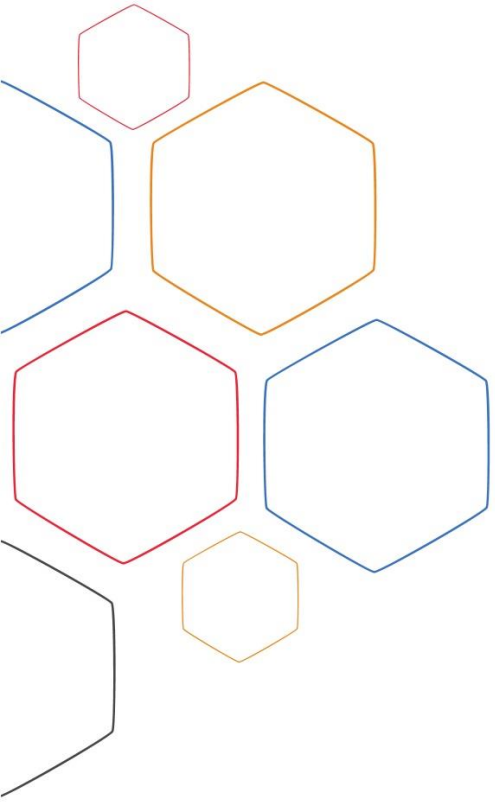
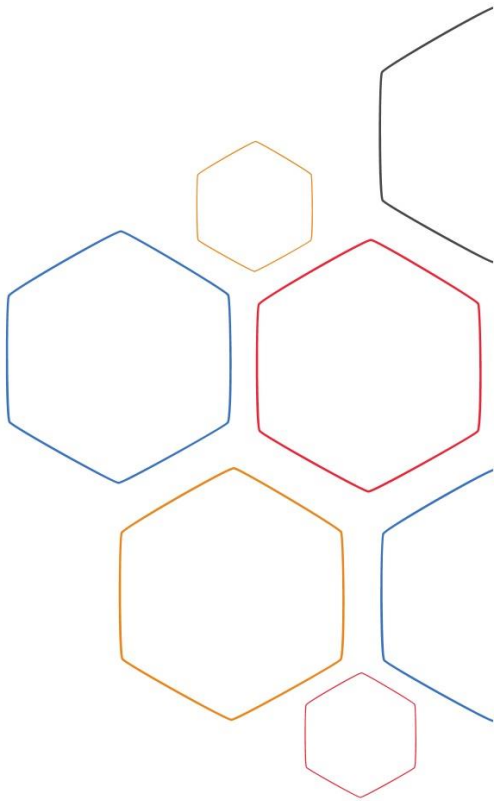
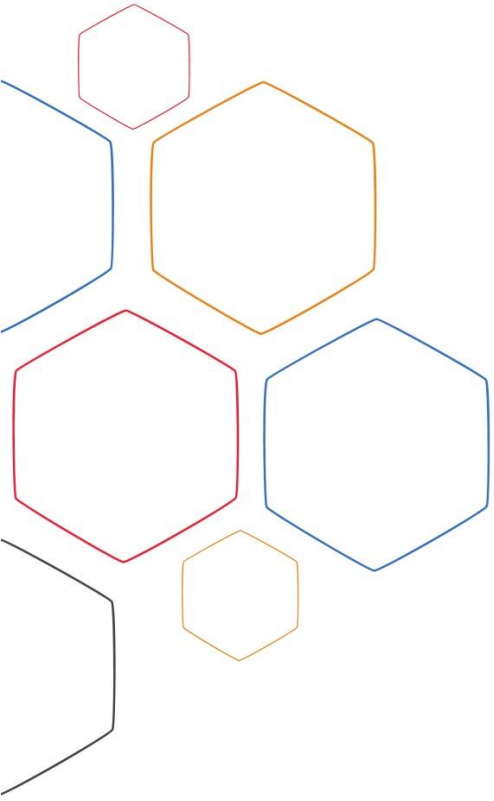


Logic locking: ten years since the SAT attack

Sophie Dupuis
LIRMM, UM, CNRS



Outline



001 **Context**

Hardware trust: why, what, how?

010 **Prevention against overproduction**

Logic locking pre SAT-attack
Logic locking post SAT-attack

DPA: attack and protection

011 **Conclusion**

Why?

A deeper dive into semiconductor foundries
https://deepforest.substack.com/p/a-deeper-dive-into-semiconductor

- Time to market constraints, complexity, cost increases etc…
  - Third-party IP cores purchased
  - Use of EDA tools
  - Fabrication outsourced
    - The cost of building a foundry was estimated at 5$ billion in 2015, 20$ billion currently (3nm production line)

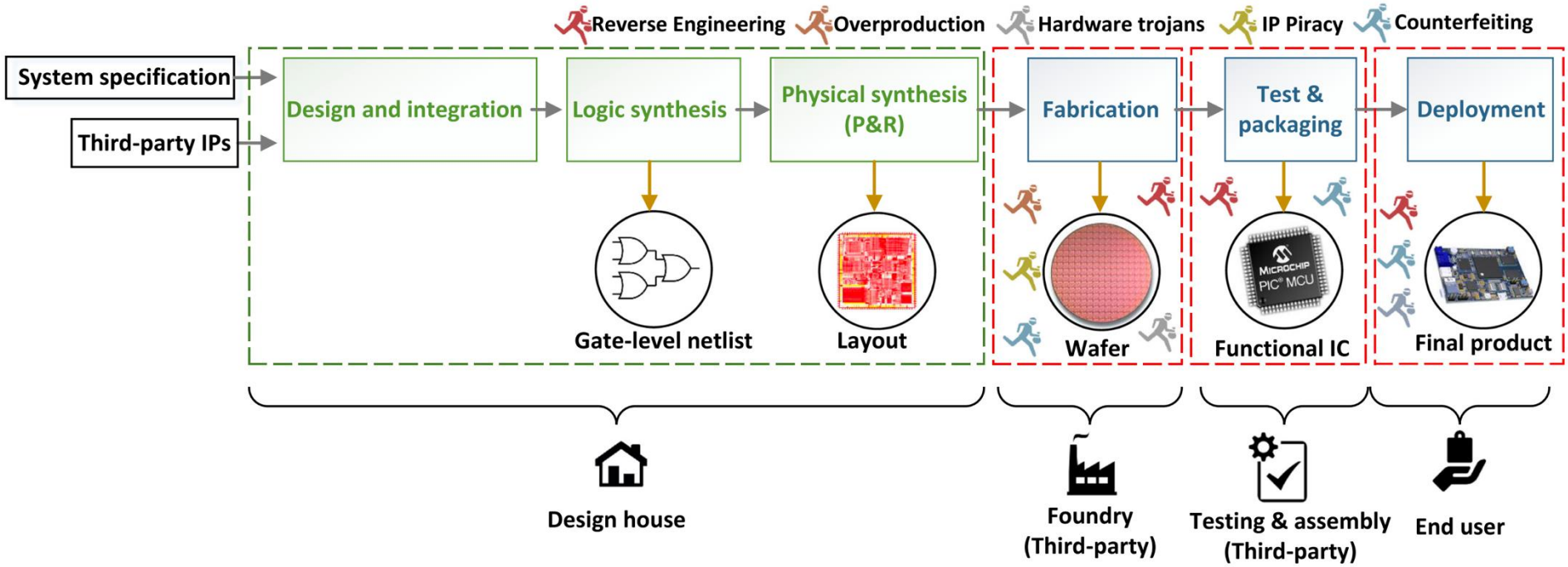
Today's integrated circuits' supply chain

=

Many (untrusted) third parties from various countries

What?

