Provably Secure Camouflaging Strategy for IC Protection

Meng Li[1] Kaveh Shamsi[2] Travis Meade[2] Zheng Zhao[1]
Bei Yu[3] Yier Jin[2] David Z. Pan[1]

[1]Electrical and Computer Engineering, University of Texas at Austin

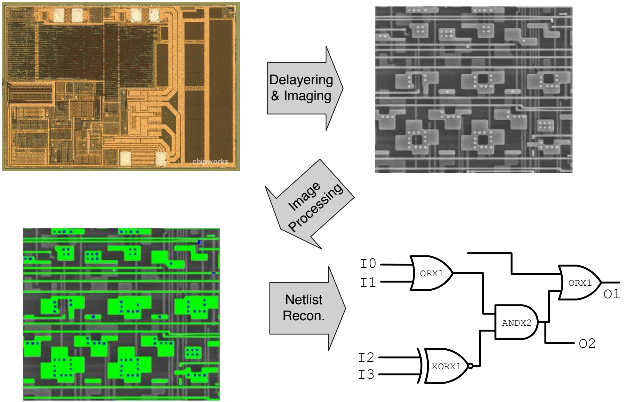
[2]Electrical and Computer Engineering, University of Central Florida

[3]Computer Science and Engineering, The Chinese University of Hong Kong

ICCAD2016 - November 07, 2016 - Austin, TX

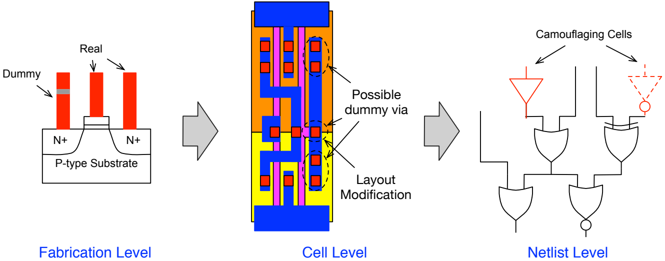
Introduction

- IP protection against reverse engineering becomes a significant concern
- Reverse engineering flow



Introduction

- IC camouflaging is proposed to hide circuit functionality
  - Layout technique
  - Create cells that look alike but have different functionalities



- Open questions to solve:
  - How to evaluate the security of a camouflaged netlist
  - How to reduce the overhead introduced by IC camouflaging

State-of-The-Art IC Camouflaging

- Fabrication level techniques:
  - Contact- and doping-based techniques [Chow+, US Patent'07]

State-of-The-Art IC Camouflaging

- Fabrication level techniques:
  - Contact- and doping-based techniques [Chow+, US Patent'07]
- Cell level designs:
  - Camouflaging lookup table [Malik+, ISVLSI'15]

State-of-The-Art IC Camouflaging

- Fabrication level techniques:
  - Contact- and doping-based techniques [Chow+, US Patent'07]
- Cell level designs:
  - Camouflaging lookup table [Malik+, ISVLSI'15]
- Netlist level camouflaging cell insertion strategy:
  - Insertion based on interference graph [Rajendran+, CCS'13]

State-of-The-Art IC Camouflaging

- Fabrication level techniques:
  - Contact- and doping-based techniques [Chow+, US Patent'07]
- Cell level designs:
  - Camouflaging lookup table [Malik+, ISVLSI'15]
- Netlist level camouflaging cell insertion strategy:
  - Insertion based on interference graph [Rajendran+, CCS'13]
- **Our contribution**
  - A provably secure criterion is proposed and formally analyzed from Machine Learning perspective
  - Two factors that improve the circuit security are revealed
  - A camouflaging framework is proposed to increase the security exponentially with linear increase of overhead

Preliminary: Reverse Engineering Attack

- Knowledge of the attacker:
  - Get camouflaged netlists
    - Include cells and connections
  - Differentiate regular and camouflaging cells
    - Don't know the specific functionality of camouflaging cells
  - Acquire a functional circuit as black box
    - Don't have access to internal signals

