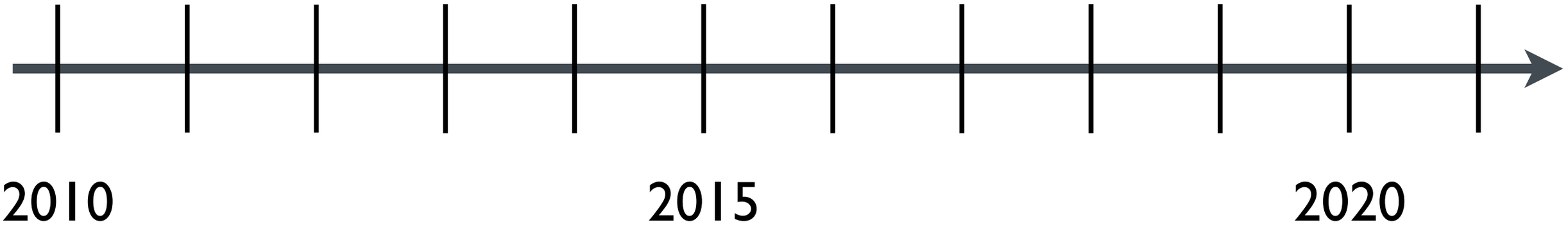
CryptoExperts

WE INNOVATE TO SECURE YOUR BUSINESS

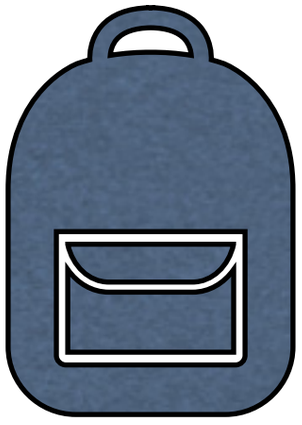
# On the road to building formally verified side-channel countermeasures

CrossFyre 2021

December 2, 2021


2010
2015
2020

# First Steps: 2010

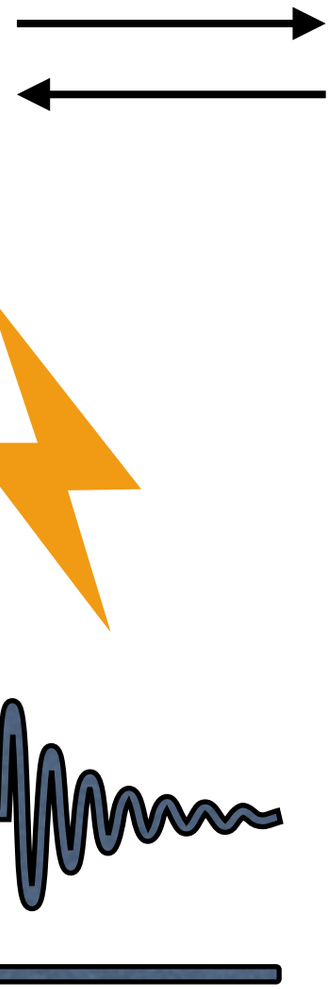


- Engineering school
- Some lectures in cryptography
- 1 internship at Thales in crypto

# First Steps: 2010

- Engineering school
- Some lectures in cryptography
- 1 internship at Thales in crypto

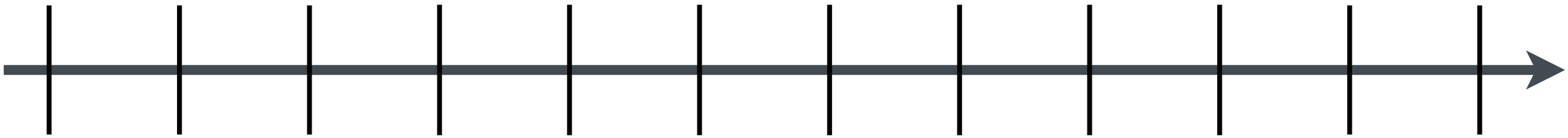
# Side-Channel Attacks

inputs / outputs

Power consumption

$$\approx \left|\left|_i \left(f(v_i) + \eta\right)\right.\right.$$

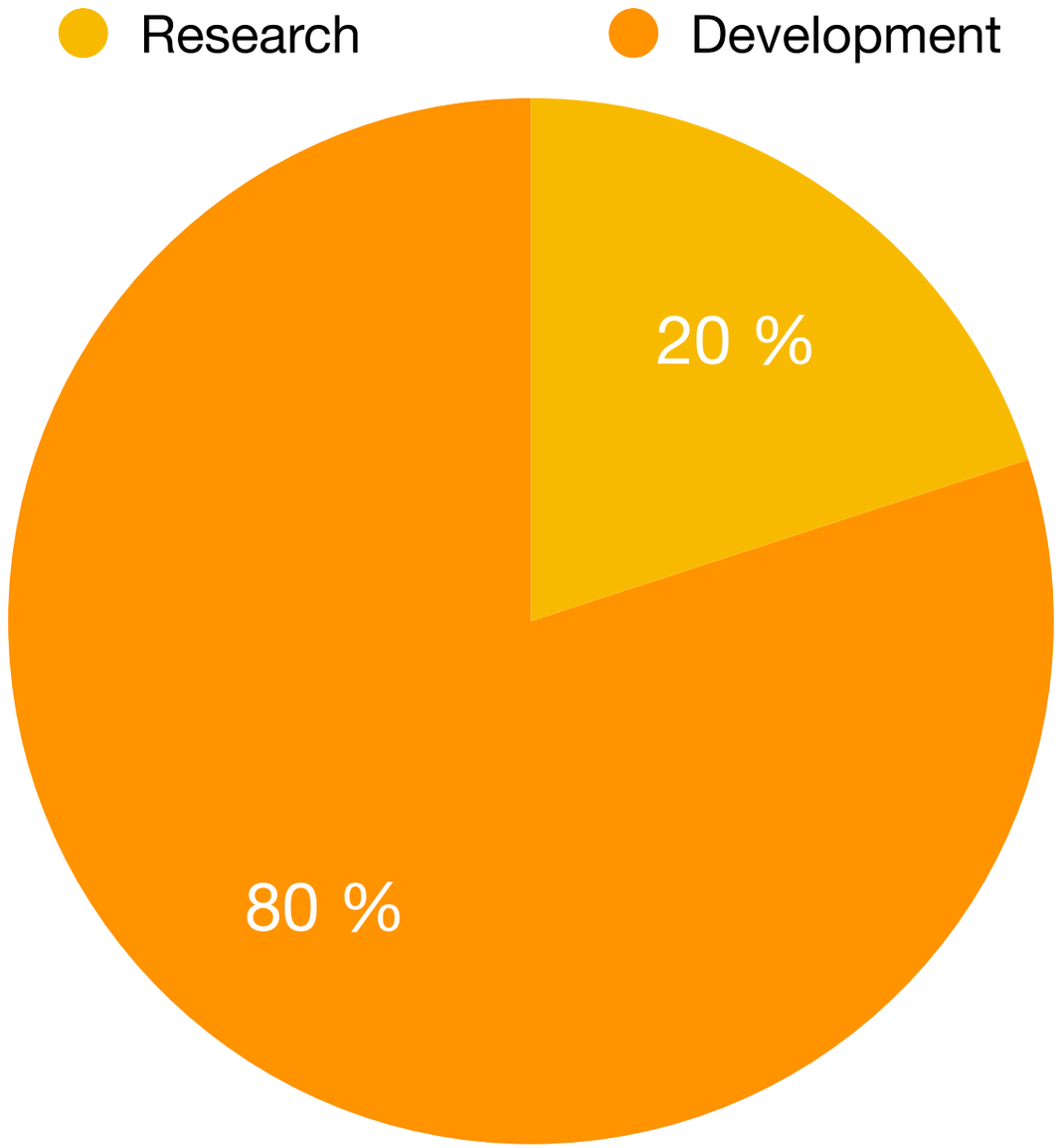

2010
2015
2020

# First Steps: 2011

- Engineering school
- Some lectures in cryptography
- 2 internships at Thales in crypto

- First working experience as an engineer
- No PhD thesis
- Some research in crypto

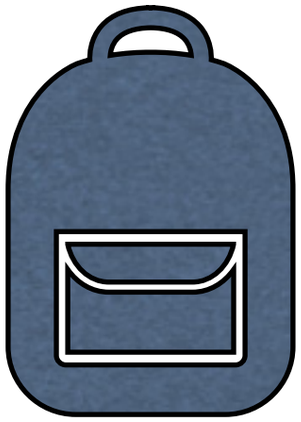
# First Working Experience

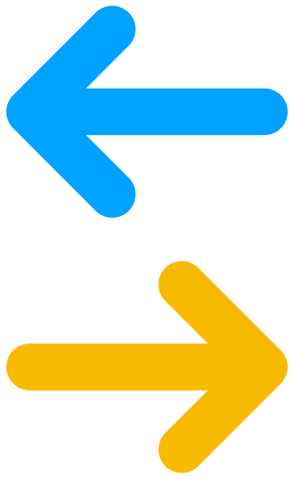


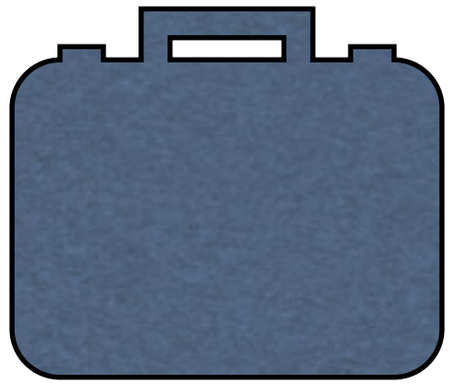

Oberthur
Technologies
Grenoble INP
ENSIMAG
2010
2015
2020

# First Steps: 2012

- Second working experience as an engineer
- PhD thesis in crypto
- Some research in crypto







- Engineering school
- Some lectures in cryptography
- 2 internships at Thales in crypto

- First working experience as an engineer
- No PhD thesis
- Some research in crypto

Oberthur Technologies

Grenoble INP ENSIMAG

PhD thesis

2010

2015

2020

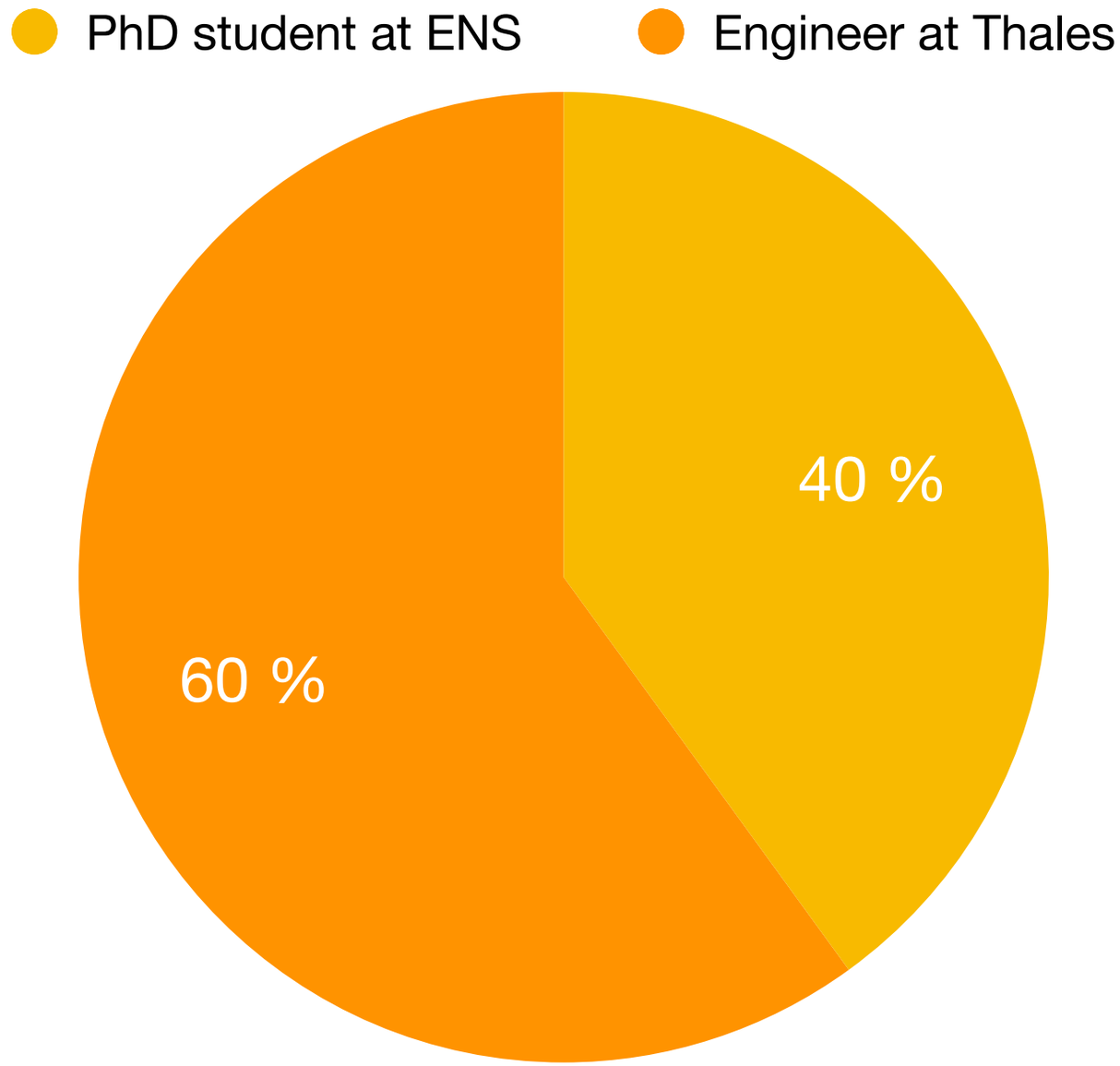
ECOLE NORMALE SUPERIEURE LETTRES SCIENCES