An Overview of FPGA-inspired Obfuscation Techniques, ACM Computing Surveys, 2024



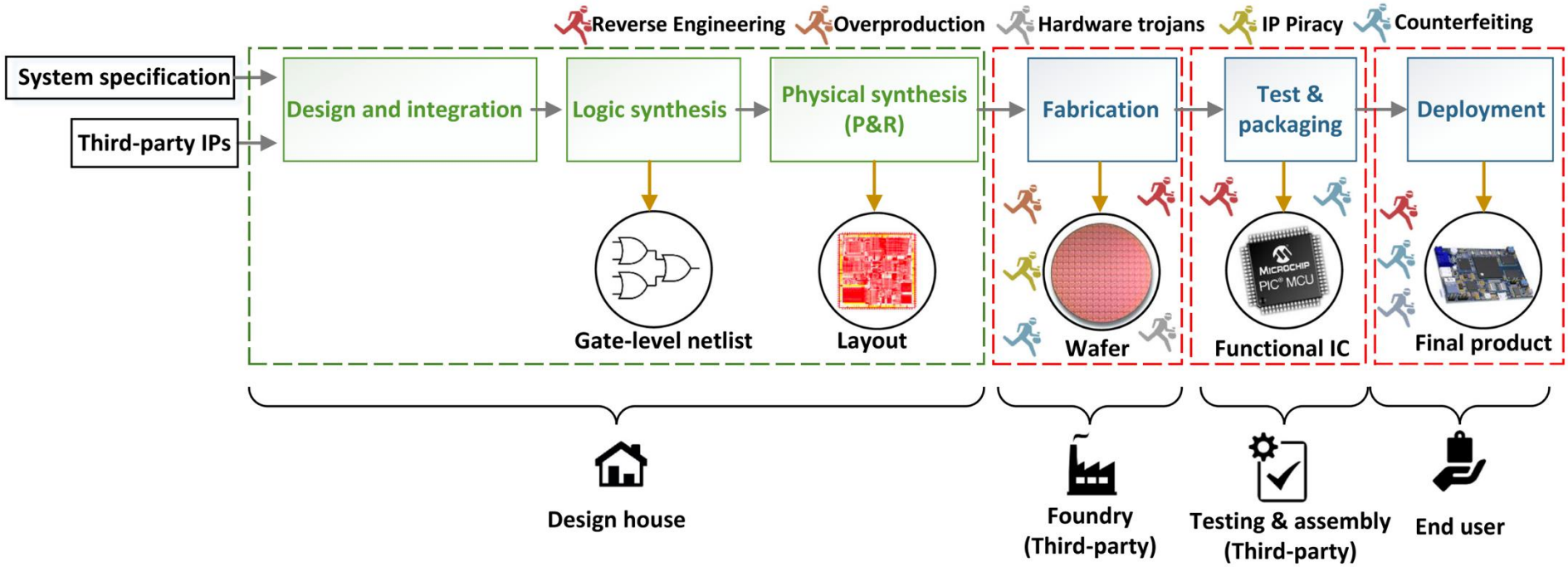
Many (untrusted) third parties from various countries

=

Many potential threats

What?

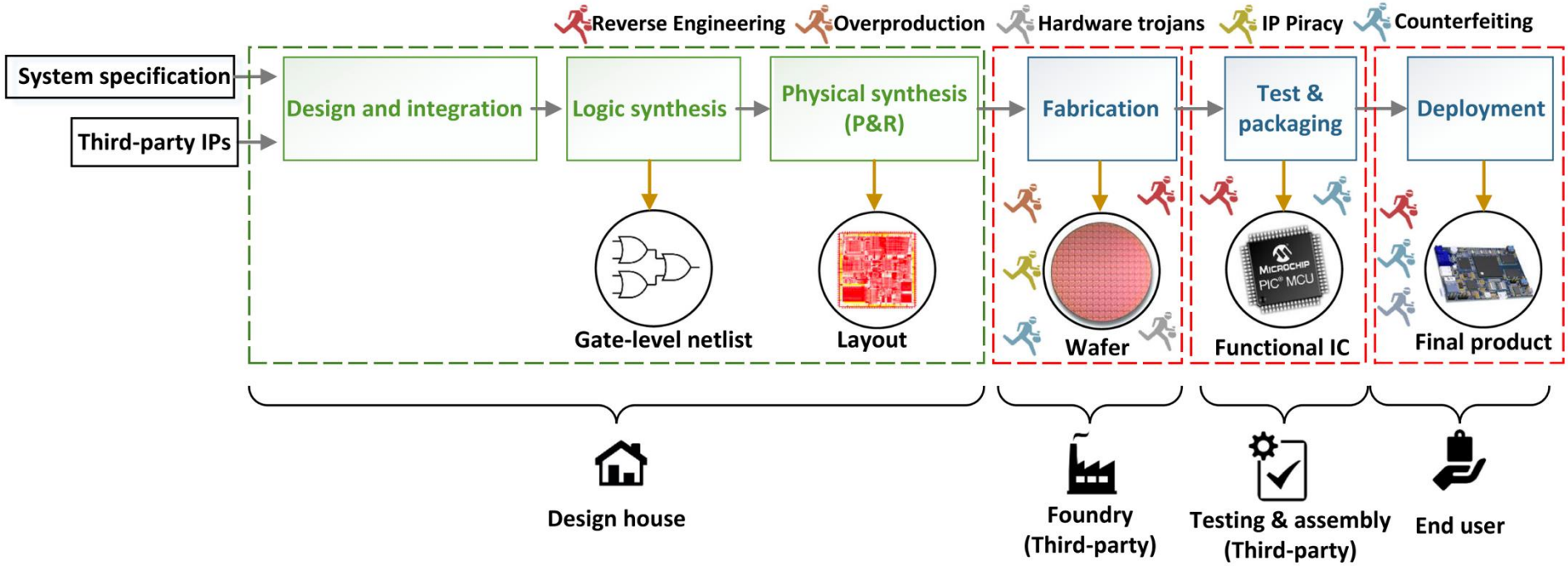
An Overview of FPGA-inspired Obfuscation Techniques, ACM Computing Surveys, 2024



- Potential threats:
  - IP theft (reverse engineering)
  - An unauthorized less reliable replica (counterfeiting)
  - An unauthorized exact copy (piracy / overproduction)
  - An IC that does not conform to the original design/performance standards (Hardware trojans)
- Counterfeit incidents since 2001, 4 times more incidents in 2014 than in 2009
- The semiconductor industry loses several billions dollars annually

What?

An Overview of FPGA-inspired Obfuscation Techniques, ACM Computing Surveys, 2024



- A foundry can produce more ICs, without (significant) investment
  - No R&D, no new masks to develop, possibility to hide their yield
- It is then “interesting” to produce more than ordered, and resell the surplus on the black market (at a very attractive price)

How?

Anti-Counterfeit techniques: from design to resign, MTV, 2013
Introduction to Hardware Security and Trust, Springer, 2012

- How can a designer prevent such a threat?

- Passive metering
  - The design house can uniquely tag / ID each IC (each ID of a genuine IC is listed in a secure database)
  - If an IC's ID is not registered, there is a high possibility that it is overproduced

  - Physically Unclonable Functions: exploit inherent random (uncontrollable and unpredictable) physical variations

- But useful only for detection, not for prevention

How?

EPIC: Ending Piracy of Integrated Circuits, DATE, 2008

- How can a designer prevent such a threat?

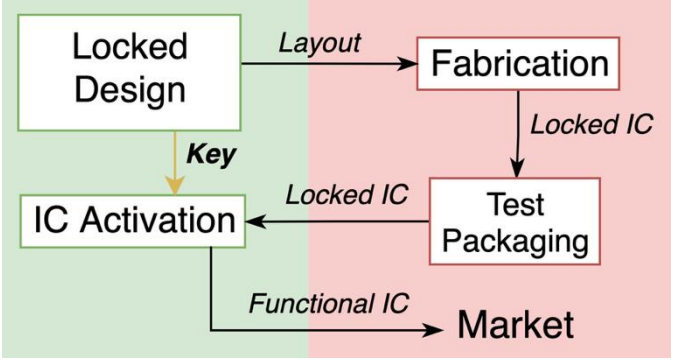
- Active metering
  - The design house – and only them – can access, enable (unlock) or disable (lock) the functionality of the ICs
  - The foundry produces non-functional – unsaleable – ICs

  - Logic locking

EPIC: Ending Piracy of Integrated Circuits, DATE, 2008

Locks the design during fabrication

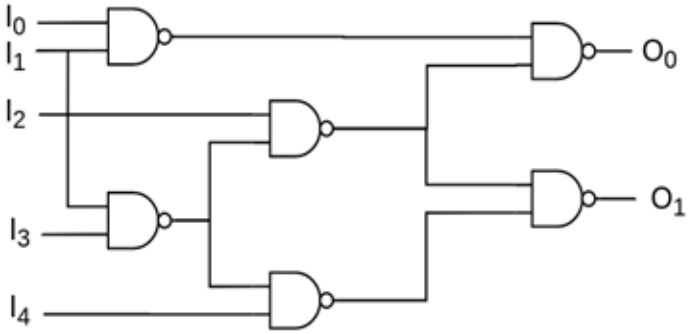
- The foundry cannot use the ICs properly, therefore preventing them from selling (non-functional) copies on the black market
- Addition of a new input (a secret key)
  - Correct key value -> correct functionality
  - Wrong key value -> incorrect functionality

