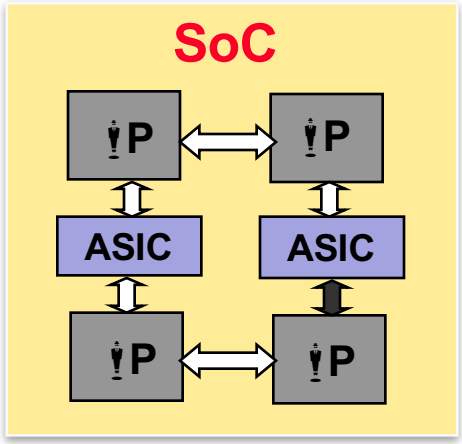
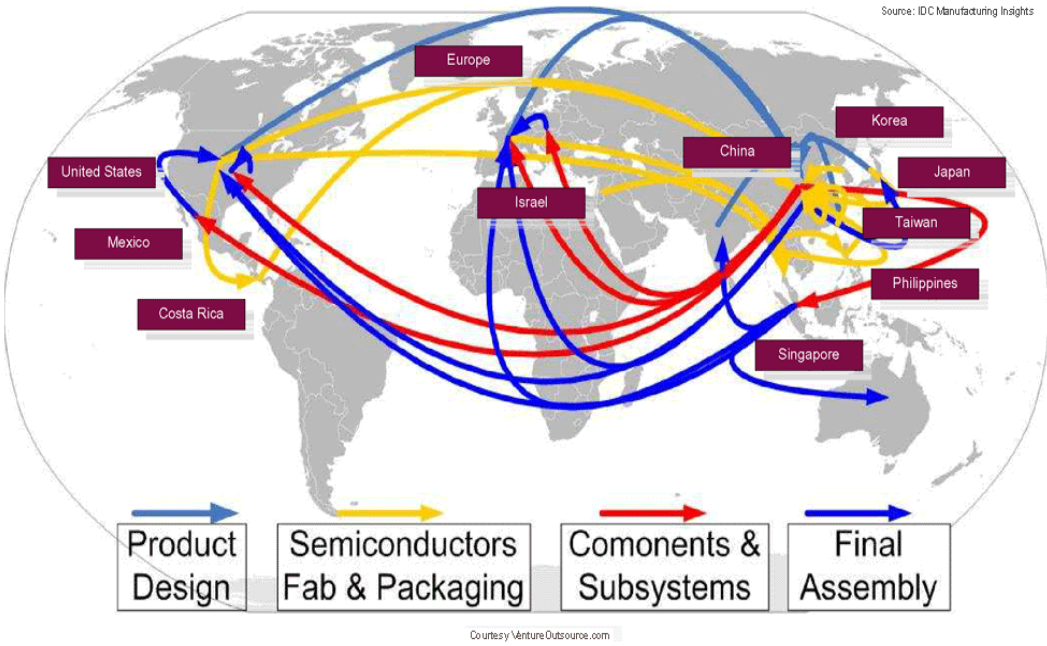
Outline

- Introduction
- Logic locking / logic encryption
- SAT attack against logic encryption
- SARLock
- SARLock + SLE
- Experiments and results
- Conclusion

Globalization of IC supply chain



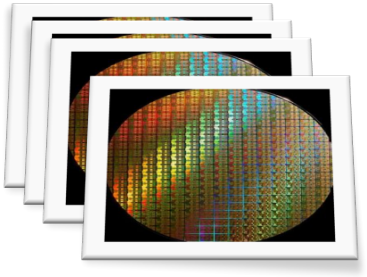


- Economic concerns
- Time-to-market
- Design complexity

Security vulnerabilities

IC Supply-chain Security

Real Fake



Impact

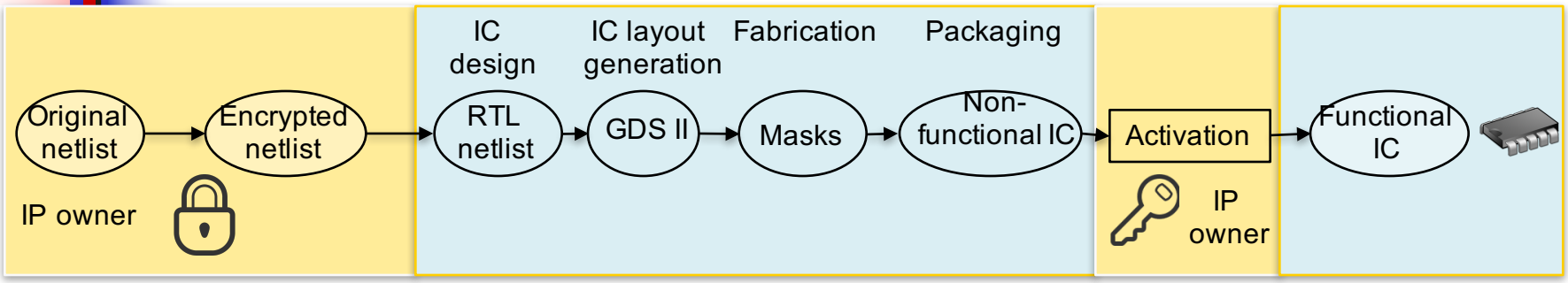
- Loss of revenue ~$4 billion annually
- Loss of trust
- Unreliable consumer electronics

Reverse Engineering

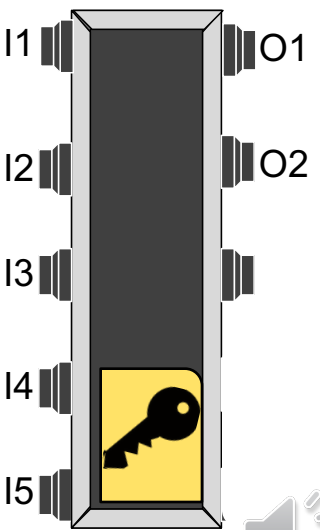
Hardware Trojans

SEMI, "Innovation is at risk as semiconductor equipment and materials industry loses up to $4 billion annually due to IP infringement," www.semi.org/en/Press/P043775, 2008.
Karri, Ramesh, et al. "Trustworthy hardware: Identifying and classifying hardware trojans." Computer 10 (2010): 39-46.
Tehranipoor, Mark Mohammad, Ujjwal Guin, and Domenic Forte. Counterfeit Integrated Circuits: Detection and Avoidance. Springer, 2015.
Torrance, Randy, and Dick James. "The state-of-the-art in IC reverse engineering." CHES 2009

Logic Encryption

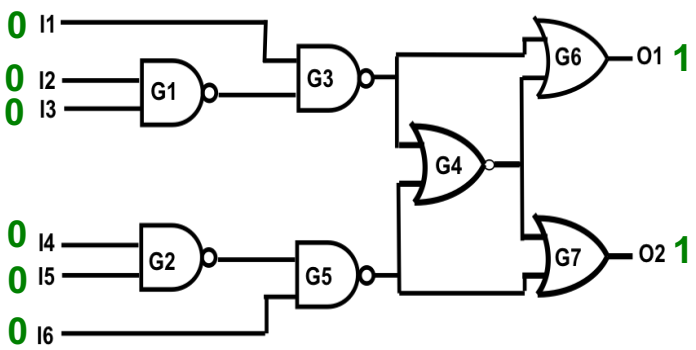


- Design-for-trust solutions
  - Watermarking
  - Fingerprinting
  - IC metering
  - Logic encryption
- Logic encryption
  - IP owner encrypts the netlist
  - IC is activated by loading the correct key