THALES

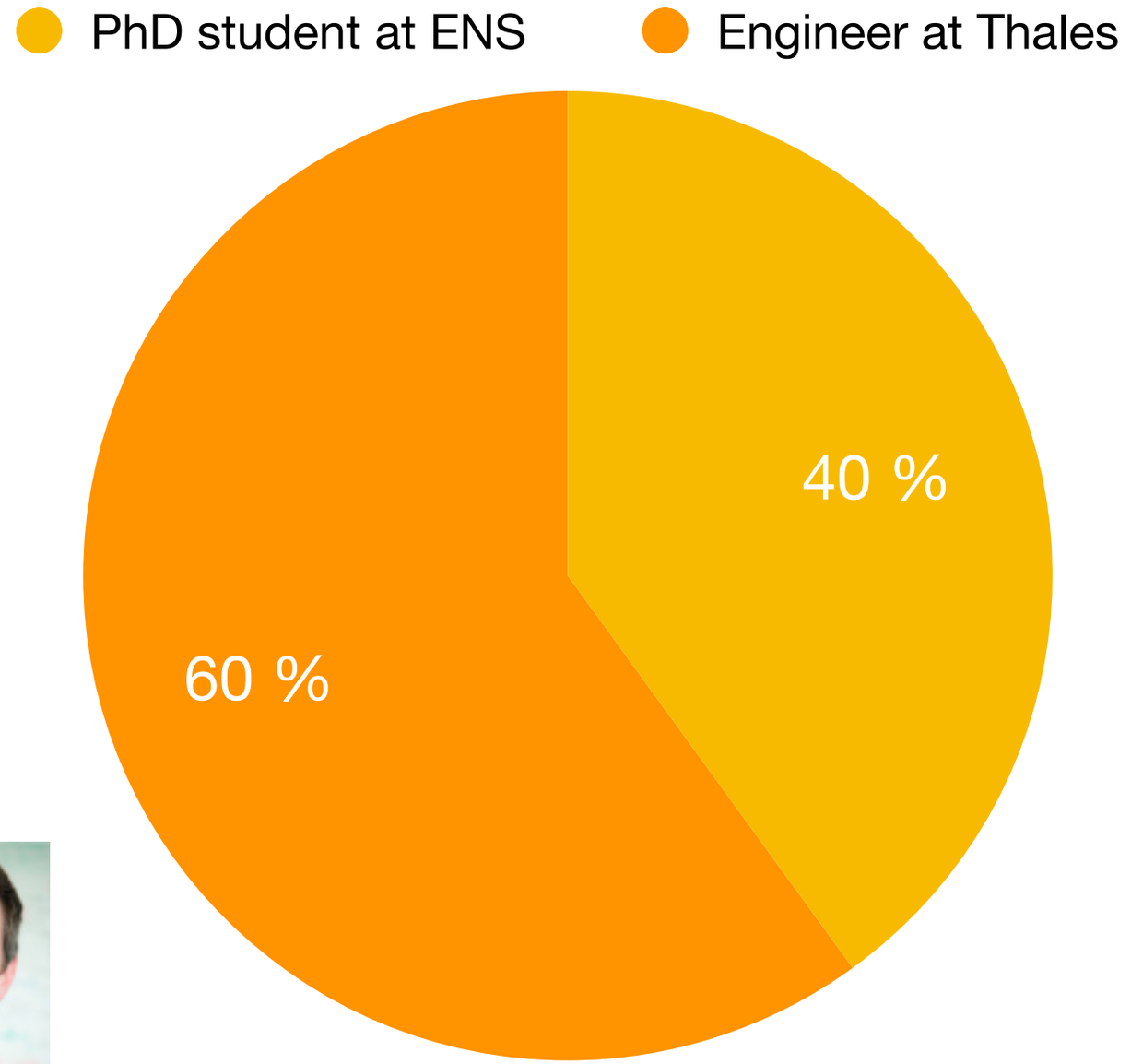


CryptoExperts

# First Steps: 2012



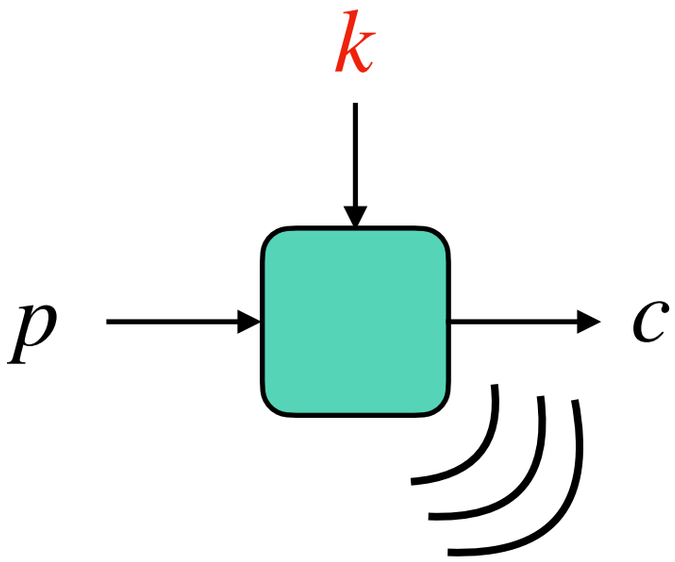
# First Steps: 2012





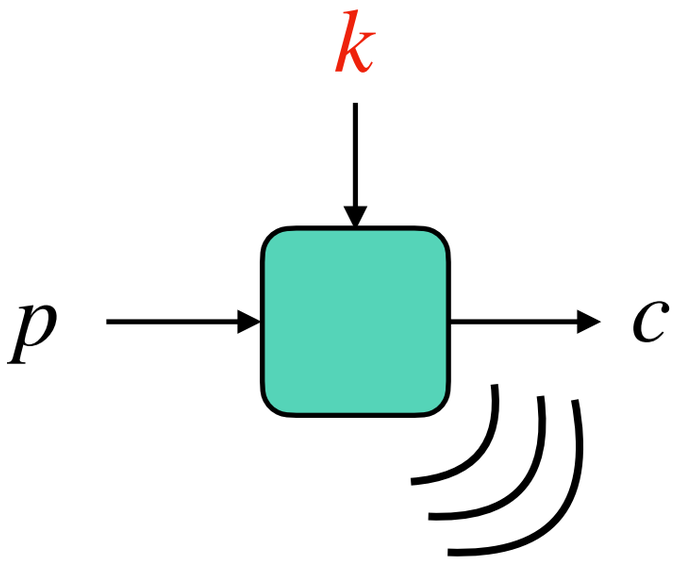

CryptoExperts

# Side-Channel Countermeasures




Problem: the leakage is key-dependent

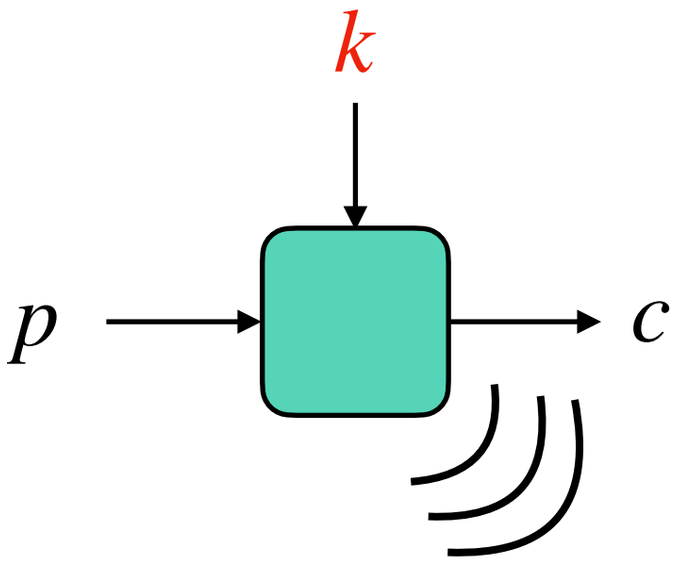
# Side-Channel Countermeasures




Problem: the leakage is key-dependent

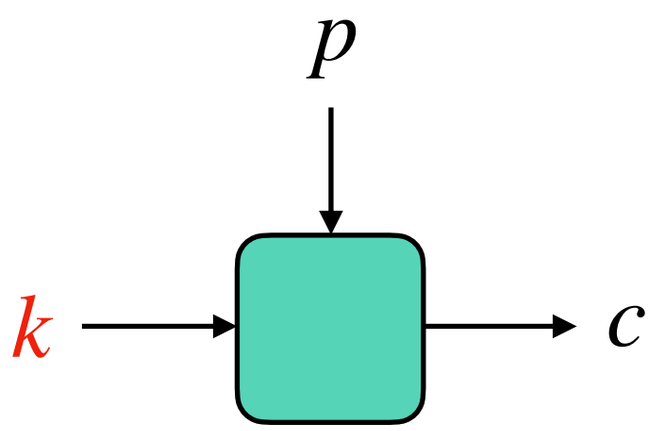
## Solution 1: Fresh Re-Keying (regularly change the key)