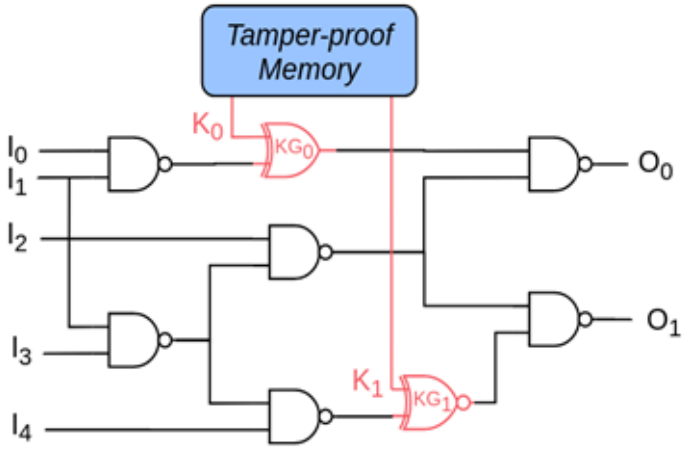


EPIC: Ending Piracy of Integrated Circuits, DATE, 2008

Locks the design during fabrication

- The foundry cannot use the ICs properly, therefore preventing them from selling (non-functional) copies on the black market
- During the design: addition of a new input (a secret key)
  - Drives additional key-gates





First era

EPIC: Ending Piracy of Integrated Circuits, DATE, 2008
Preventing IC Piracy using reconfigurable Logic Barriers, IEEE Design & Test of Computers, 2010

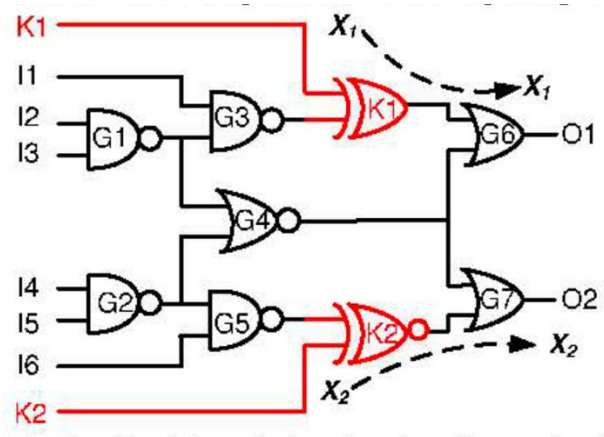
- Random insertion of XOR/XNOR gates
  - Potential problem in effectiveness : the effect of the corrupted signal may not be propagated to an output
  - Attacks may be able to retrieve the secret key value

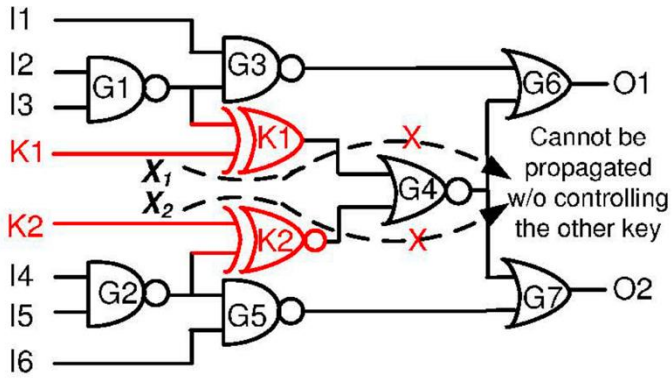
- Different choices of key-gates
  - XOR/XNOR, AND/OR, LUT, Muxes

First era

Security Analysis of logic obfuscation, DAC, 2012

- Different algorithms to choose wisely insertion locations
  - To prevent ATPG-based attacks
  - Insertion algorithm maximizing the interference between key gates





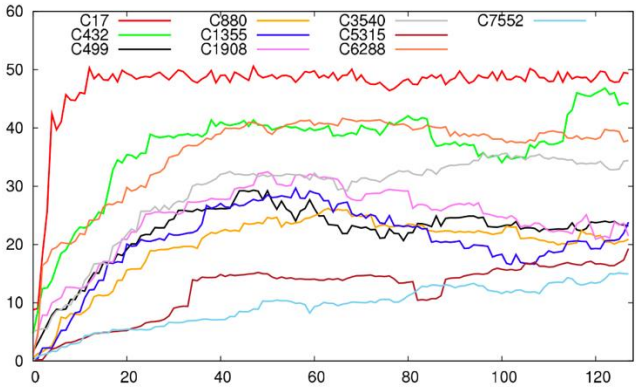
First era

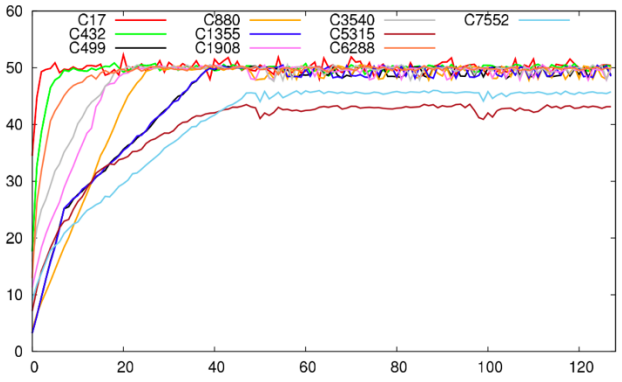
Logic Encryption: A Fault Analysis Perpective, DATE, 2012

- Different algorithms to choose wisely insertion locations
  - To improve output corruption
  - New quality metric introduced: Output corruption
    - Hamming distance: Number of outputs corrupted (50%)
    - Number of input patterns that lead to corruption (100%)

  - Insertion algorithm based on the fault impact

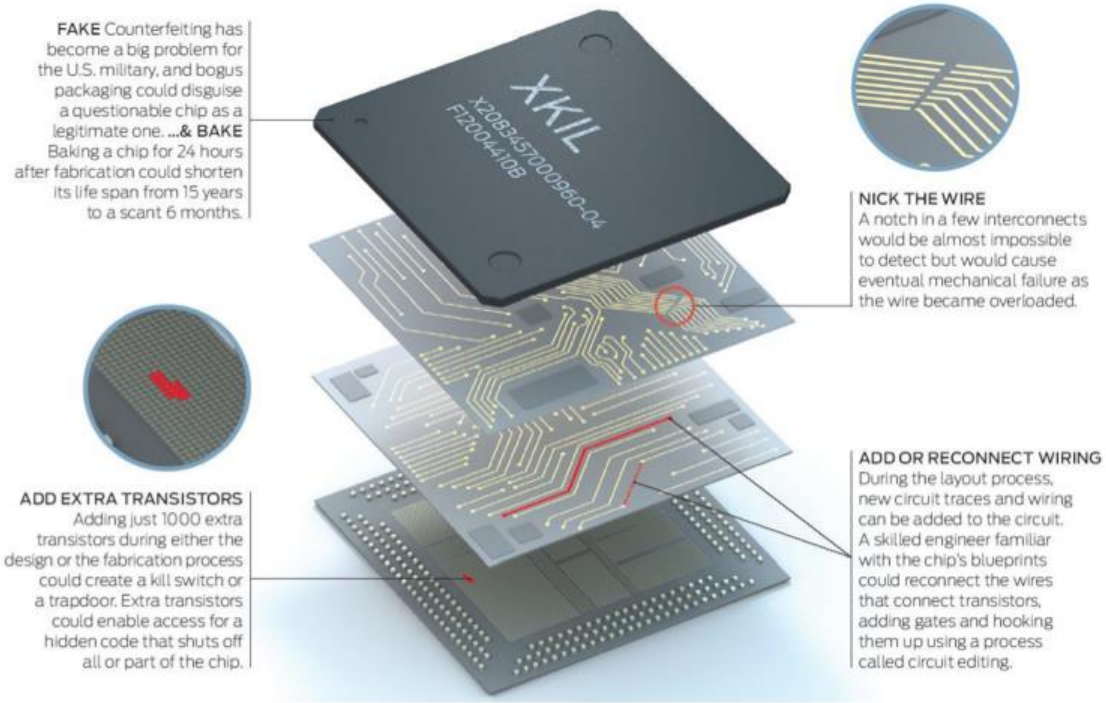
$$\text{Fault impact} = (\text{No. of Test Patterns}_{\text{s-a-0}} \times \text{No. of Outputs}_{\text{s-a-0}}) + (\text{No. of Test Patterns}_{\text{s-a-1}} \times \text{No. of Outputs}_{\text{s-a-1}})$$





A novel hardware logic encryption technique for thwarting illegal overproduction and hardware Trojans, IOLTS, 2014