Preliminary: Reverse Engineering Attack

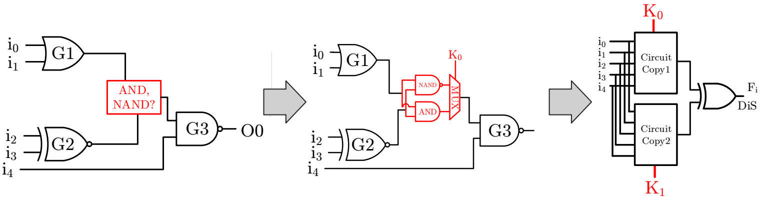
- Knowledge of the attacker:
  - Get camouflaged netlists
    - Include cells and connections
  - Differentiate regular and camouflaging cells
    - Don't know the specific functionality of camouflaging cells
  - Acquire a functional circuit as black box
    - Don't have access to internal signals
- The attacker aims to recover the circuit functionality by querying the black-box functional circuit

Preliminary: Reverse Engineering Attack

- Knowledge of the attacker:
  - Get camouflaged netlists
    - Include cells and connections
  - Differentiate regular and camouflaging cells
    - Don't know the specific functionality of camouflaging cells
  - Acquire a functional circuit as black box
    - Don't have access to internal signals
- The attacker aims to recover the circuit functionality by querying the black-box functional circuit
- Attacker query strategy:
  - Brute force attack
  - Testing-based attack [Rajendran+, CCS'13]
  - SAT-based attack [Massad+, NDSS'15]

Preliminary: SAT-based Attack

- Key idea:
  - Only query black box with input patterns that can help remove false functionalities
- No existing camouflaging strategy demonstrates enough resilience



IC De-camouflaging Modeled As a Learning Problem

- IC de-camouflaging can be modeled as a learning problem
  - Functions of camouflaged circuit $\leftrightarrow$ A set of boolean functions
  - Original circuit $\leftrightarrow$ Target boolean function
  - Input-output pairs $\leftrightarrow$ Samples
- Different attack methods correspond to different sampling strategies
  - Brute force attack $\leftrightarrow$ Random sampling
  - SAT-based attack $\leftrightarrow$ Query by disagreement
  - *SAT-based attack requires asymptotically less number of input-output pairs compared with brute force attack*

IC Camouflaging Security Analysis

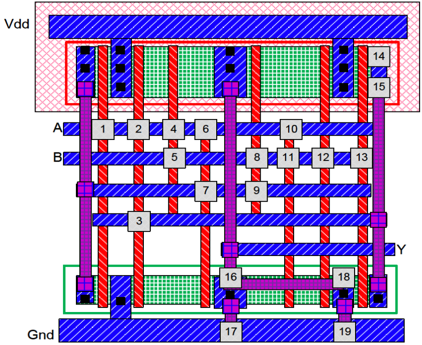
- De-camouflaging complexity (DC)
  - Number of input patterns the attacker needs to query to resolve circuit functionality
  - Independent of how the de-camouflaging problem is formulated
- Then, de-camouflaging complexity is

$$DC \sim O(\theta d \log(\frac{1}{\epsilon}))$$

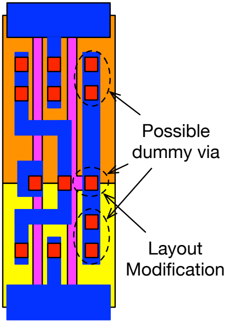
  - d: characterize the total number of functionalities
  - θ: characterize the number of functionalities that can be pruned by each input pattern
  - ϵ: output error probability for the resolved circuit
  - Intrinsic trade-off between DC and output error probability
- Need to increase θ and d to enhance security

Novel Camouflaging Cell Generation Strategy

- Target at increasing d for better security
- To increase d
  - Increase the number of functionalities of the camouflaging cells
  - Increase the number of cells inserted into the netlist



NAND/NOR/XOR



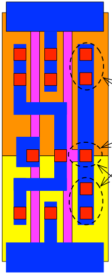
BUF/INV

Novel Camouflaging Cell Generation Strategy

- Observation:
  - Overhead of a cell depends on its functionality
- Cell design strategy:
  - Build cells with negligible overhead for certain functionality
- Two different types:
  - Dummy contact-based camouflaging cells
  - Stealth doping-based camouflaging cells