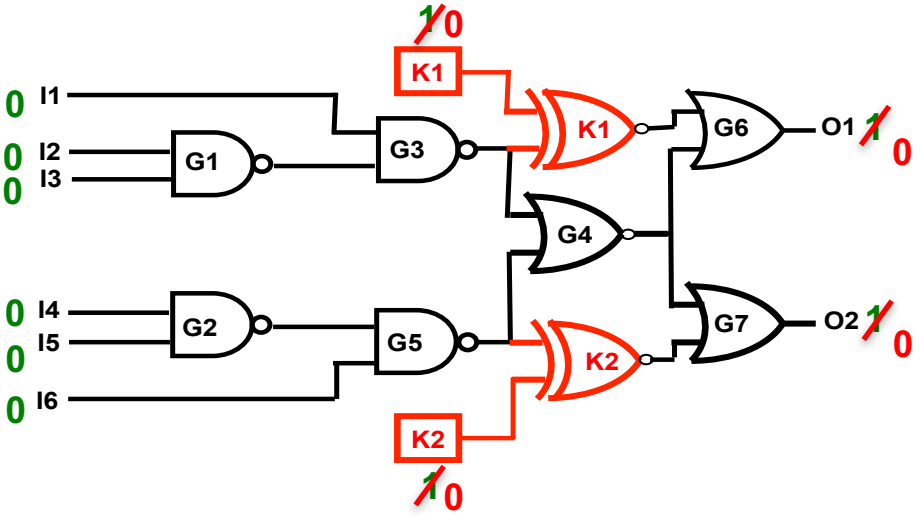


Roy, Jarrod A. et al.,"EPIC: Ending Piracy of Integrated Circuits," DATE, 2008.
Koushanfar, Farinaz. "Hardware metering: A survey." Introduction to Hardware Security and Trust. Springer New York, 2012. 103-122.
Liu, Bao, and Gang Qu. "VLSI supply chain security risks and mitigation techniques: A survey."VLSI Integration, 2016.
Chang, Chip-Hong, Miodrag Potkonjak, and Li Zhang. "Hardware IP Watermarking and Fingerprinting." Secure System Design and Trustable Computing. Springer, 2016.

Logic Encryption



Original netlist



Encrypted netlist

- Adds key gates (e.g., XOR, XNOR, MUXes)
- Correct key → Correct outputs
- Incorrect key → Incorrect outputs

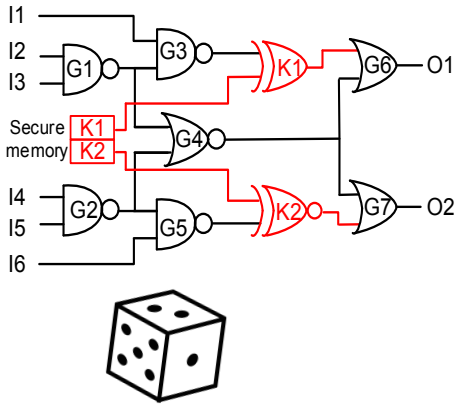
Roy, Jarrod A. et al.,"EPIC: Ending Piracy of Integrated Circuits," DATE, 2008.

Logic Encryption Techniques

Random (RLE)

Falsely claim defect-free ICs to be defective

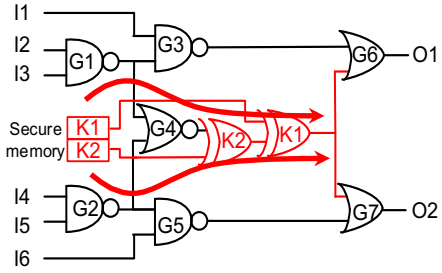
Key-gates uniformly distributed in the netlist



Fault analysis (FLE)

Key-gates at the most "influential" locations in the netlist

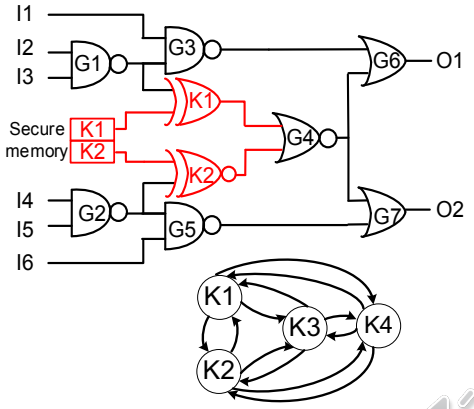
Key-gates tend to be localized and mostly back-to-back



Strong (SLE)

Key-gates to hamper sensitization of individual key bits

Key-gates localized in the netlist



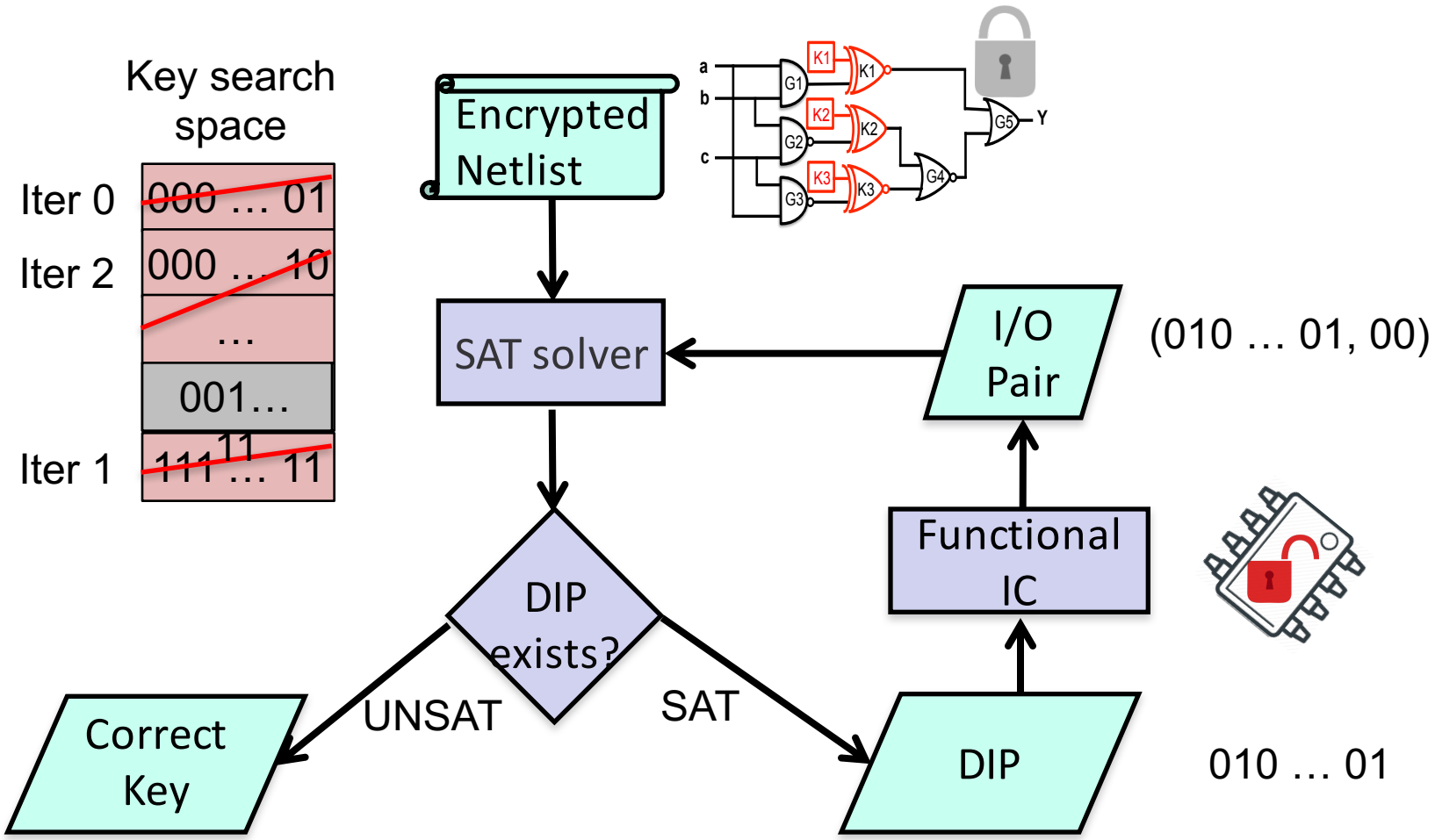
Roy, Jarrod A. et al. ,"EPIC: Ending Piracy of Integrated Circuits," DATE, 2008.
Rajendran, Jeyavijayan et al. ,"Fault-Anallysis based Logic Encryption", TCOMP 2015
Rajendran, Jeyavijayan et al. ,"Security Analysis of Logic Obfuscation", DAC 2012
Yasin, Muhammad et al. "On Improving the Security of Logic Locking.", TCAD 2015

