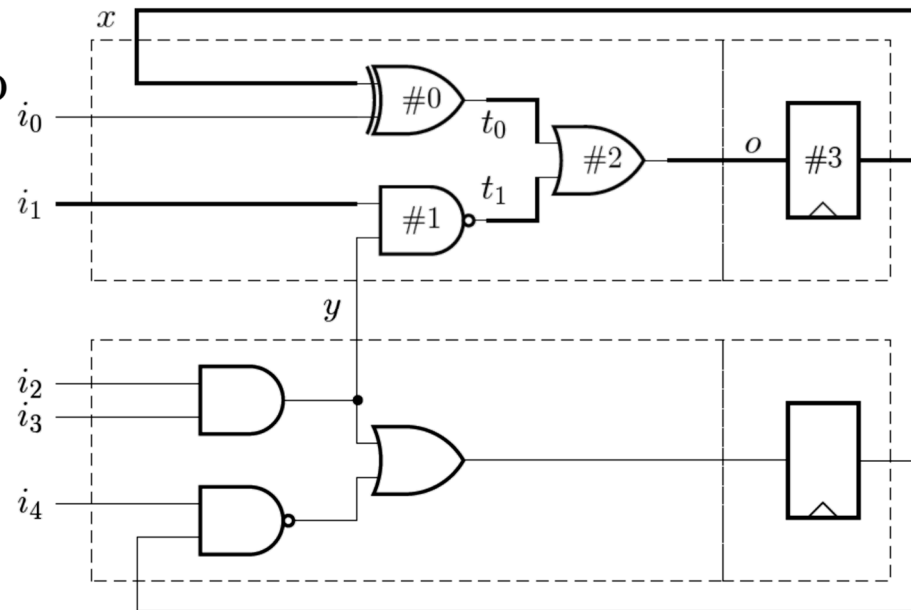
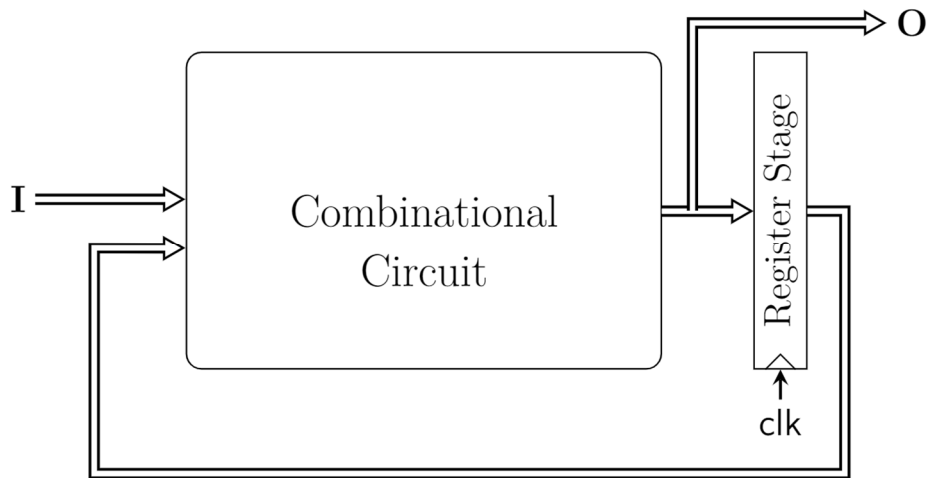


