Random Active Shield

FDTC'2012 FDTC 2012 Fault Diagnosis and Tolerance in Cryptography, Leuven, Belgium.

Sébastien BRIAIS[1], Jean-Michel CIORANESCO[2,3], Jean-Luc DANGER[1,4], Sylvain GUILLEY[1,4], David NACCACHE[3,5] and Thibault PORTEBOEUF[1].

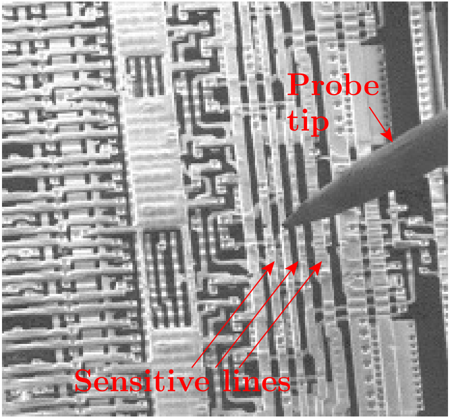
[1]**Secure-IC S.A.S.**, *37/39 rue Dareau, 75 014 Paris, France and 80 avenue des Buttes de Coësmes, 35 700 Rennes, France.*
[2]**Altis Semiconductor**, *224 Boulevard John Kennedy, 91 100 Corbeil-Essonnes, France.*
[3]**Sorbonne Universités** – *Université Paris II, 12 place du Panthéon, 75 231, Paris Cedex 05, France.*
[4]**Institut MINES-TELECOM**, *TELECOM-ParisTech, CNRS LTCI (UMR 5141), 46 rue Barrault, 75 634 Paris Cedex 13, France.*
[5]**École normale supérieure**, *Département d'informatique, 45 rue d'Ulm, 75 230, Paris Cedex 05, France.*

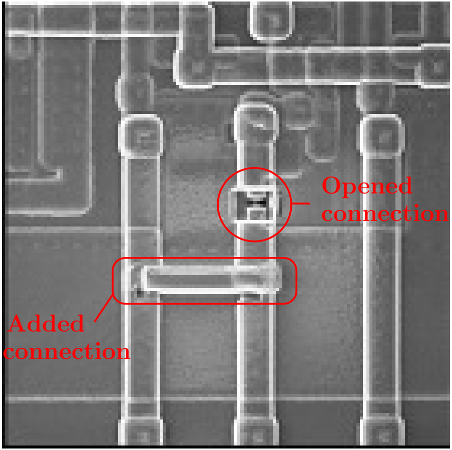
Sunday, September 9, 2012.

