



**HAL**  
open science

# Logic Locking a Design-for-Trust IC Design Technique

Marie-Lise Flottes

► **To cite this version:**

Marie-Lise Flottes. Logic Locking a Design-for-Trust IC Design Technique. FETCH 2020 - École d'hiver Francophone sur les Technologies de Conception des Systèmes Embarqués Hétérogènes, Feb 2020, Montréal, Canada. lirmm-03125660

**HAL Id: lirmm-03125660**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-03125660>**

Submitted on 29 Jan 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# LOGIC LOCKING A DESIGN-FOR-TRUST IC DESIGN TECHNIQUE

**M.-L. Flottes**

*LIRMM (CNRS - Université de Montpellier)*

*France*

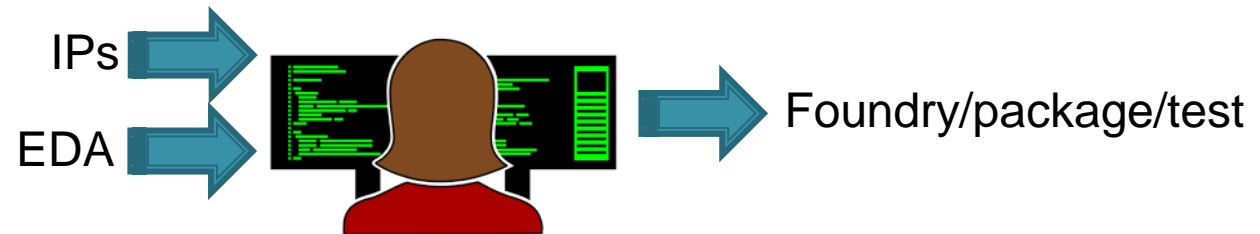
**FETCH 2020**

**École d'hiver Francophone sur les Technologies de Conception  
des Systèmes Embarqués Hétérogènes**

# COUNTERFEIT ICS: SOURCES & ISSUES

---

- Source: profit + globalization



- Issues: Financial loss/Reliability/Security
  - Miss out \$100 billion/year
  - Reported counterfeit parts have been quadrupled since 2009
  - Many sectors are impacted (computers, telecom, automotive, .... military systems)
  - Dramatic consequences on critical systems

[0-3]



# TAXONOMY

---

- Recycled/remarked components
  - Old components sold as new
  - New components sold with higher specification
    - commercial grade → industrial grade → defense grade
- Overproduction: Fabrication outside contract
  - Extra ICs or defective/out-of-spec components
- Cloning: Design copy
  - Reverse Engineering / IPs obtain illegally
- Tampered type: Hardware/Software Trojans (HT/ST)
  - Inserted at any level
  - Time bomb / back door



# COUNTERFEIT DETECTION

---

- Physical detection
  - X-Ray, SEM
- Electrical detection
  - Parametric Tests / Functional tests



Time/Cost and  
Confidence  
issues



# COUNTERFEIT AVOIDANCE

---

- *“Need for development of innovative avoidance mechanisms to be incorporated in the design”*
- (e.g. RO-Based) Sensors: Prevent die and IC recycling [15-16]
- Split manufacturing: Prevent overproduction [17]
- IC camouflaging: Prevent reverse engineering [18]
- Hardware watermarking: Secure IPs [19]
- Hardware metering:
  - Passive methods
    - Digitally stored serial numbers (nonfunctional identification)
    - PUF (functional identification)
  - Active methods: lock each IC until key is provided by the IP holder
    - Initialize IC to a locked state on power up
    - Add an FSM to unlock with the correct sequence to Initial State
    - Logic locking