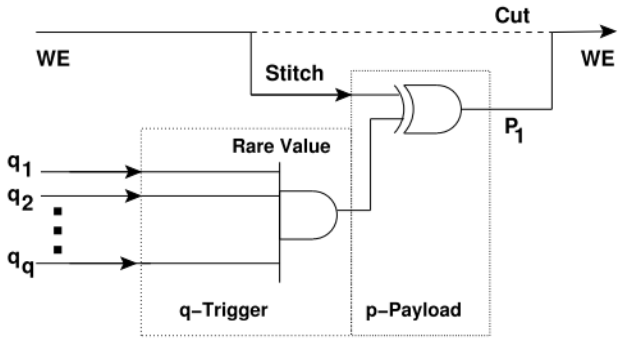
- Proposal to counteract hardware Trojan horses insertion also
  - A malicious, intentional modification of a circuit design that masquerades as something benign until the moment of activation



A novel hardware logic encryption technique for thwarting illegal overproduction and hardware Trojans, IOLTS, 2014

- Proposal to counteract hardware Trojan horses insertion also
  - Logic locking, by definition, modifies / "obfuscates" signals probabilities, thereby "annoys" the attacker in their choice for trigger input
  - Logic locking to specifically help detection methods based on logic test

- Algorithm targeting low controllable signals / signals with unbalanced probabilities?

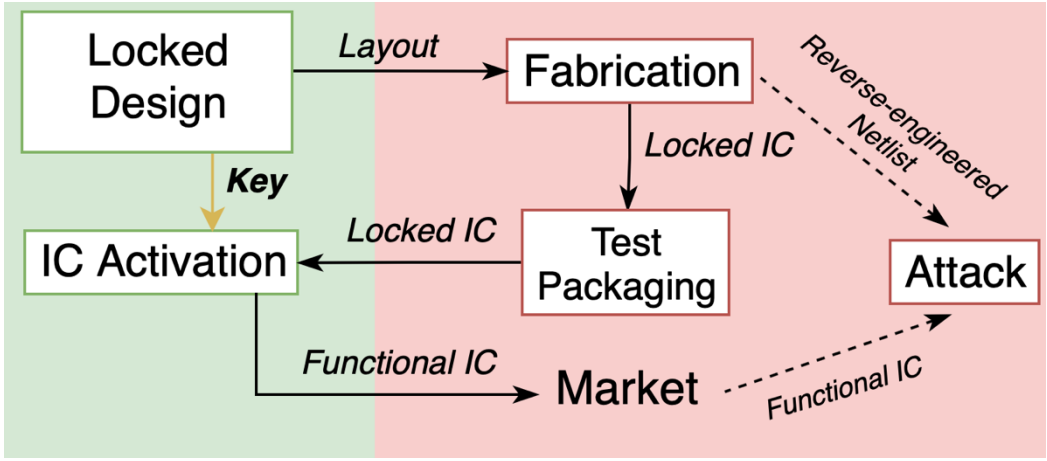


- The trigger is a crafted extremely rare condition not detectable by classical test procedure that activates the Trojan

The SAT attack

Evaluating the security of logic encryption algorithms, HOST, 2015

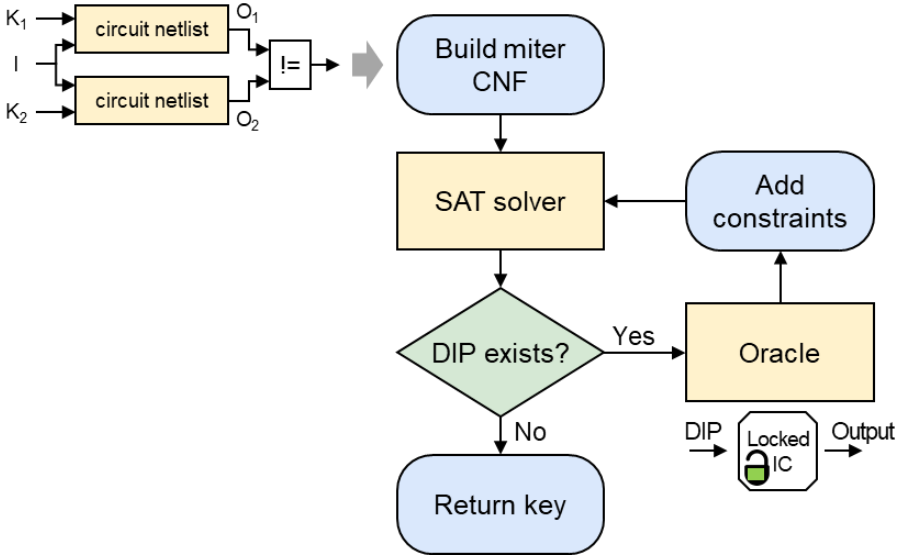
- Goal: Retrieve the secret key value
- Threat model: oracle-guided-attack
  - The attacker has a locked netlist and an oracle
    (= a functional IC programmed with the correct key)



The SAT attack

Evaluating the security of logic encryption algorithms, HOST, 2015

- SAT-attack: based on a Satisfiability solver
  - At each iteration:
    - Identify a Distinguishing Input Pattern: an input pattern for which two different keys give to different outputs
    - Prune out wrong key values (thanks to the oracle)



Second era

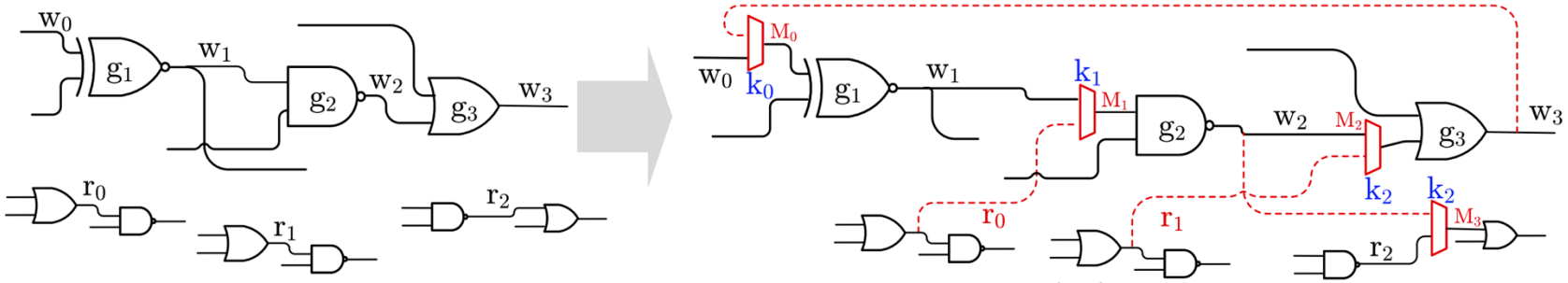
- How to counteract the attack?
  - Prevent the attack from being launched
  - Increase the computation time
    - Increase the time of one iteration
    - Increase the number of iterations
  - New types of attacks
    - Oracle-guided algorithmic (SAT-based) attacks
    - Oracle-less structural-based attacks
  - Protections improvements to be resilient against “every attacks”
    - Point function-based
    - CAC-based
  - And so on and so forth…

Second era

Cyclic Obfuscation for Creating SAT-Unresolvable Circuits, GLSVLSI, 2017
CycSAT: SAT-Based Attack on Cyclic Logic Encryptions, ICCAD, 2017

- How to counteract the attack?
  - Prevent the attack from being launched
  - Increase the computation time
    - Increase the time of one iteration
    - Increase the number of iterations

- Circuit modeled as a Boolean function in Conjunctive Normal Form
- Dummy paths added to the circuit to create logical loops
- SAT-based attack improvement:
  - Identify a key with which the circuit does not contain loop