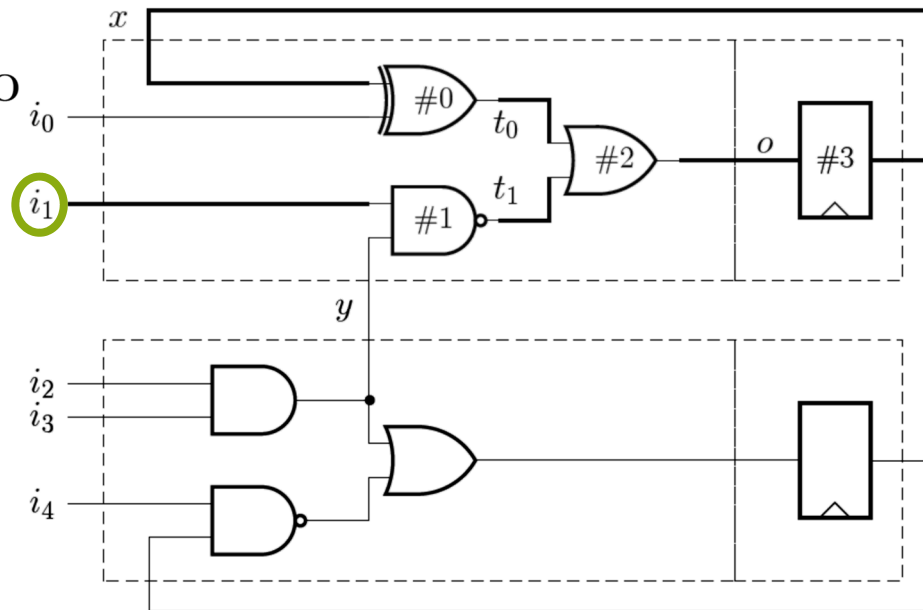
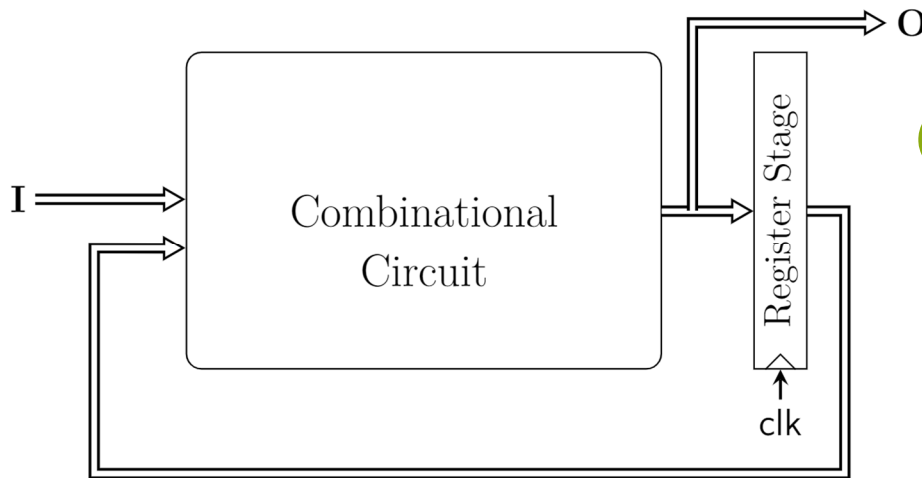
# How to Make a Secure Design?

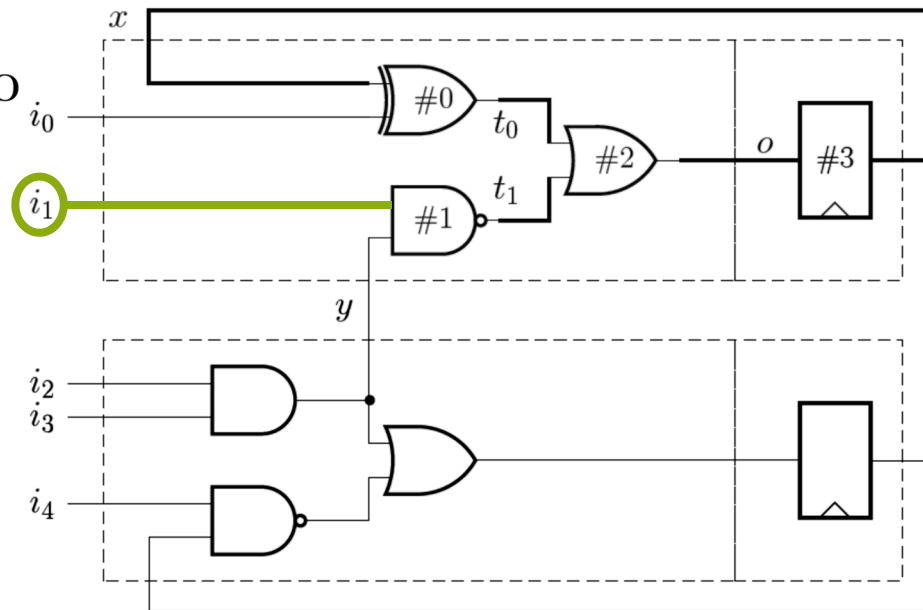
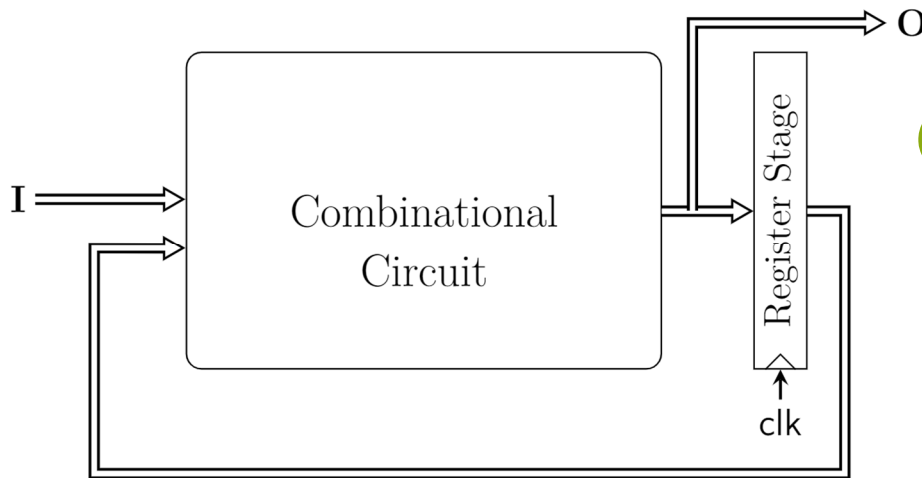