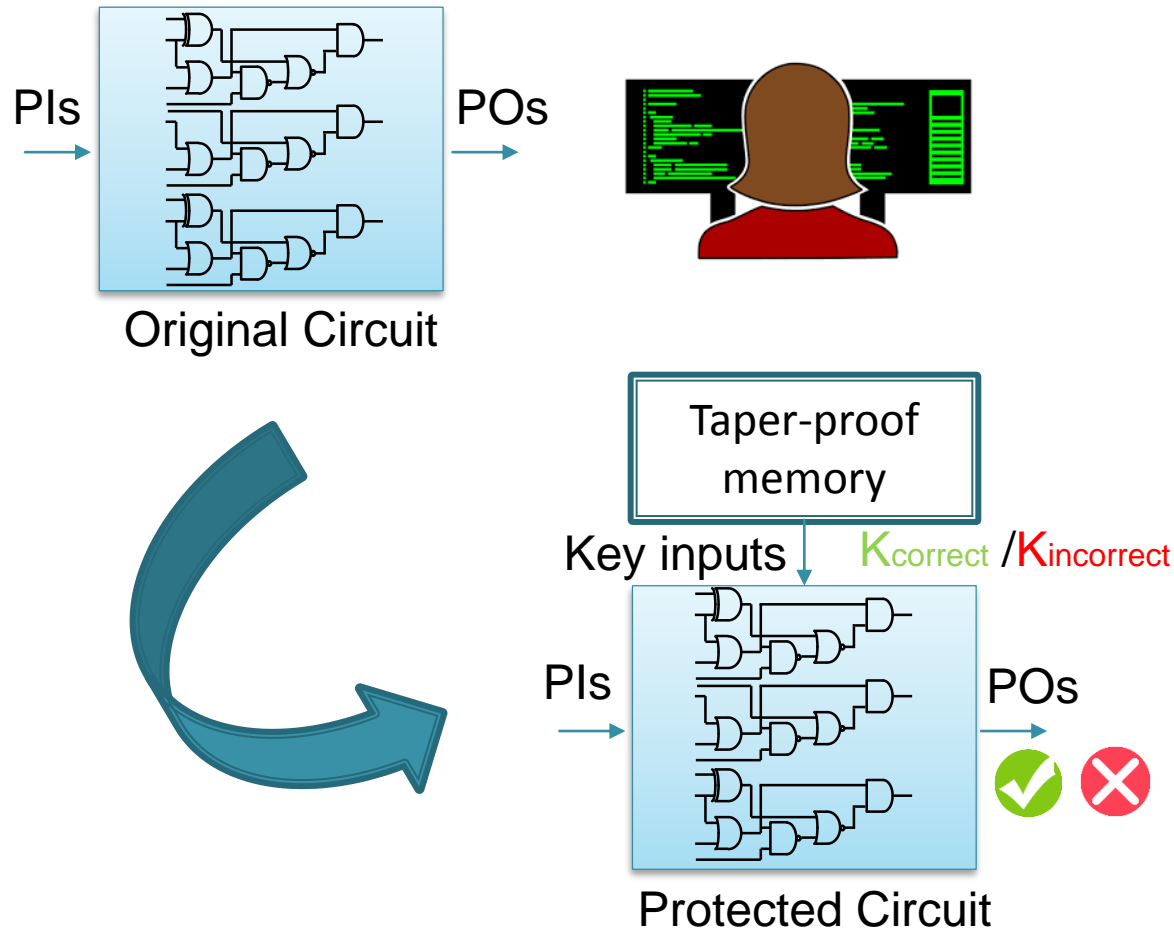
# OUTLINE

---

- Principle
- Implementations
- SAT Attack on logic locking
- Improvement on logic locking solutions and other attacks
- Conclusions

