Synthesis for Security

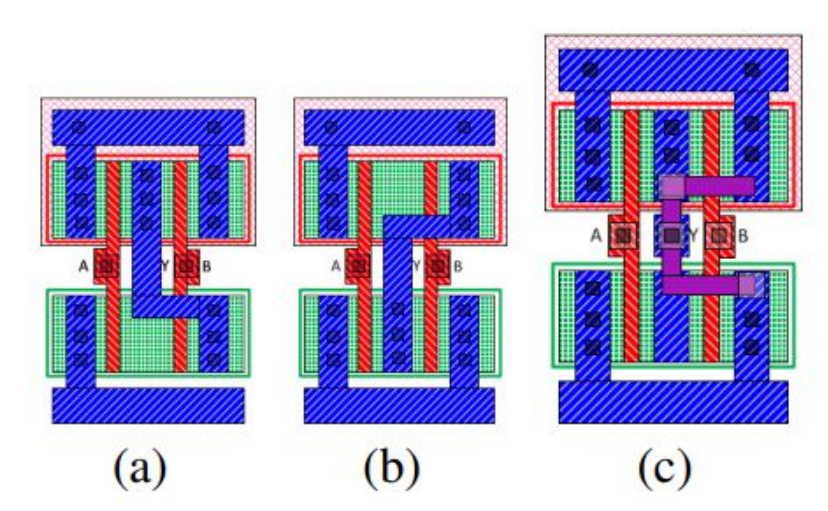
2023.12.26

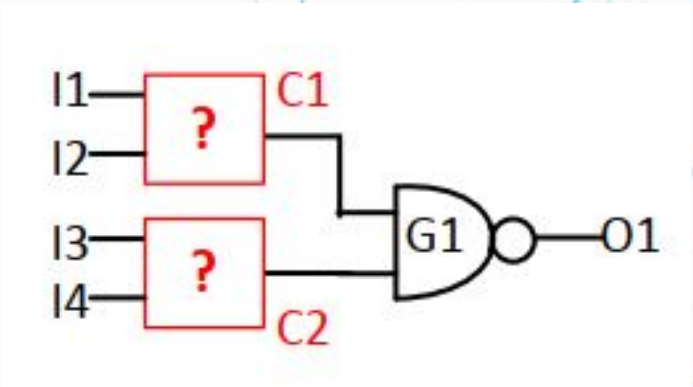
R12943102 **王濰紳**
R12943103 **張芯慈**

Outline

- IC camouflaging
- Logic Obfuscation
- Statistics

Layout-level Approach: IC Camouflaging





- Replace some gates with camouflaged gates
- Attacker cannot verify functionality from layout directly

Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. 2013. **Security analysis of integrated circuit camouflaging**. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13). Association for Computing Machinery, New York, NY, USA, 709–720.

IC Camouflaging – Threat Model

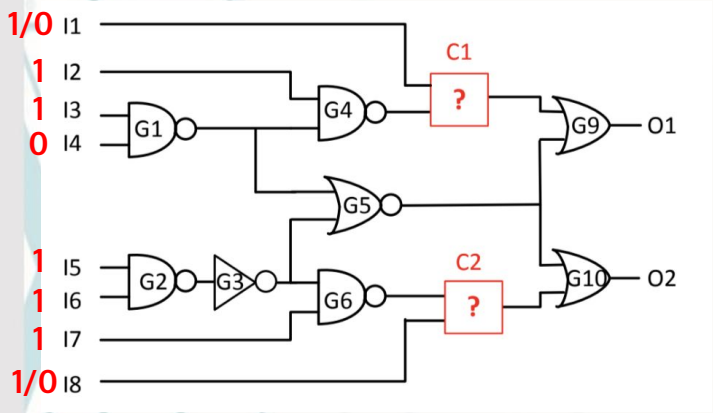
- The attacker can delayer the IC by optical microscope and image processing tool
- The attacker can differentiate regular/camouflaged standard cells
- The attacker knows the list of possible functions a camouflaged cell can implement. E.g. {XOR, NAND, NOR}