Presentation Outline

Presentation Outline

Introduction: Invasive Attacks

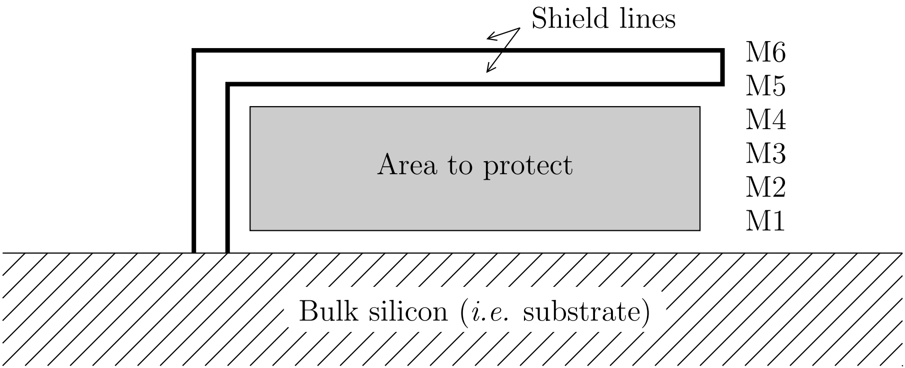




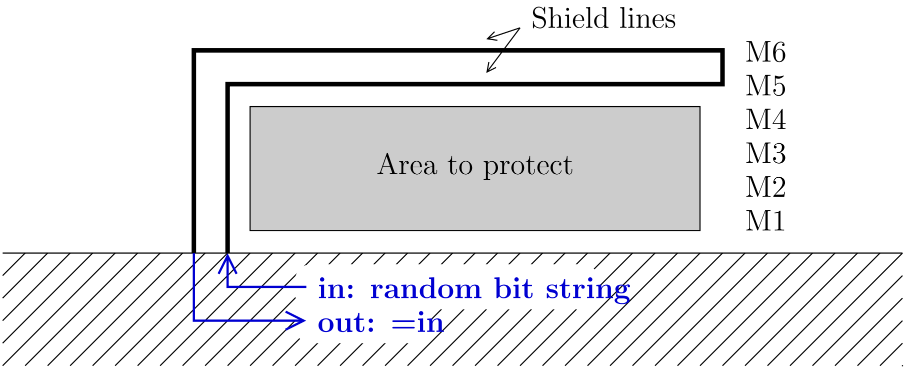
- Probing thanks to prober tip;
- Read or force sensitive variables.

- Edition thanks to a FIB;
- Unlock access to a memory.

General structure of a shield (sagittal view)

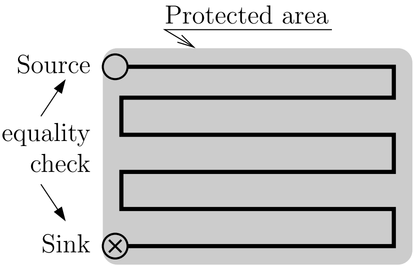


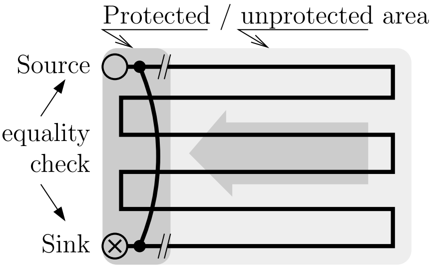
General structure of a shield (sagittal view)



Rerouting attacks

Theory

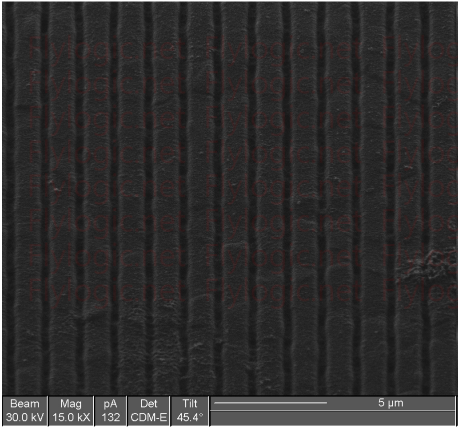


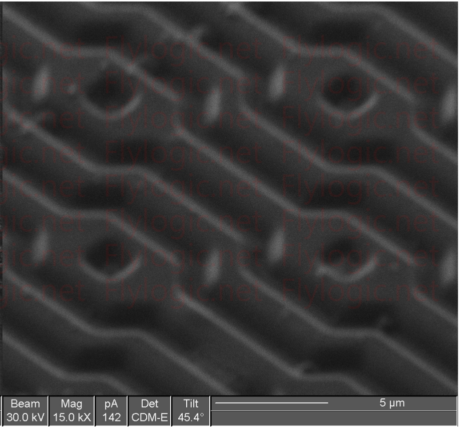


Cuts // and connections • are introduced by FIB.

Rerouting attacks

Practice



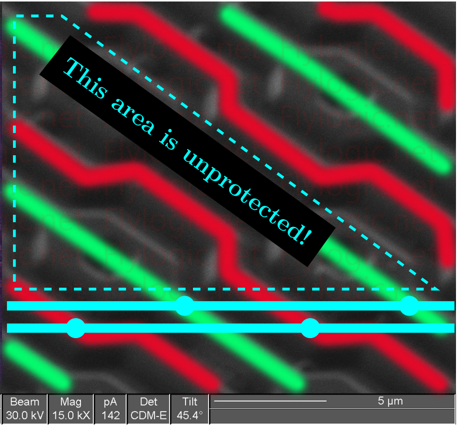


$\times 15,000$

Pictures courtesy of FlyLogic [Tar10]

