Does logic locking work with EDA tools?

Zhaokun Han, Muhammad Yasin, Jeyavijayan “JV” Rajendran
{hzhk0618, myasin, jv.rajendran}@tamu.edu
Department of Electrical and Computer Engineering
Texas A&M University

Supply Chain Security of Hardware

Threat \ Countermeasure	Watermarking	Split manufacturing	IC camouflaging	Logic locking
IP piracy & Overbuilding	✓	✓	✓	✓
Reverse engineering	✗	✓	✓	✓
IC counterfeiting	✓	✗	✗	✓
Hardware Trojan	✗	✓	✗	✓

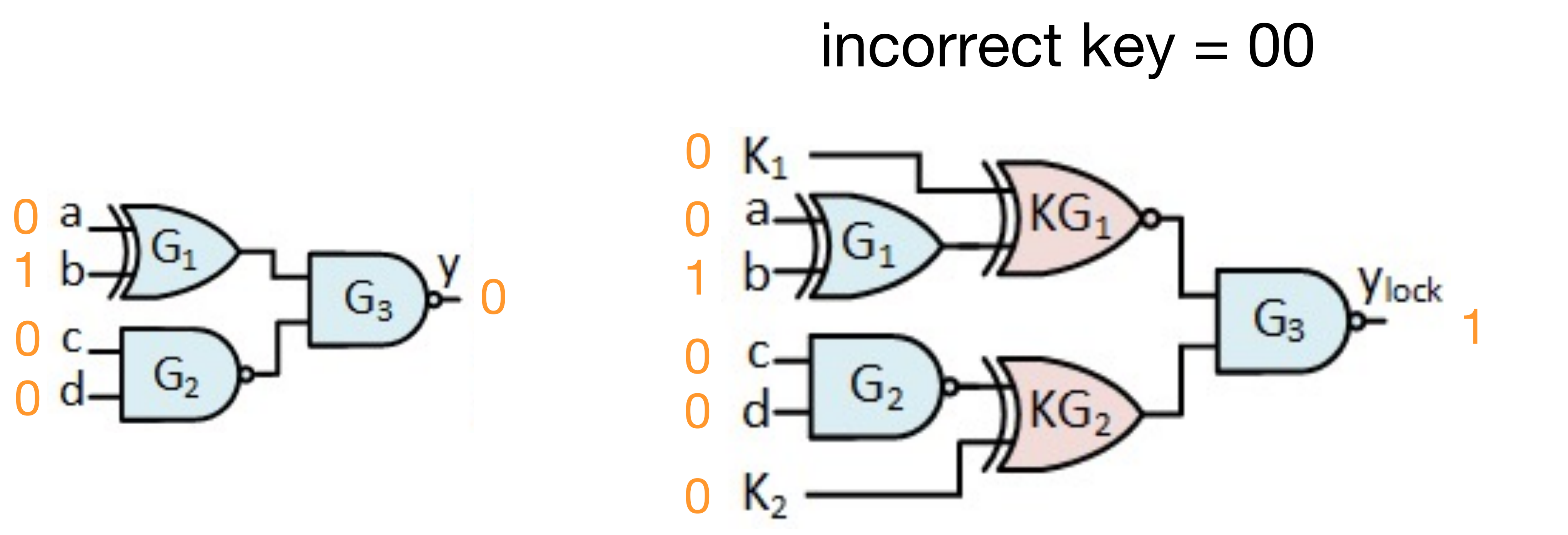
✓ denotes a successful defense, ✗ denotes an unsuccessful defense

Logic locking can defend against all these threats

Kahng et al., DAC'98; Imeson et al., USENIX'13; Roy et al., DATE'08.