IC Camouflaging – Objective

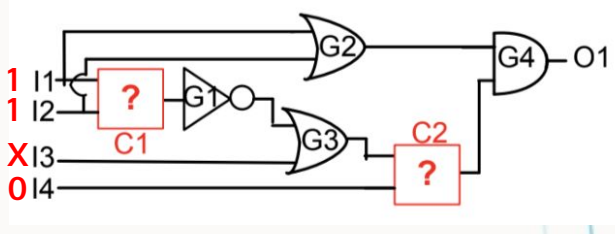
- Given the original circuit, choose a set of standard cells for camouflaging
- Goal: effectively increase the complexity of resolving the functionality of each camouflaged gate
- Design overhead should be considered

⇒ Given the original circuit, choose a certain percentage of standard cells for camouflaging

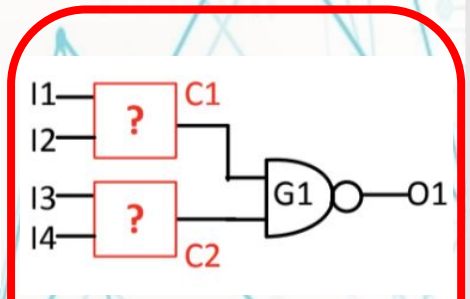
Different Status of Camouflaged Gates



Isolated



Partially resolvable



Non-resolvable

Easy to Attack

Hard to Attack

Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. 2013. **Security analysis of integrated circuit camouflaging**. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13). Association for Computing Machinery, New York, NY, USA, 709–720.

Output Corruptibility

$$\text{Output corruptibility} = \sum_{\#\ \text{I/P patterns}} (\#\ \text{O/Ps with ambiguity activated and propagated})$$

- Higher output corruptibility

  ⇒ More impact on incorrect function selection

Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. 2013. **Security analysis of integrated circuit camouflaging**. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13). Association for Computing Machinery, New York, NY, USA, 709–720.

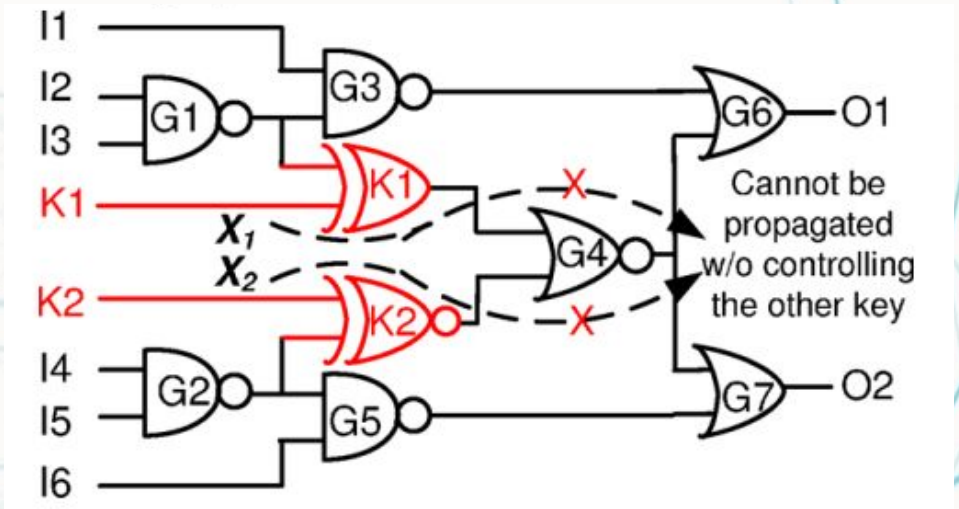
IC Camouflaging – Defender Strategy

- Assume desired number of camouflaged gates = N
- First, camouflage the gate with highest output corruptibility
- For i = 2 to N,
  - Identify all non-resolvable uncamouflaged gates based on the camouflaged gates as candidates
  - Among all candidates, select the one with highest output corruptibility