Attacks against Logic Encryption

Sensitization attack [DAC'12]	Test-data mining attack [DATE'16]	SAT attack [HOST'15]
Attacker's capabilities Encrypted netlist Functional IC	**Attacker's capabilities** Encrypted netlist Test data	**Attacker's capabilities** Encrypted netlist Functional IC
Attack method Sensitize individual key bits to primary outputs	**Attack method** Use test data to extract the secret key for pre-test activation	**Attack method** Eliminate incorrect keys using "distinguishing input patterns"
Defense Strong logic encryption	**Defense** Post-test activation	**Defense ?** Our work

Rajendran, Jeyavijayan et al. "Security analysis of logic obfuscation", DAC 2012
Yasin et al. "Activation of Logic Encrypted Chips: Pre-Test or Post-Test?.", DATE 2016
Subramanyan et al. "Evaluating the Security of Logic Encryption Algorithms", HOST 2015
Plaza, Stephen M., and Igor L. Markov. "Protecting integrated circuits from piracy with test-aware logic locking." ICCAD 2014.

SAT Attack: Overview



Subramanyan, Pramod et al. "Evaluating the Security of Logic Encryption Algorithms." HOST 2015

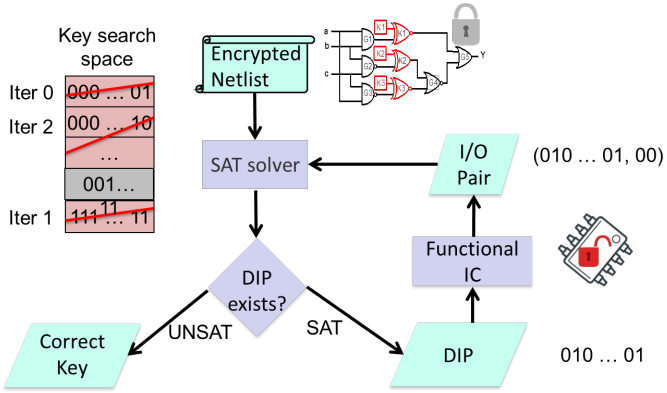
SAT Attack: Distinguishing Ability

					Output Y for different key values								
No.	a	b	c	Y	k0	k1	k2	k3	k4	k5	k6	k7	Pruned key values
0	0	0	0	0	1	1	1	1	1	1	0	1	
1	0	0	1	0	1	1	1	1	1	1	0	1	
2	0	1	0	0	1	1	1	1	1	1	0	1	
3	0	1	1	1	1	1	1	1	0	1	1	1	Iteration 1: k4
4	1	0	0	0	1	1	1	1	1	1	0	1	Iteration 4: all incorrect
5	1	0	1	1	1	1	1	1	1	1	1	0	Iteration 3: k7
6	1	1	0	1	1	1	0	1	1	1	1	1	
7	1	1	1	1	1	0	1	1	1	1	1	1	Iteration 2: k1

- Each DIP eliminates different number of keys
- Number of key combinations pruned out ↑
  - Number of iterations ↓

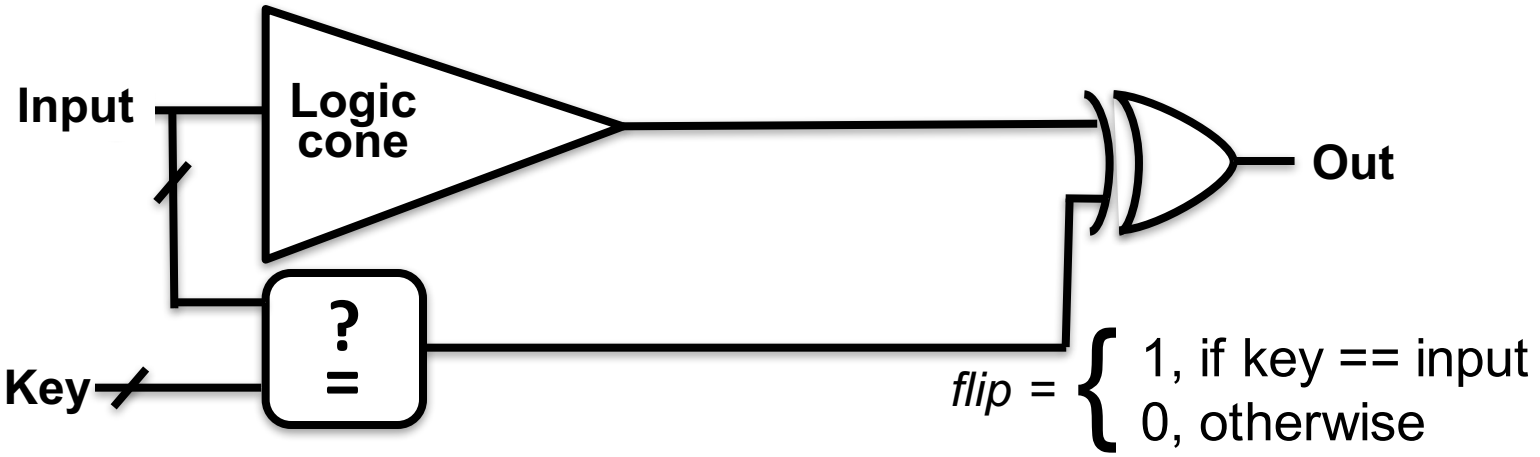
Resisting SAT Attacks – Key Idea

					Output Y for different key values							
No.	**a**	**b**	**c**	**Y**	**k0**	**k1**	**k2**	**k3**	**k4**	**k5**	**k6**	**k7**
0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	1	0	0	0	0
3	0	1	1	1	1	1	0	1	1	1	1	1
4	1	0	0	0	0	0	0	0	0	1	0	0
5	1	0	1	1	1	1	1	1	0	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	0



- Desired: Each DIP eliminates one key value
- Number of DIPs = Number of input combinations