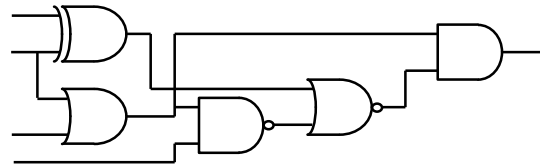


# TECHNICAL PRINCIPLE: KEYING MECHANISM

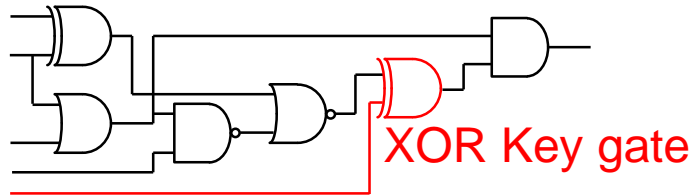




# TECHNICAL PRINCIPLE: KEY GATES & KEY BITS



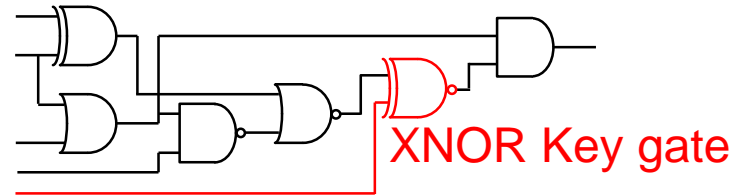
Original Circuit



XOR Key gate

Key bit K1

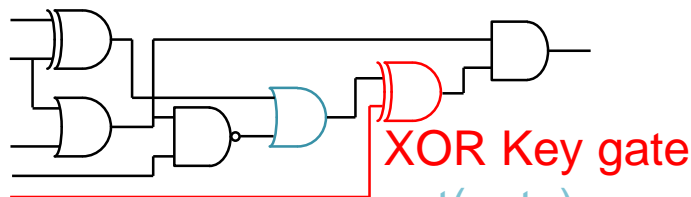
K1=0 ✓ K1=1 ✗



XNOR Key gate

Key bit K1

K1=1 ✓ K1=0 ✗

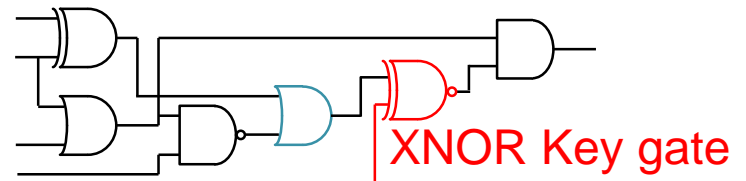


XOR Key gate

not(gate)

Key bit K1

K1=1 ✓ K1=0 ✗



XNOR Key gate

not(gate)