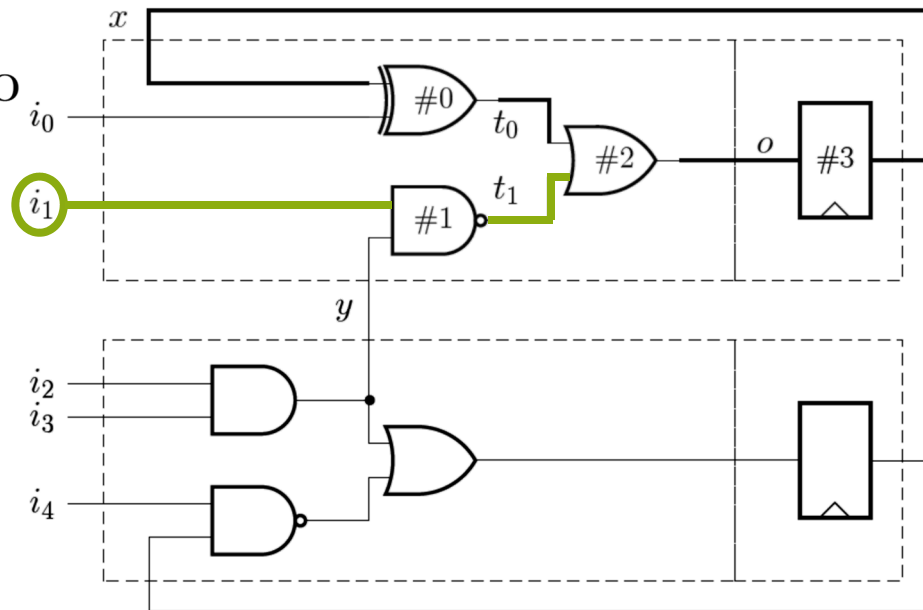
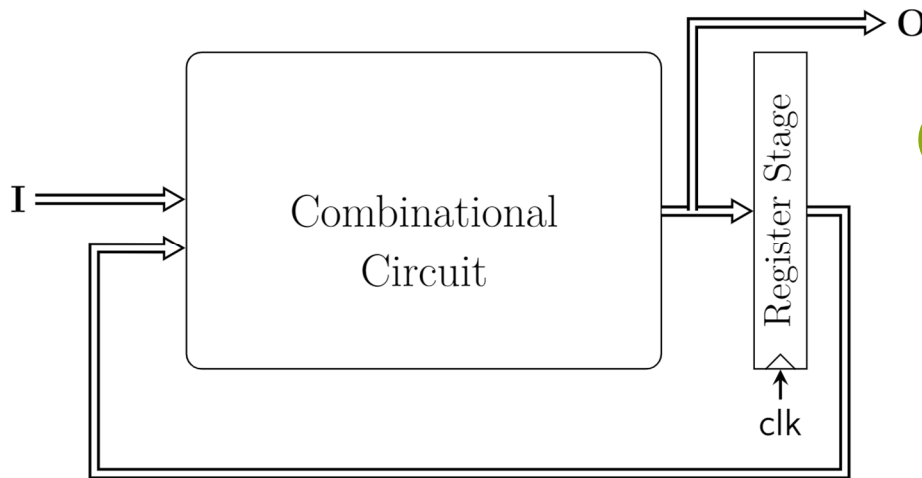