SARLock: SAT-attack Resistant Logic Encryption

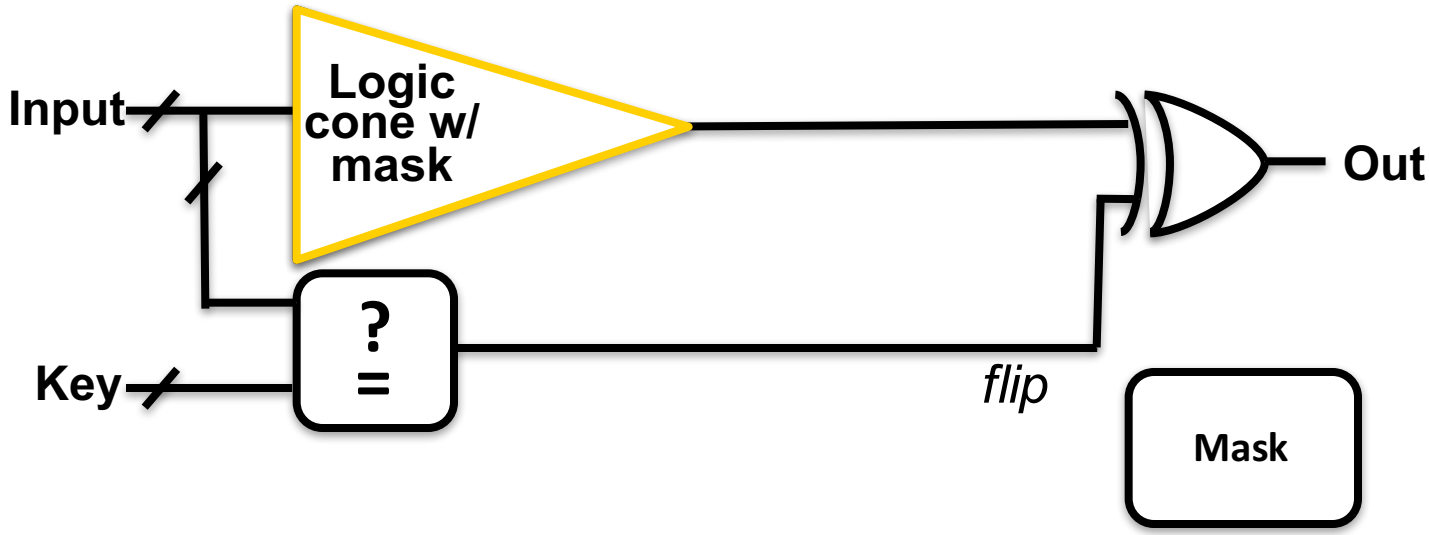


- Flip is asserted for
  - Correct key value
  - Diagonal entries of table

Number of DIPs = 2^K -1

					Output Y for different key values							
No.	a	b	c	Y	k0	k1	k2	k3	k4	k5	k6	k7
0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	0	0	0
2	0	1	0	0	0	0	1	0	0	0	0	0
3	0	1	1	1	1	1	1	0	1	1	1	1
4	1	0	0	0	0	0	0	0	1	0	0	0
5	1	0	1	1	1	1	1	1	1	0	1	1
6	1	1	0	1	1	1	1	1	1	1	0	1
7	1	1	1	1	1	1	1	1	1	1	1	0

SARLock: SAT-attack Resistant Logic Encryption



flip = 0 for correct key value

- Mask logic
  - Can be incorporated into logic cone
  - Ensures correct circuit operation

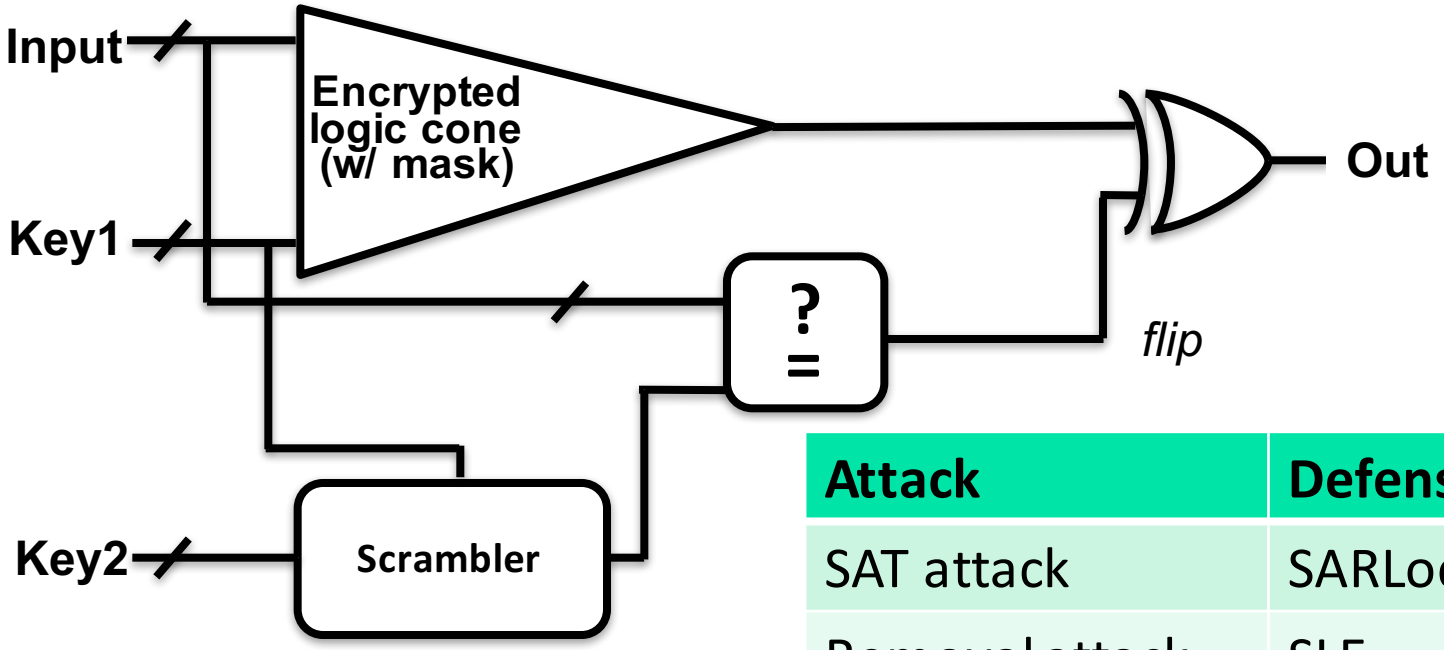
Is SARLock alone Sufficient?

Feature	SARLock
SAT attack resistant?	✓
Removal attack resistant?	✗
Sensitization attack resistant?	?