Key bit K1

K1=0 ✓ K1=1 ✗



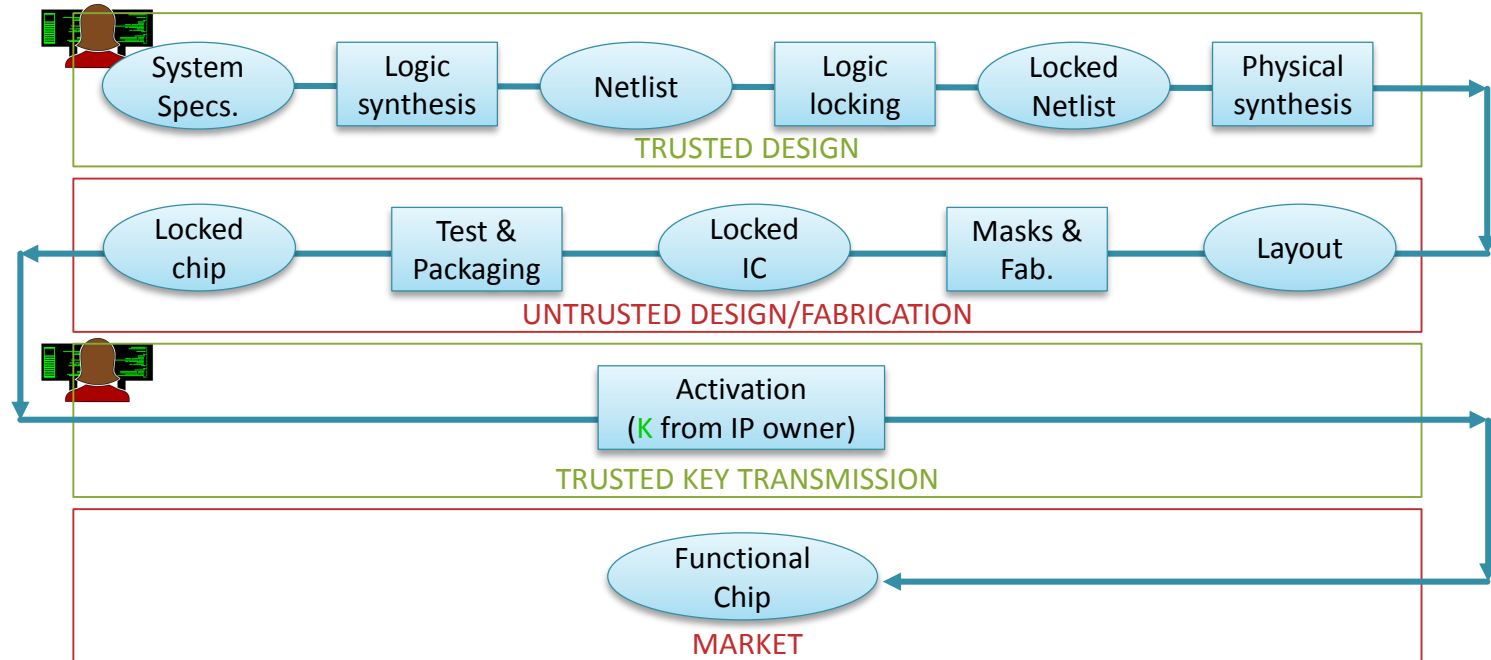
# EVALUATION

---

- Output corruptibility
  - HD(corret outputs, incorrect outputs)
  - Optimum HD = 50% (maximal ambiguity)
- Security
  - Possibilities to penetrate the system using techniques available to an attacker



# APPLICATION PRINCIPLE IN THE IC DESIGN FLOW

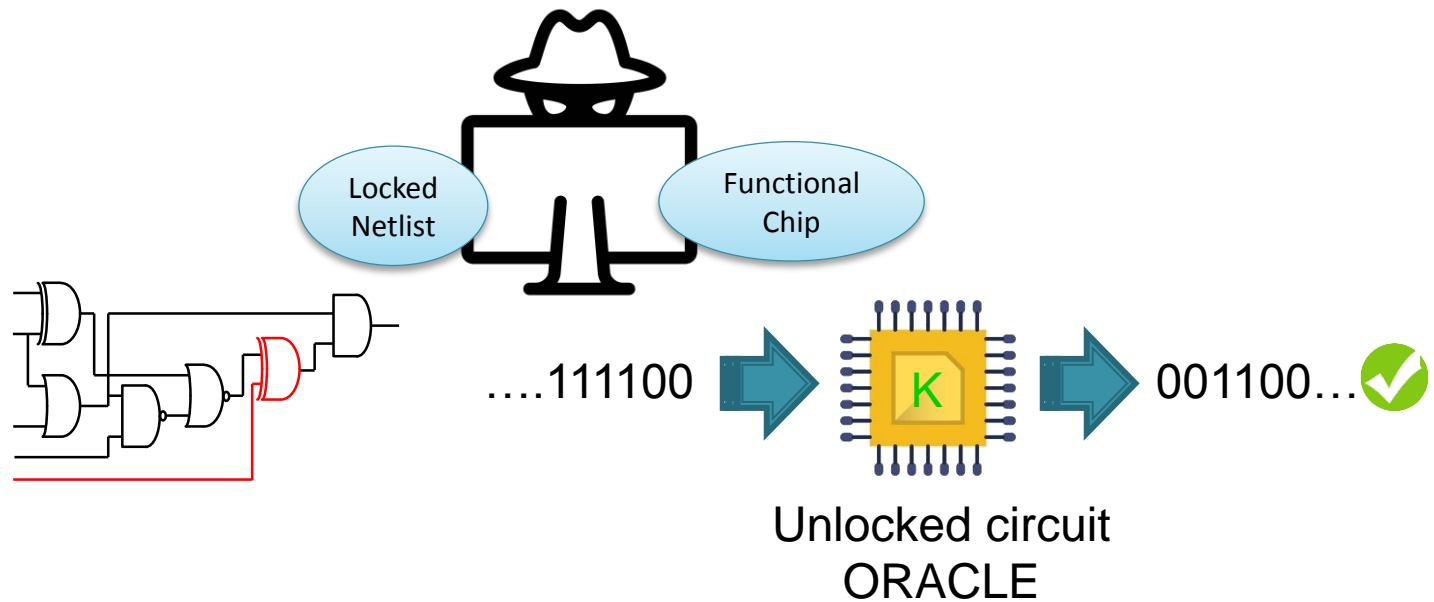


- ✓ Prevents from Reverse Engineering
- ✓ Prevents from Overproduction
- ✓ Makes harder identification of 'safe place' for HT insertion