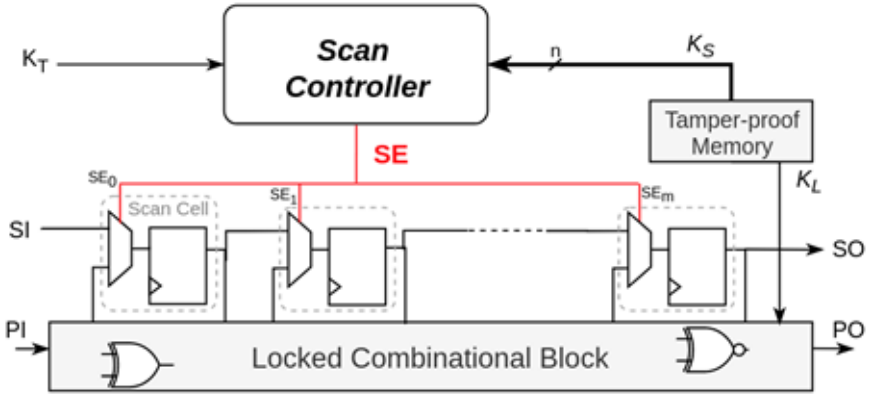


Second era

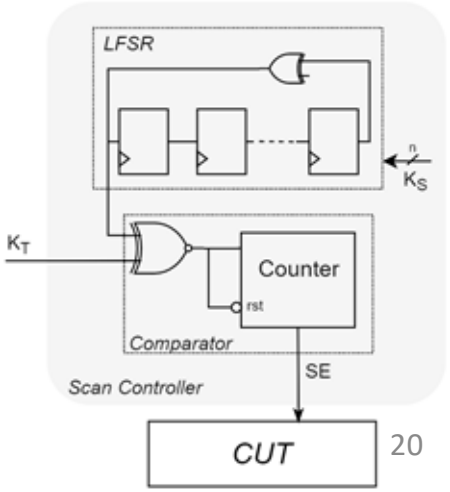
A secure scan controller for protecting logic locking, IOLTS, 2020

- How to counteract the attack?
  - Prevent the attack from being launched
  - Increase the computation time
    - Increase the time of one iteration
    - Increase the number of iterations

- Attack possible only on combinational circuits (or sequential ciruits with a scan access)
  - Block shift operation with a « scan access » key



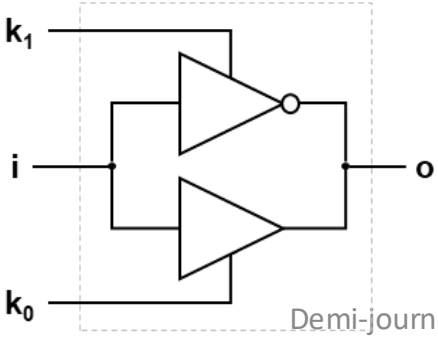
- Key input KT
- Linear feedback shift register initialized with KS
- Comparator LFSR and KT
- SE = 1 if matched for n consecutive cycles



- How to counteract the attack?
  - Prevent the attack from being launched
  - Increase the computation time
    - Increase the time of one iteration
    - Increase the number of iterations

- New type of key-gate unmodelable
  - Tristate-based key-gate: composed of a tristate buffer and a tristate inverter
    - Acts as a buffer, an inverter, creates a high impedance state on an » unintended » output

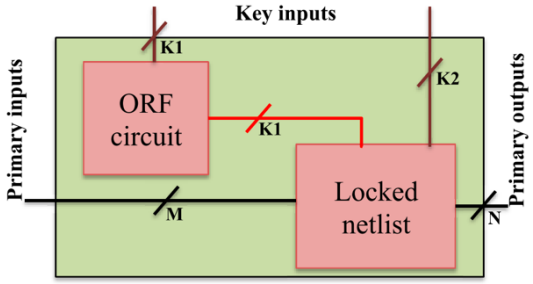
k_1	k_0	O
0	0	Hi-Z
0	1	i
1	0	not i
1	1	UNK



Second era

On improving the security of logic locking, IEEE TCAD, 2016

- How to counteract the attack?
  - Prevent the attack from being launched
  - Increase the computation time
    - Increase the time of one iteration
    - Increase the number of iterations
  - Use of “SAT hard” blocks, such as a one way-random function (such as an AES)



Second era

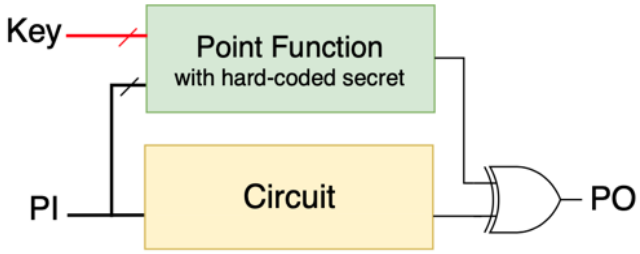
- How to counteract the attack?
  - Prevent the attack from being launched
  - Increase the computation time
    - Increase the time of one iteration
    - Increase the number of iterations

Point functions

SARLock: SAT attack resistant logic locking, HOST, 2016
Mitigating SAT attack on logic locking, CHES, 2016

- Point function-based logic locking
  - Each iteration can rule out one unique key value
  - This leads to an exponential number of iterations (2^n-1)

- How?
  - Addition of a new block creating a point-function i.e. a Boolean function that produces 1 at exactly one point
  - SARLock: The added block
    - Flips a signal for a specific key/input pair (key == input)
    - Prevents the flip, only for the correct key

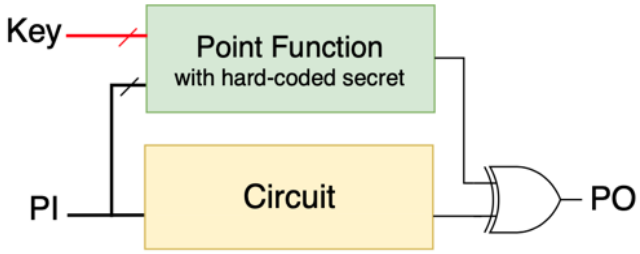


- Extremely low output corruption…

New attacks

Double DIP: Re-Evaluating security of logic encryption algorithms, GLSVLSI, 2017
AppSAT: Approximately deobfuscating integrated circuits, HOST, 2017
Security analysis of Anti-SAT, ASP-DAC, 2017

- Removal attacks
  - Based on structural analysis
  - Aim to remove (or disconnect) the locking logic
  - Oracle-less, but need to change the GDSII for the attacker



Corruption?

SARLock: SAT attack resistant logic locking, HOST, 2016

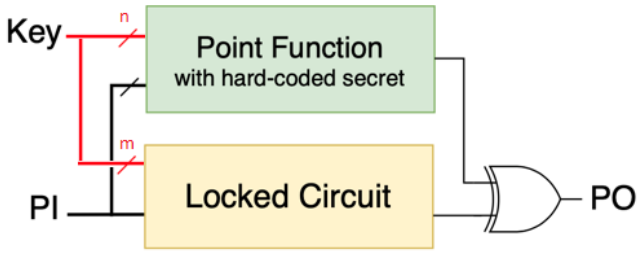
- Compound scheme

Trade-off

between output corruption

and resilience against attacks

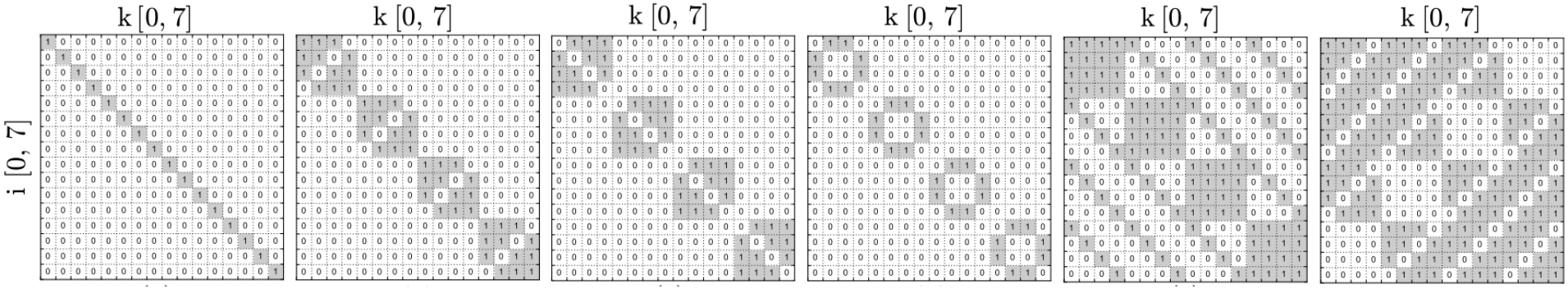
- Approximate attacks
  - Aim to find a key value minimizing corruption
  - Circuit only partially unlocked



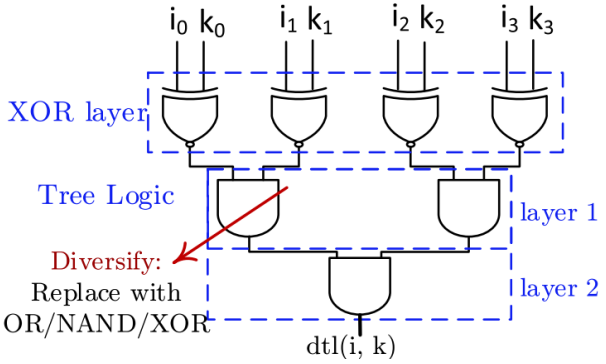
New point functions

On the approximation resilience of logic locking and IC camouflaging schemes, IEEE TIFS, 2019