# Side-Channel Countermeasures

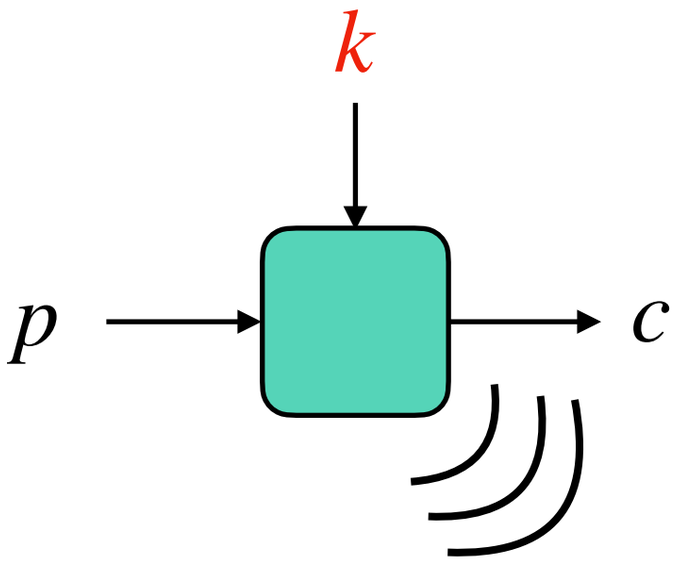



Problem: the leakage is key-dependent

## Solution 1: Fresh Re-Keying (regularly change the key)

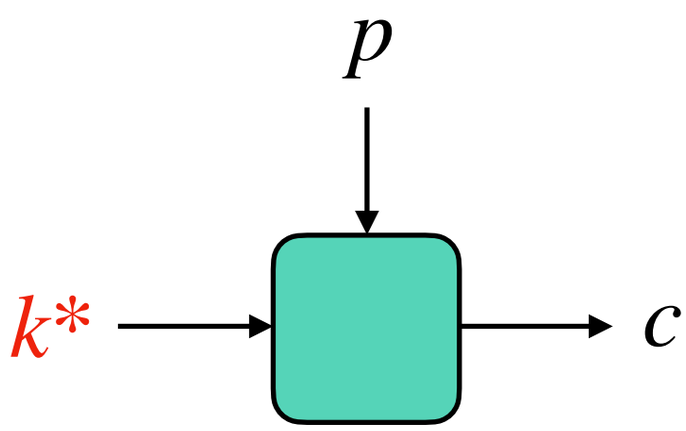


# Side-Channel Countermeasures

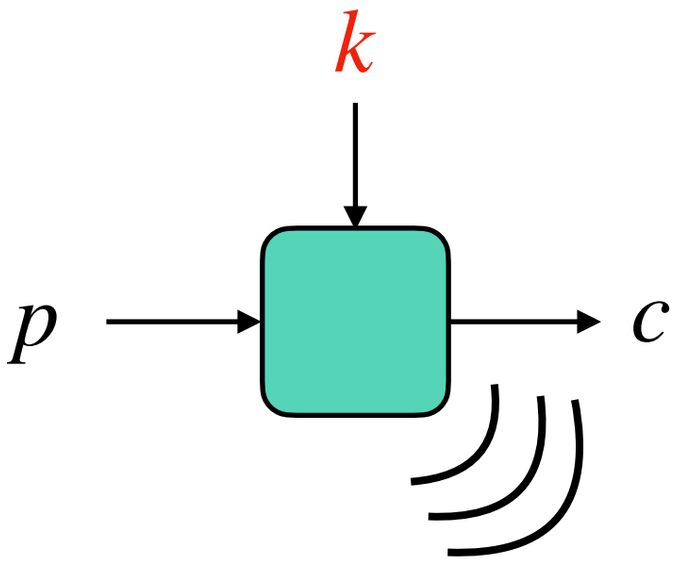



Problem: the leakage is key-dependent

## Solution 1: Fresh Re-Keying (regularly change the key)

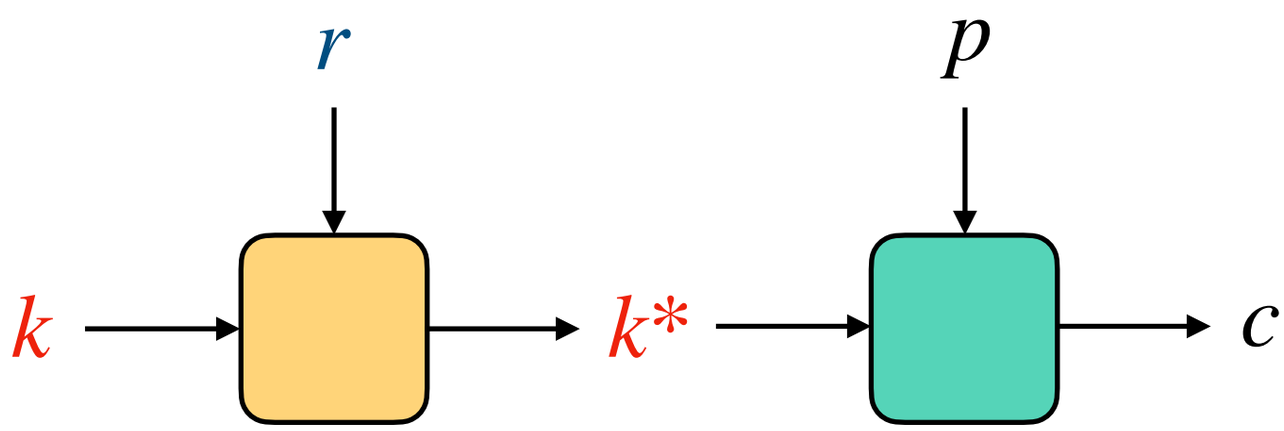


# Side-Channel Countermeasures

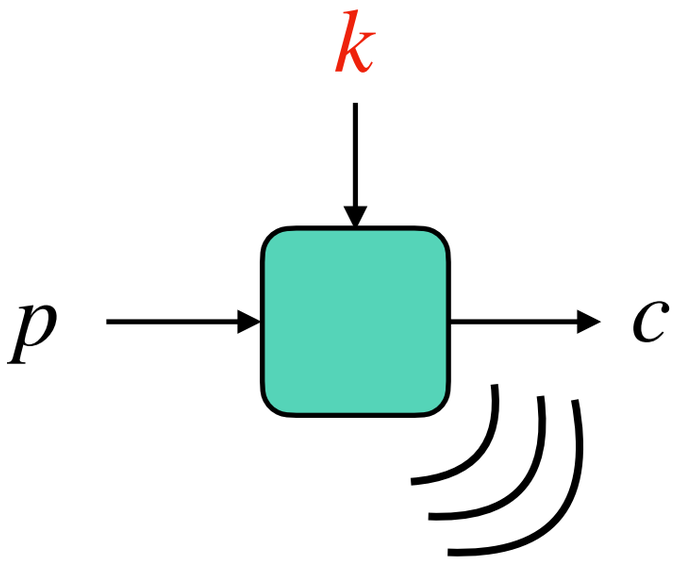



Problem: the leakage is key-dependent

## Solution 1: Fresh Re-Keying (regularly change the key)



# Side-Channel Countermeasures

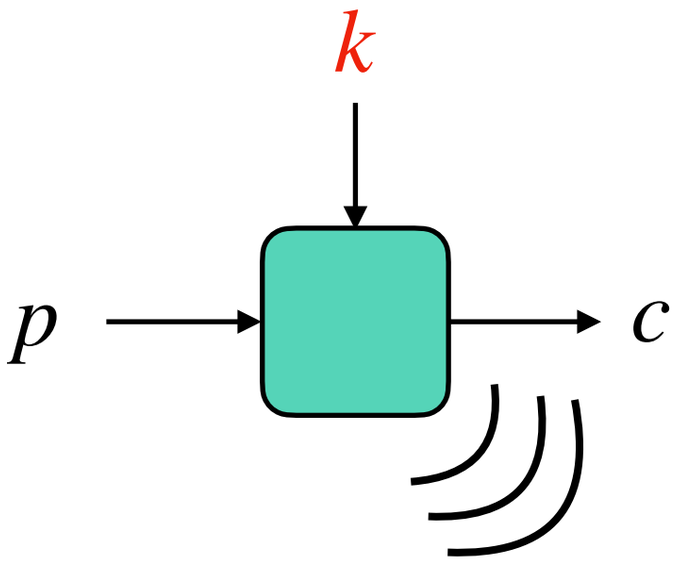



Problem: the leakage is key-dependent

First research idea:
*Discussion with a researcher at the conference CARDIS'12*

> Further developed with my supervisors and then published at CHES'13

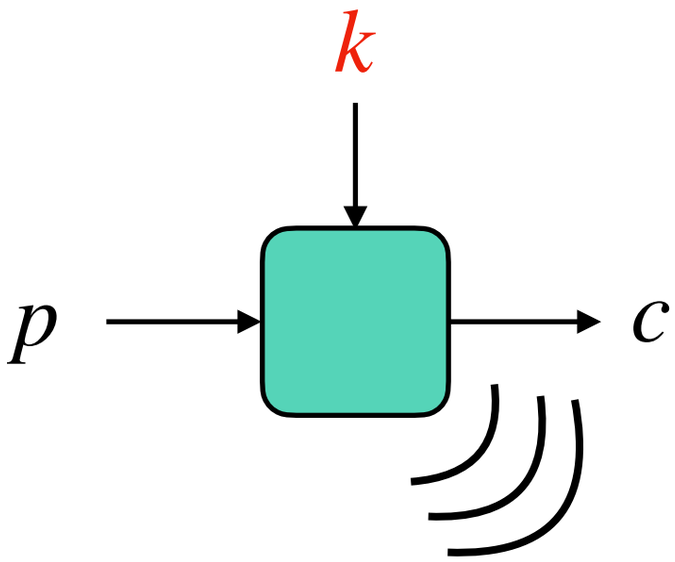
# Side-Channel Countermeasures




Problem: the leakage is key-dependent

## Solution 2: Masking (make the leakage random)

# Side-Channel Countermeasures

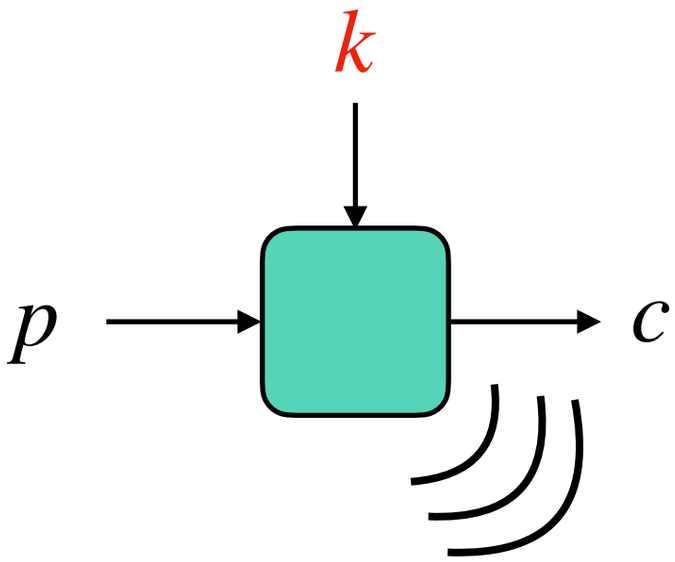



Problem: the leakage is key-dependent

## Solution 2: Masking (make the leakage random)

for each sensitive value $v \leftarrow f(p, k)$

# Side-Channel Countermeasures



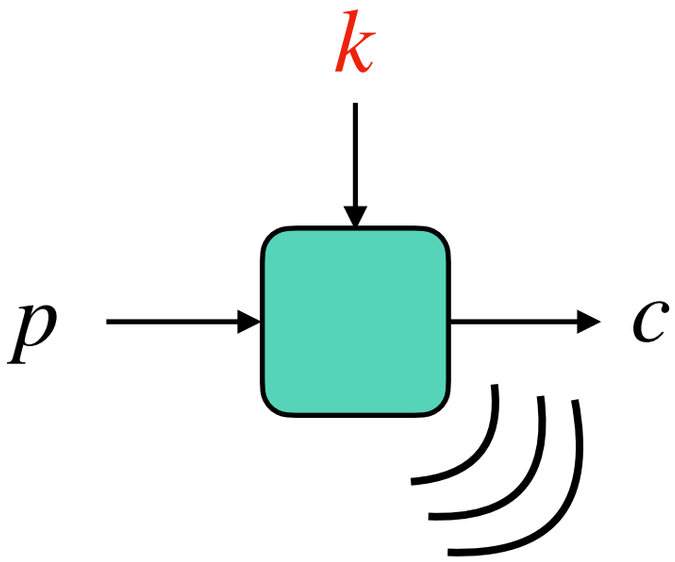

Problem: the leakage is key-dependent

## Solution 2: Masking (make the leakage random)

for each sensitive value $v \leftarrow f(p, k)$

$$v_1 \leftarrow \$ \qquad v_2 \leftarrow \$ \qquad \cdots \qquad v_{n-1} \leftarrow \$$$

# Side-Channel Countermeasures



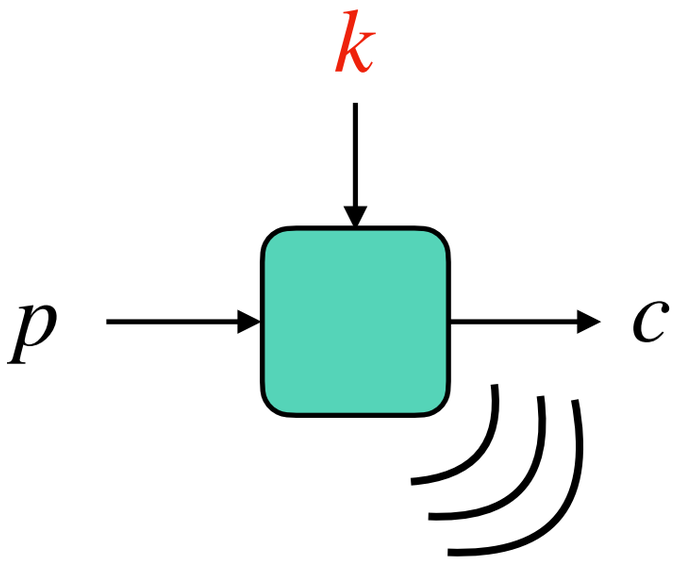

Problem: the leakage is key-dependent

## Solution 2: Masking (make the leakage random)

for each sensitive value $v \leftarrow f(p, k)$

$$v_0 \leftarrow v \oplus \left( \bigoplus_{i=1}^{n-1} v_i \right) \qquad v_1 \leftarrow \$ \qquad v_2 \leftarrow \$ \qquad \cdots \qquad v_{n-1} \leftarrow \$$$

# Side-Channel Countermeasures




Problem: the leakage is key-dependent

Research ideas from:

- my supervisors
- collaborative projects
- discussion with other researchers (seminars, conferences)

Oberthur Technologies

Grenoble INP ENSIMAG

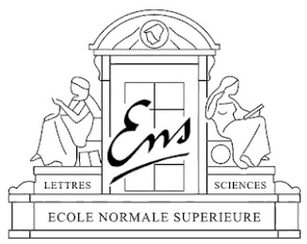
*Summer 2014*

PhD thesis

2010

2015

2020




THALES



CryptoExperts

# Mid-PhD thesis

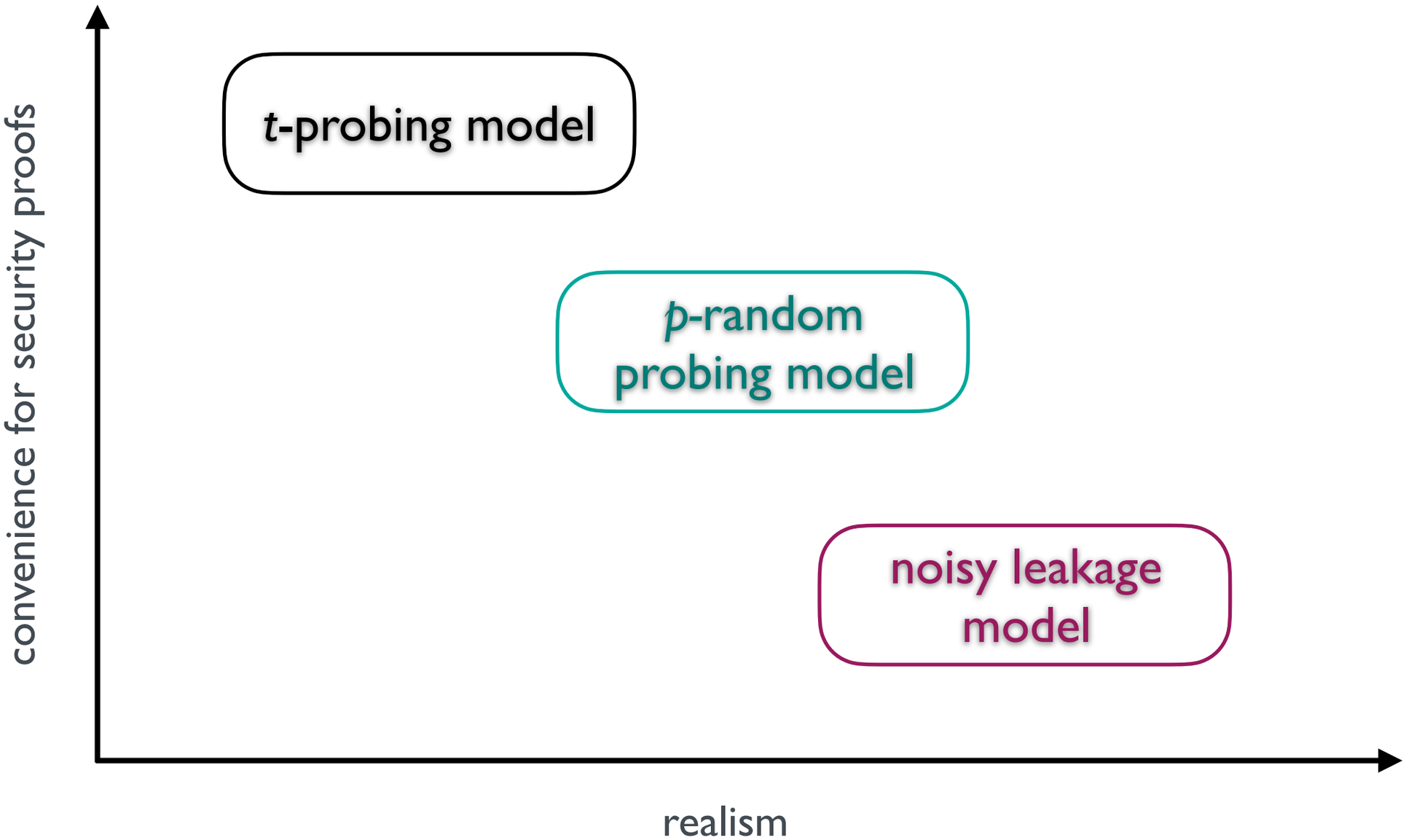
# Mid-PhD thesis

IMDEA+ team: Gilles Barthe, François Dupressoir, Benjamin Grégoire and Pierre-Yves Strub

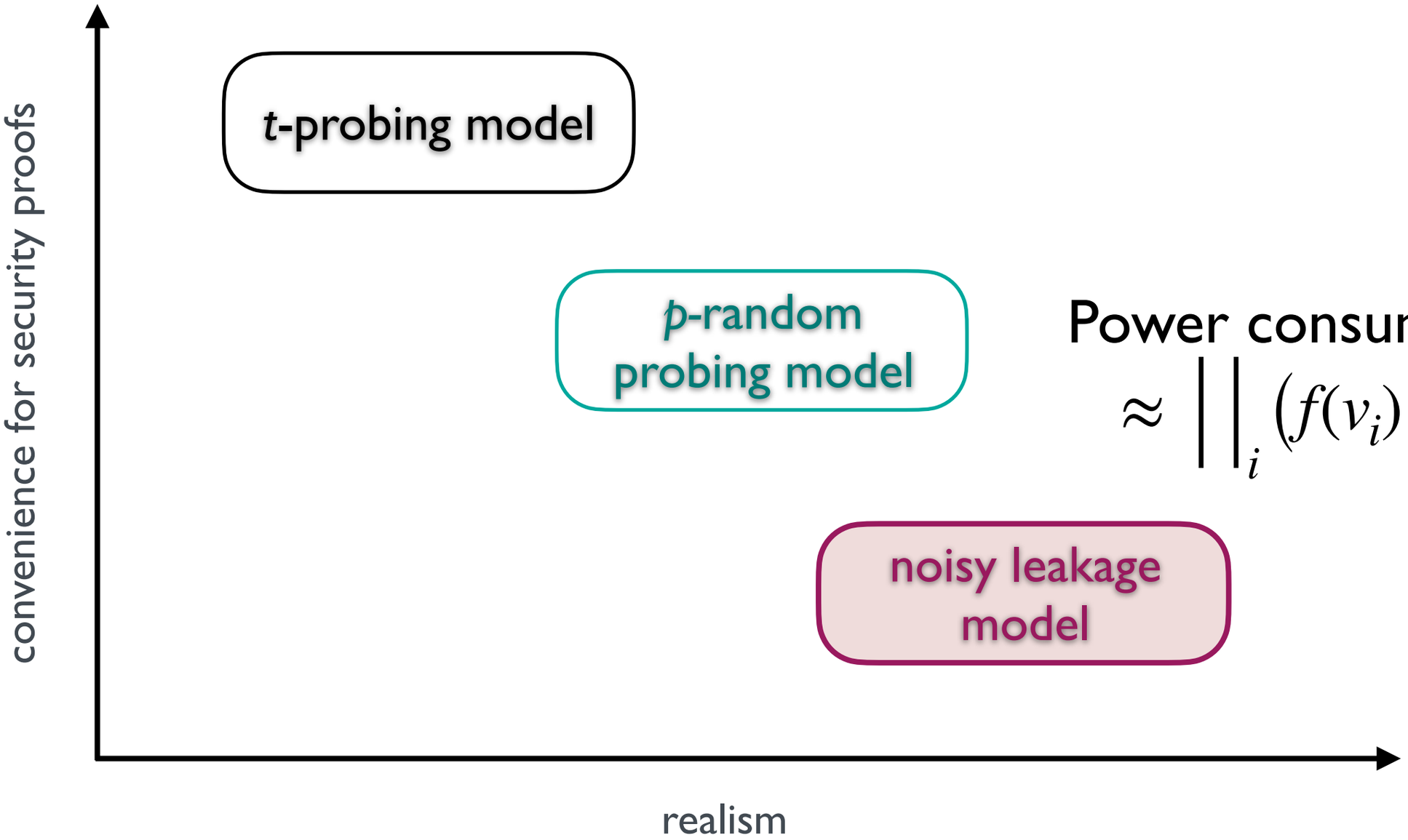
# How to Use Formal Methods for Masking?