Logic Locking

- Modifying logic and adding key inputs
- Only with correct key, the output is correct
- Secret key is stored in tamper-proof memory



Original circuit

Locked circuit, with correct key=10

Original circuit

Circuit protected by Logic Locking

Existing Logic Locking Techniques

Attack / Defense	Query-based attack			Structural attack		
	Sensitization	SAT, SMT	AppSAT, 2-DIP	SPS, ATR	FALL	SAIL
XOR-based (random, strong, fault-based LUT-based)	✗	✗	✗	✓	✓	✗
Point-function (AND-tree, SARLock, Anti-SAT)	✓	✓	✓	✗	✓	✓
CAC-HD, CAC-flex	✓	✓	✓	✓	✗	✓
CAC-rem	✓	✓	✓	✓	✓	✓

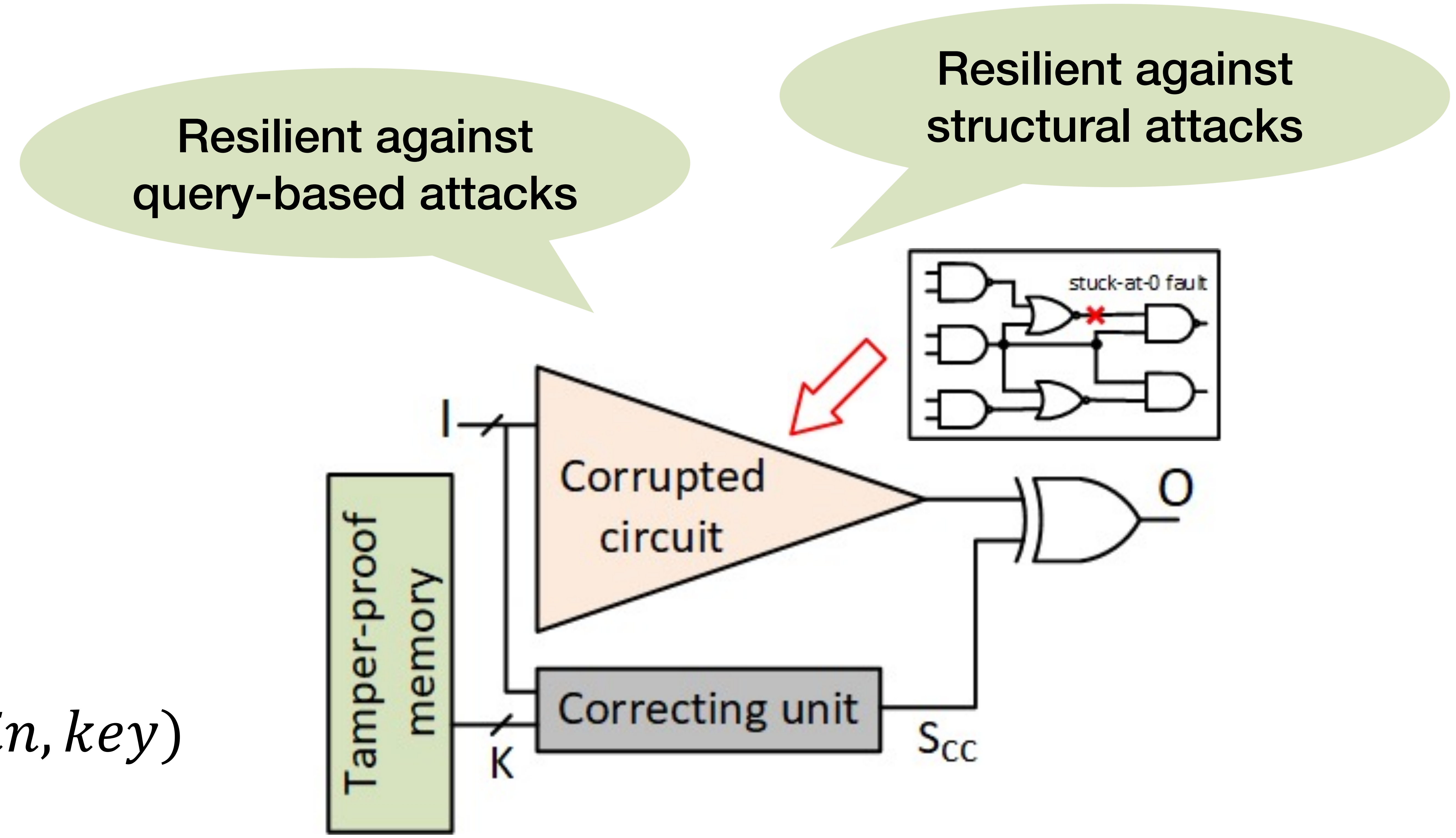
✓ denotes a successful defense, ✗ denotes an unsuccessful defense

- Point-function and CAC techniques are resilient to query-based attacks.
- CAC-rem is the only technique that defends against all existing attacks.