Gate-level Approach: Logic Obfuscation

- Introduction: Key-Gates insertion
- Key-Gates Structures
- Defend Algorithim

Gate-level Approach: Logic Obfuscation

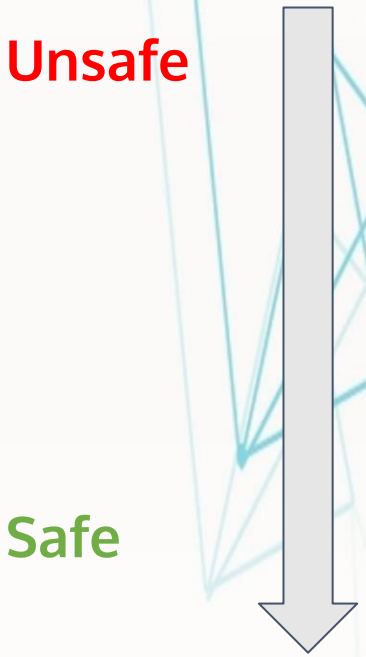


- Insert XOR/XNOR gates
- With incorrect keys, circuit behavior would differ from the original one

J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12

Key-Gates Structures

- Runs of key-gates
- Isolated key-gates
- Mutable key-gates
  - Dominating key-gates
  - Concurrent key-gates
  - Sequential key-gates
- Non-Mutable key-gates



Runs of key-gates

- Meaningless PPA overhead
- Single key-bit from attacker perspective

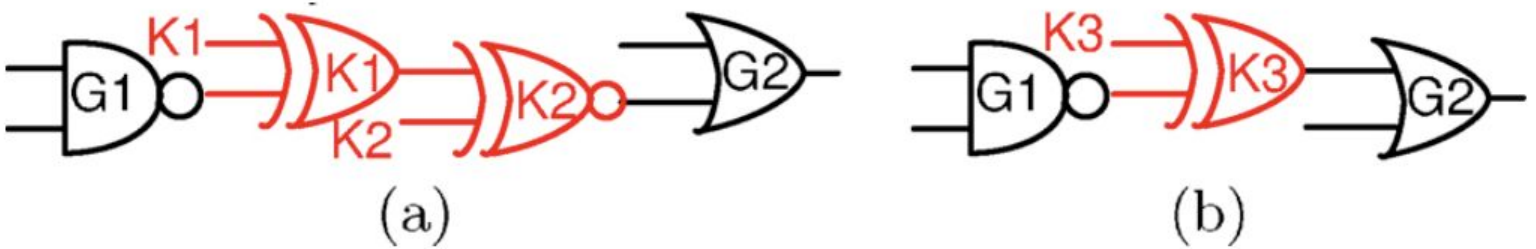
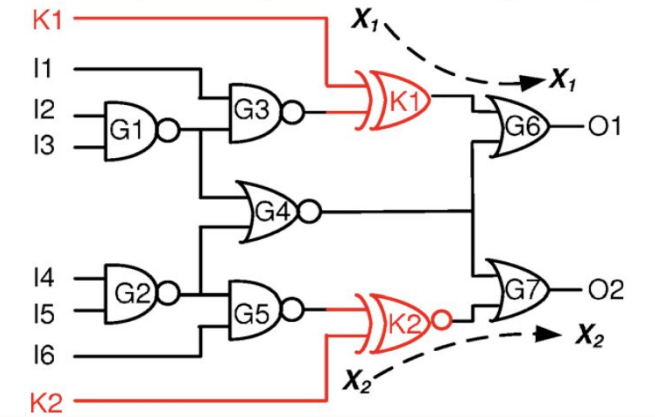


Figure 4: (a) A run of two key-gates K1 and K2. (b) K3 replaces K1 and K2.