Integration with SLE to:

- Leverage the strengths of SLE /other logic encryption
- Thwart sensitization and removal attacks

Two-layer Logic Encryption: SARLock + SLE

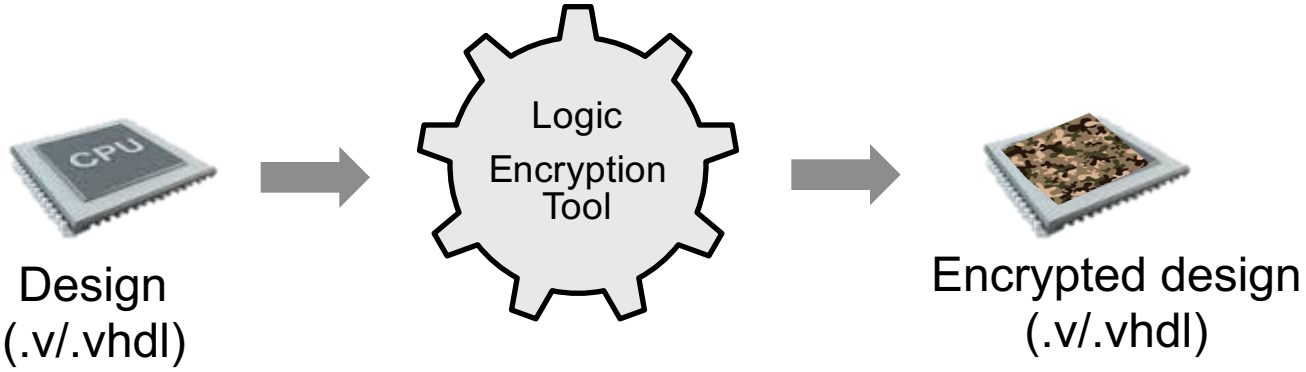


Attack	Defense
SAT attack	SARLock
Removal attack	SLE
Both	SARLock+SLE

- Scrambler
  - Creates key dependencies to thwart removal attack
  - E.g., Hash functions

Experimental Results

- ISCAS'85 benchmark circuits
- OpenSPARC microprocessor controllers
- Lingeling SAT solver
- 6-core Intel Xeon W3690 CPU
  - Running at 3.47GHz with 24 GB RAM



Biere, Armin. "Lingeling, plingeling, picosat and precosat at SAT race 2010." FMV Report Series Technical Report 10.1, 2010
Rajendran, Jeyavijayan et al. ,"Security Analysis of Logic Obfuscation", DAC 2012
"OpenSPARC T1 Processor," http://www.oracle.com/technetwork/systems/ opensparc/opensparc-t1-page-1444609.html
Hansen, Mark C. et al. "Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering." IEEE Design & Test of Computers ,1999

Experimental Results

- SARLock+SLE
  - Number of DIPs = $2^{|K|}$
  - SARLock **exponentially** increases number of DIPs

Design	#DIPs				Execution Time (s)			
	10	11	12	13	10	11	12	13
s5378	1024	2048	4096	8191	54.1	190.6	619.7	4351.8
c5315	1024	2049	4096	8191	75.4	252.9	829.1	4778.2
c7552	1025	2049	4096	8191	78.3	234.1	757	3165.3
s9234	1027	2049	4102	8195	77.2	247.9	864.1	3225.7
IFU	1023	2056	4100	8206	55.2	166.7	789.5	2309.8
LSUrw	1025	2049	4096	8194	58.2	152	626.9	1802.6
FPUin	1025	2049	4097	8194	28.4	135	1359.6	4497.6
LSUex	1024	2049	4096	8194	52.8	268.3	1137.2	3101.3

SARLock+SLE resists SAT attack

Overhead and Comparison

- SLE+SARLock vs. other techniques
  - Number of key gates: |K| = 64
  - Number of DIPs and execution time are extrapolated

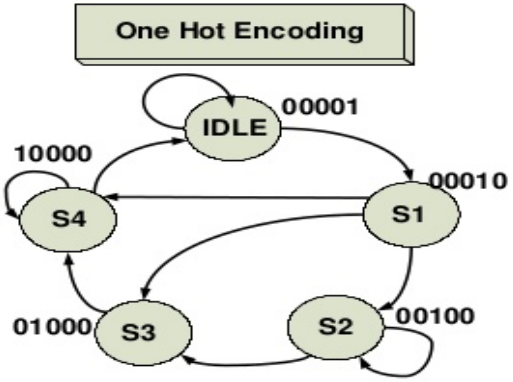
Metrics	RLE	SLE	SarLock+SLE
#DIPs	19	26	4.3E09
Attack time (s)	0.4	0.7	3.1E09
Area overhead (%)	29.6	32.2	35.2
Power overhead (%)	45.0	59.2	61.0
Delay overhead (%)	15.1	17.2	9.3

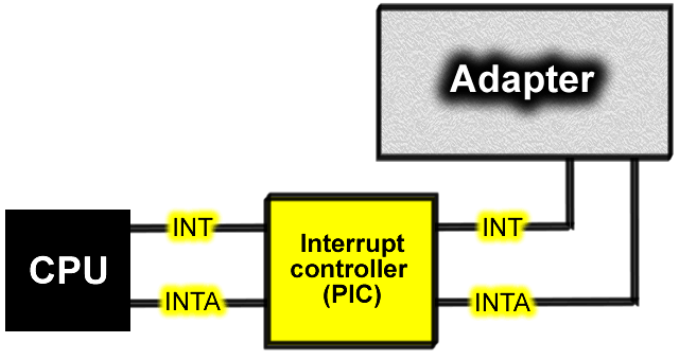
Attack effort increases exponentially SARLock+SLE

Roy, Jarrod A. et al.,"EPIC: Ending Piracy of Integrated Circuits," DATE, 2008.
Rajendran, Jeyavijayan et al. ,"Security Analysis of Logic Obfuscation", DAC 2012

Applications of SARLock+SLE

- Password checkers
- FSMs with one-hot encoding
- Interrupt controllers
- Network controllers





Koushanfar, Farinaz. "Hardware metering: A survey." Introduction to Hardware Security and Trust. Springer New York, 2012. 103-122.

Conclusion

- SARLock+SLE thwarts SAT attack
  - Limits distinguishing ability of DIPs
- SARLock+SLE thwarts
  - Reverse engineering and piracy
  - Removal and sensitization attacks

- As the key size increases
  - Attack effort increases **exponentially**
  - Overhead increases **linearly**
- Limitations/Assumptions
  - Logic synthesis performs “indistinguishable obfuscation”
  - Hard to differentiate design, mask, and scrambler logic