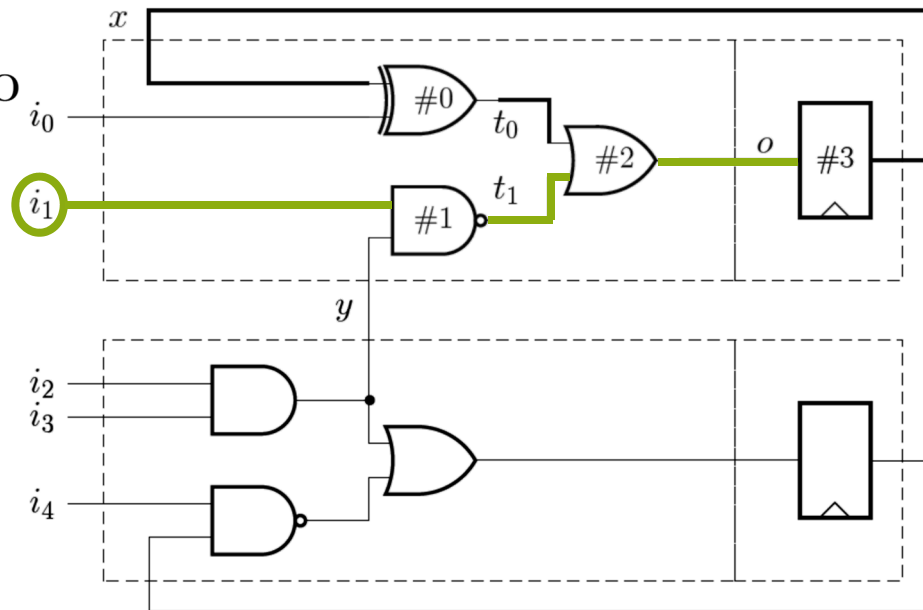
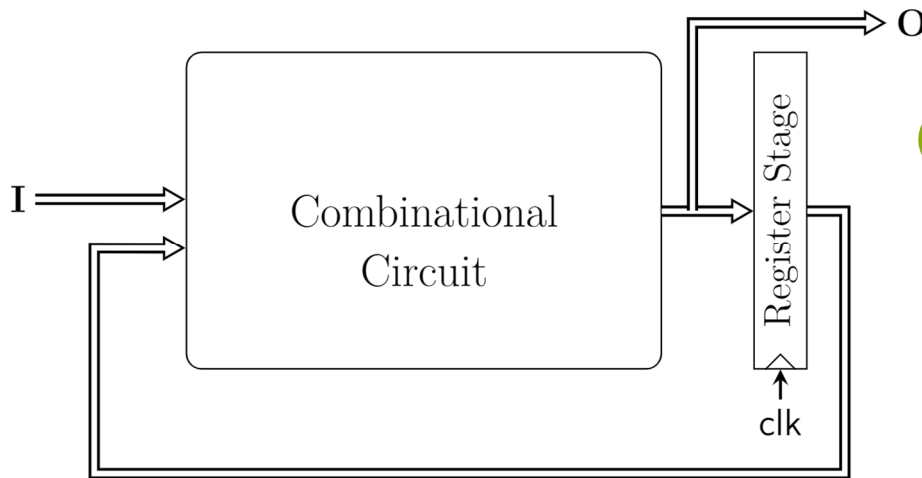


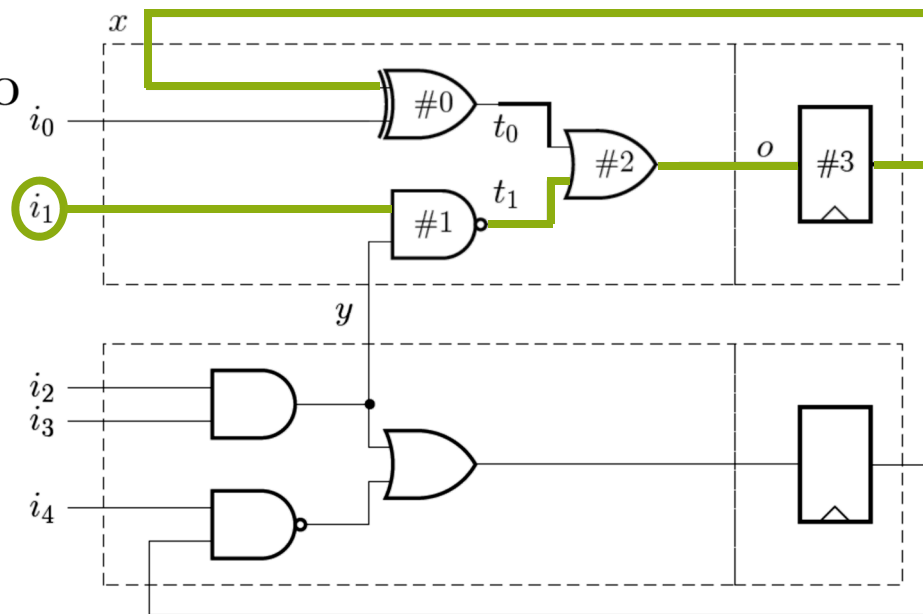
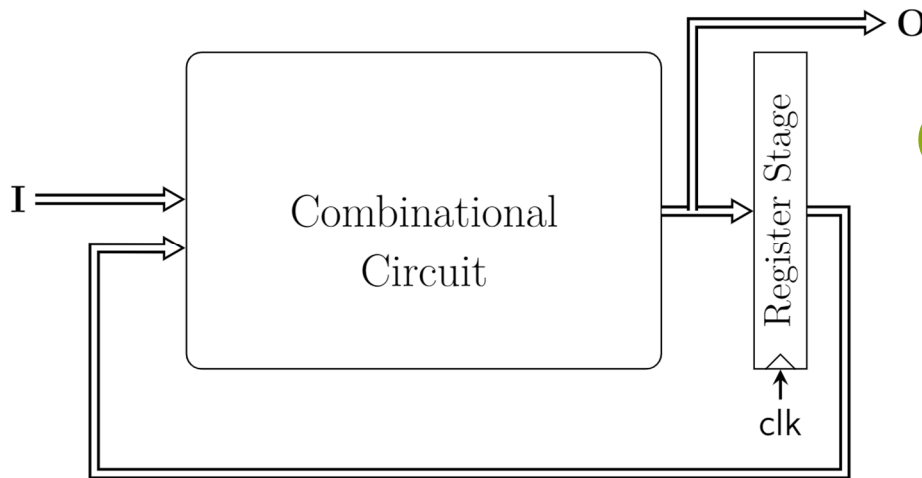