# First Step: How to Prove the Security of the Masked Implementations?

# Leakage Models



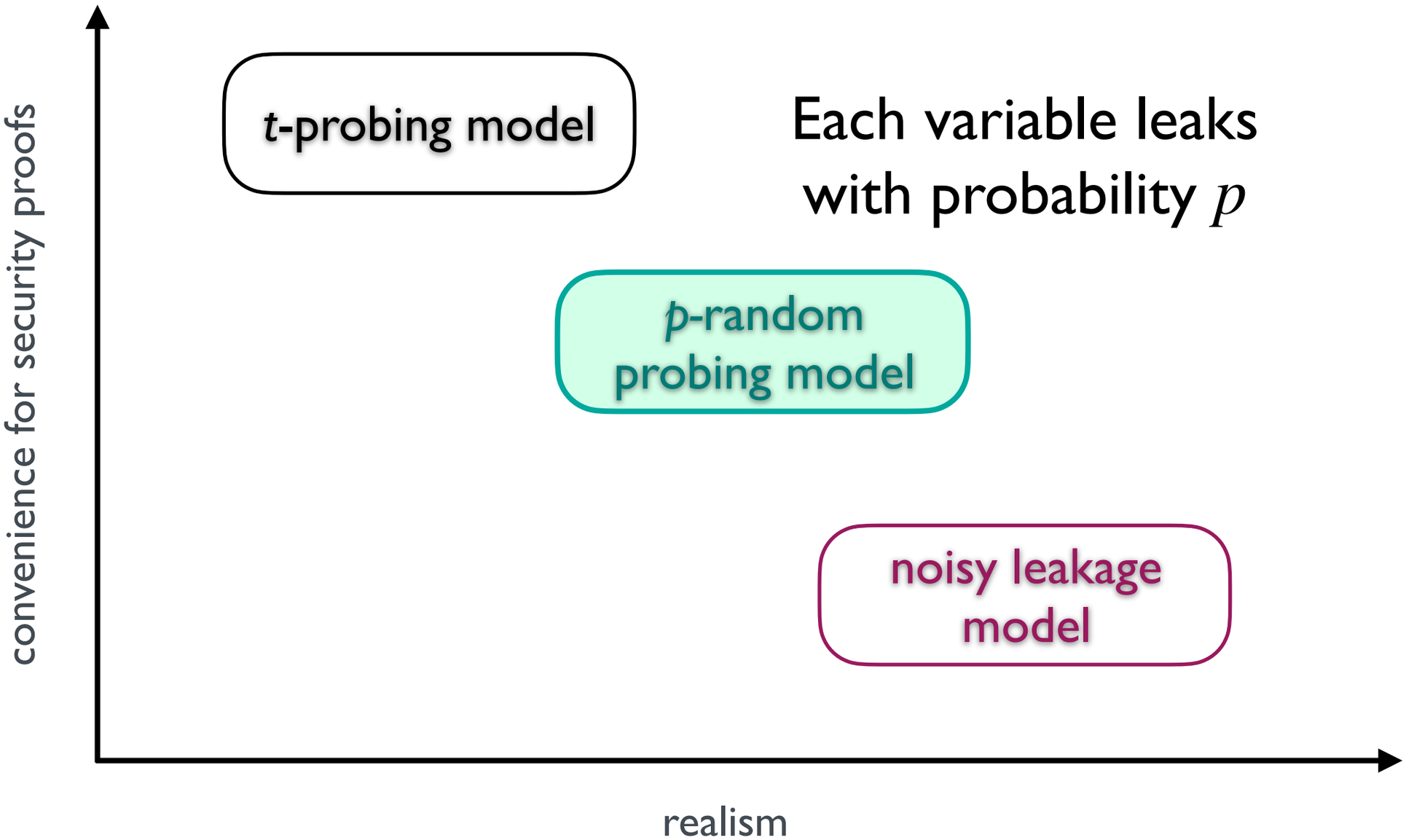
# Leakage Models



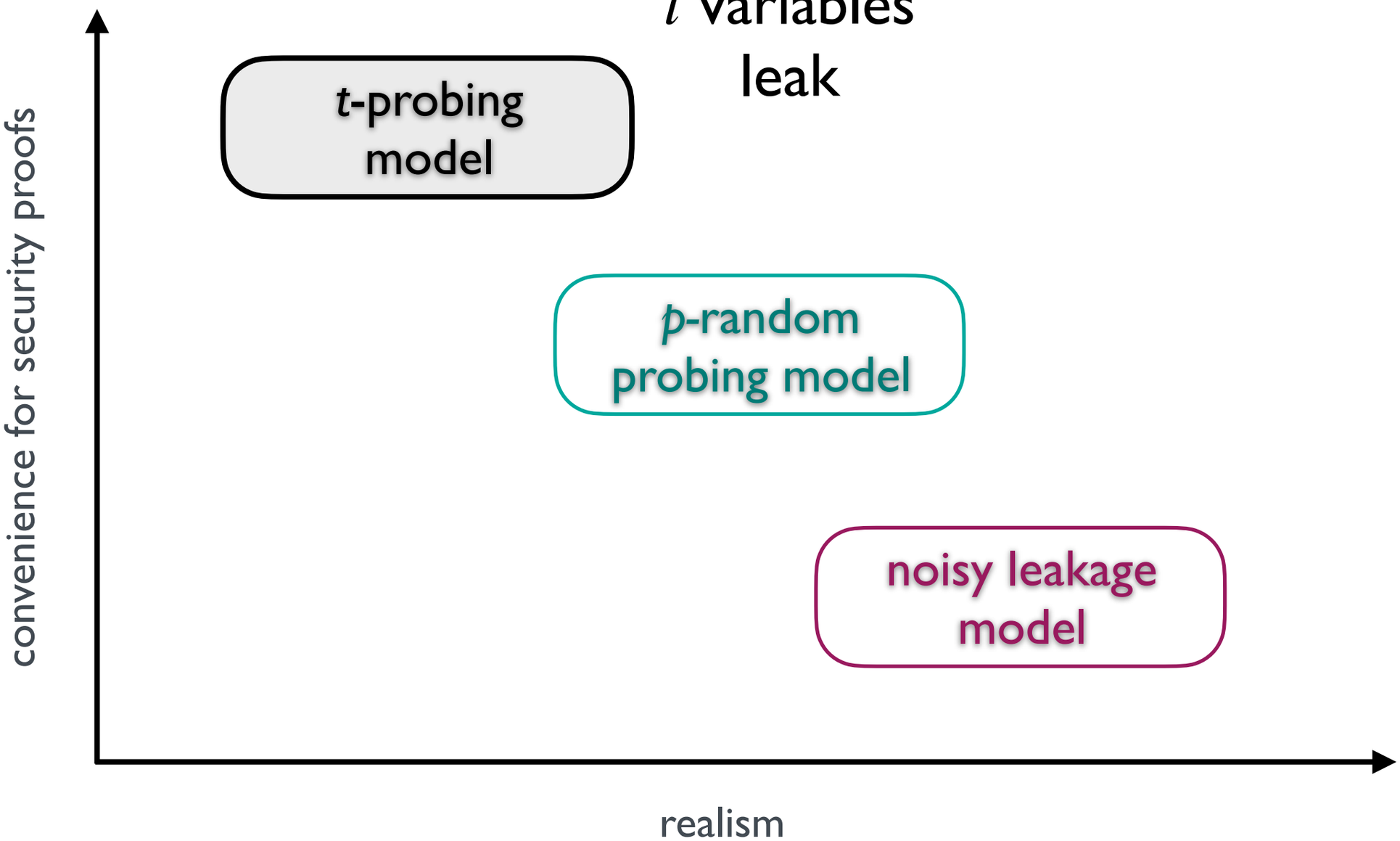

Power consumption

$$\approx \left|\left|_i \left(f(v_i) + \eta\right)\right.\right.$$

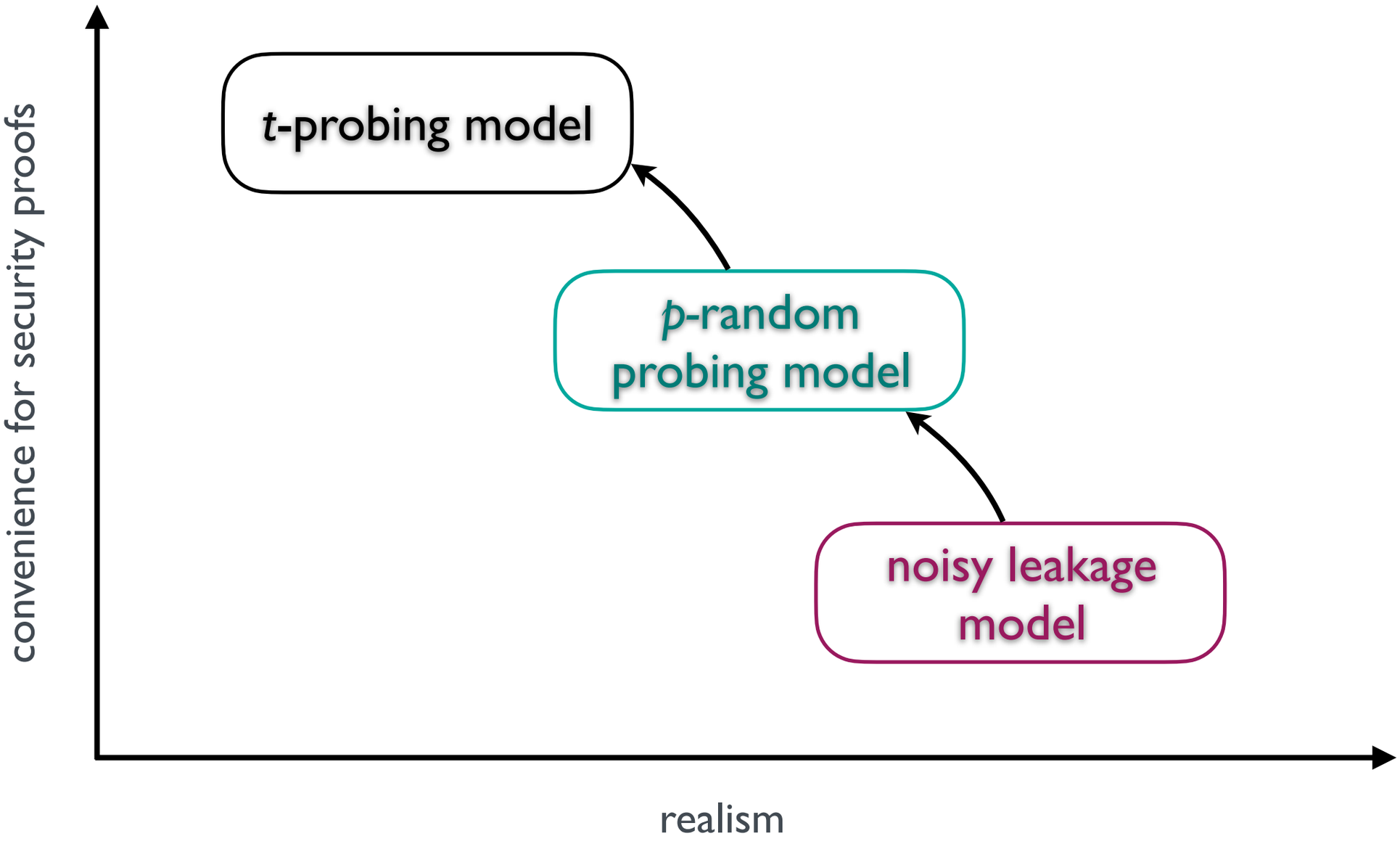
# Leakage Models



# Leakage Models



# Leakage Models




**[EC:DDF14]** *Unifying Leakage Models: From Probing Attacks to Noisy Leakage*. Alexandre Duc, Stefan Dziembowski, Sebastian Faust. Eurocrypt 2014

# Probing Security [ISW03]

Attacker model: adversary gets the exact values of $t$ intermediate variables

Security: any set of *at most* $t$ intermediate variables must be independent of the secrets

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

2 shares

function example$(a_0, a_1, b_0, b_1)$

$r \leftarrow \$$

$u \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow u \oplus r$

$v \leftarrow a_1 \cdot b_1$

$x \leftarrow a_0 \cdot b_1$

$w \leftarrow v \oplus x$

$y \leftarrow w \oplus r$

$z \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow y \oplus z$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

2 shares

1-probing secure?

function example$(a_0, a_1, b_0, b_1)$

$r \leftarrow \$$

$u \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow u \oplus r$

$v \leftarrow a_1 \cdot b_1$

$x \leftarrow a_0 \cdot b_1$

$w \leftarrow v \oplus x$

$y \leftarrow w \oplus r$

$z \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow y \oplus z$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

Independent from secrets?

function example$(a_0, a_1, b_0, b_1)$

$r \leftarrow \$$

$u \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow u \oplus r$

$v \leftarrow a_1 \cdot b_1$

$x \leftarrow a_0 \cdot b_1$

$w \leftarrow v \oplus x$

$y \leftarrow w \oplus r$

$z \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow y \oplus z$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

Independent from secrets?

function example$(a_0, a_1, b_0, b_1)$

$r \leftarrow \$$

$u \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow u \oplus r$

$v \leftarrow a_1 \cdot b_1$

$x \leftarrow a_0 \cdot b_1$

$w \leftarrow v \oplus x$

$y \leftarrow w \oplus r$

$z \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow y \oplus z$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

Independent from secrets?

function example$(a_0, a_1, b_0, b_1)$

$r \leftarrow \$$

$u \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow u \oplus r$

$v \leftarrow a_1 \cdot b_1$

$x \leftarrow a_0 \cdot b_1$

$w \leftarrow v \oplus x$

$y \leftarrow w \oplus r$

$z \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow y \oplus z$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

Independent from secrets?

function example$(a_0, a_1, b_0, b_1)$

$r \leftarrow \$$

$u \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow u \oplus r$

$v \leftarrow a_1 \cdot b_1$

$x \leftarrow a_0 \cdot b_1$

$w \leftarrow v \oplus x$

$y \leftarrow w \oplus r$

$z \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow y \oplus z$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

Independent from secrets?

function example$(a_0, a_1, b_0, b_1)$

$r \leftarrow \$$

$u \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow u \oplus r$

$v \leftarrow a_1 \cdot b_1$

$x \leftarrow a_0 \cdot b_1$

$w \leftarrow v \oplus x$

$y \leftarrow w \oplus r$

$z \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow y \oplus z$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

Independent from secrets?

function example$(a_0, a_1, b_0, b_1)$

$r \leftarrow \$$

$u \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow u \oplus r$

$v \leftarrow a_1 \cdot b_1$

$x \leftarrow a_0 \cdot b_1$

$w \leftarrow v \oplus x$

$y \leftarrow w \oplus r$

$z \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow y \oplus z$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

Independent from secrets?

$$w = v \oplus x$$

$$w = a_1 \cdot b_1 \oplus a_0 \cdot b_1$$

$$w = a \cdot b_1$$

function example$(a_0, a_1, b_0, b_1)$

$$r \leftarrow \$$$

$$u \leftarrow a_0 \cdot b_0$$

$$c_0 \leftarrow u \oplus r$$

$$v \leftarrow a_1 \cdot b_1$$

$$x \leftarrow a_0 \cdot b_1$$

$$w \leftarrow v \oplus x$$

$$y \leftarrow w \oplus r$$

$$z \leftarrow a_1 \cdot b_0$$

$$c_1 \leftarrow y \oplus z$$

return $(c_0, c_1)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

3 shares

function example$(a_0, a_1, a_2, b_0, b_1, b_2)$

$r_{00}, r_{01}, r_{02}, r_{12} \leftarrow \$$
$t \leftarrow a_0 \cdot b_0$
$c_0 \leftarrow t \oplus r_{00}$
$t \leftarrow a_0 \cdot b_1$
$t \leftarrow t \oplus r_{01}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_0 \cdot b_2$
$t \leftarrow t \oplus r_{02}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_1 \cdot b_0$
$c_1 \leftarrow t \oplus r_{01}$
$t \leftarrow a_1 \cdot b_1$
$c_1 \leftarrow c_1 \oplus t$
$\ldots$

return $(c_0, c_1, c_2)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

3 shares

33 intermediate variables

$\binom{33}{2} = 528$ couples to verify

function example$(a_0, a_1, a_2, b_0, b_1, b_2)$

$r_{00}, r_{01}, r_{02}, r_{12} \leftarrow \$$
$t \leftarrow a_0 \cdot b_0$
$c_0 \leftarrow t \oplus r_{00}$
$t \leftarrow a_0 \cdot b_1$
$t \leftarrow t \oplus r_{01}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_0 \cdot b_2$
$t \leftarrow t \oplus r_{02}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_1 \cdot b_0$
$c_1 \leftarrow t \oplus r_{01}$
$t \leftarrow a_1 \cdot b_1$
$c_1 \leftarrow c_1 \oplus t$
$\cdots$
return $(c_0, c_1, c_2)$

# Proof in the Probing Model

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret

3 shares

33 intermediate variables

$\binom{33}{2} = 528$ couples to verify

$\binom{n}{t}$ tuples to verify

function example$(a_0, a_1, a_2, b_0, b_1, b_2)$

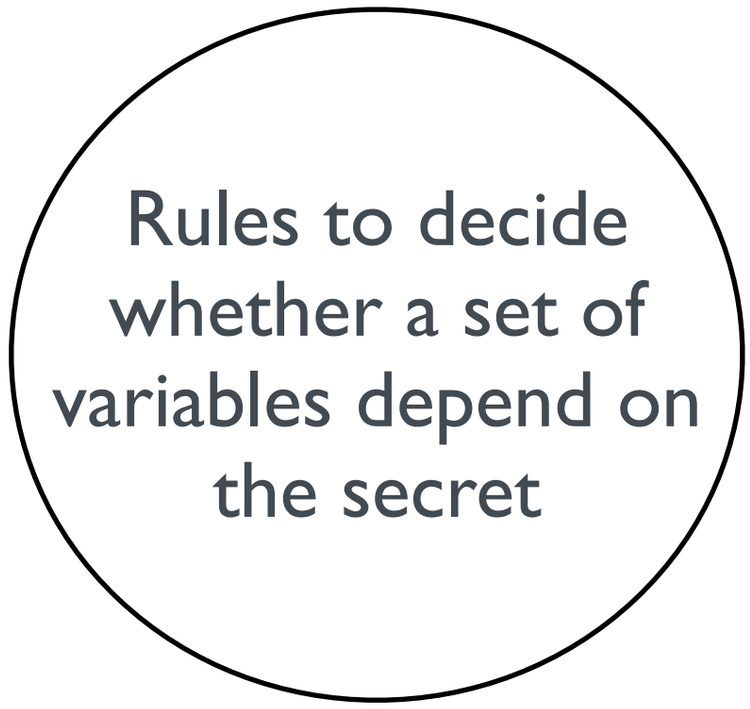
$r_{00}, r_{01}, r_{02}, r_{12} \leftarrow \$$
$t \leftarrow a_0 \cdot b_0$
$c_0 \leftarrow t \oplus r_{00}$
$t \leftarrow a_0 \cdot b_1$
$t \leftarrow t \oplus r_{01}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_0 \cdot b_2$
$t \leftarrow t \oplus r_{02}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_1 \cdot b_0$
$c_1 \leftarrow t \oplus r_{01}$
$t \leftarrow a_1 \cdot b_1$
$c_1 \leftarrow c_1 \oplus t$
$\cdots$
return $(c_0, c_1, c_2)$