Thank you!
Questions

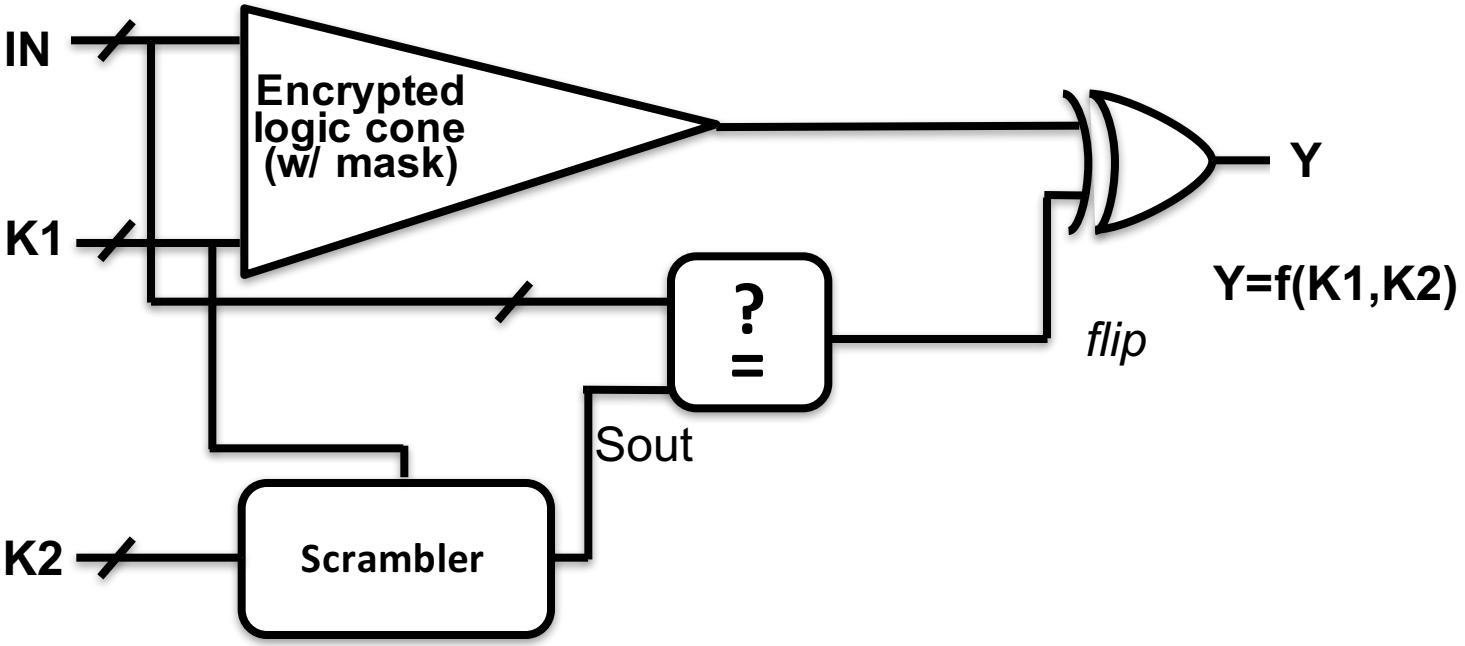
Experimental Results

- Strong logic encryption (SLE)
  - Can be broken using a small number of DIPs

Design	#DIPs					Execution Time (s)				
	10	11	12	13	14	10	11	12	13	14
s5378	8	9	9	10	13	0.2	0.2	0.2	0.2	0.2
c5315	4	3	4	5	3	0.3	0.3	0.3	0.3	0.3
c7552	8	9	9	9	12	0.7	0.5	0.5	0.5	0.5
s9234	7	13	13	10	12	0.2	0.3	0.3	0.3	0.3
IFU	8	8	9	13	11	0.1	0.1	0.1	0.1	0.1
LSUrw	4	5	5	7	9	0.1	0.1	0.1	0.1	0.1
FPUin	6	7	8	5	9	0.1	0.1	0.1	0.1	0.1
LSUex	5	5	8	8	6	0.1	0.1	0.1	0.1	0.1

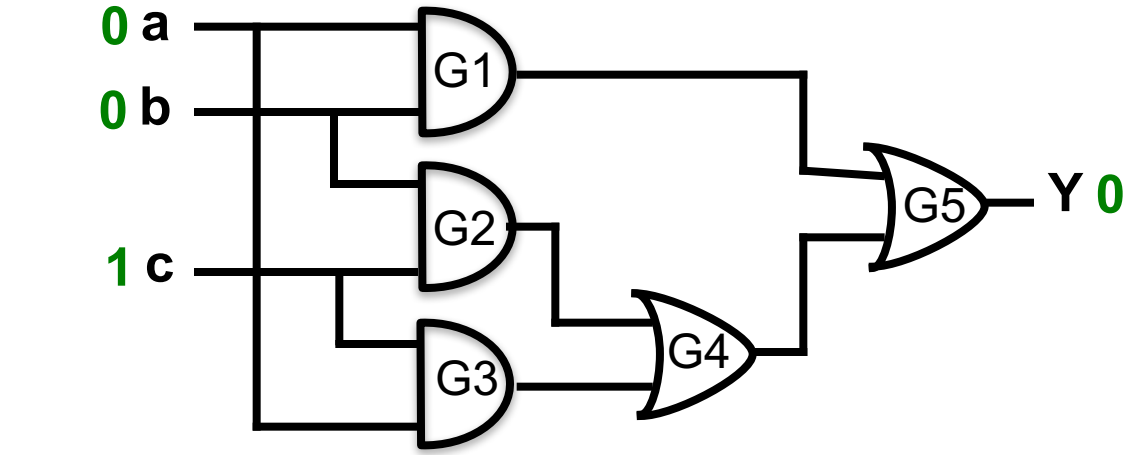
SLE is vulnerable to SAT attack!

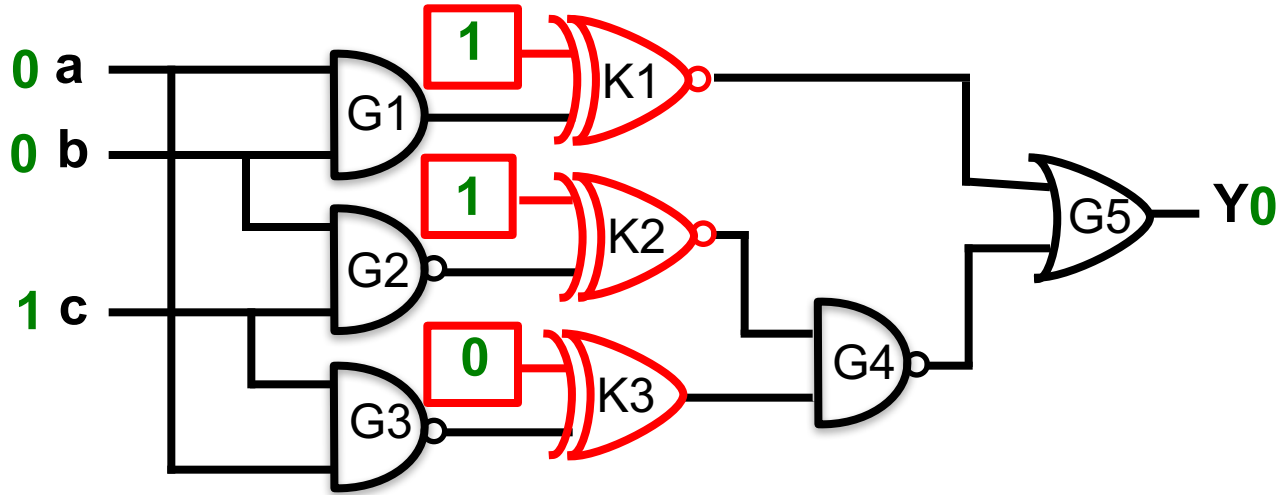
Thwarting Removal Attack



Attack	Thwarting Mechanism
SAT attack on K2 (exploit SLE vulnerability)	K1 must be known to attack K2
Removal attack on K1 (detach SARLock)	K2 must be known to attack K1