Rerouting attacks

Practice



$\times 15,000$

Pictures courtesy of FlyLogic [Tar10]

Presentation Outline

Total Coverage

Second requirement

The shield should cover the whole surface without leaving holes

Different shielding strategies

- **Passive shielding**: detects with an analog sensor a change of the shield
  - Pros: constraints the rerouting attack to be at constant capacitive load
  - Cons: difficult to define the threshold for a successful/unsuccessful attack
- **Active shielding**: detects digitally any topological change of the shield
  - Pros: logical countermeasure, more robust [BCC+12], and also more portable
  - Cons: successful attacks do not need to balance the rerouting

Total Coverage

Second requirement

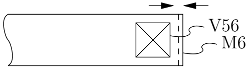
The shield should cover the whole surface without leaving holes

Different shielding strategies

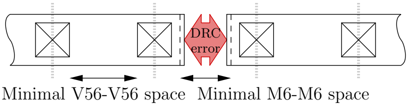
- **Passive shielding**: detects with an analog sensor a change of the shield
  - Pros: constraints the rerouting attack to be at constant capacitive load
  - Cons: difficult to define the threshold for a successful/unsuccessful attack
- **Active shielding**: detects digitally any topological change of the shield
  - Pros: logical countermeasure, more robust [BCC+12], and also more portable
  - Cons: successful attacks do not need to balance the rerouting

Manufacturability

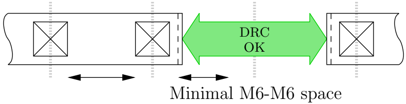
(a) **Mandatory extension after a via**



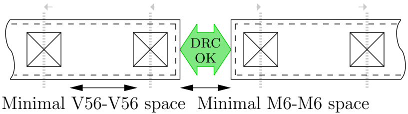
(b) **One via site is lost at every via**



(c) **Solution #1: skip a via**



(d) **Solution #2: fatten the wire and space the vias**



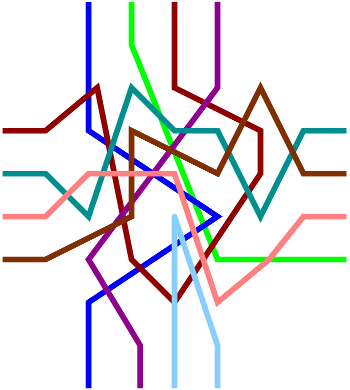
Design Rule Checks (DRC)

- **Metal extension beyond a via at end of lines**
- **Metal maximal parallel run length**
- **Density considerations**
- **Antennae rules check**

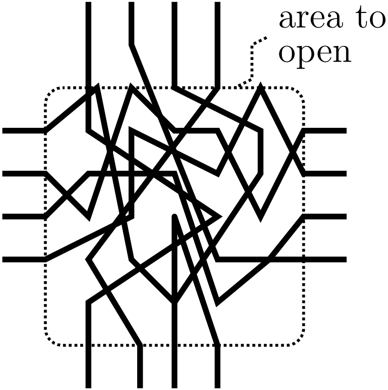
Presentation Outline

Idea

Designer's view



Attacker's view



Formalization

Algorithm 1 Dense Random Spaghetti Routing.

Input: N: number of different interleaved equipotentials.

Output: A random shield made up of N equipotentials.

1: Build a graph whose vertices consist in free via slots and edges in the free routing slots.

2: Label each edge by a random number.

3: Solve the Traveling Salesman Problem (TSP) to get one Hamiltonian circuit.