Novel Camouflaging Cell Generation Strategy

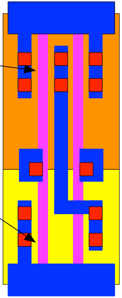
- Dummy contact-based camouflaging cells



	BUF		AND2		OR2	
Function	BUF	INV	AND2	NAND2	OR2	NOR2
Timing	1.0x	2.0x	1.0x	1.5x	1.0x	1.9x
Area	1.0x	1.5x	1.0x	1.3x	1.0x	1.3x
Power	1.0x	1.5x	1.0x	0.9x	1.0x	1.1x

Novel Camouflaging Cell Generation Strategy

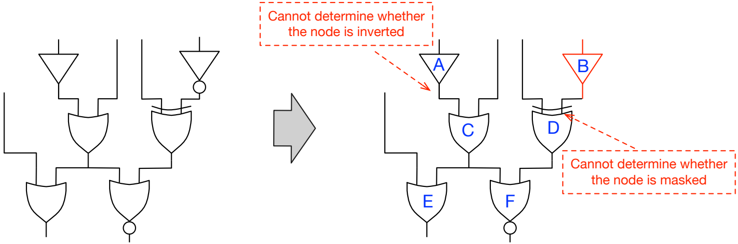
- Stealth doping-based camouflaging cells



	AND2		OR2		NAND2	
Function	AND2	BUF	OR2	BUF	NAND2	INV
Timing	1.0x	1.4x	1.0x	1.4x	1.0x	1.6x
Area	1.0x	1.3x	1.0x	1.3x	1.0x	1.5x
Power	1.0x	1.2x	1.0x	1.2x	1.0x	1.5x

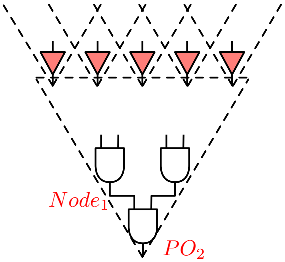
Novel Camouflaging Cell Generation Strategy

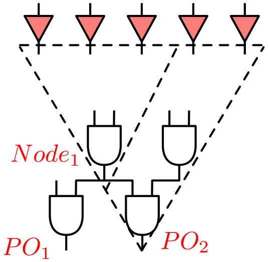
- Characteristics of two type camouflaging cells:
  - Dummy contact-based cell: error probability is 1
  - Stealth doping-based cell: enable dummy wire connection
- Contact and doping technique can be further combined to increase the number of functionalities



AND-Tree Camouflaging Strategy

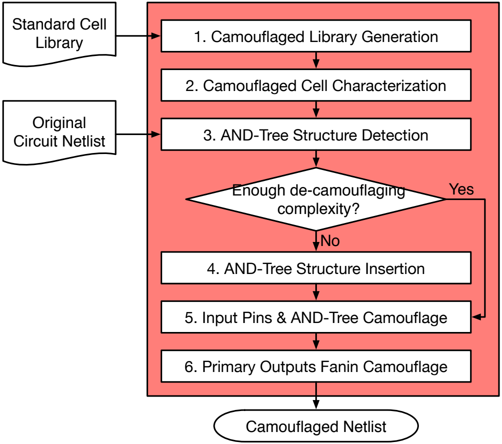
- Target at increasing θ for better security
- AND-Tree achieves high resilience against SAT-based attack
  - Represent a class of circuits with output 0/1 for only one input
- We find θ increases exponentially for ideal AND-Tree
  - Unbiased primary inputs: i.i.d binary distribution
  - Non-decomposability





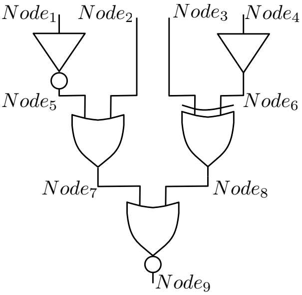
Overall Camouflaging Framework

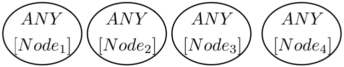
- Combine the proposed camouflaging strategy
  - Leverage camouflaging cells to insert AND-Tree