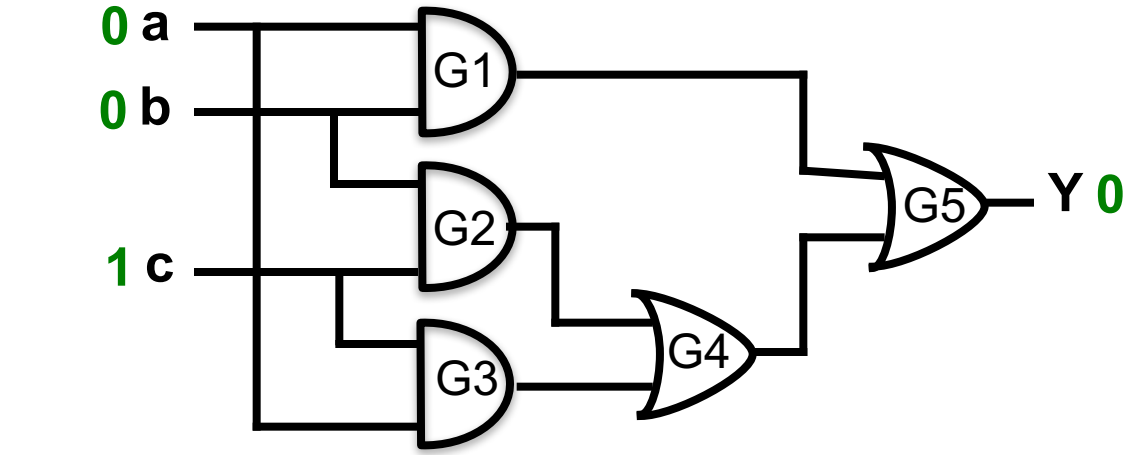
Logic locking

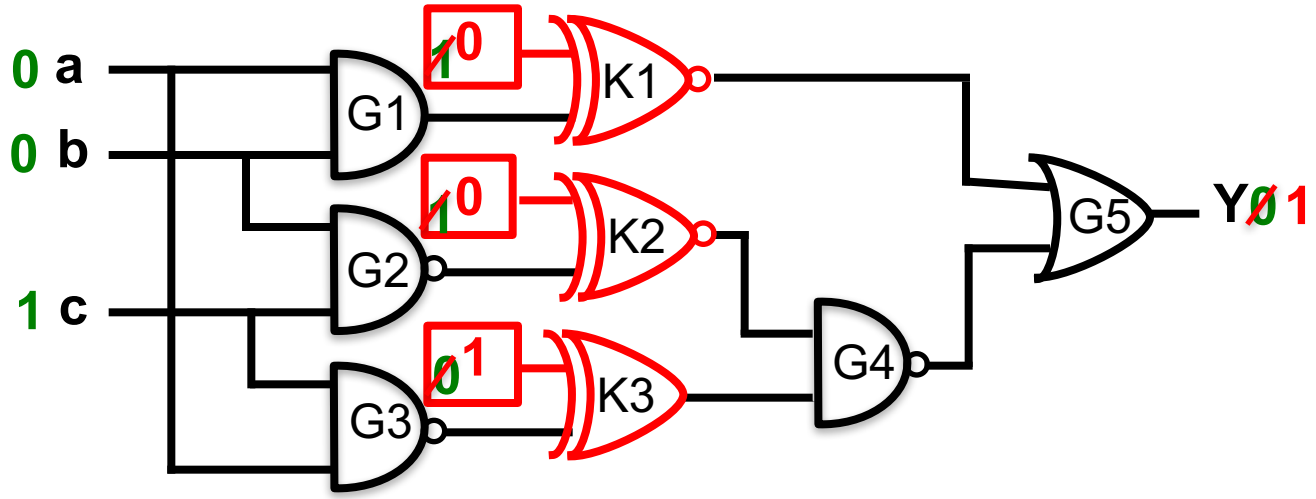




Correct key ➔ Correct output

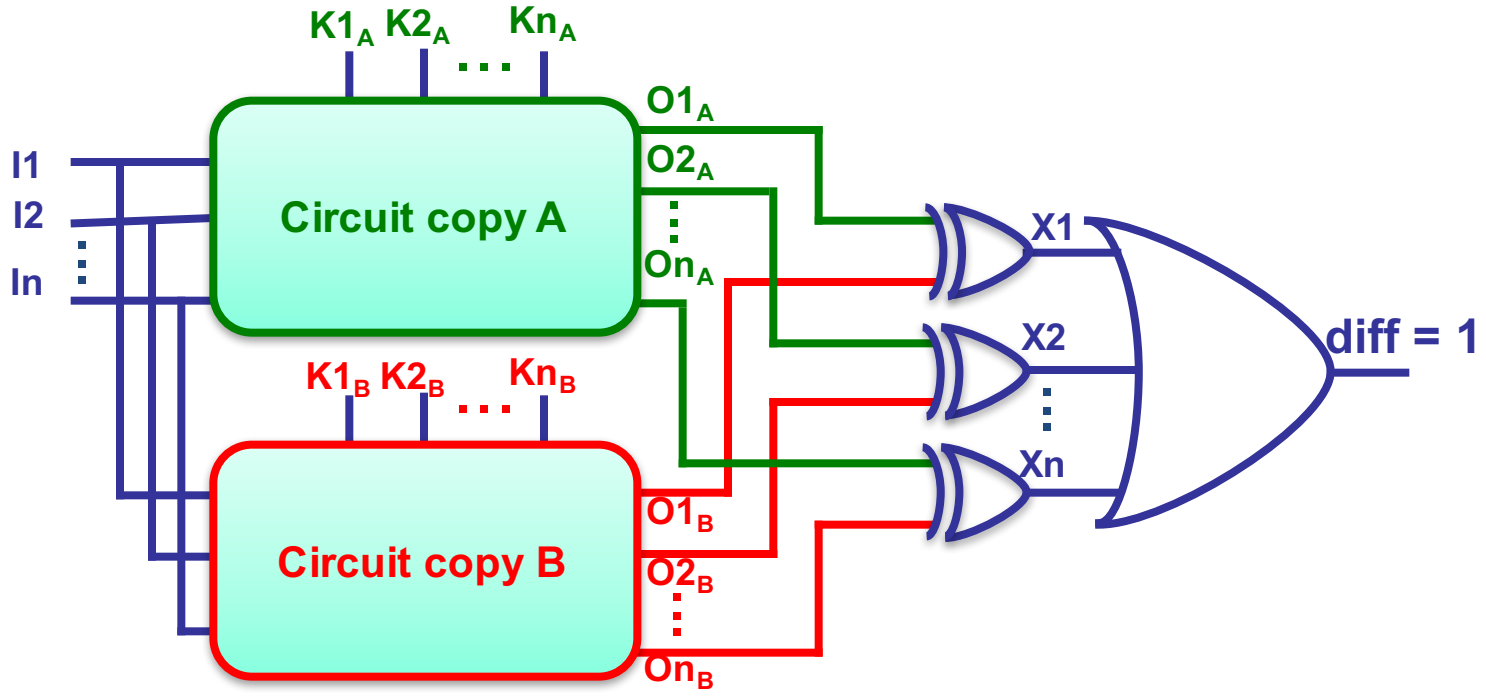
Logic locking





Incorrect key ➔ Incorrect output

SAT attack: Computing DIPs



- Diff = 1 ➔ output differs
  - At least one key value is incorrect

1. Subramanyan et al. "Evaluating the Security of Logic Encryption Algorithms." HOST 2015

Provably secure obfuscation

- One-point function
  - for each K_{incorr}, output differs only at one input w.r.t. K_{corr}, i.e., $F(IN, K_{corr}) \oplus F(IN, K_{corr} \oplus \alpha), \alpha \in \{0,1\}^n \setminus \{0^n\}$ is a one point function

- Point functions can be provable obfuscated[1]

$$F(IN, K_{corr}) \oplus F(IN, K_{corr} \oplus \alpha) = \begin{cases} 1, if\ r^{IN} = r^{K_{corr} \oplus \alpha} \\ 0, otherwise \end{cases}$$