Rajendran et al., DAC'12; Subramanyan et al., HOST'15; Azar et al., TCHES'18; Shamsi et al., HOST'17; Shen et al., GLSVLSI'17; Yasin et al., ASP-DAC'16; Li et al., ICCAD'16; Sirone et al., DATE'19; Chakraborty et al., AsianHOST'18; Roy et al., DATE'08; Rajendran et al., DAC'12; Baumgarten et al., DTC'10; Li et al., ICCAD'16; Yasin et al., HOST'16; Xie et al., CHES'16; Yasin et al., CCS'17; Sengupta et al., TCAD'20.

Corrupted and Corrected (CAC)

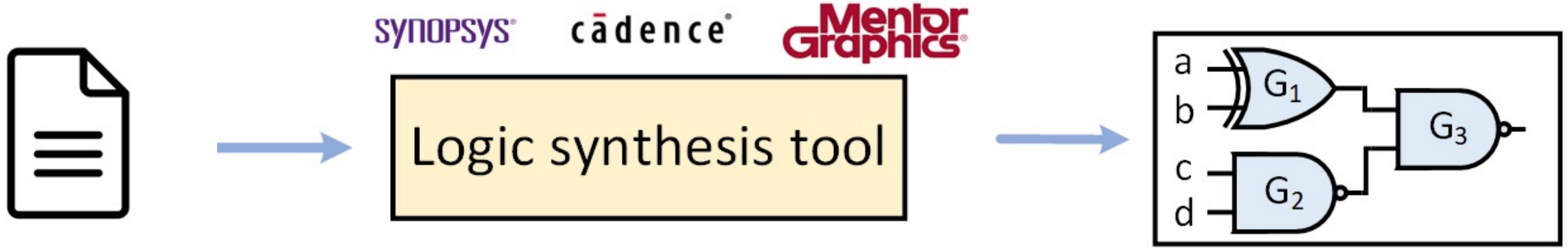
- Aka stripped-functionality logic locking
- Components in CAC-locked circuit
  - Corrupted circuit
    - ❑ Protected input pattern (PIP)
    - ❑ $in \notin PIP \Leftrightarrow f_{orig}(in) \neq f_{cp}(in)$
  - Correcting unit
    - ❑ $key = key_c \Leftrightarrow \forall in, f_{orig}(in) = f_{lock}(in, key)$
- CAC-rem is unbroken since 2020



[1]. Yasin et al., "Provably-secure logic locking: From theory to practice." CCS 2017.
[2]. Sengupta et al., "Truly stripping functionality for logic locking: A fault-based perspective." TCAD 2020.