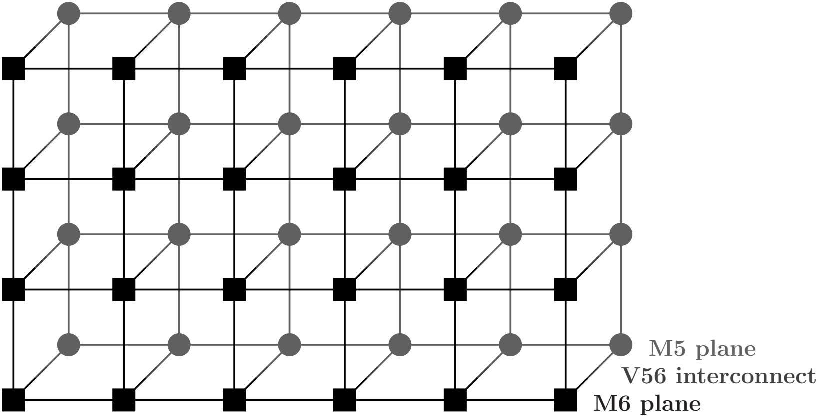
4: Cut the Hamiltonian circuit into N sub-paths, and return those.

Shield Objectives

- it must cover the circuit uniformly,
- it must resist against alteration.
  - Our strategy is to make the identification phase very chancy.
  - Somehow, it is a *security by obscurity* solution, often encountered in CC.

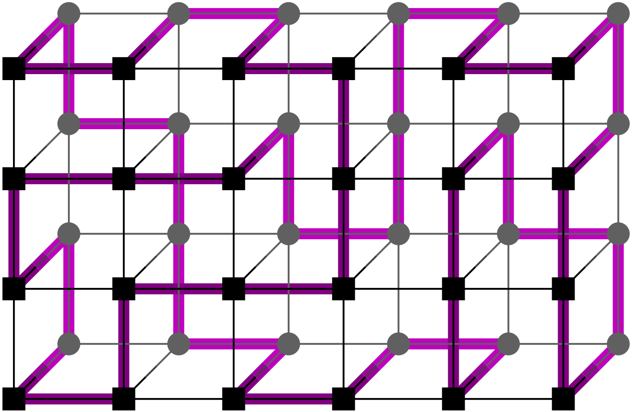
Example with $N = 3$

(a)

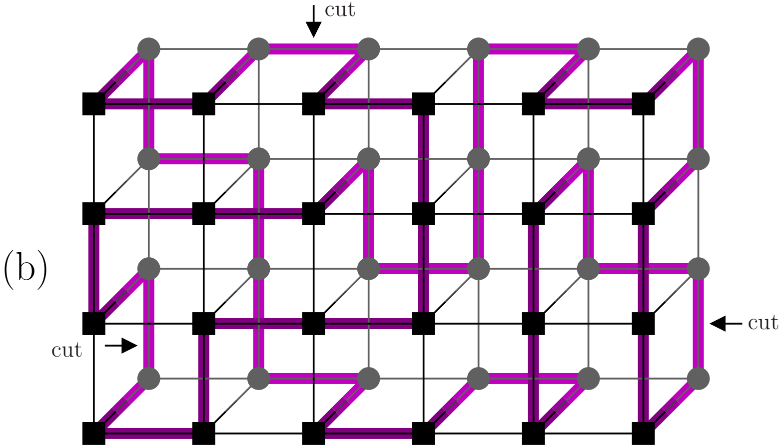


Example with $N = 3$

(b)