AND-Tree Detection

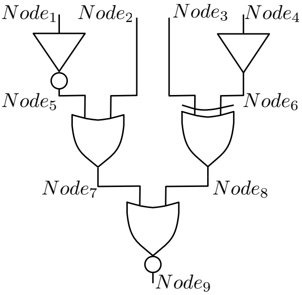
- Detect existing AND-Tree structure in the netlist
- Important criterion:
  - AND-Tree size
  - AND-Tree input bias (distance with ideal distribution)
  - AND-Tree de-composability
- Example:

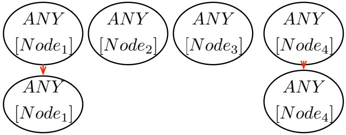




AND-Tree Detection

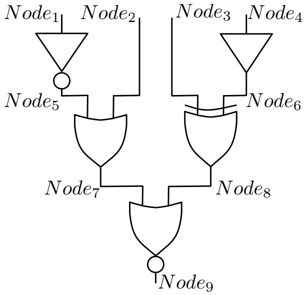
- Detect existing AND-Tree structure in the netlist
- Important criterion:
  - AND-Tree size
  - AND-Tree input bias (distance with ideal distribution)
  - AND-Tree de-composability
- Example:

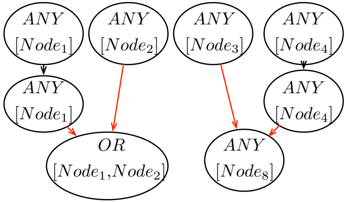




AND-Tree Detection

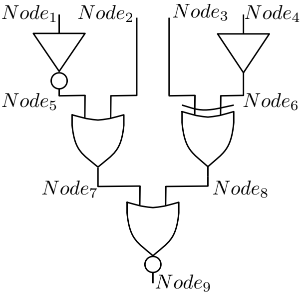
- Detect existing AND-Tree structure in the netlist
- Important criterion:
  - AND-Tree size
  - AND-Tree input bias (distance with ideal distribution)
  - AND-Tree de-composability
- Example:

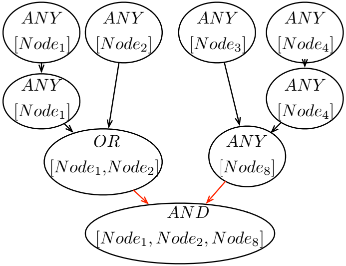




AND-Tree Detection

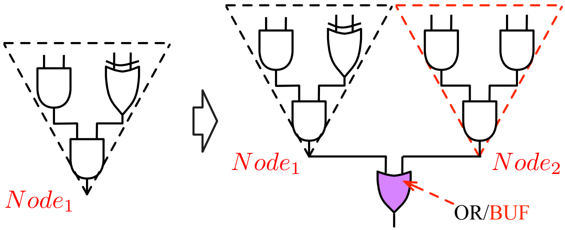
- Detect existing AND-Tree structure in the netlist
- Important criterion:
  - AND-Tree size
  - AND-Tree input bias (distance with ideal distribution)
  - AND-Tree de-composability
- Example:





AND-Tree Insertion

- Insert AND-Tree when no trees exist in original netlist
  - Guarantee non-decomposable
  - Guarantee unbiasedness by connecting tree inputs to primary inputs
- To insert AND-Tree into the netlist



- θ increases exponentially as the inserted AND-Tree size

AND-Tree Insertion

- Node selection criterion for AND-Tree insertion
  - Consider timing/Power overhead, error impact
- Define insertion score (IS) for each node

$$IS = \frac{\alpha \times SA - \beta \times P_{ob}}{N_O}$$

  - SA: switching probability
  - P_{ob}: observe probability
  - N_O: number of outputs in the fanout cone
- Select nodes iteratively until AND-Tree exists in the fanin cone of each output

Experimental Results

- **Experimental setup**
  - SAT-based de-camouflaging attack [Subramanyan+, HOST'15]
  - Runtime limit 1.5×10^5s
  - Camouflaging framework implemented in `C++`
  - Timing/Power analysis with `Primetime/Primetime-PX`
  - Benchmark: ISCAS'85 and MCNC