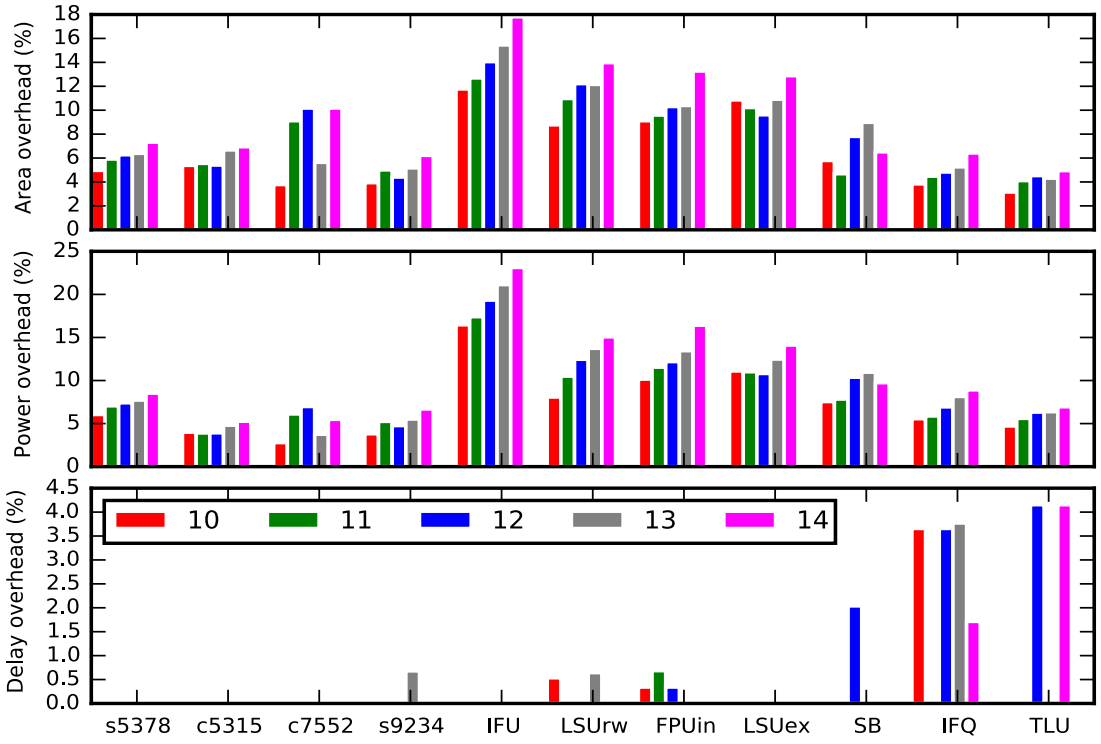
- High overhead[2]
  - 100K+ gates for 1024-bit exponentiation alone

1. R. Canetti, "Towards Realizing Random Oracles: Hash Functions That Hide All Partial Information," in Proc. Annual International Cryptology Conference, 1997, pp. 455–469.

2. G.D.Sutter,J.-P.Deschamps, and J.L.Imaňa,"Modular Multiplication and Exponentiation Architectures for Fast RSA Cryptosystem Based on Digit Serial Computation," IEEE Trans. Ind. Electron., vol. 58, no. 7, pp. 3101–3109, 2011.

Experimental results

- SARLock
  - Exponential security gain at linear increase in cost



Minimal delay overhead

Experimental results

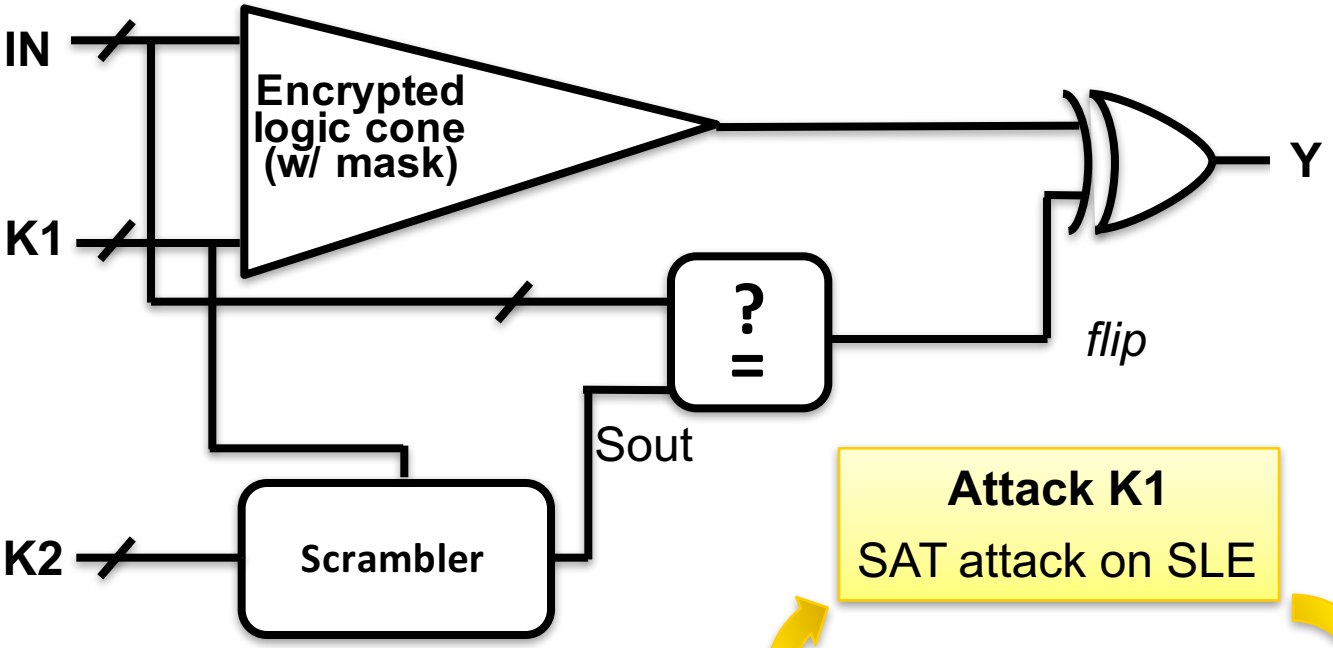
- SARLock
  - #DIPs ≈ $2^{|K|}$
  - Key size ↑ ➔ Execution time (3x-4x) ↑

Benchmark	#DIPs					Execution Time (s)				
	10	11	12	13	14	10	11	12	13	14
s5378	1023	2047	4095	8191	16383	62.6	177.9	996.1	2710.2	9374.6
c5315	1023	2047	4095	8191	16383	79.6	247.4	1188.1	3441.4	11122.5
c7552	1023	2047	4095	8191	16383	73.1	311.1	126.6	3561.3	11761.5
s9234	1023	2047	4095	8191	16383	77.7	279	1235.2	3491.2	12330.9
IFU	1023	2047	4095	8191	16383	32.5	104.3	589.6	2216.8	6650.8
LSUrw	1023	2047	4095	8191	16383	37.3	120.5	618	1762.8	6133
FPUin	1023	2047	4095	8191	16383	34.9	112.8	650	1898.6	6390
LSUex	1023	2047	4095	8191	16383	36	130.4	690.4	1941	7104.9
SB	1023	2047	4095	8191	16383	50.8	149.3	825.2	2412.9	7845.9
IFQ	1023	2047	4095	8191	16383	67.9	232.6	1135.9	2958.1	10521.1
TLU	1023	2047	4095	8191	16383	81.2	271.7	1198.7	3341.4	11628.5

SARLock resists SAT attack

e32r

Thwarting Removal Attack



Attack K1
SAT attack on SLE

Y=f(K1,K2)
K2 ?

SAT attack resistant

Attack K2
Removal attack

Sout=f(K1,K2)
K1 ?