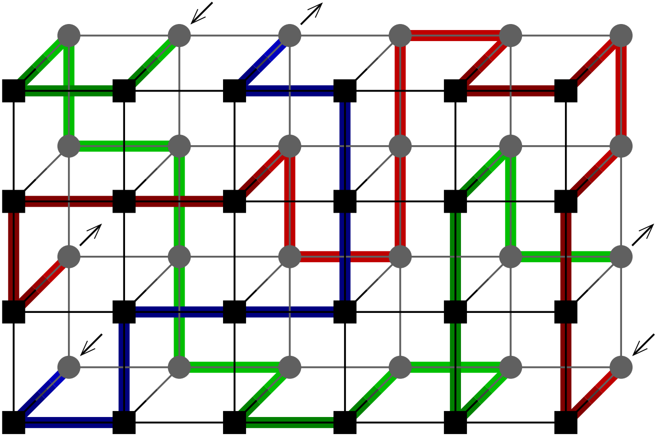


Example with $N = 3$

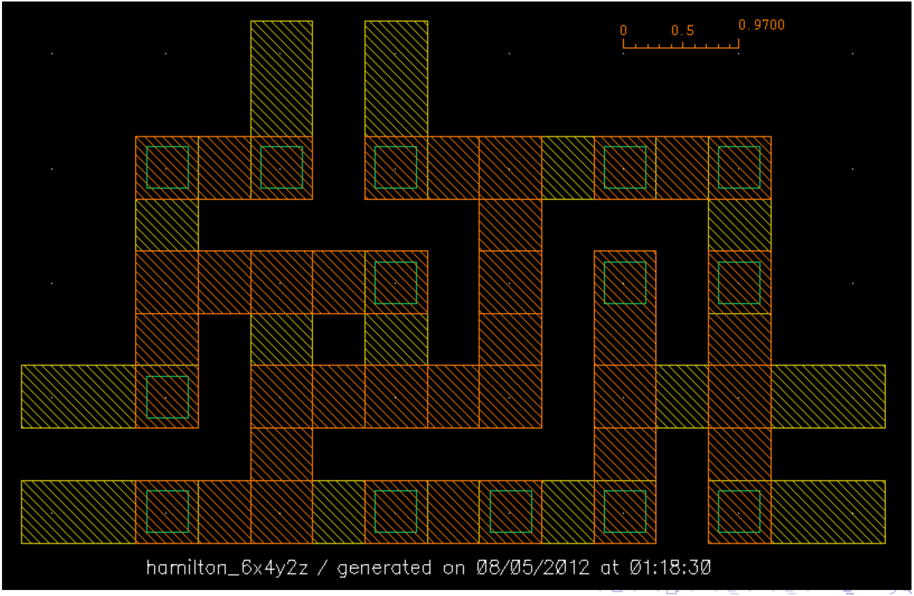


Example with $N = 3$

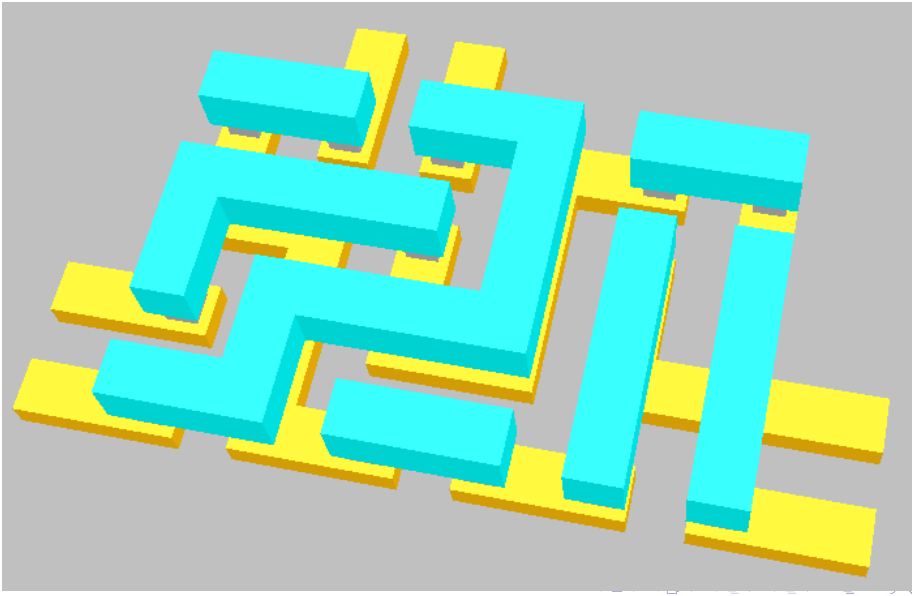
(c)