J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12

Isolated key-gates

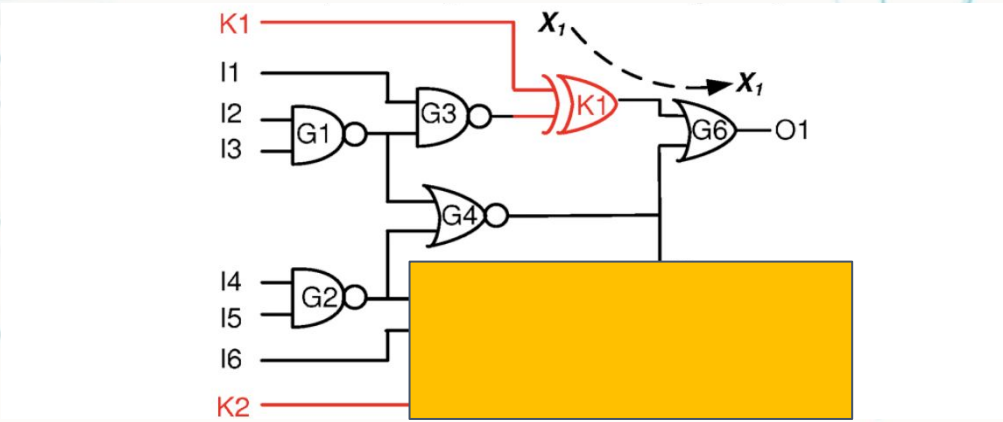
- No path to all other key-gates
- Can be solved as one bit key



J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12

Isolated key-gates

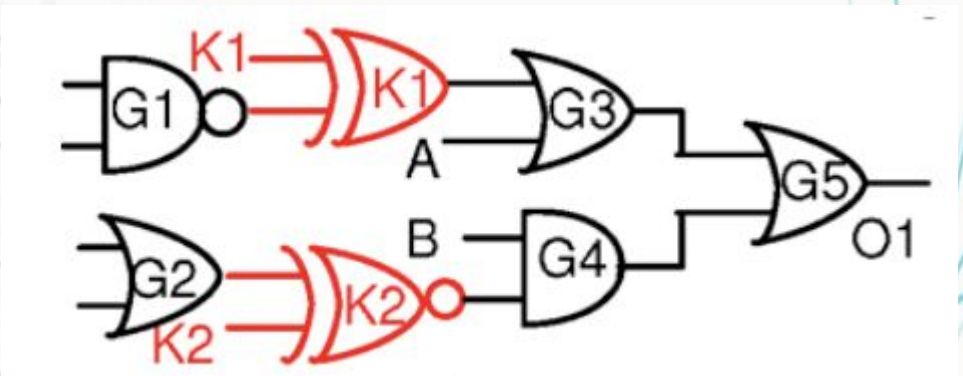
- No path to all the other key-gates
- Can be solved as one bit key