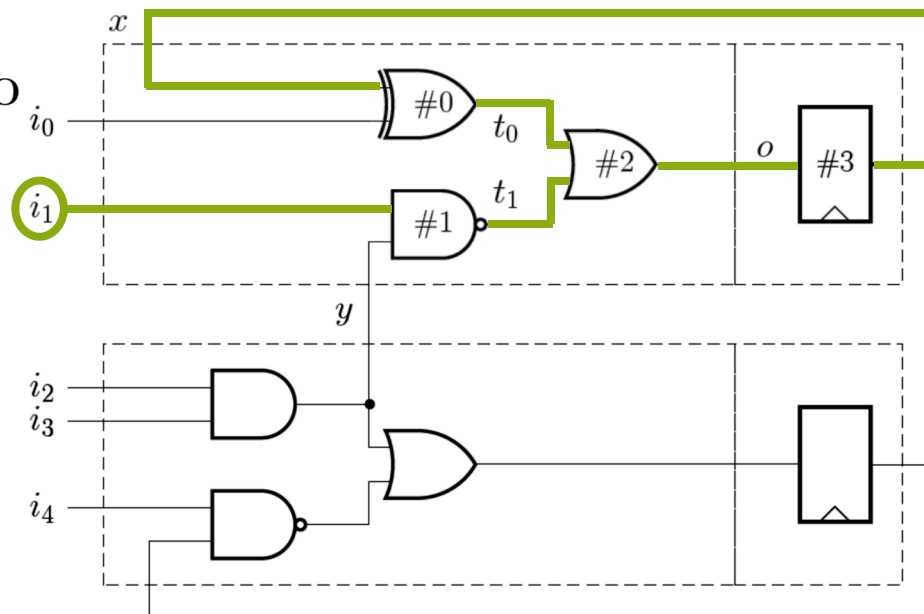
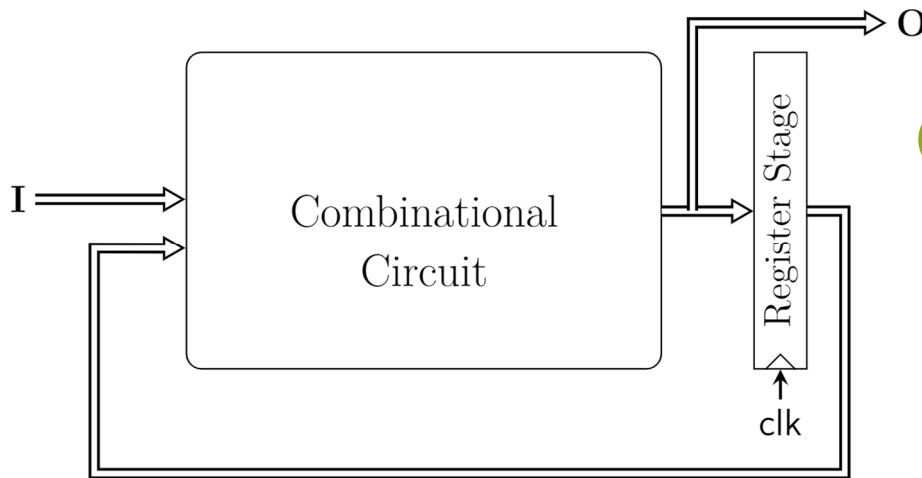