Example with $N = 3$

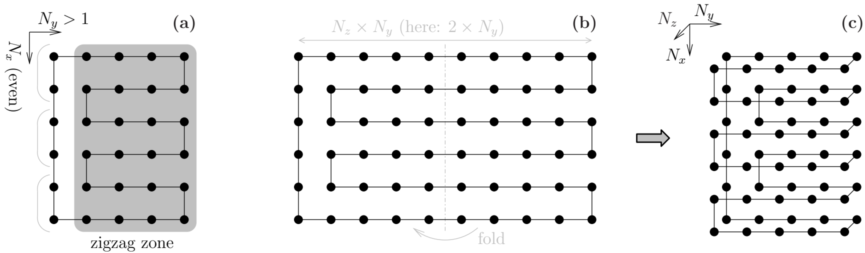


Example with $N = 3$



How to generate *quickly* a random Hamiltonian circuit?

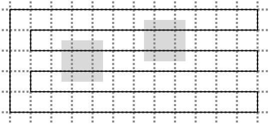
1/2: Start from a regular Hamiltonian circuit



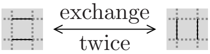
How to generate *quickly* a random Hamiltonian circuit?

2/2: Randomize it

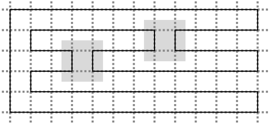
Step 1: build a trivial circuit



Step 2: invariant transform

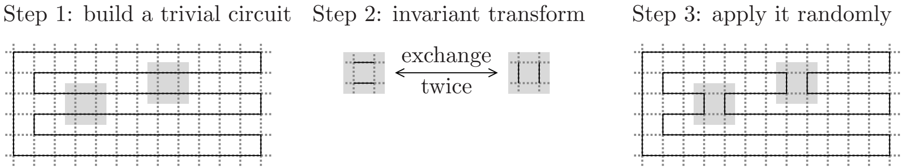


Step 3: apply it randomly



Metric: entropy $H(C) = \sum_{d\in\{x,y,z\}} -P(d) \cdot \log_2 P(d)$

2/2: Randomize it ($\Rightarrow$ *make it as much isotropic as possible*)



Step 1

- $P(x) = 68/78$ and $P(y) = 10/78$;
- $H(C) \simeq 0.55$ bit.

Step 2 $\longrightarrow$

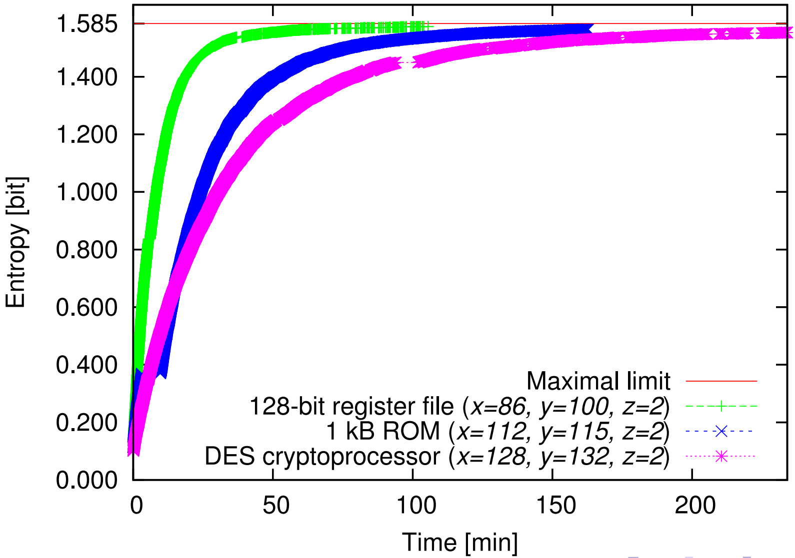
Step 3

- $P(x) = 64/78$ and $P(y) = 14/78$;
- $H(C) \simeq 0.68$ bit.

Computation time to generate a Hamiltonian circuit.