- Improvement of the added block
  - To prevent removal attacks
  - Or/and to improve output corruption



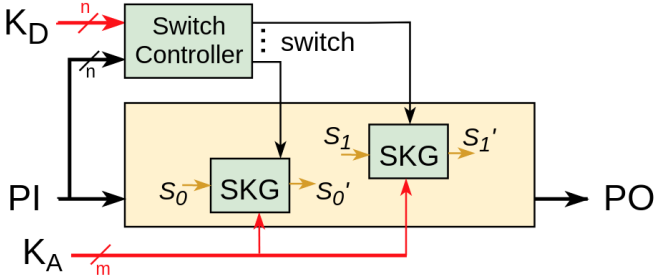
"Classical" AND tree / 1 NAND in the 1st layer / 1 OR in the 1st layer / 1 XOR in the 1st layer / 1 OR in the 2nd layer / 2 OR in the 1st layer



New point functions

On preventing SAT attack with decoy key-inputs, ISVLSI, 2021
SKG-Lock+: A Provably Secure Logic Locking Scheme Creating Significant Output corruption, MDPI Electronics, 2022

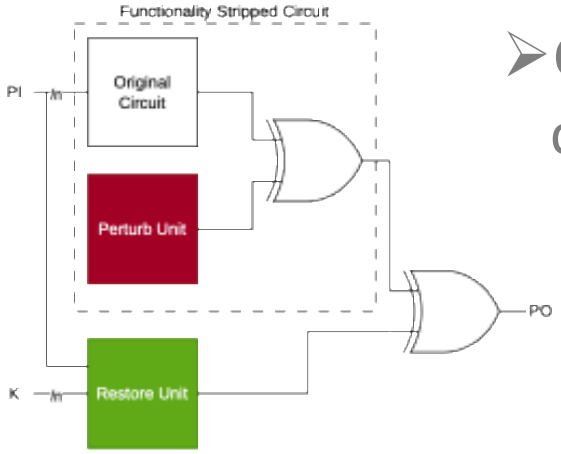
- Use of switchable key-gates with tunable corruptibility, driven by a switch controller
  - Switchable key-gates
    - Driven by key K_A and the switch signal
    - Corrupt the circuit only if K_A is correct and SW=1
  - Switch controller
    - Driven by decoy key K_D
    - Generates the switch signals
    - Allows tuning output corruption



CAC schemes

What to lock? Functional and parametric locking, GLSVLSI, 2017
Provably-Secure logic locking: From theory to practice, CCS, 2017

- New concept of CAC protection schemes: Corrupt And Correct
- To prevent removal attacks
  - The FSC (perturb unit resynthetised with the original circuit) inverts the output if the input = the secret key
  - The Restore unit creates a second inversion if the correct key is applied (it is therefore a comparator)

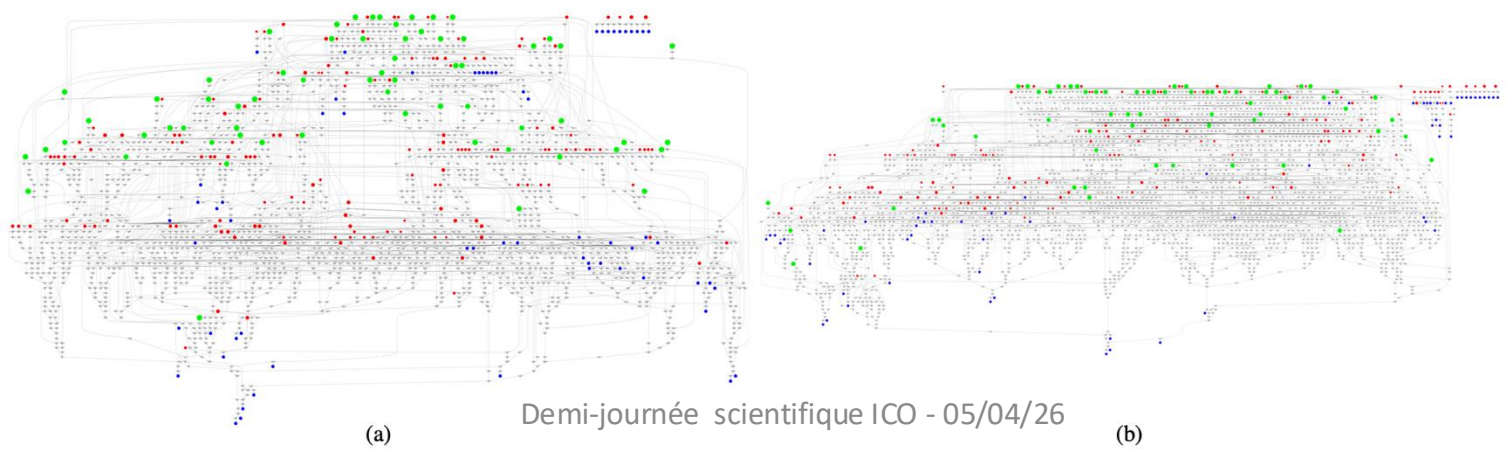


- CAC schemes improvements: to increase output corruption
  - Parameter controlling the Hamming distance between the input and the secret key
  - The restore unit becomes a Hamming distance checker

New attacks

Resynthesis-based attacks against Logic locking, ISQED, 2023

- Resynthesis-based pre-attack
  - Oracle-less
  - Resynthesis of the locked netlist using different synthesis parameters leading to a large number of functionally equivalent but structurally different locked circuits
  - Some variants of the locked circuits may be vulnerable to existing attacks
  - Best attack at the time on CAC schemes
    - But still no 100% key bits discovered



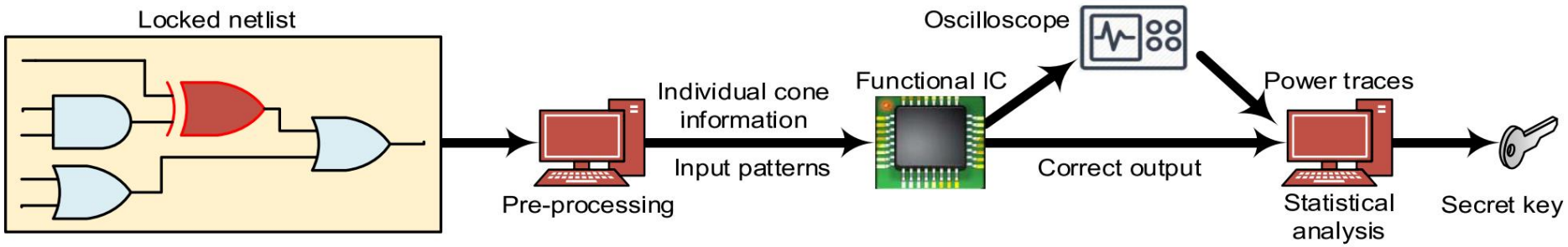
(a) (b)

Side-channels

Logic locking with provable security against analysis attacks, IEEE TCAD, 2020

- First attempt at DPA against logic locking
  - Challenging because of key-aliasing
  - Pre-SAT methods:
    - 60% key bits discovered in 60% of the benchmarks locked by 32-bit keys
    - only 40% for 64-bit keys
  - Post-SAT methods: No key bits discovered

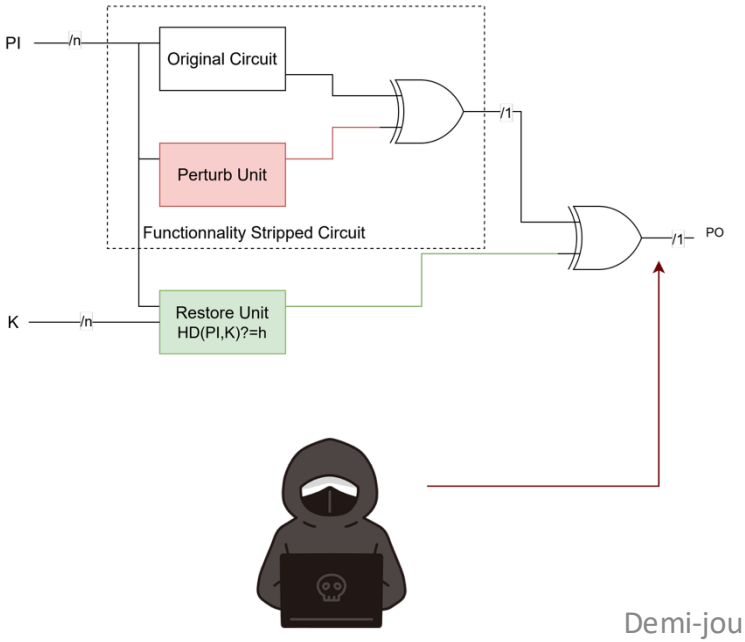
Strong relation between DPA resistance and SAT resilience (?)