J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12

Mutable key-gates

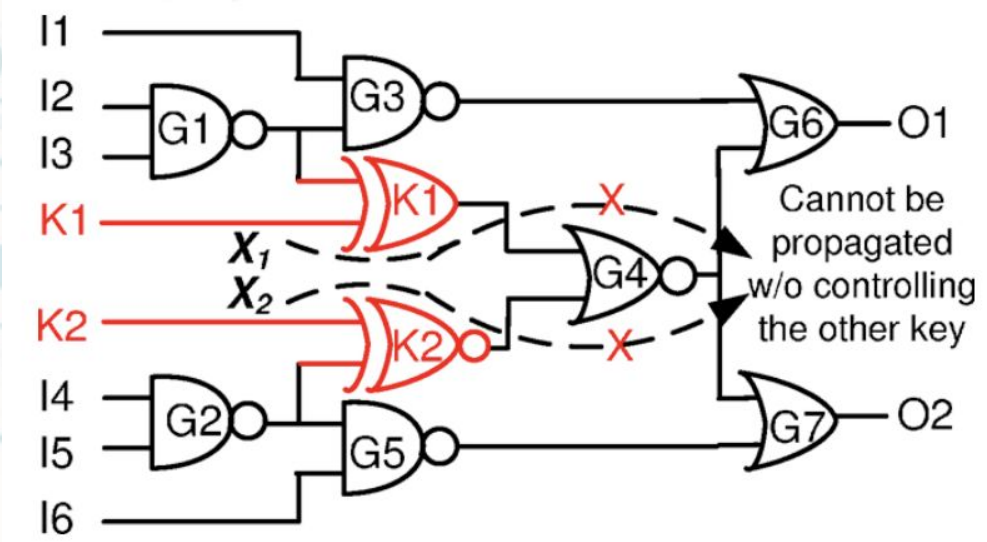
- A=1 or B=0 mute Key1 or Key2



J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12

Non-mutable key-gates

- Attacker must apply brute force

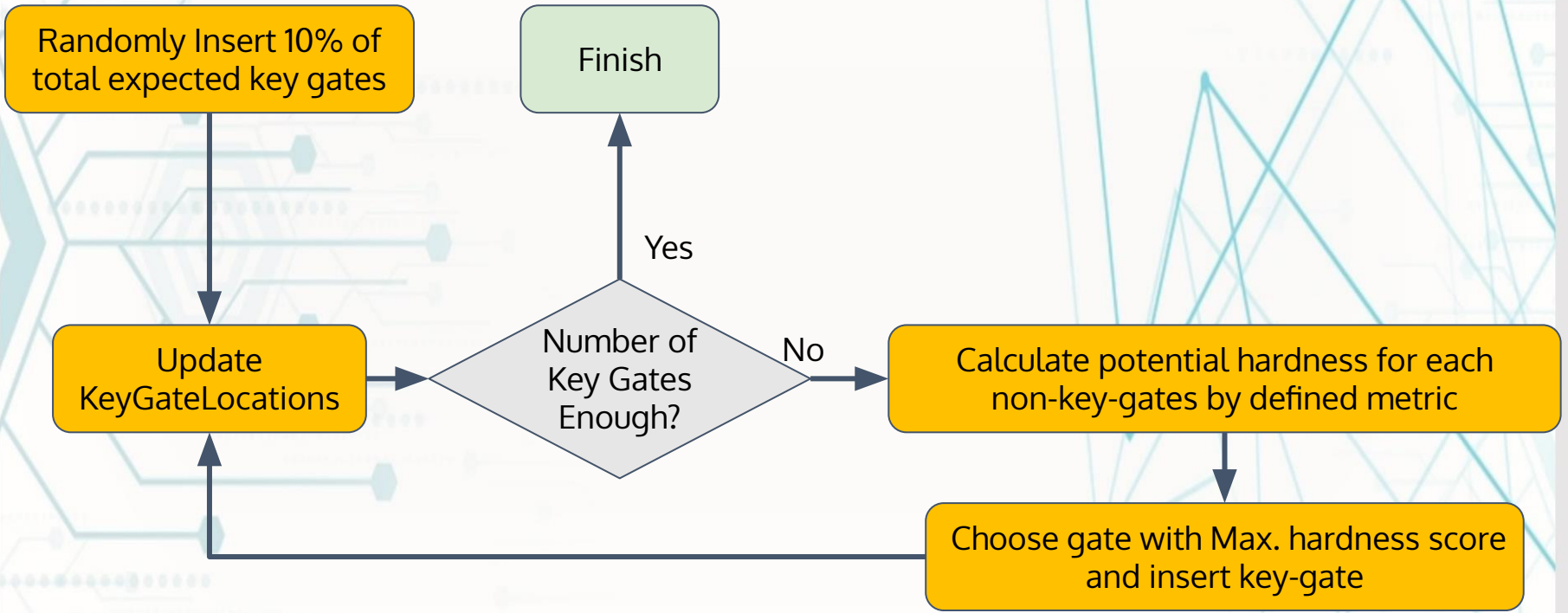


J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12

Defend Algorithm

- Assumption of attacker
  - Know the location of key gate
  - Able to brute-force all the PI-PO relations
  - NOT able to rewire or observe internal logic
- Strategy on key-gates
  - No run of key-gates
  - Isolated key-gates (metric = 0)
  - Mutable key-gates (metric = 1)
    - Dominating, Concurrent, Sequential
  - Non-mutable key-gates (metric = 2)