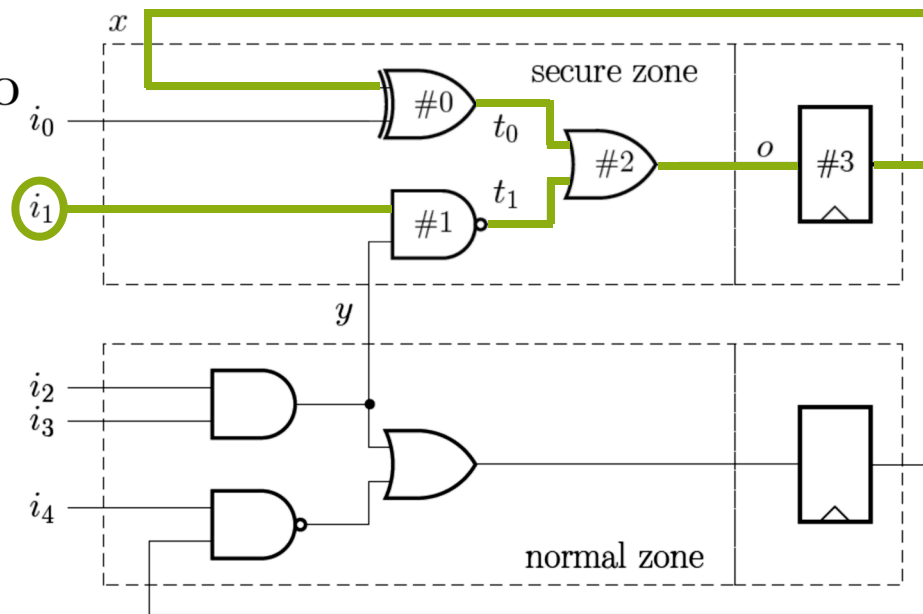
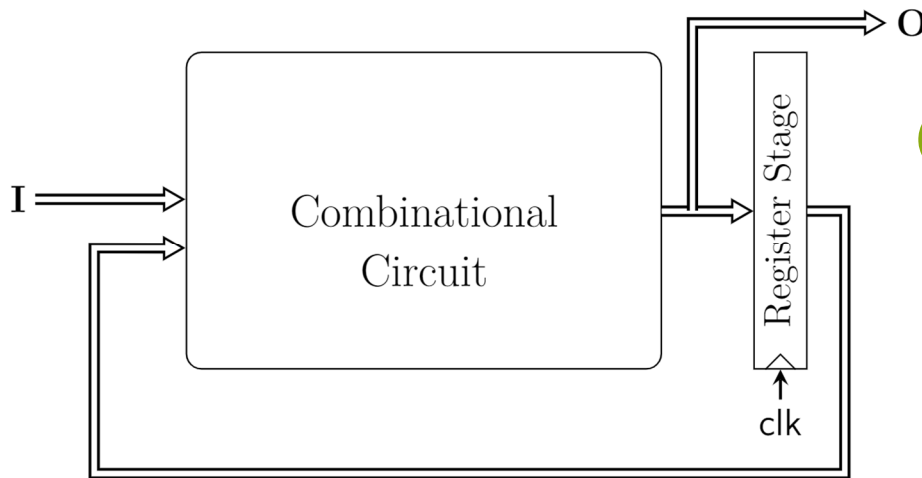