# ASSUMPTION ON LOGIC LOCKING ATTACKS

- Acker knows the locked netlist / has an unlocked circuit (K inside)



# OUTLINE

---

- Principle
- Implementations
- SAT Attack on logic locking
- Improvement on logic locking solution sand other attacks
- Conclusions



# IMPLEMENTATION(S)

---

- First 2010
  - [6] « EPIC : Ending Piracy of Integrated Circuits »
    - **RLL: Random Logic locking**
    - Introduce k XOR/NXOR key-gates at random locations (while meeting timing constraints)
  - [7] « Preventing IC Piracy Using Reconfigurable Logic Barriers »
    - **LUT-based locking** (Correct/incorrect LUT programming provide modification of the information flow)
    - Introduce LUT at chosen location for maximum attacker effort (low-controllable nodes), and for optimal output corruption (high observable nodes)



# IMPLEMENTATION(S) CONT'D

---

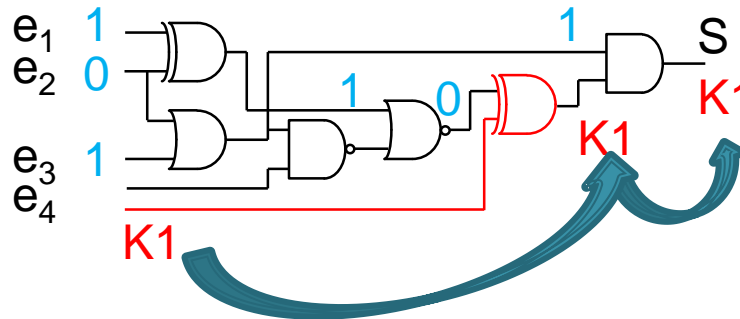
- First improvements (output corruption)
  - [8] 2015 « Fault Analysis-Based Logic Encryption »
    - **FLL: Fault-Analysis-based logic locking**
    - Introduce  $k$  XOR/NXOR key-gates at chosen locations for optimal output corruption
    - Metric (maximal number of patterns  $NC$  to control the node & maximal number of affected primary outputs  $NO$ )
    - Highest FI =  $NC_0 \times NO_0 + NC_1 \times NO_1$
  - [9] Variante 2017
    - **WLL: Weighted logic locking**
    - XOR key-gates fed by multiple key-bits through additional AND/OR gates which leads to a higher output corruptibility



# IMPLEMENTATION(S) CONT'D

- First improvements (security)
  - [10-11] 2012-2016

- Issue



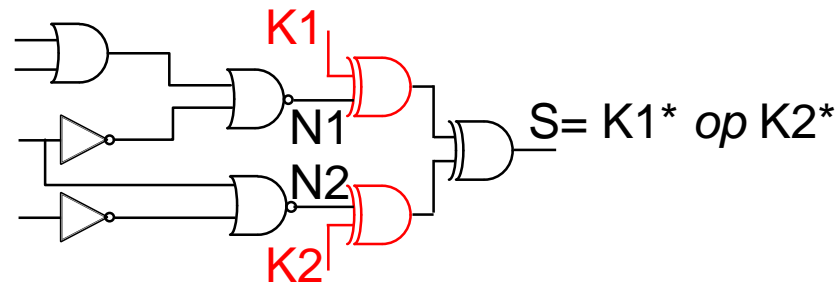
Input Patterns  
 $(e_1, e_2, e_3, e_4) = (1, 0, 1, x)$



$S = K_1 !$

- SLL: Strong Logic Locking

Introduce XOR/NXOR key-gates at chosen locations for ensuring interdependence among key bits