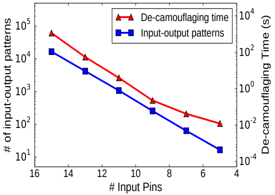
Experimental Results

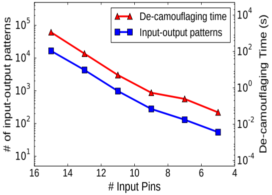
- Examination of cell generation strategy
  - Use the proposed camouflaging cells to rebuild the benchmarks

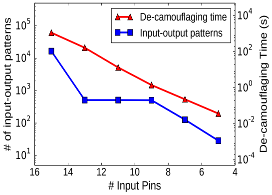
bench		# input	# output	# gate	time (s)	# iter
ISCAS	c432	36	7	203	1.758	80
	c880	60	23	466	1.2×10^4	148
	c1908	33	25	938	N/A	N/A
	c2670	233	64	1490	N/A	N/A
	c3540	50	22	1741	N/A	N/A
	c5315	178	123	2608	N/A	N/A
MCNC	i4	192	6	536	1.9×10^3	743
	apex2	39	3	652	N/A	N/A
	ex5	8	63	1126	6.9×10^2	139
	i9	88	63	1186	2.1×10^4	81
	i7	199	67	1581	1.5×10^2	225
	k2	46	45	1906	N/A	N/A

Experimental Results

- Examination of AND-Tree structure
  - Ideal AND-Tree
  - Impact of decomposability and input bias

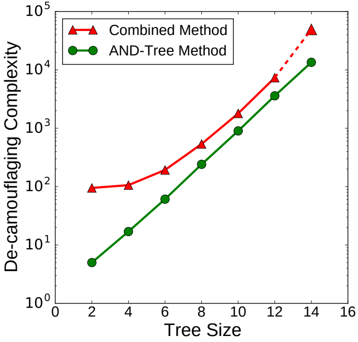


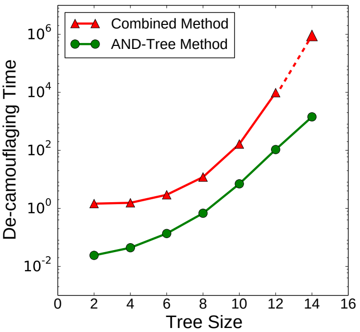




Experimental Results

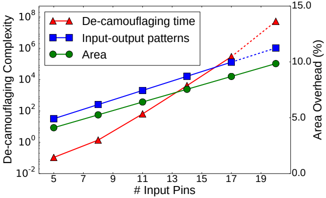
- De-camouflaging complexity of the proposed framework
  - Combined strategy v.s. AND-Tree strategy





Experimental Results

- Overhead of the proposed framework



bench	# gate	area (%)	power (%)	timing (%)
c432	203	16.7	14.1	0.30
c499	275	5.83	4.32	0.00
c880	466	9.85	10.8	0.06
i4	536	12.0	8.73	0.00
i7	1581	5.41	4.02	0.15
ex5	1126	4.15	3.73	0.11
ex1010	5086	0.75	1.06	0.00
des	6974	0.64	0.23	0.00
sparc_exu	27368	0.22	0.05	0.00

Conclusion

- The security criterion is formally analyzed based on the equivalence to active learning
- Two camouflaging techniques are proposed to enhance the security of circuit netlist
- A provably secure camouflaging framework is developed to combine two techniques
- Effectiveness of the framework is verified with experiments and demonstrate good resilience achieved with small overhead

Thanks for your attention!

Back Up: Cell Generation Strategy Comparison

- Comparison with two different cell generation strategies
- Assume
  - Circuit size: N
  - Number of functions of each camouflaging cells
    - Previous method: m_1
    - Our method: m_2
  - Number of modified cells: n
- Number of possible functionalities
  - Previous method: $\sim {m_1}^n$
  - Our method: $\sim C_N^n {m_2}^n$
- If $N = 1000, m_1 = 8, m_2 = 2, n = 10$, then
  - Previous method: $\sim 10^9$
  - Our method: $\sim 10^{26}$

Back Up: AND-Tree Camouflaging

- To camouflage the inserted AND-Tree
  - Functional camouflaging with BUF/INV cell
  - Structural camouflaging to hinder removal attack