Logic Synthesis v.s. Logic Locking

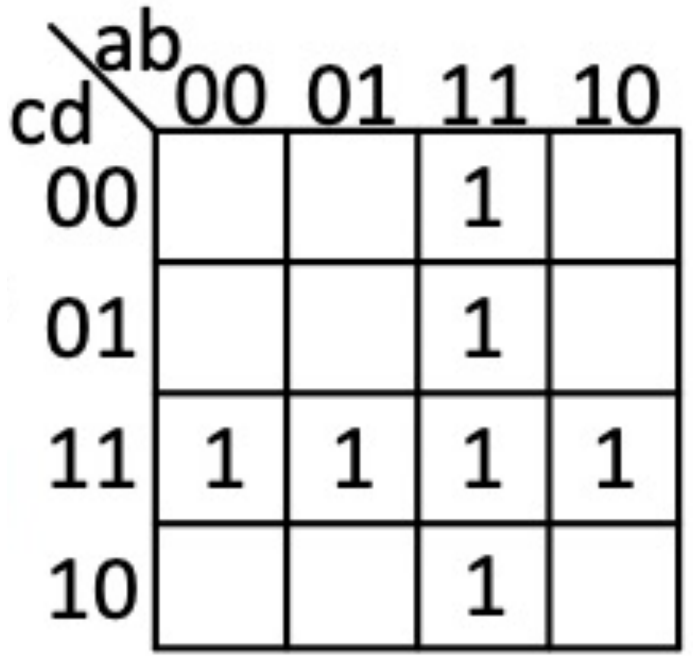
- Logic synthesis process



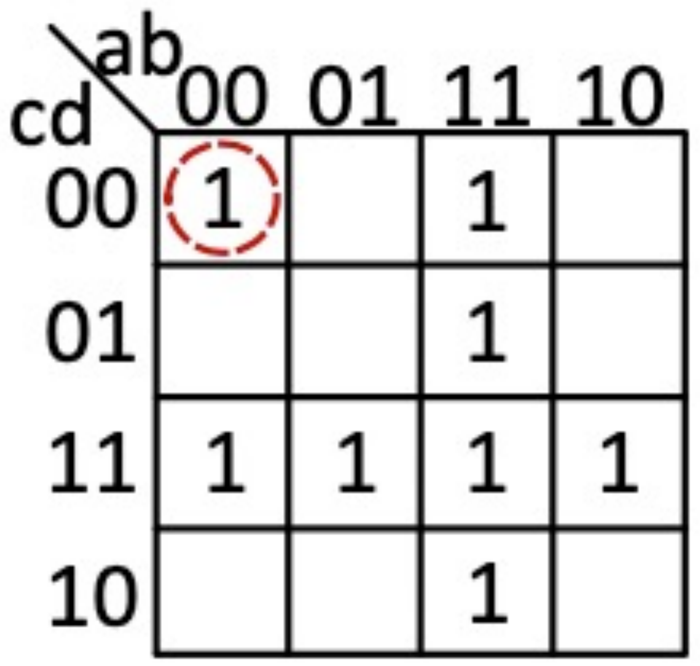
RTL design

Netlist

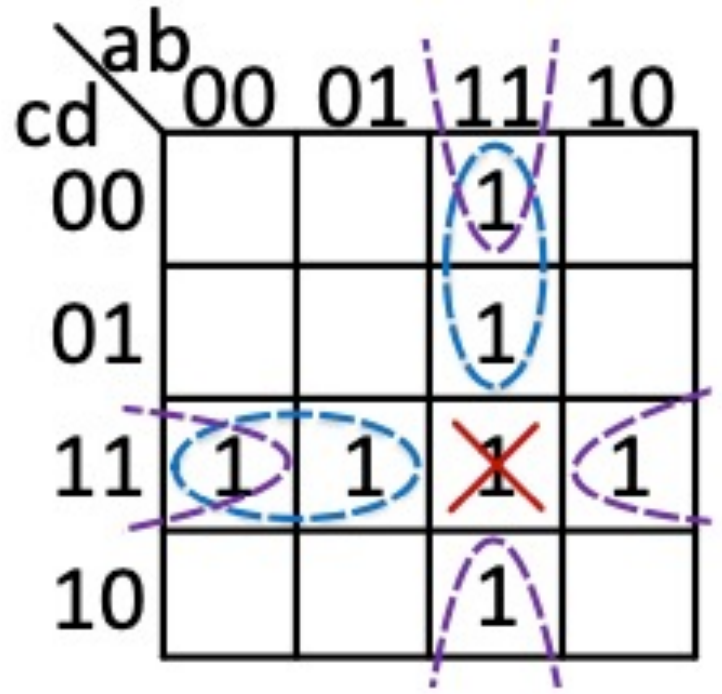
- Example of logic synthesis in logic locking