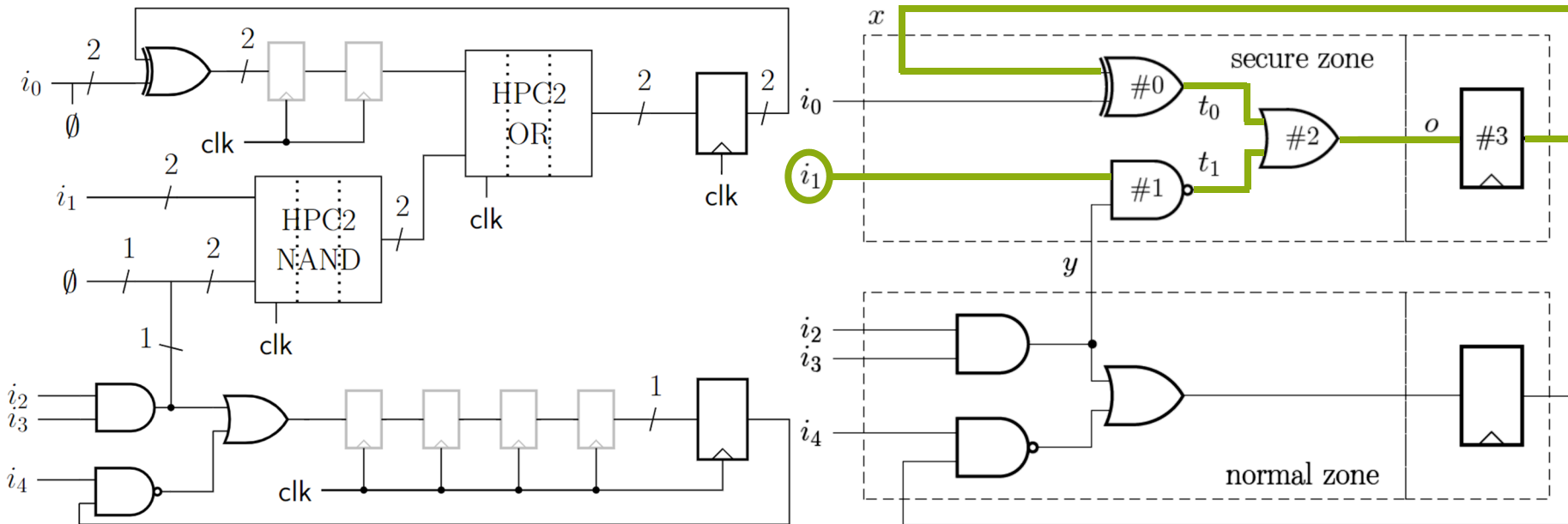




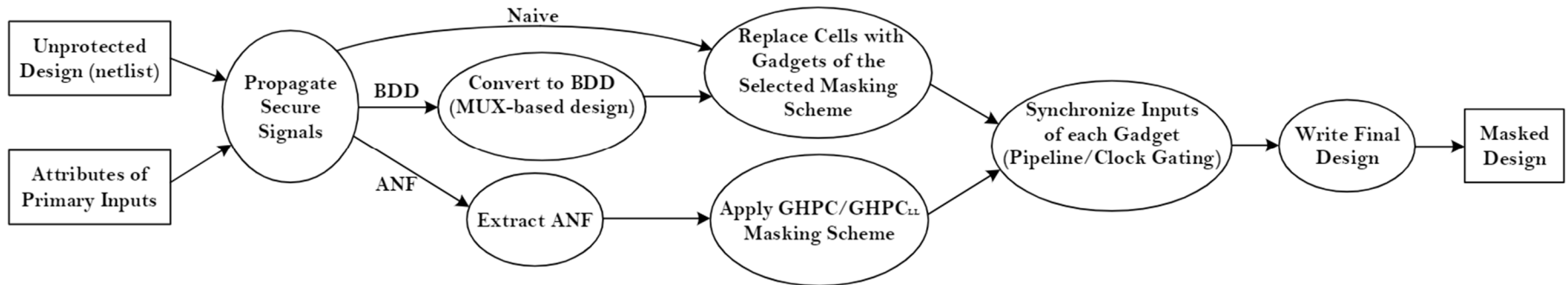




# Example



# General Procedure



- **Requirements**
  - Composable security
    - A secure circuit is not necessarily secure when composed
    - PINI (Probe-Isolating Non-Interference)
  - PINI gadgets of essential gates
    - AND/NAND/OR/NOR/...

- **Requirements**

- Composable security
  - A secure circuit is not necessarily secure when composed
  - PINI (Probe-Isolating Non-Interference)
- PINI gadgets of essential gates
  - AND/NAND/OR/NOR/...

- **Efficiency**

- Provable security
  - As long as the gadgets are PINI
- Extendable to any arbitrary order
- Not as efficient as manually-crafted designs
  - Larger, higher latency, higher demand for fresh masks
- Any engineer can make secure designs
  - <https://github.com/Chair-for-Security-Engineering/AGEMA>