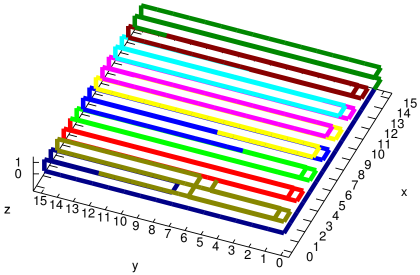
Circuit	Area	Number of vertices	Time for the generation	Entropy
128-bit reg--ister file	$10,000\ \mu\text{m}^2$	$17,200$	1 h 45 min	1.574 bit
1 kB ROM	$15,000\ \mu\text{m}^2$	$25,760$	2 h 43 min	1.564 bit
DES crypto accelerator	$21,000\ \mu\text{m}^2$	$33,792$	3 h 54 min	1.554 bit

Convergence rate of three real-world random active shields



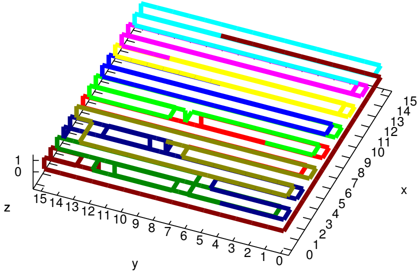
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$H = 0.550$ bit, $T = 37$ ms.



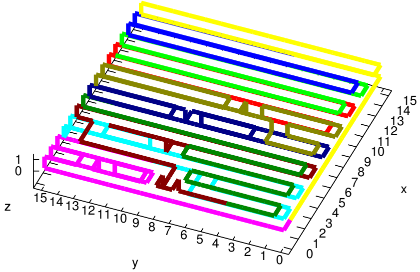
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$H = 0.673$ bit, $T = 129$ ms.



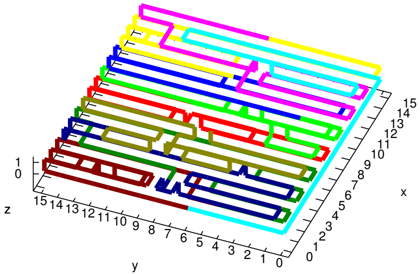
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$$H = 0.783 \text{ bit}, \; T = 213 \text{ ms}.$$



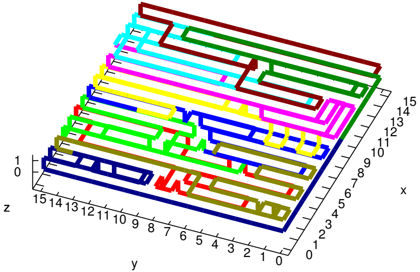
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$H = 0.903$ bit, $T = 316$ ms.



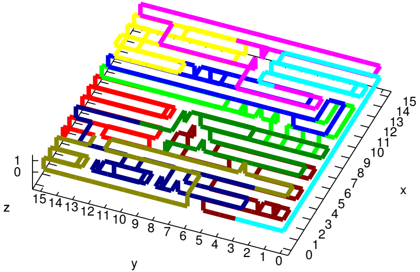
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$H = 1.014$ bit, $T = 438$ ms.



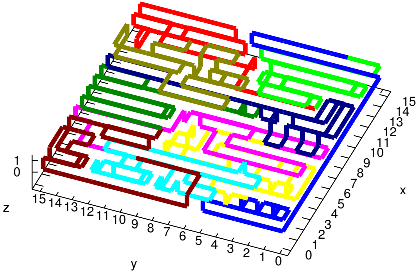
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$$H = 1.126 \text{ bit}, \ T = 599 \text{ ms}.$$



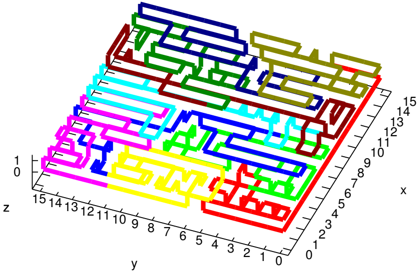
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$H = 1.240$ bit, $T = 940$ ms.



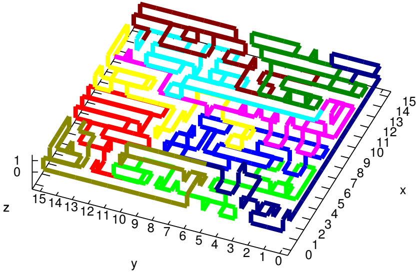
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$H = 1.349$ bit, $T = 1381$ ms.