Defend Algorithm



Implementation Setup

- Defend Algorithm Metric used: isolated=0, non-isolated=1
- Area: abc gate count
- Delay: abc levels

Example: before obfuscation

```
module c17(N1, N2, N3, N6, N7, N22, N23);
    input N1, N2, N3, N6, N
    output N22, N2
    wire N1, N2, N3, N6, N7, N10, N16, N11, N19, N22, N23;
    assign N10 = ~( N1 & N3 );
    assign N16 = ~( N2 & N11 );
    assign N11 = ~( N3 & N6 );
    assign N19 = ~( N11 & N7 );
    assign N22 = ~( N10 & N16 );
    assign N23 = ~( N16 & N19 );
endmodule
```

Example: after obfuscation

```
module c17(N1, N2, N3, N6, N7, key_0, key_1, key_2, N22_key, N23_key);
    input N1, N2, N3, N6, N
    input7key_0, key_1, key_2;
    output N22_key, N23_key;
    wire N1, N2, N3, N6, N7, N10, N16, N11, N19, N22, N23;
    wire N23_key, N22_key, N1_ke
    wirey key_0, key_1, key_2;
    assign N10 = ~( N1_key & N3 );
    assign N16 = ~( N2 & N11 );
    assign N11 = ~( N3 & N6 );
    assign N19 = ~( N11 & N7 );
    assign N22 = ~( N10 & N16 );
    assign N23 = ~( N16 & N19 );
    assign N23_key = ~( N23 ^ key_0 );
    assign N22_key = ~( N22 ^ key_1 );
    assign N1_key = ~( N1 ^ key_2 );
endmodule
```

Implementation Statistics

	Original		Obfuscated		Overhead	
Circuit	Area	Delay	Area	Delay	Area	Delay
c880	325	25	397	31	18.14%	19.35%
c1355	494	25	584	33	15.41%	24.24%
c3540	1034	41	1097	50	5.74%	18.00%
c5315	1774	37	2146	38	17.33%	2.63%
c6288	2337	120	2445	122	4.42%	1.64%

Reference

[1] J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri, "Security analysis of logic obfuscation," DAC Design Automation Conference 2012, San Francisco, CA, USA, 2012, pp. 83-89

[2] Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. 2013. Security analysis of integrated circuit camouflaging. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13). Association for Computing Machinery, New York, NY, USA, 709–720.

[3] Jarrod A. Roy, Farinaz Koushanfar, and Igor L. Markov. 2008. EPIC: ending piracy of integrated circuits. In Proceedings of the conference on Design, automation and test in Europe (DATE '08). Association for Computing Machinery, New York, NY, USA, 1069–1074.

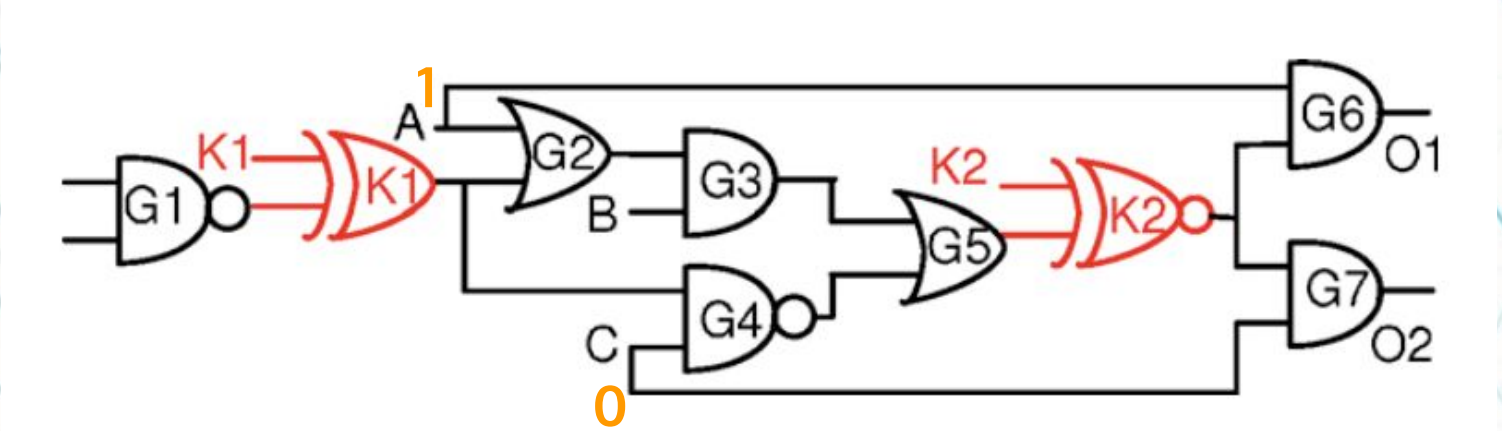
Thanks for your attention !