# OUTLINE

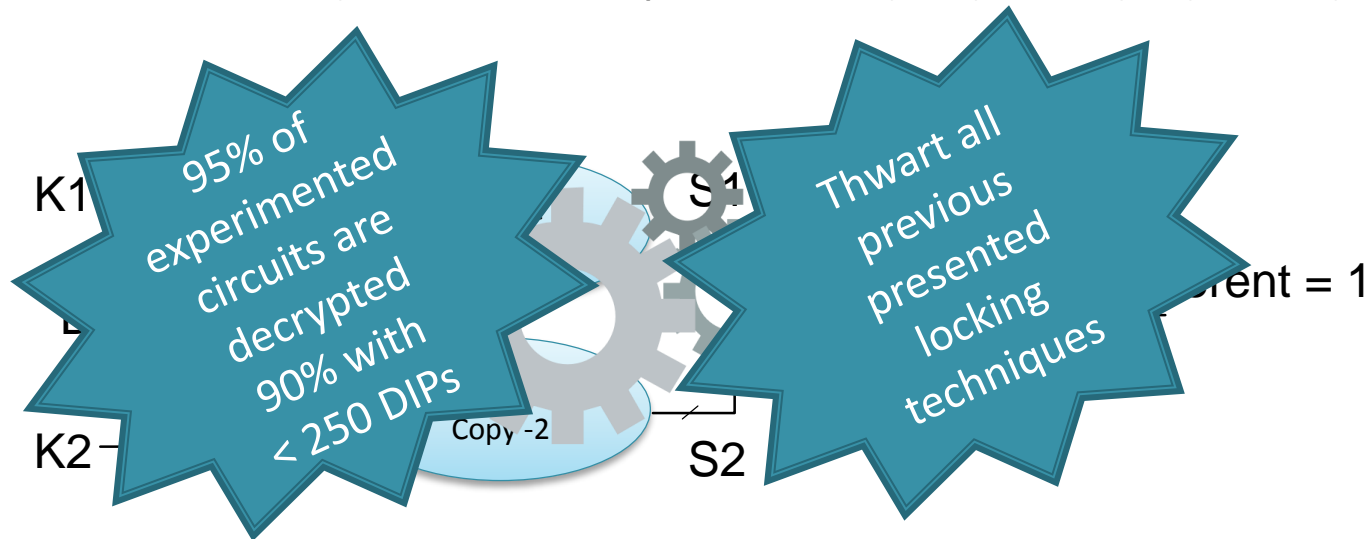
---

- Principle
- Implementations
- SAT Attack on logic locking
- Improvement on logic locking solutions and other attacks
- Conclusions



# SAT ATTACK

- Boolean Satisfiability attack (SAT attack [12] 2015): Iteratively rules out incorrect keys
  - 1/ Found a DIP (Differential Input Pattern) /  $f(\text{DIP}, K1) \neq f(\text{DIP}, K2)$



- 2/ Compare  $f(\text{DIP}, K_i)$  with Oracle(DIP)
  - If  $f(\text{DIP}, K_i) \neq \text{Oracle}(\text{DIP})$ ,  $K_i$  can be rejected
- 3/ Iterate until no more DIP is found
  - All incorrect keys have been rejected



# OUTLINE

---

- Principle
- Implementations
- SAT Attack on logic locking
- Improvement on logic locking solutions and other attacks
- Conclusions



# POST-SAT-ATTACK SOLUTIONS

- Resisting the SAT-attack by increasing its Execution time

$$\text{SAT Execution Time: } ET = \sum_{i=1}^{iter} t_i$$

- ⇒ Controlling the distinguishing ability of DIPs
- ⇒ Rule out at most one incorrect key per DIP

Inputs			Original O	O for k <sub>i</sub>							
I1	I2	I3		K0	K1	K2	K3	K4	K5	K6	K7
0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	1	0	1	1	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1
0	1	1	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0
1	1	0	1	1	1	1	1	1	1	0	1
1	1	1	0	0	0	0	0	0	0	0	1