# With Formal Methods

Formally verify security in the probing model with EasyCrypt

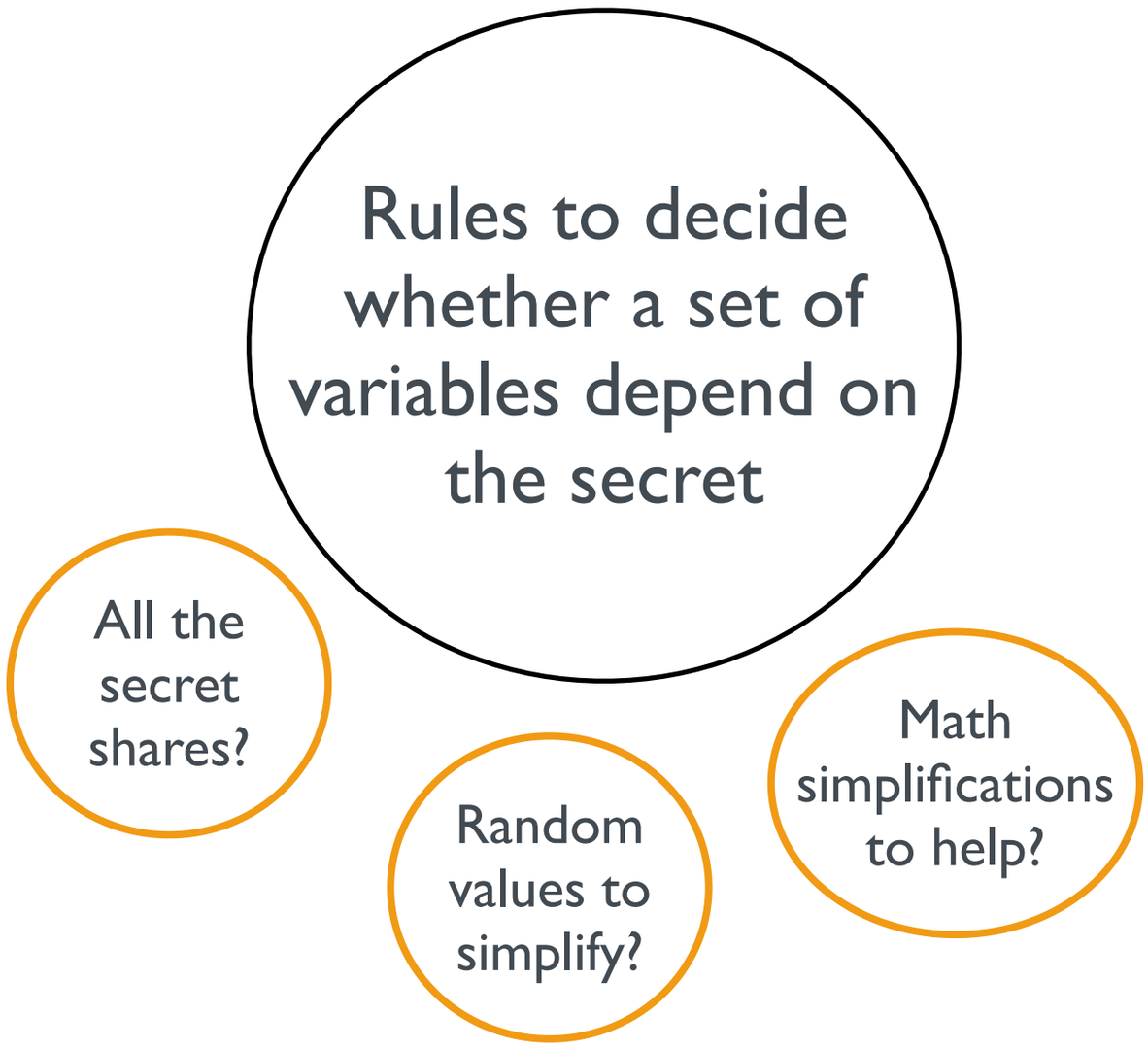
# With Formal Methods

Formally verify security in the probing model with EasyCrypt



# With Formal Methods

Formally verify security in the probing model with EasyCrypt



# With Formal Methods

Formally verify security in the probing model with EasyCrypt

Rules to decide whether a set of variables depend on the secret

All the secret shares?

Random values to simplify?

Math simplifications to help?

Instead of verifying all the sets of $t$ variables
=> verifying bigger sets

# With Formal Methods

maskVerif

# With Formal Methods

Security order $t$

function example$(a_0, a_1, a_2, b_0, b_1, b_2)$

$r_{00}, r_{01}, r_{02}, r_{12} \leftarrow \$$
$t \leftarrow a_0 \cdot b_0$
$c_0 \leftarrow t \oplus r_{00}$
$t \leftarrow a_0 \cdot b_1$
$t \leftarrow t \oplus r_{01}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_0 \cdot b_2$
$t \leftarrow t \oplus r_{02}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_1 \cdot b_0$
$c_1 \leftarrow t \oplus r_{01}$
$t \leftarrow a_1 \cdot b_1$
$c_1 \leftarrow c_1 \oplus t$
$\ldots$
return $(c_0, c_1, c_2)$

maskVerif

# With Formal Methods

Security order $t$

function example$(a_0, a_1, a_2, b_0, b_1, b_2)$

$r_{00}, r_{01}, r_{02}, r_{12} \leftarrow \$$

$t \leftarrow a_0 \cdot b_0$

$c_0 \leftarrow t \oplus r_{00}$

$t \leftarrow a_0 \cdot b_1$

$t \leftarrow t \oplus r_{01}$

$c_0 \leftarrow c_0 \oplus t$

$t \leftarrow a_0 \cdot b_2$

$t \leftarrow t \oplus r_{02}$

$c_0 \leftarrow c_0 \oplus t$

$t \leftarrow a_1 \cdot b_0$

$c_1 \leftarrow t \oplus r_{01}$

$t \leftarrow a_1 \cdot b_1$

$c_1 \leftarrow c_1 \oplus t$

…

return $(c_0, c_1, c_2)$

maskVerif

Security proof or potential attacks

# With Formal Methods

Security order $t$

function example$(a_0, a_1, a_2, b_0, b_1, b_2)$

$r_{00}, r_{01}, r_{02}, r_{12} \leftarrow \$$
$t \leftarrow a_0 \cdot b_0$
$c_0 \leftarrow t \oplus r_{00}$
$t \leftarrow a_0 \cdot b_1$
$t \leftarrow t \oplus r_{01}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_0 \cdot b_2$
$t \leftarrow t \oplus r_{02}$
$c_0 \leftarrow c_0 \oplus t$
$t \leftarrow a_1 \cdot b_0$
$c_1 \leftarrow t \oplus r_{01}$
$t \leftarrow a_1 \cdot b_1$
$c_1 \leftarrow c_1 \oplus t$
...
return $(c_0, c_1, c_2)$

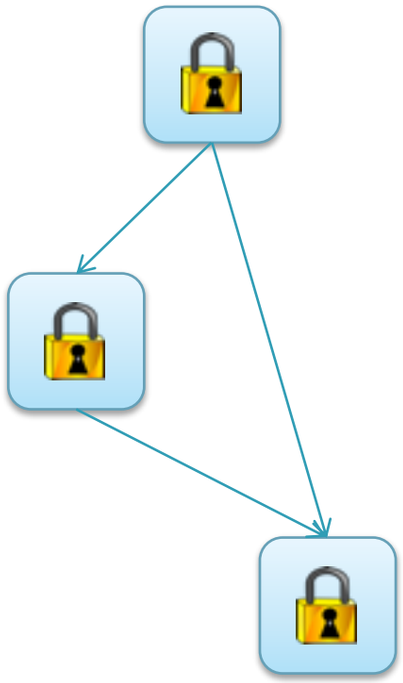
→ maskVerif →

Security proof or potential attacks

**[EC:BBDFGS15]** *Verified Proofs of Higher-Order Masking.* Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, and Pierre-Yves Strub. Eurocrypt 2015

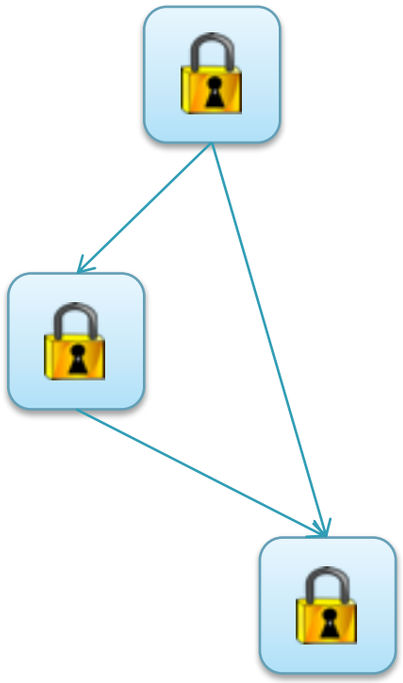
# New Ideas?

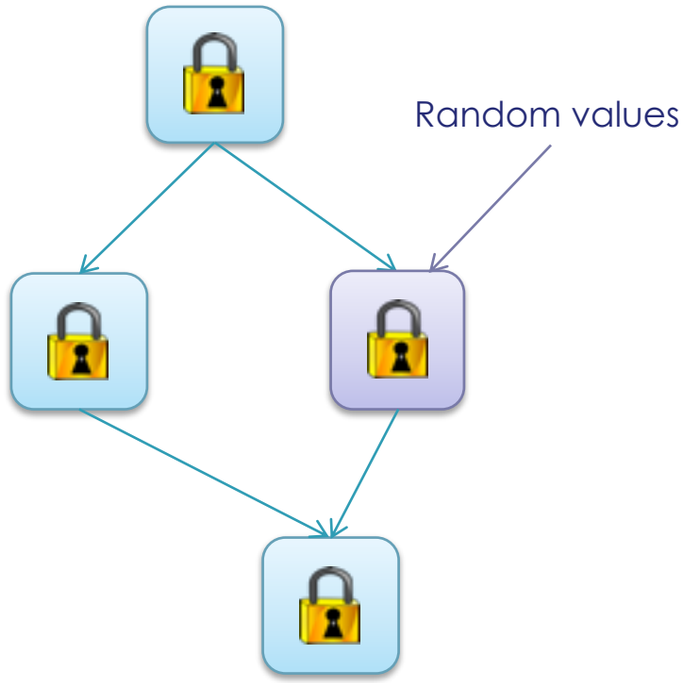
# Composition of gadgets



[1] Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. CRYPTO 2003

# Composition of gadgets

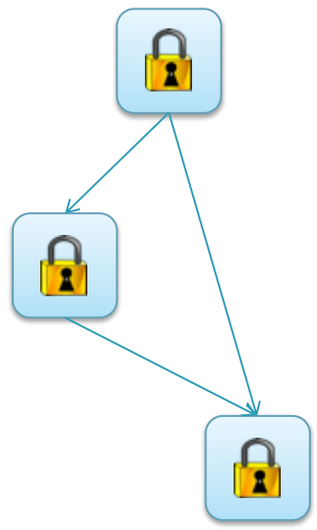





[1] Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. CRYPTO 2003
[2] M. Rivain and E. Prouff. Provably secure higher-order masking of AES. CHES 2010

# Composition of gadgets

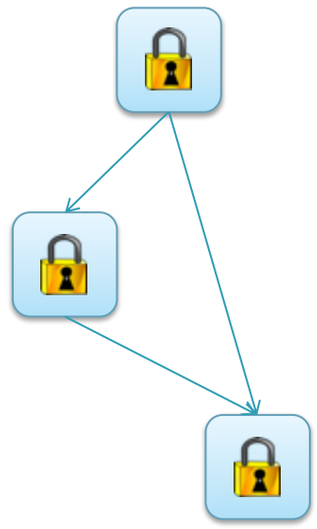
- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret
- How to reason on composition?



# Composition of gadgets

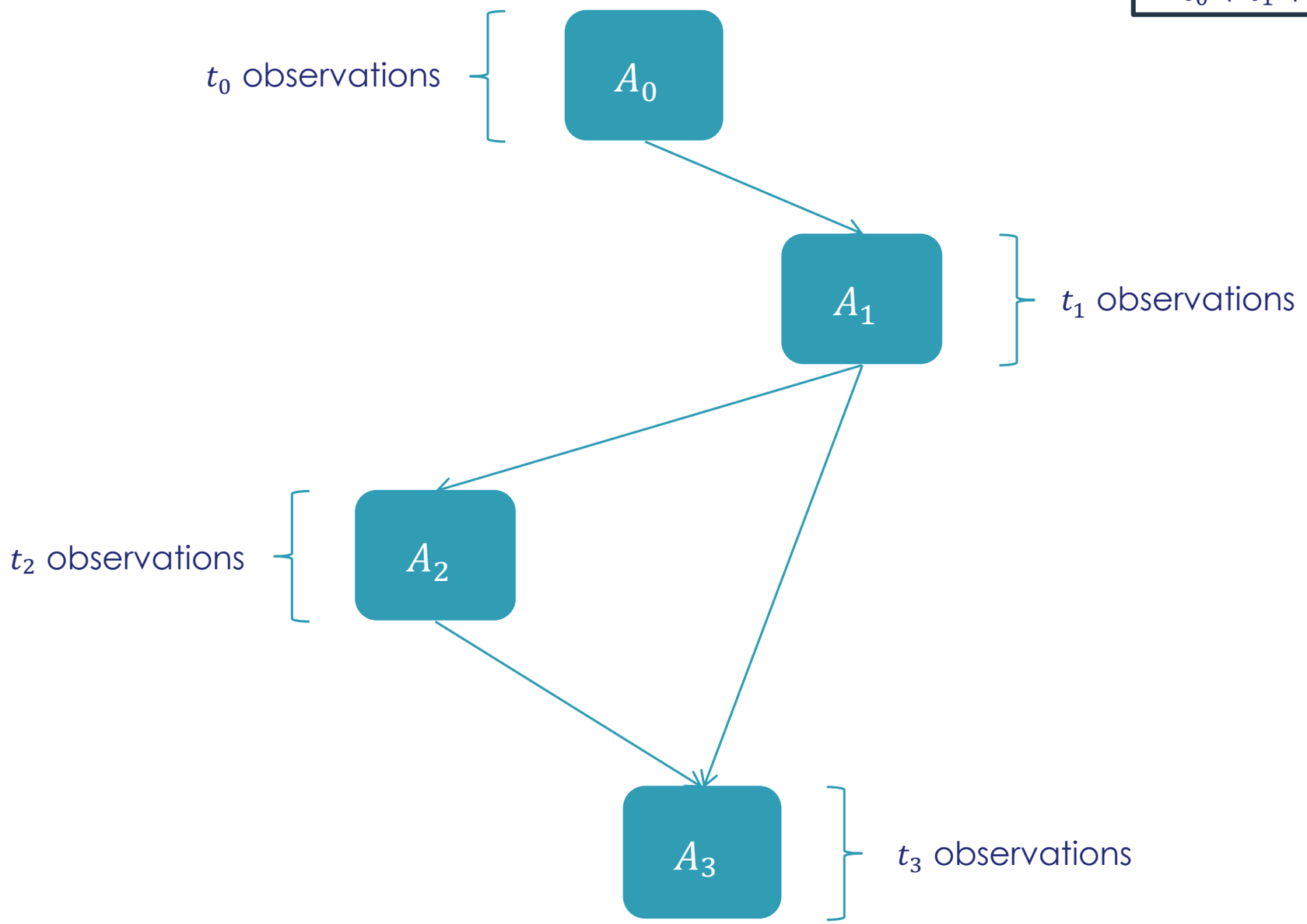
- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret
- How to reason on composition?
  - Stronger property: non-interference

any set of *t* variables can be simulated with at most *t* input shares



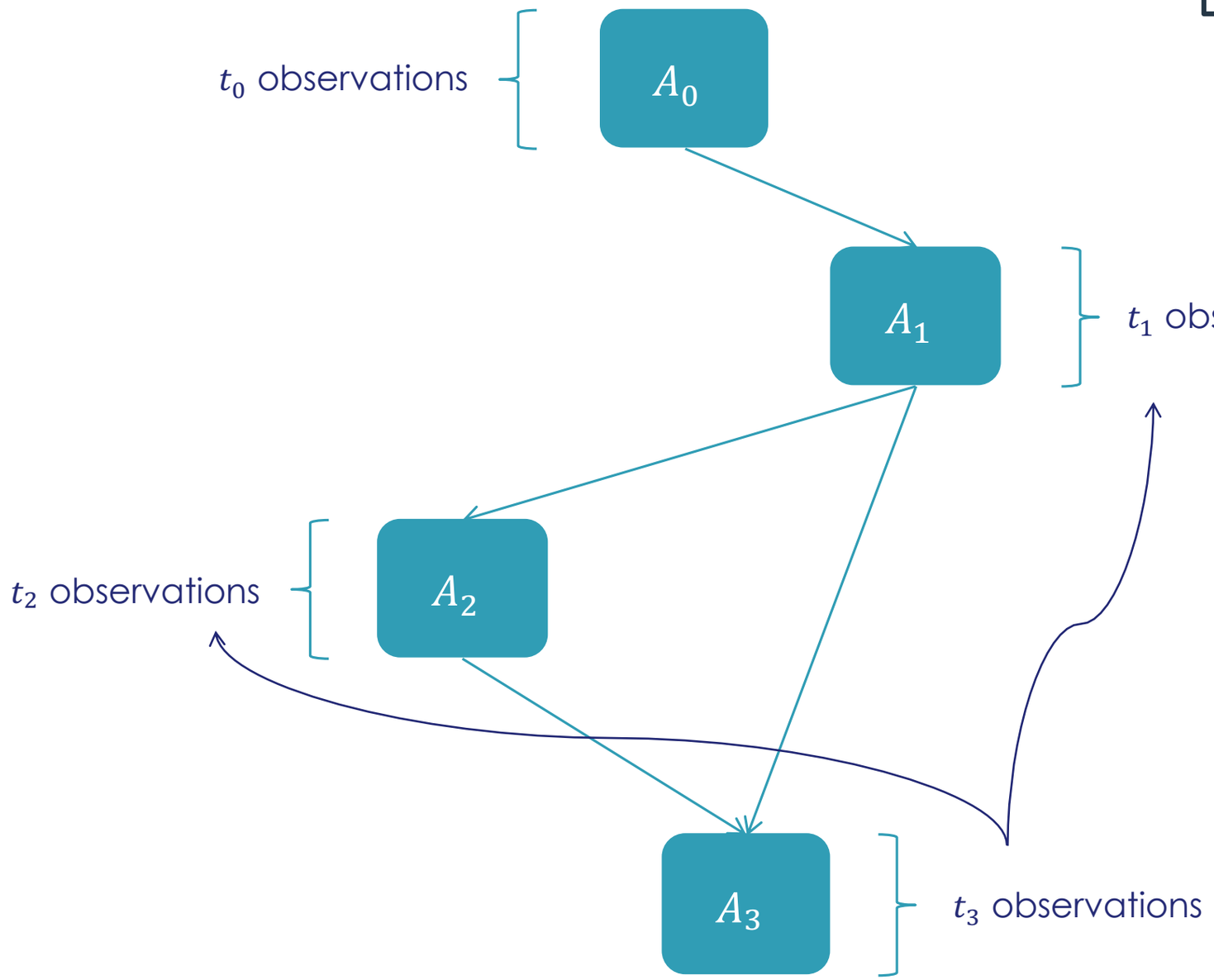
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 \leq t$$