Proposed DPA

Power Analysis attack against post-SAT logic locking schemes, ETS, 2024

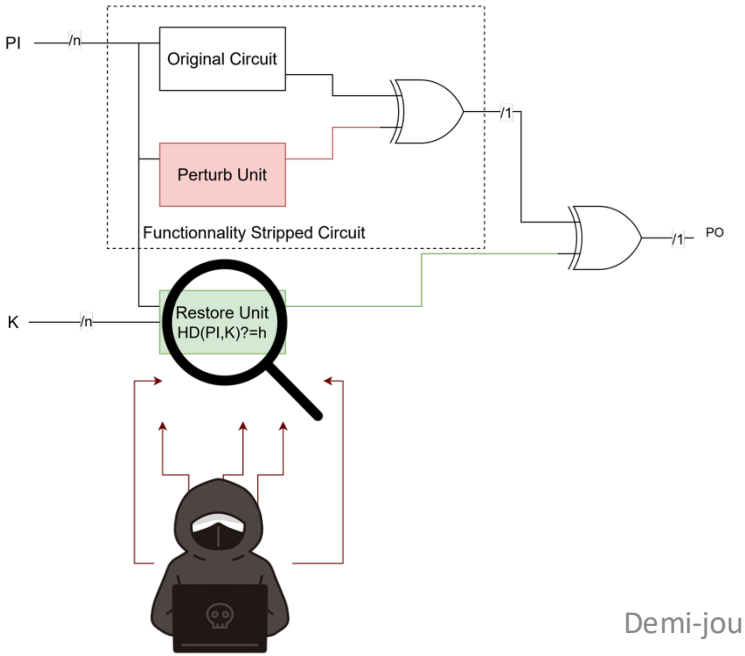
- Attack on a primary output
  - Not enough corruption to distinguish DoM values
  - Brute force attack (the PO is controlled from all key bits)

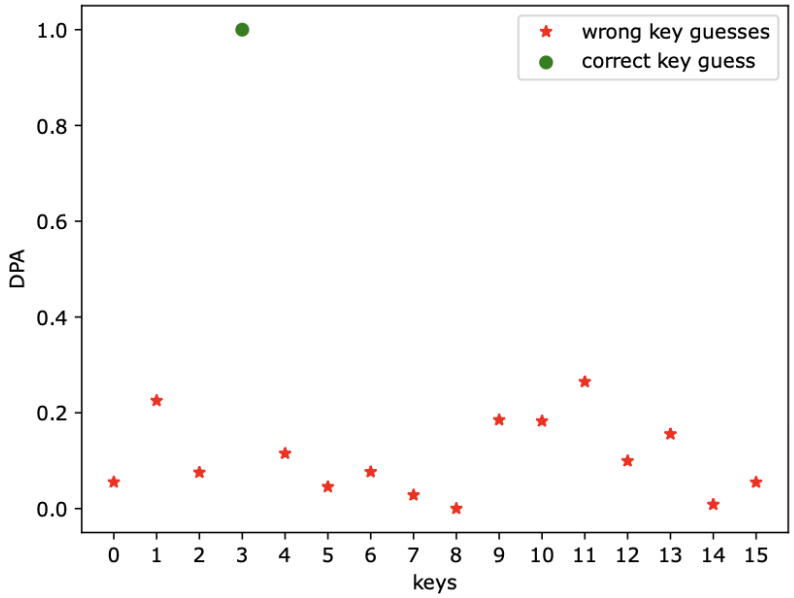


Proposed DPA

Power Analysis attack against post-SAT logic locking schemes, ETS, 2024

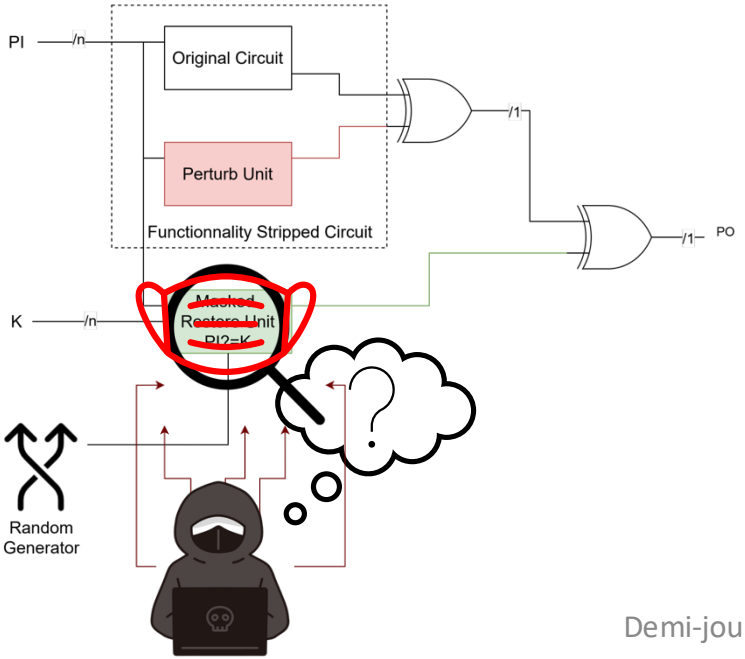
- Attack on the restore unit
  - Divide and Conquer approach
  - Attacks on 4-bit comparators
  - 100% bits discovery on 5 ISCAS85 benchmarks

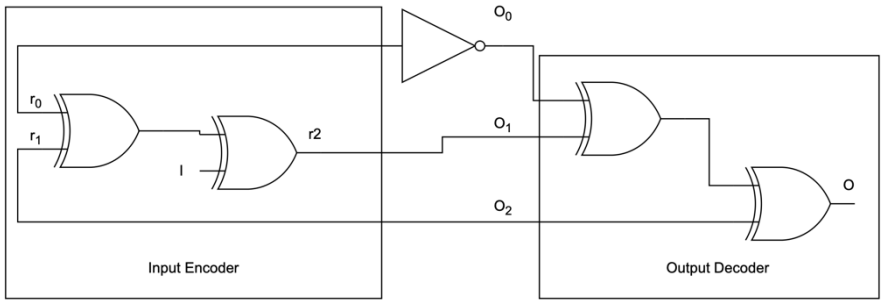




Late Breaking Results: Enhancing CAC-based logic locking with masking for resilience against DPA, ETS, 2026

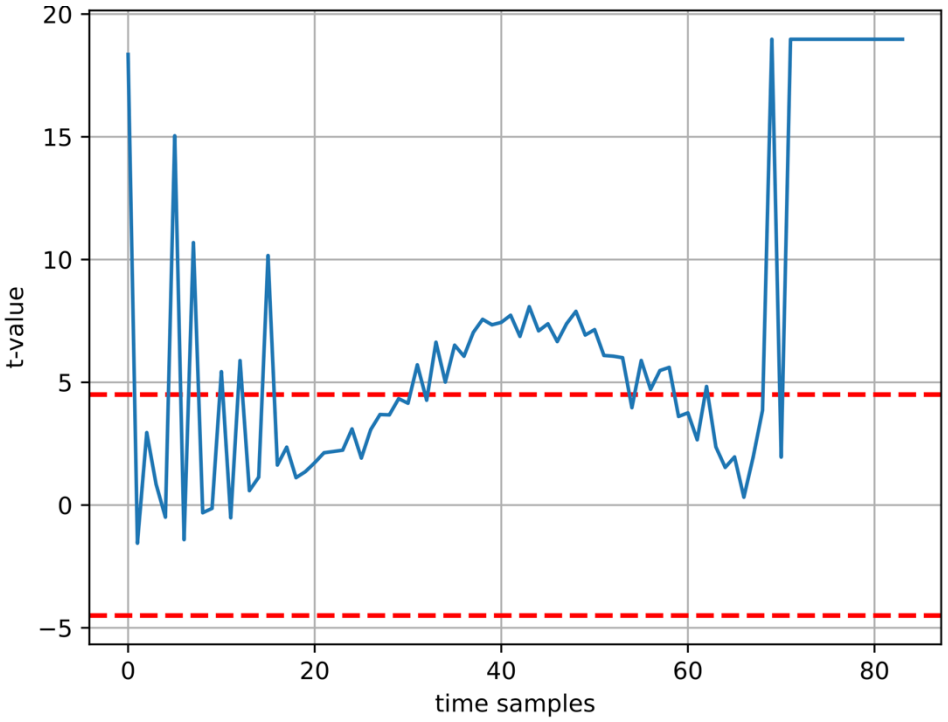
- Randomness introduced
- Information split into several shares
  - X divided into 3 shares:
    - X = R1 xor R2 xor R3,
      with R1 and R2 being randomly generated bits
  - Need of an encoder, a decoder, a resynthesis (using NOT, XOR and AND gates) and masking of each gate