K-map of original circuit



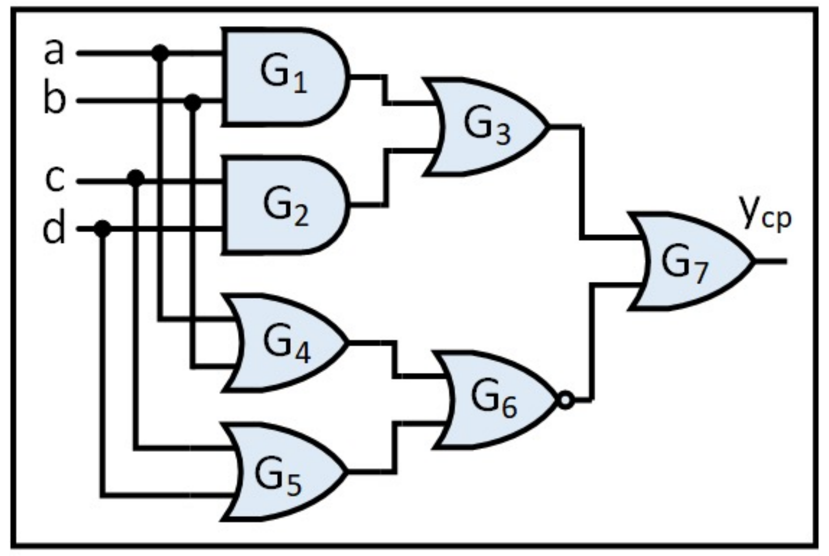
Adding a minterm, 0000



Removing a minterm, 1111

Sparse Prime Implicant (SPI) Attack

- Using prime implicant table (PIT) to search for PIPs
  - Implicant: A cube that only covers ON-set minterms
  - Prime implicant: The implicant cannot be covered by any other implicant
  - Sparse Prime implicant: PIs are "far" away in rest of PIs in PIT
- SPI attack process



Corrupted circuit

a b c d	y_{cp}
0 0 0 0	1
1 1 - -	1
- - 1 1	1

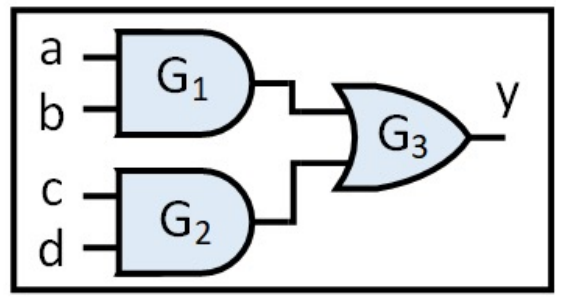
Corrupted circuit's PIT

a b c d
0 0 0 0

Verified PIP

a b c d	y
1 1 - -	1
- - 1 1	1

Original circuit's PIT



Original circuit

Results

Attack \ Circuit	Competition-small						Competition-large		
	b10	b11	b12	b13	b15	b17	b17L	b20	b22
SAT	✓	✓	✓	✓	✓	✗	✗	✗	✗
AppSAT	✗	✗	✗	✓	✓	✗	✗	✗	✗
ATR, SPS, FALL	✗	✗	✗	✗	✗	✗	✗	✗	✗
SPI (proposed)	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓ denotes a successful attack, ✗ denotes an unsuccessful attack

- CAC-rem locked circuit are from CSAW'19 logic locking competition
    - No one broke it during the competition
- SPI attack breaks all the locked circuits

More Details (in the paper)

- Scalability of SPI attacks
- What makes SPI attacks hard?
  - PIPs that are far away from PITs of corrupted circuit (aka D2PIPs)
- Conventional benchmark circuits have only few D2PIPs (<100)
  - They are not secure; should we even use them for logic locking research?

Future work

- Encode Boolean circuits such that # of D2PIPs are increased
- Tradeoff between security and overhead

Thank you!

Zhaokun Han

sites.google.com/a/tamu.edu/zhaokun-han/

hzhk0618@tamu.edu

TAMU Secure and Trustworthy Hardware (SETH) Lab: seth.engr.tamu.edu/