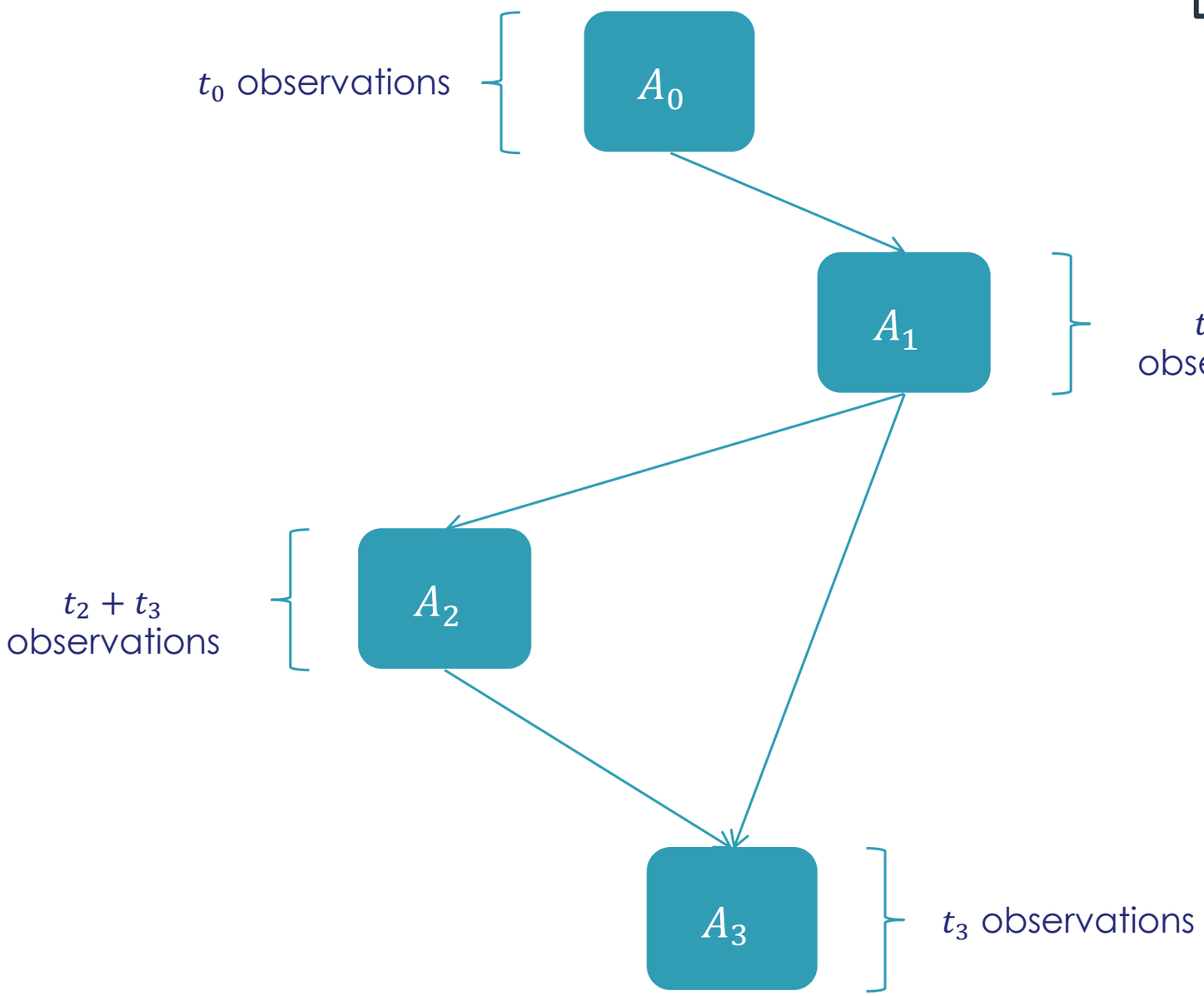
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 \leq t$$



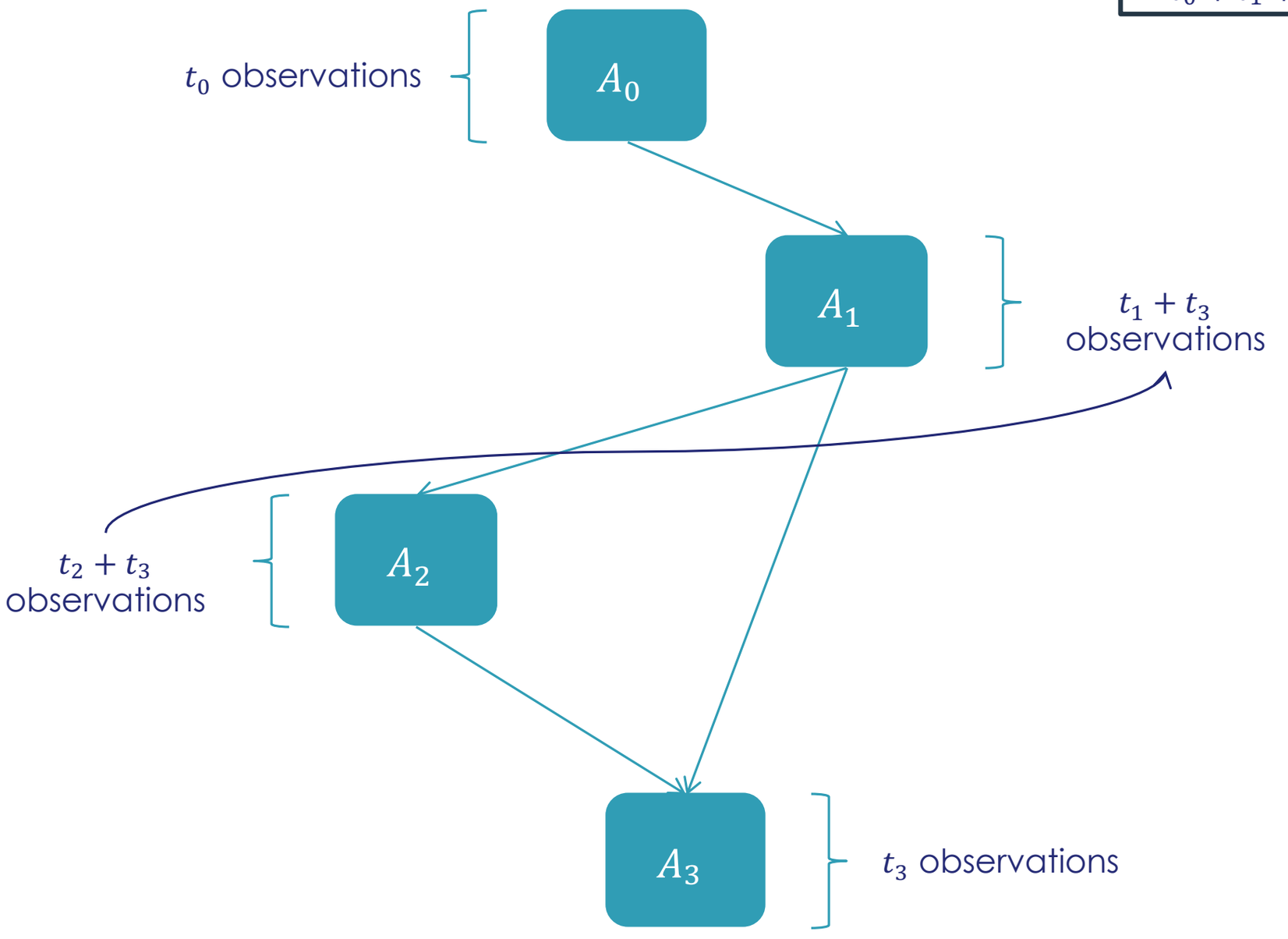
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 \leq t$$



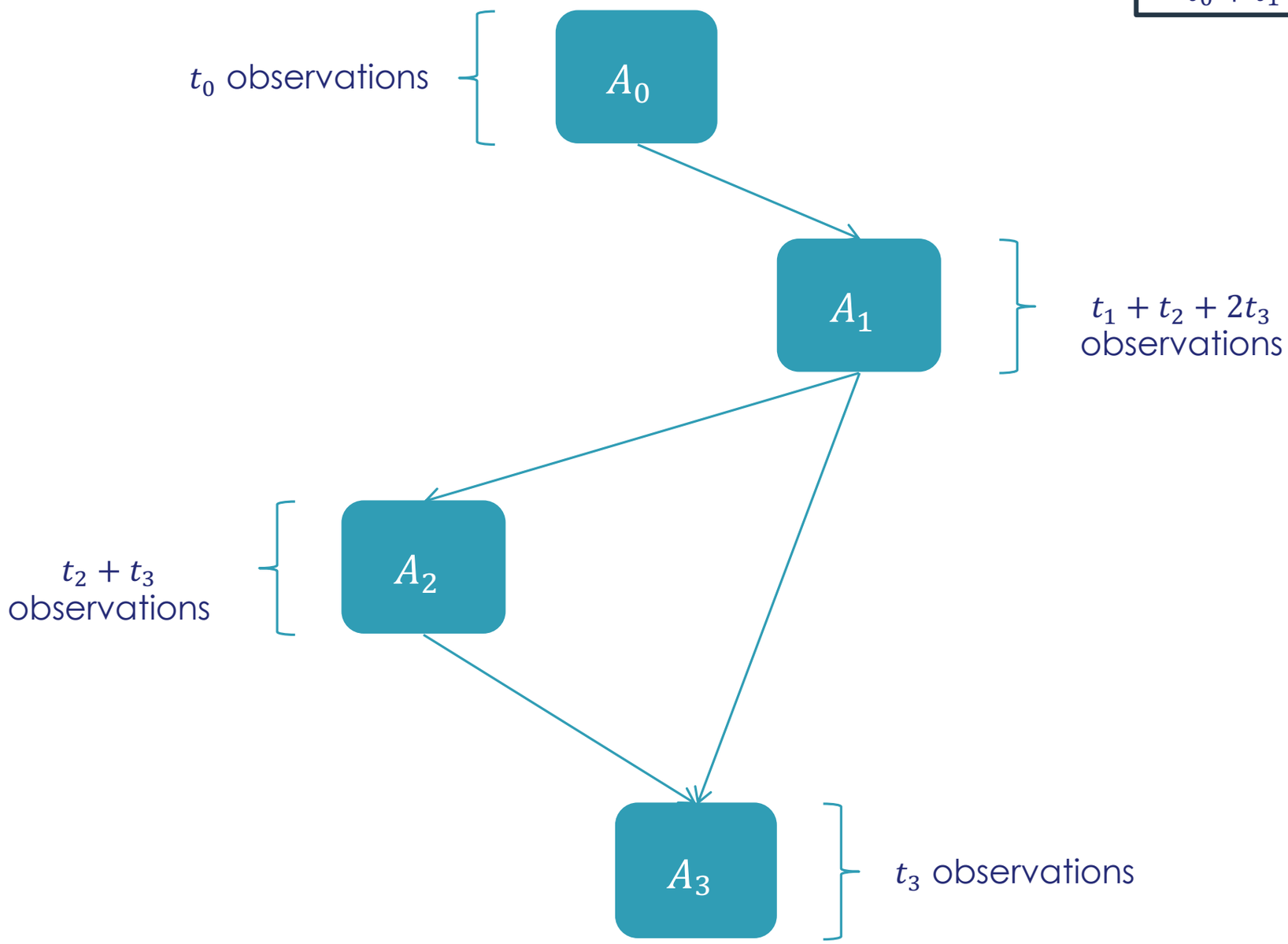
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 \leq t$$



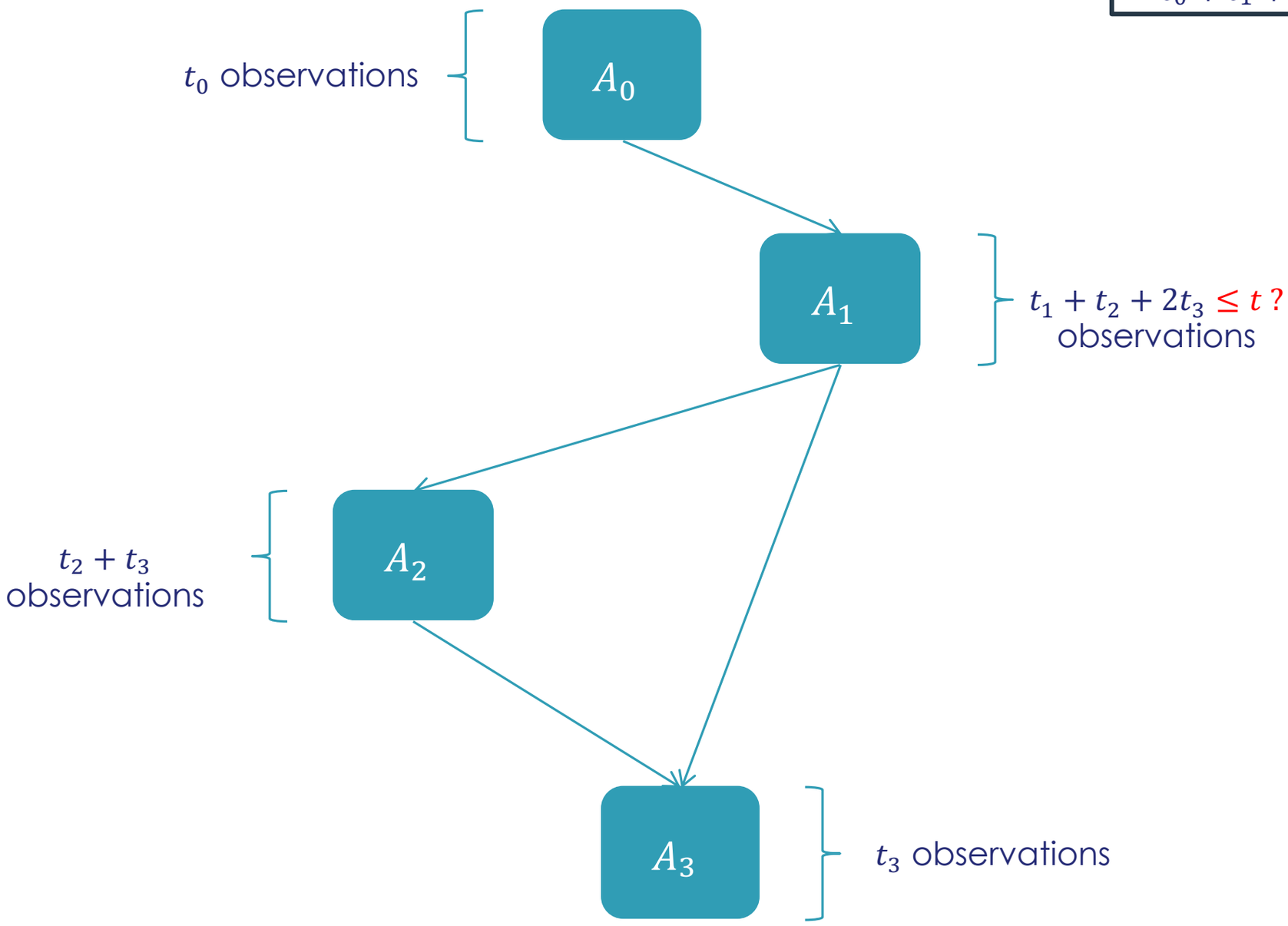
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 \leq t$$