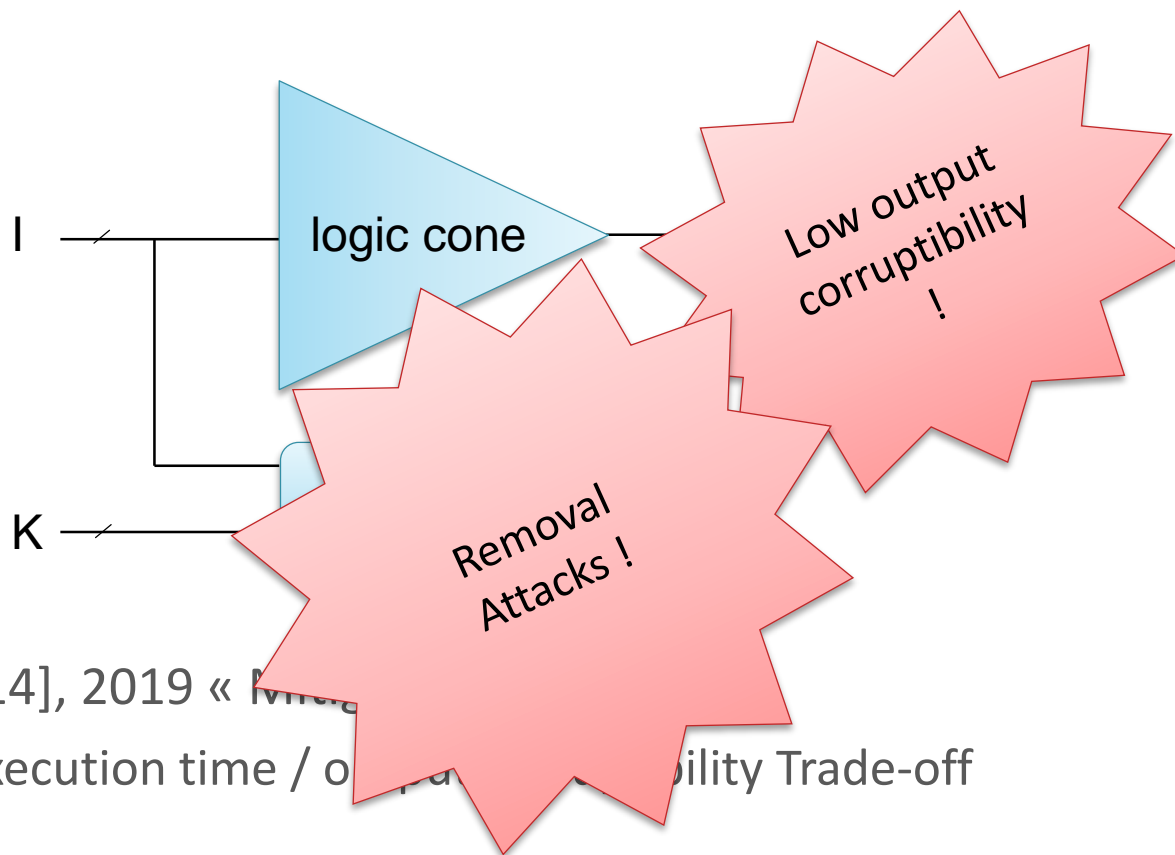
$2^k - 1$  DIPs to succeed !



# POST-SAT-ATTACK SOLUTIONS (CONT'D)

---

- SARLock [13], 2016 « SAT Attack Resilient logic locking »



- Anti-SAT, [14], 2019 « Mitigating SAT Attacks on Logic Locking »
  - SAT Execution time / output corruptibility Trade-off



# OTHER ATTACKS ON LOGIC LOCKING

---

- Removal attacks
  - remove locking mechanisms from the studied netlist
- Approximate attacks on compound logic locking techniques (eg SARLock+FLL)
  - returns an approximate key (only FLL key bits are extracted) leaving the low-corruptability constituent in the netlist (SARLock countermeasure)
- Power side-channel attacks
- Oracle-less attacks (e.g. redundancy identification)



# CONCLUSION

---

## ○ Design for Trust (DfTr)

- Watermarking that embeds a designer's signature into the design
- Passive metering that enables tracking of individual ICs throughout their lifetime
- Camouflaging that introduces look-alike structures at the layout-level
- Split manufacturing that involves partial fabrication at two separate foundries
- And...