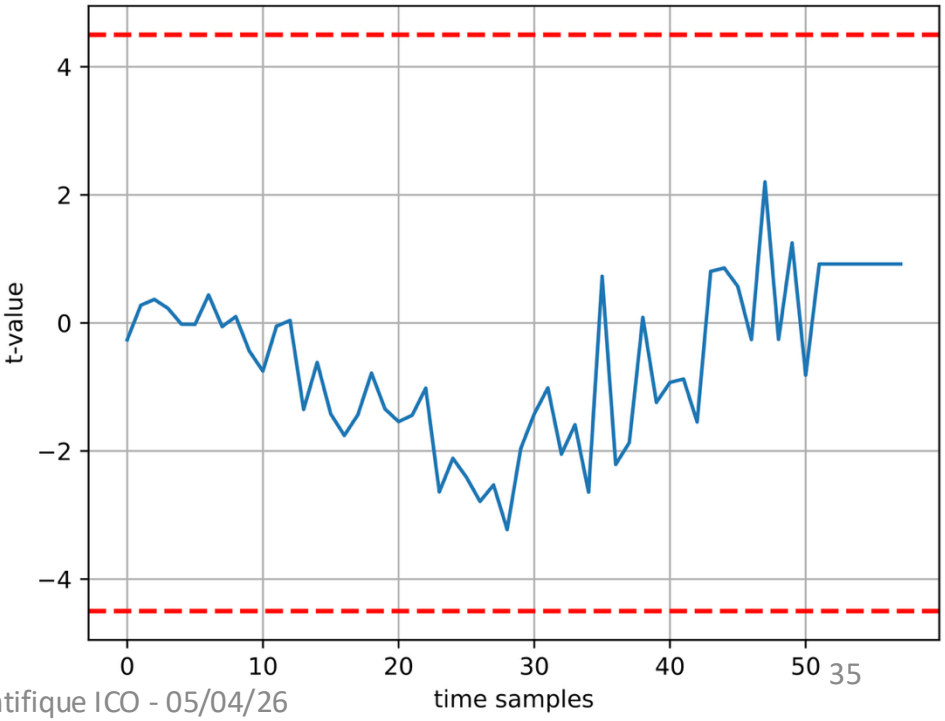


Late Breaking Results: Enhancing CAC-based logic locking with masking for resilience against DPA, ETS, 2026

- Experimental results:
  - Five ISCAS'85 benchmarks locked with masked SFLL-HD^0
- Information leakage: T-test



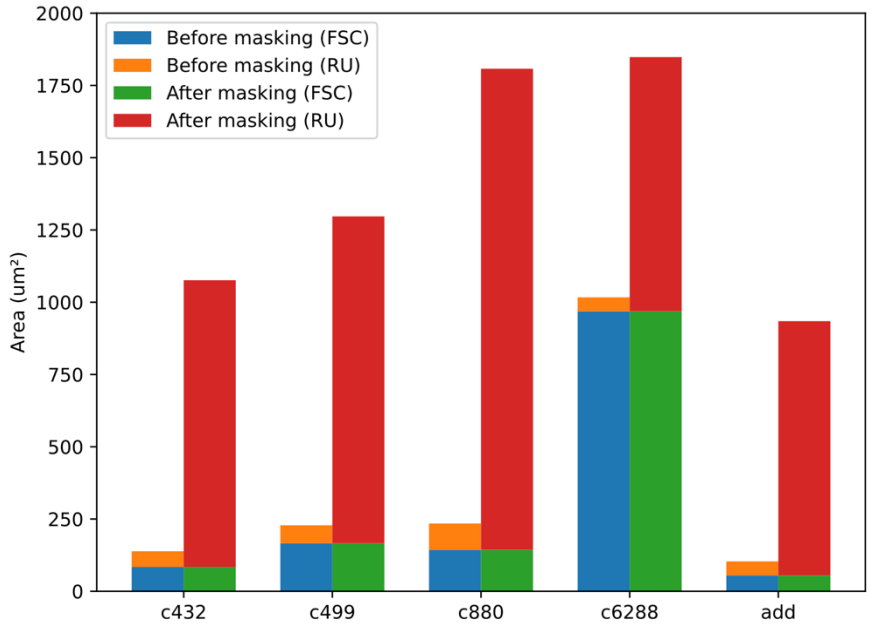
- Locked adder
- Masked Restore Unit

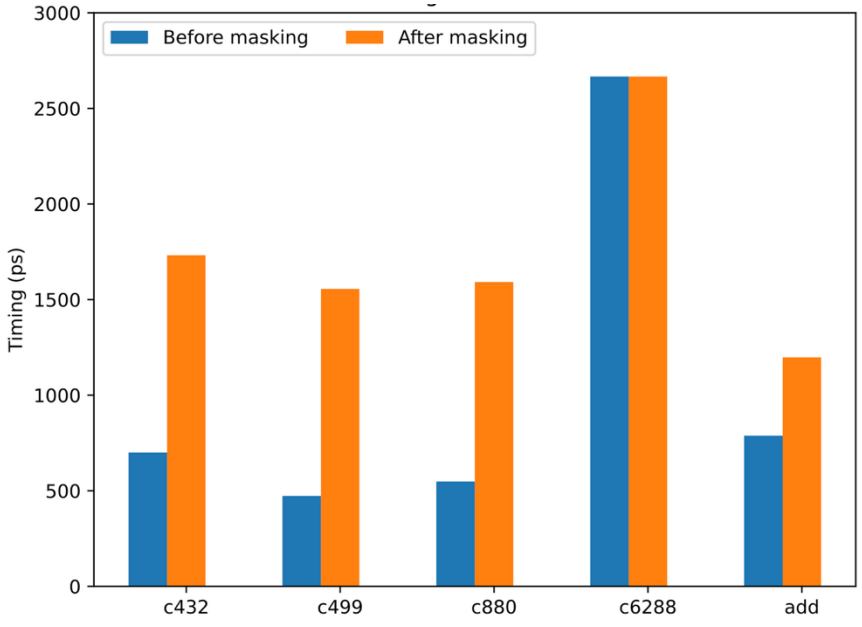


Masking

Late Breaking Results: Enhancing CAC-based logic locking with masking for resilience against DPA, ETS, 2026

- Experimental results :
  - Five ISCAS'85 benchmarks locked with masked SFLL-HD^0
- PPA overheads
  - From +182% to +903% in area





Conclusion

- Logic locking is a very popular technique in the field of hardware trust, which has been attacked and improved numerous times over the last decade
- Each improvement becomes increasingly expensive (and struggles to maintain the quality criterion of corruption)

- At LIRMM, we have worked on this subject within three projects / theses
- The most recent one advanced the state of the art:
  - by attacking, for the first time using DPA, the most secure schemes at that time,
  - and then proposing a countermeasure to ensure security against DPA

- Merci à Marie-Lise, Bruno, Giorgio, Florent, Pascal, Loïc, Papa-Sidy, Linh, Nassim, Clemy…

- 2012-2016: HOMERE+ FUI Project
  (Hardware trOjans: Menaces et robustEsse des ciRcuits intEgrés)
  - 2013/2016: P.-S. Ba PhD
- 2018-2023: MOOSIC ANR Project
  (Multi-Objective Optimised Synthesis to Improve Cybersecurity)
  - 2018/2021: Q.-L. Nguyen PhD
- 2021-2023: SAFEST European twinning project
  (Secure and Assured hardware: Facilitating Estonia's digital society)
- 2022-2025: ARSENE PEPR-Cybersécurité-PP7 Project
  (ARchitecture SEcurisées pour le Numérique Embarqué)
  - 2022/2025: N. Riadi PhD

LIRMM

UNIVERSITÉ DE MONTPELLIER

cnrs

LIRMM

Thank you for your attention!

Any question?

- More ressources on these topics

UNIVERSITÉ DE MONTPELLIER

cnrs