# Composition of gadgets

$t_0 + t_1 + t_2 + t_3 \leq t$



# Composition of gadgets

- Reminder: an implementation is *t*-probing secure iff any set of at most *t* variables is independent from the secret
- How to reason on composition?
  - Stronger property: non-interference

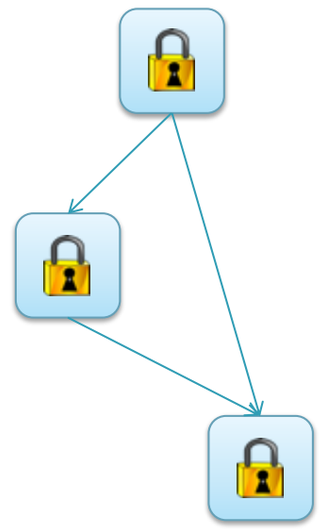
any set of $t$ variables can be simulated with at most $t$ input shares

  - **Stronger** property: strong non-interference

any set of

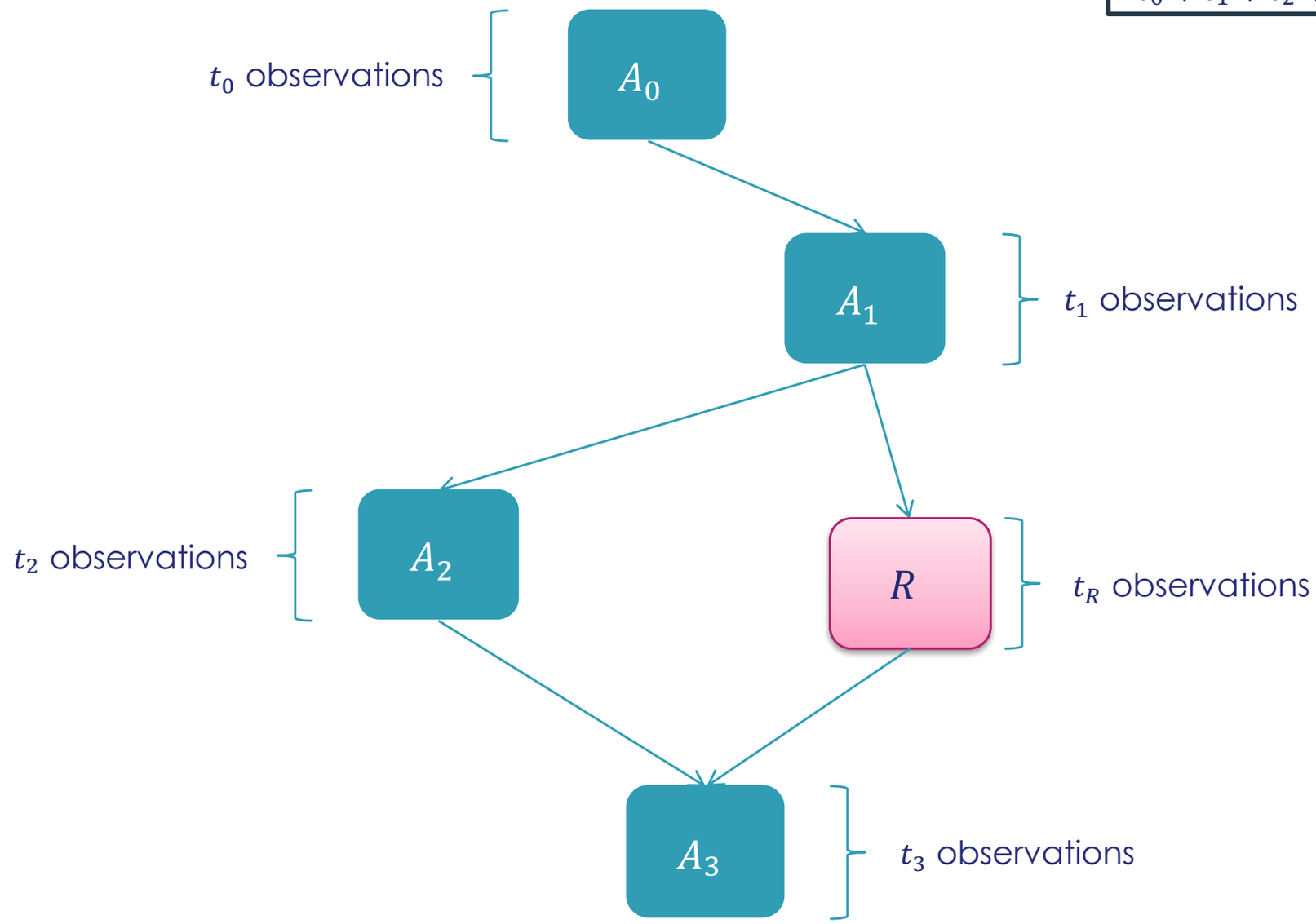
- $t_1$ internal variables
- $t_2$ output variables

can be simulated with at most $t_1$ input shares



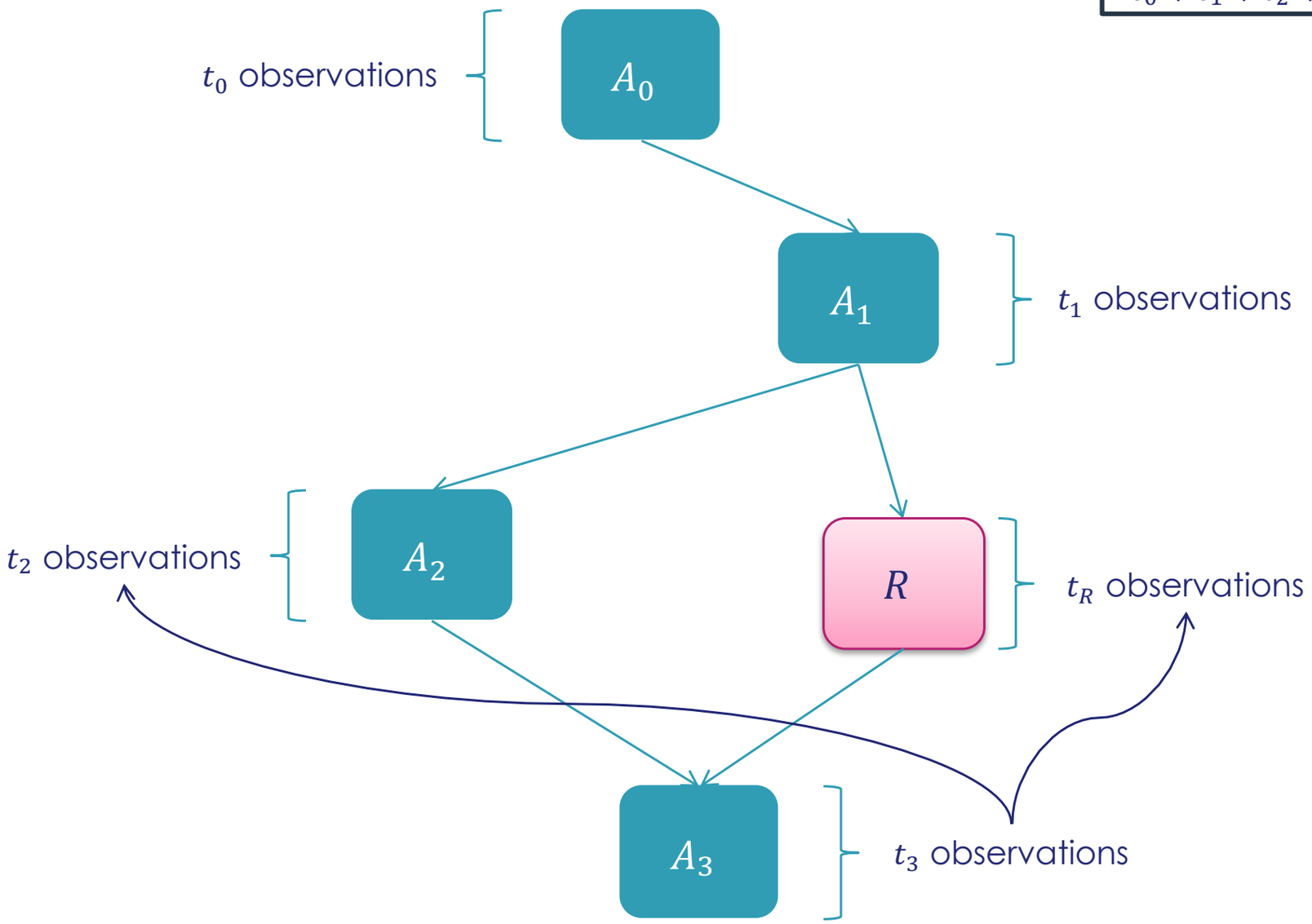
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 + t_R \leq t$$



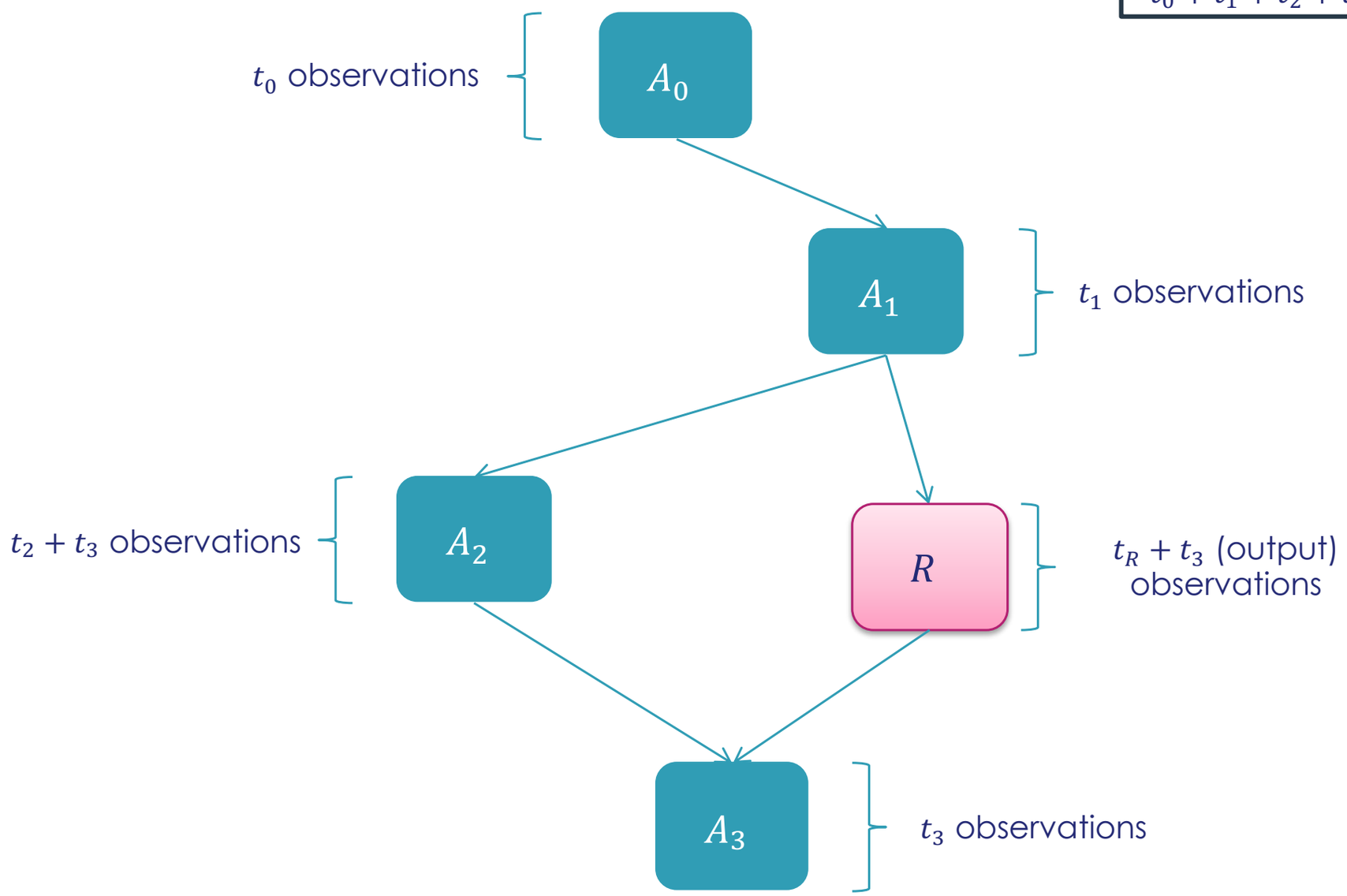
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 + t_R \leq t$$



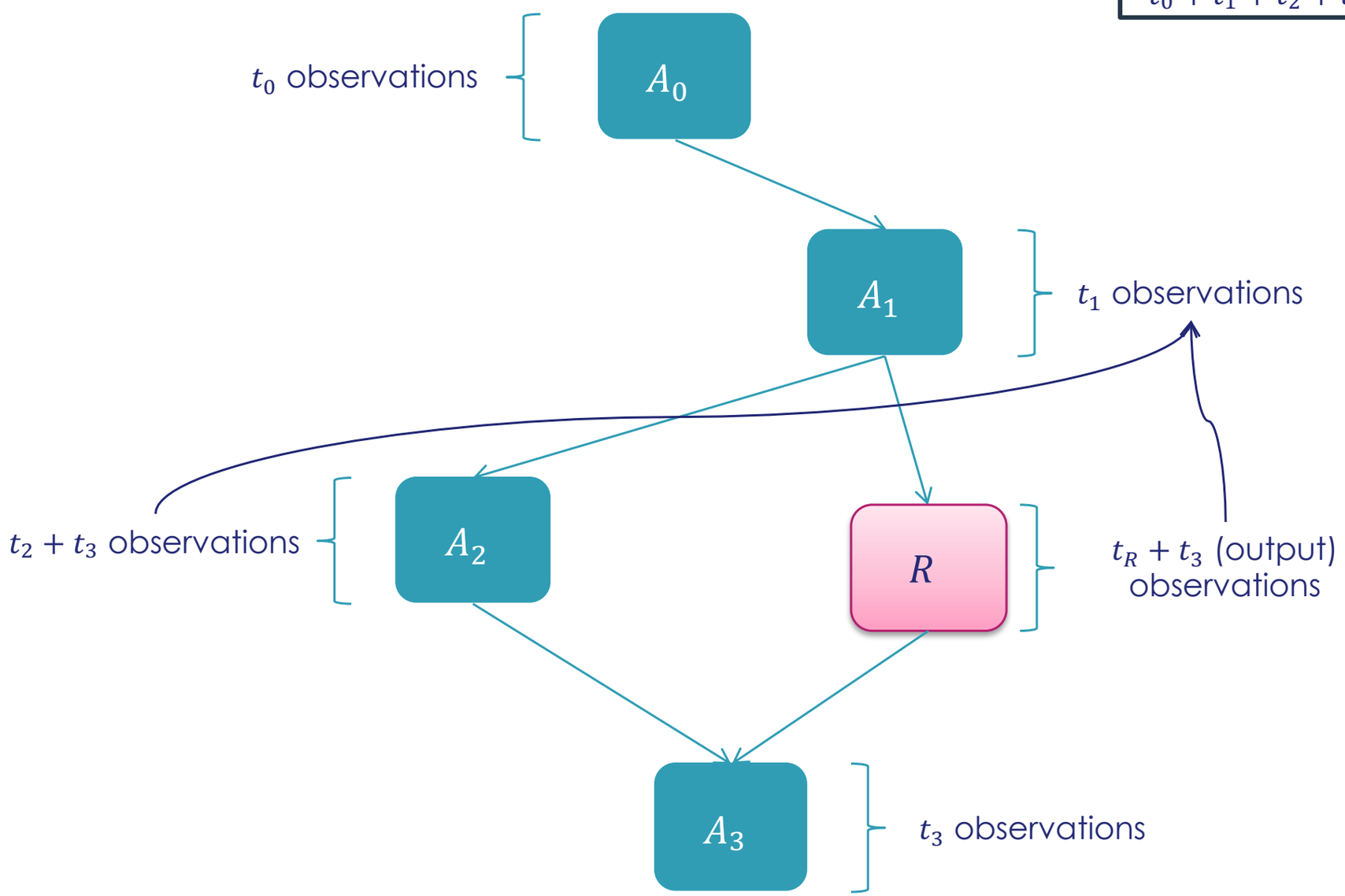
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 + t_R \leq t$$