## ○ Logic locking

- Locks a design with key-controlled protection logic
- Protection anywhere in the supply chain
  - Rogue SoC integrator (IP reuse)
  - Untrusted foundry (overproduction, HT)
  - Untrusted test facility (sell defective parts, recycling)
  - Malicious end-user (replicate)



# WORK IN PROGRESS

---

- All logic Locking solutions exhibit specific weakness
- No metrics
- May exhibit vulnerabilities after implementation
- Implementation Cost





---

# Merci !



# REFERENCES

---

---

- [0] <http://www.blogpresidentcnac.fr/lutter-contre-la-contrefacon-de-composants-electroniques/>
- [1] U.S. Senate Committee on Armed Services, "Inquiry into counterfeit electronic parts in the Department of Defence supply chain," May 2012. [Online]. Available: <http://www.armedservices.senate.gov/Publications/Counterfeit%20Electronic%20Parts.pdf>
- [2] U.S. Department of Commerce, "Defense industrial base assessment: Counterfeit electronics," Jan. 2010.
- [3] "Reports of counterfeit parts quadruple since 2009, challenging U.S. Defence Industry and National Security," Apr. 2012. [Online]. Available: <http://www.ihs.com/images/IHS-iSuppli-Reports-Counterfeit-Parts-Quadruple-Since-2009.pdf>.
- [4] Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain, Ujjwal Guin et al., Proceedings of the IEEE, Vol. 102, No. 8, August 2014
- [5] G. Contreras, T. Rahman, and M. Tehranipoor, "Secure split-test for preventing IC piracy by untrusted foundry and assembly," in Proc. Int. Symp. Defect Fault Tolerance VLSI Syst., 2013, pp. 196–203.
- [6] J. Roy, F. Koushanfar, and I. Markov, "Ending Piracy of Integrated Circuits," IEEE Computer, vol. 43, pp. 30–38, 2010.
- [7] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," IEEE Design & Test of Computers, vol. 27, no. 1, pp. 66–75, 2010.
- [8] J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption," IEEE Transactions on Computer, vol. 64, no. 2, pp. 410–424, 2015.
- [10] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security Analysis of Logic Obfuscation," IEEE/ACM Design Automation Conference, pp. 83–89, 2012.
- [11] M. Yasin, J. Rajendran, O. Sinanoglu, and R. Karri, "On Improving the Security of Logic Locking," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 35, pp. 1411–1424, 2016.
- [12] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," IEEE International Symposium on Hardware Oriented Security and Trust, pp. 137–143, 2015.
- [13] M. Yasin, B. Mazumdar, J. Rajendran, and O. Sinanoglu, "SARLock: SAT Attack Resistant Logic Locking," IEEE International Symposium on Hardware Oriented Security and Trust, pp. 236–241, 2015.
- [14] Yang Xie, Ankur Srivastava, Anti-SAT: Mitigating SAT Attack on Logic Locking, IEEE Trans. On CAD of inte. Circuits and systems, VOL. 38, NO. 2, feb. 2019
- [15] X. Zhang, N. Tuzzio, and M. Tehranipoor, "Identification of recovered ICS using fingerprints from a light-weight on-chip sensor," in Proc. IEEE Design Autom. Conf., Jun. 2012, pp. 703–708.
- [16] X. Zhang and M. Tehranipoor, "Design of on-chip lightweight sensors for effective detection of recycled ICs," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 5, pp. 1016–1029, May 2014.
- [17] R. Jarvis and M. G. McIntyre, "Split manufacturing method for advanced semiconductor circuits," U.S. Patent 7 195 931, 2004.



# REFERENCE (CONT')

---

- [18] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in Proc. ACM Conf. Comput. Commun. Security, 2013, pp. 709–720.
- [19] Watermarking Techniques for Intellectual Property Protection A. B. Kahng, J. Lach<sup>†</sup>, W. H. Mangione-Smith<sup>†</sup>, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker<sup>‡</sup>, H. Wang and G. Wolfe, IEEE DAC 776-781, 1998
- [A] J. Roy, F. Koushanfar, and I. Markov, "Ending Piracy of Integrated Circuits," IEEE Computer, vol. 43, pp. 30–38, 2010.