Why Synthesis for Security

- Prevent IP piracy
- Add ambiguity to increase the difficulty for reverse engineering

Dominating key-gates (mutable)

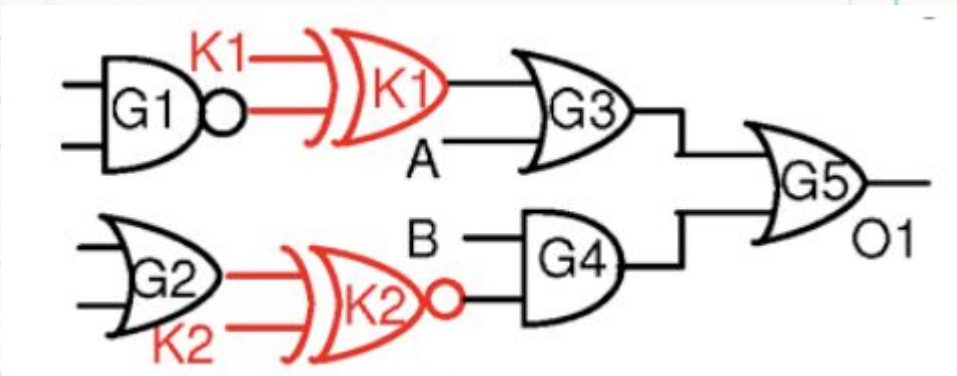
- Key2 dominates Key1
- **Mute** Key1 to solve Key2



J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12

Concurrent key-gates (mutable)

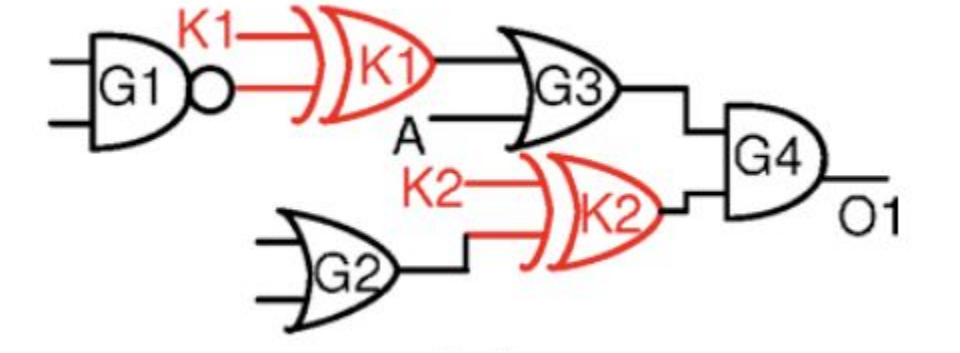
- A=1 or B=0 mute Key1 or Key2 (<=>)



J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12

Sequential key-gates (mutable)

- A=1, Mute key1 to solve key2 (=>)
- Key2 not mutable



J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri. **Security analysis of logic obfuscation.** DAC'12