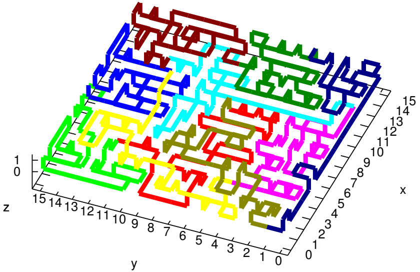
Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

$H = 1.454$ bit, $T = 2228$ ms.



Evolution of a $x = 16$, $y = 16$, $z = 2$ shield with $N = 10$ segments

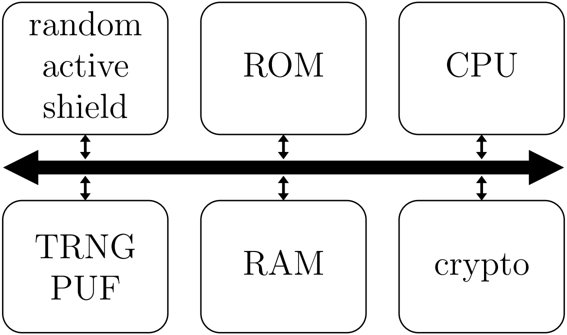
$$H = 1.556 \text{ bit}, \; T = 4303 \text{ ms}.$$



Cost of the Control

AMBA / APB slave, in Xilinx

- Cost for 4,000 segments:
  - 3,607 slices
- Comparison with a crypto SoC with a 32-bit RISC CPU
  - 13,244 slices



Presentation Outline

Conclusions

- First random active shield concept
- DRC compliant
- Slave on a system bus

Perspectives

- Generalization in 3D technologies (*front-* & *back-side*)
- Dynamic routes, which makes static imaging techniques (*e.g.* voltage contrast analysis) futile [BCC+12]

Acknowledgements

- **TOISE**, *Trusted Computing for European Embedded Systems*, funded under European grant ENIAC-2010-1,
- **MARSHAL+**, *Mechanism Against Reverse Engineering for Secure Hardware and Algorithm*, funded under French grant FUI12.

Sébastien Briais, Stéphane Caron, Jean-Michel Cioranesco, Jean-Luc Danger, Sylvain Guilley, Jacques-Henri Jourdan, Arthur Milchior, David Naccache, and Thibault Porteboeuf.

3D Hardware Canaries.

In *CHES*, Lecture Notes in Computer Science. Springer, September 9-12 2012. Leuven, Belgium. Full version in ePrint Archive, Report 2012/324 (http://eprint.iacr.org/2012/324/).

Christopher Tarnovsky.

Infineon / ST Mesh Comparison, February 14th 2010.

http://www.flylogic.net/blog/?p=86.

Random Active Shield

FDTC'2012 FDTC 2012 Fault Diagnosis and Tolerance in Cryptography, Leuven, Belgium.

Sébastien BRIAIS[1], Jean-Michel CIORANESCO[2,3], Jean-Luc DANGER[1,4], Sylvain GUILLEY[1,4], David NACCACHE[3,5] and Thibault PORTEBOEUF[1].

[1]**Secure-IC S.A.S.**, *37/39 rue Dareau, 75 014 Paris, France and 80 avenue des Buttes de Coësmes, 35 700 Rennes, France.*
[2]**Altis Semiconductor**, *224 Boulevard John Kennedy, 91 100 Corbeil-Essonnes, France.*
[3]**Sorbonne Universités** – *Université Paris II, 12 place du Panthéon, 75 231, Paris Cedex 05, France.*
[4]**Institut MINES-TELECOM**, *TELECOM-ParisTech, CNRS LTCI (UMR 5141), 46 rue Barrault, 75 634 Paris Cedex 13, France.*
[5]**École normale supérieure**, *Département d'informatique, 45 rue d'Ulm, 75 230, Paris Cedex 05, France.*

Sunday, September 9, 2012.