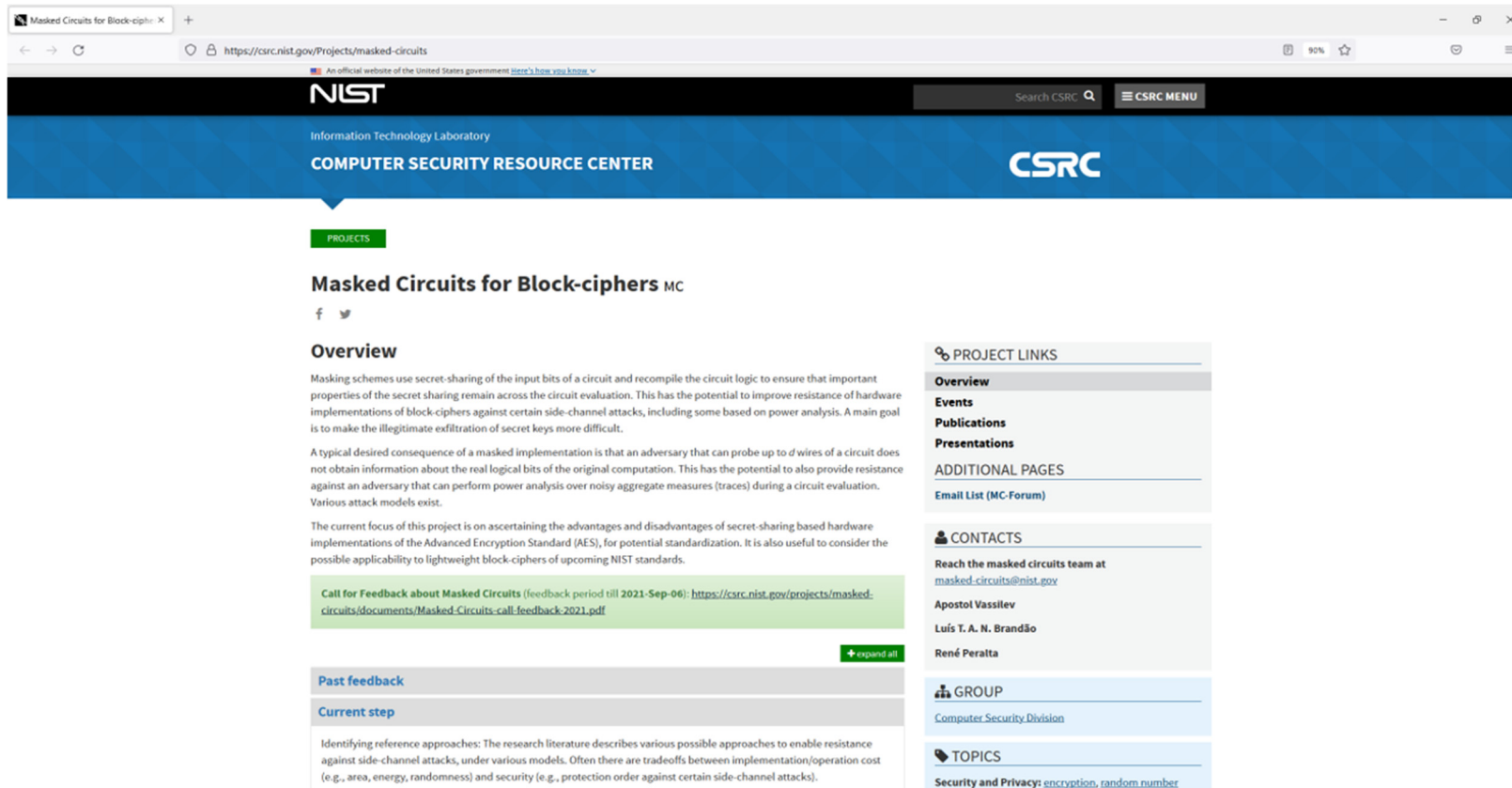


Thanks!  
Any Questions?

[amir.moradi@rub.de](mailto:amir.moradi@rub.de)

# Standardization Process

<https://csrc.nist.gov/Projects/masked-circuits>



The screenshot shows a web browser displaying the NIST CSRC website. The page title is "Masked Circuits for Block-ciphers MC". The navigation bar includes the NIST logo, "Information Technology Laboratory", "COMPUTER SECURITY RESOURCE CENTER", and the CSRC logo. A search bar and "CSRC MENU" are also present.

**PROJECTS**

## Masked Circuits for Block-ciphers MC

f t

### Overview

Masking schemes use secret-sharing of the input bits of a circuit and recompile the circuit logic to ensure that important properties of the secret sharing remain across the circuit evaluation. This has the potential to improve resistance of hardware implementations of block ciphers against certain side-channel attacks, including some based on power analysis. A main goal is to make the illegitimate exfiltration of secret keys more difficult.

A typical desired consequence of a masked implementation is that an adversary that can probe up to  $d$  wires of a circuit does not obtain information about the real logical bits of the original computation. This has the potential to also provide resistance against an adversary that can perform power analysis over noisy aggregate measures (traces) during a circuit evaluation. Various attack models exist.

The current focus of this project is on ascertaining the advantages and disadvantages of secret-sharing based hardware implementations of the Advanced Encryption Standard (AES), for potential standardization. It is also useful to consider the possible applicability to lightweight block-ciphers of upcoming NIST standards.

**Call for Feedback about Masked Circuits** (feedback period till 2021-Sep-06): <https://csrc.nist.gov/projects/masked-circuits/documents/Masked-Circuits-call-feedback-2021.pdf>

[+ expand all](#)

#### Past feedback

#### Current step

Identifying reference approaches: The research literature describes various possible approaches to enable resistance against side-channel attacks, under various models. Often there are tradeoffs between implementation/operation cost (e.g., area, energy, randomness) and security (e.g., protection order against certain side-channel attacks).

#### PROJECT LINKS

- Overview
- Events
- Publications
- Presentations

#### ADDITIONAL PAGES

- Email List (MC-Forum)

#### CONTACTS

Reach the masked circuits team at [masked\\_circuits@nist.gov](mailto:masked_circuits@nist.gov)

- Apostol Vassilev
- Luís T. A. N. Brandão
- René Peratta

#### GROUP

Computer Security Division

#### TOPICS

Security and Privacy: [encryption](#), [random number](#)