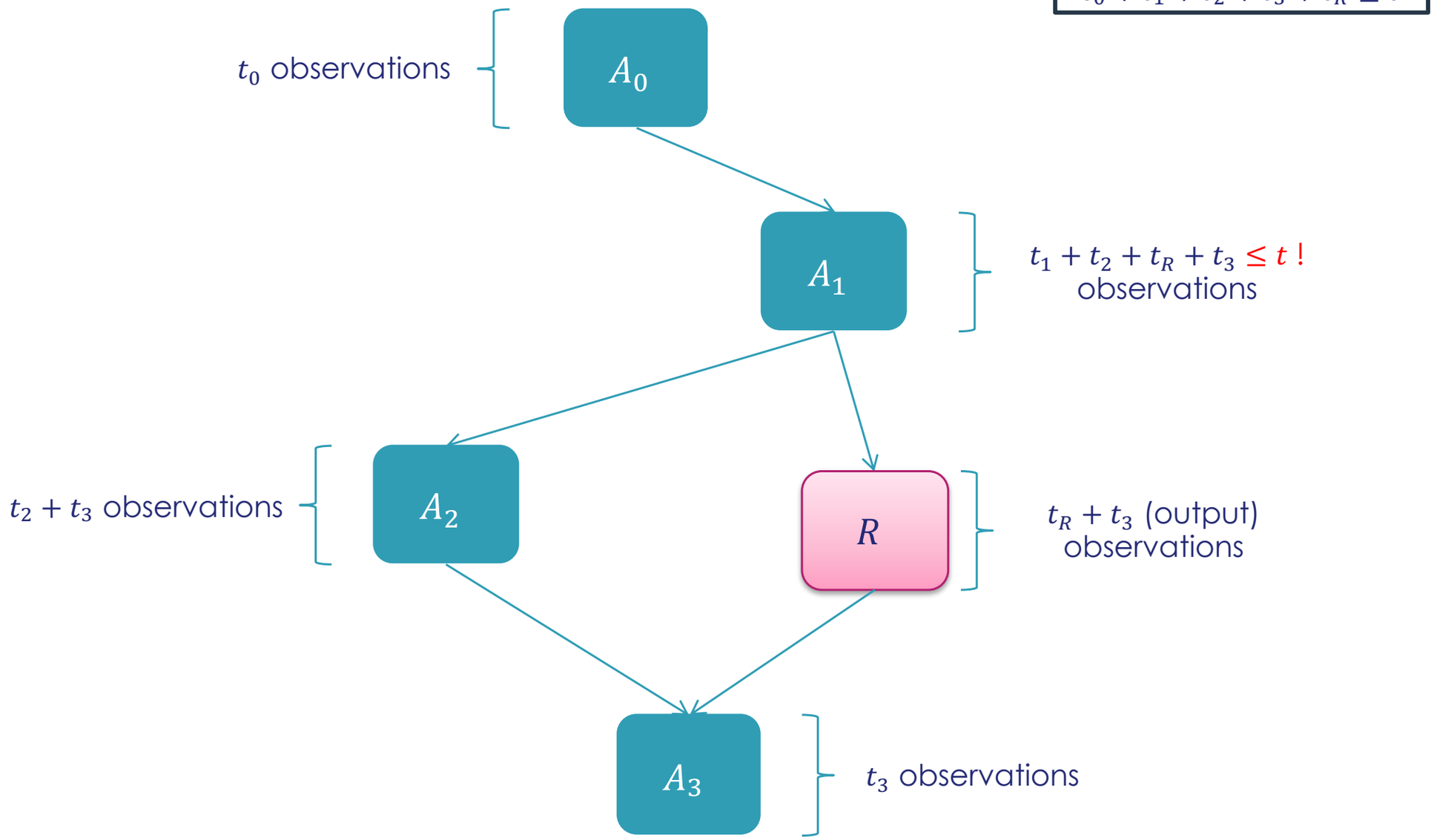
# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 + t_R \leq t$$



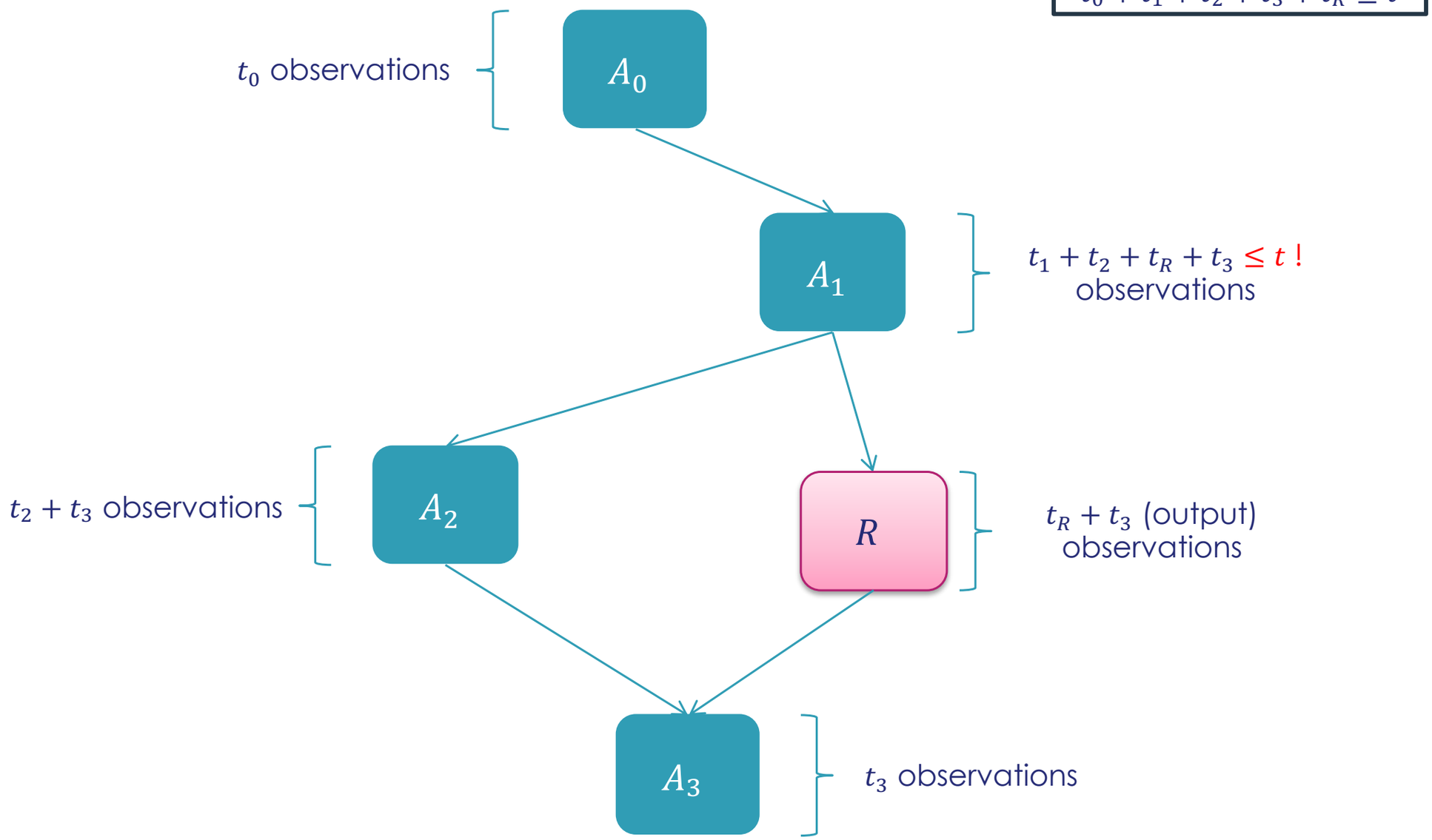
# Composition of gadgets

$t_0 + t_1 + t_2 + t_3 + t_R \leq t$



# Composition of gadgets

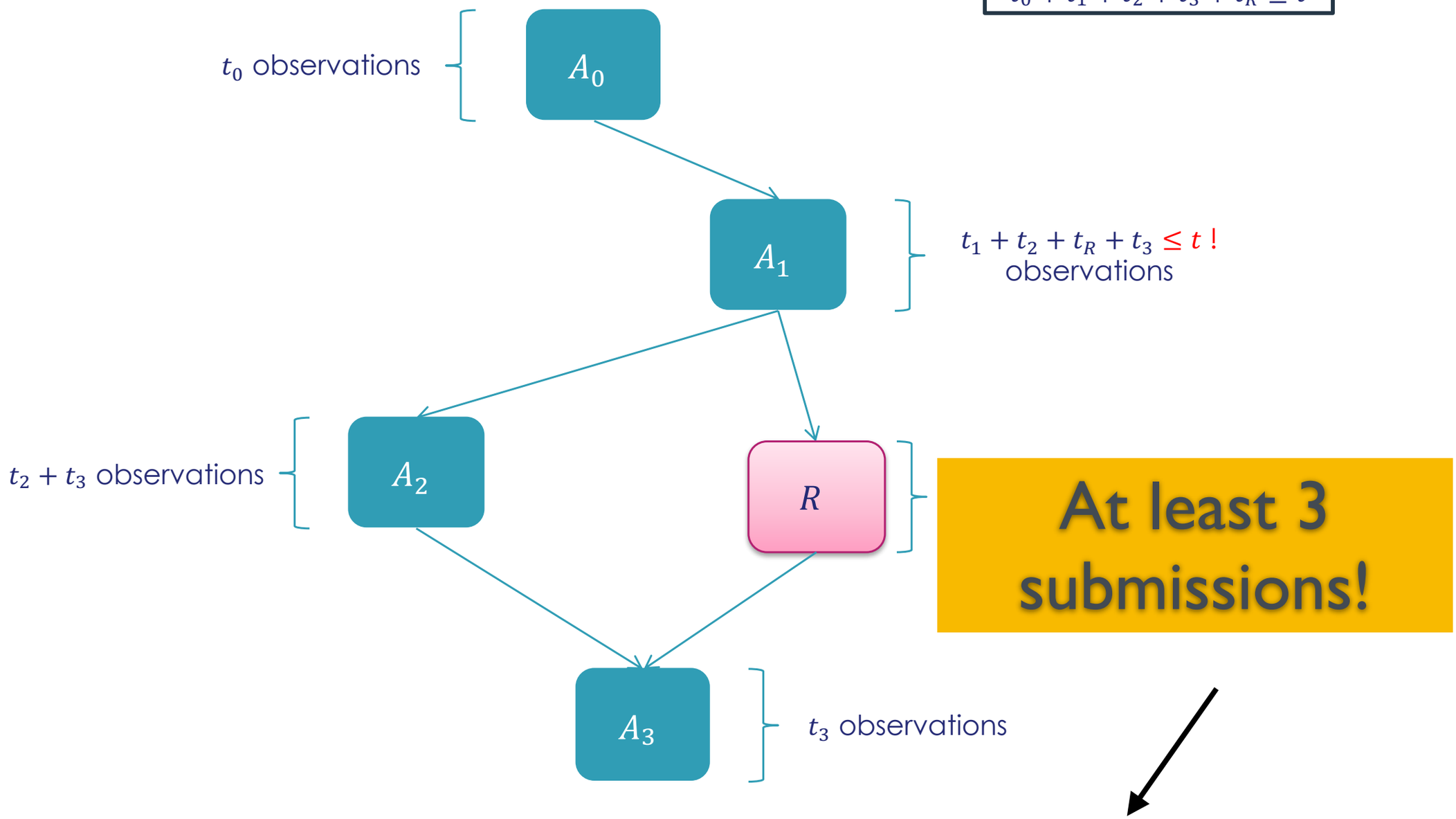
$t_0 + t_1 + t_2 + t_3 + t_R \leq t$




**[CCS:BBDFGS+16]** *Strong Non-Interference and Type-Directed Higher-Order Masking.* Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub and Rebecca Zucchini. CCS 2016.

# Composition of gadgets

$$t_0 + t_1 + t_2 + t_3 + t_R \leq t$$




**[CCS:BBDFGS+16]** *Strong Non-Interference and Type-Directed Higher-Order Masking.* Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub and Rebecca Zucchini. CCS 2016.

Oberthur Technologies

Grenoble INP ENSIMAG

PhD thesis

2010

2015

2020

LETTRES SCIENCES
ECOLE NORMALE SUPERIEURE

THALES



CryptoExperts

Oberthur Technologies

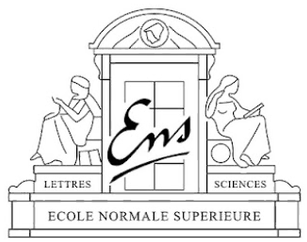
Grenoble INP ENSIMAG

PhD thesis

2010

2015

2020




THALES

- Masking Lattice-Based Signature Schemes [IMDEA++]
- Building New Generic Masked Multiplications [ENS]



CryptoExperts

Oberthur Technologies

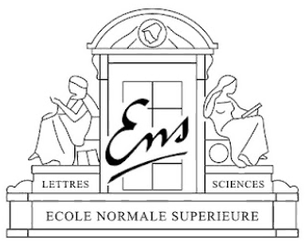
Grenoble INP ENSIMAG

PhD thesis

2010

2015

2020




CRYPTO EXPERTS

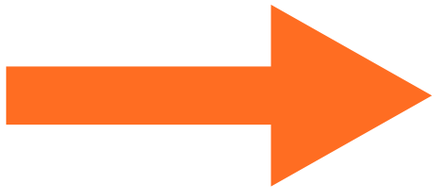
THALES



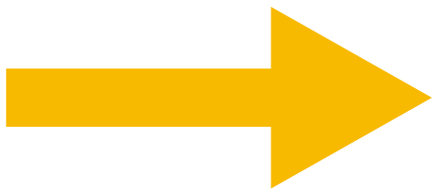
CRYPTOEXPERTS

# New Missions, New Organization

# Two New Sequences of Work



Building more efficient composition schemes



Generate and verify masking in the *random probing model*

Oberthur Technologies

Grenoble INP ENSIMAG

PhD thesis

2010

2015

2020

ECOLE NORMALE SUPERIEURE

CRYPTO EXPERTS

THALES



CRYPTOEXPERTS

Oberthur Technologies

Grenoble INP ENSIMAG

PhD thesis

2010

2015

2020

ECOLE NORMALE SUPERIEURE

CRYPTO EXPERTS

THALES



CRYPTOEXPERTS

# Maternity Leave (4 months)

More flexibility at CRX

Delayed reviews for CHES

People offered to wait for papers

Possible 1-year delay for ERC

Oberthur Technologies

Grenoble INP ENSIMAG

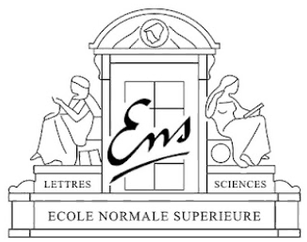
2021!

PhD thesis

2010

2015

2020




CRYPTO EXPERTS

THALES



CRYPTOEXPERTS

Oberthur Technologies

Grenoble INP ENSIMAG

Many collaborations from the people I met:
- Companies
- PhD thesis
- Conferences, seminar, projects

2021!

PhD thesis

2010

2015

2020

LETTRES SCIENCES
ECOLE NORMALE SUPERIEURE

CRYPTO EXPERTS

THALES



CRYPTOEXPERTS

Oberthur Technologies

Grenoble INP ENSIMAG

Many collaborations from the people I met:
- Companies
- PhD thesis
- Conferences, seminar, projects

2021!

Always new challenges:
- New papers
- Talks in conferences, seminar

2010

2020

LETTRES SCIENCES
ECOLE NORMALE SUPERIEURE

CRYPTO EXPERTS

THALES



CRYPTOEXPERTS