DummyWire

Our first camouflage tool is derived from [DUM]. This technique obfuscates circuit by utilizing “two types of contacts/vias for interconnects: magnesium (Mg) that contacts/vias for the real interconnects in the original design of the circuit; and magnesium oxide (MgO) that contacts/vias for the dummy interconnects deliberately introduced for the purpose of obfuscation”. As far as logic level, the author introduces several extra wires to an input of a gate, so the complexity will rise exponentially with the number of camouflaged gates.

In our implementation, for a certain gate, we replace the original wire with a 4 inputs multiplexer, in which, one of the inputs for the multiplexer is the original wire while the other inputs are chosen from wires that belong to its previous level gates. (figure 1). And we randomly select gate to camouflage.

Obfuscate

Our next camouflage tool is inspired from [OBF]. It modifies standard library to an obfuscation gate called “Obfusgate” by using “Obfuscell” (figure 2). The so-called “Obfuscell” is either an inverter, a buffer, or it outputs a constant 1 or 0, and depending on how the Obfuscells are configured, the Obfusgate can have many different logic functions. Similar to DummyWire camouflage tool, the complexity will increase exponentially as the number of camouflaged gates.

In our implementation, for a certain gate, we replace the original input with a four inputs multiplexer, which inputs are constant “1”, constant “0”, the original input and the inverted original input (figure 3). The constant “1” or constant “0” are realized by informing solver and create a unit constrain regarding this input.

MUX2

The MUX2 camouflage tool is based on the idea that multiplexer can be used as replacement to original input to make confusion. So, in implementation, we create a camouflage cell which is constituted of a multiplexer with a NAND gate, a NOR gate, and a XOR gate as candidates to connect to multiplexer. It can be observed from figure 4, the candidates share two inputs, so multiplexer can generate two keys to select the actual function. Intuitively, the complexity increase exponentially.

Mux4

This camouflage technique provides the most challenging where layout reveals nothing about the logic function of a gate. In the previous kinds of techniques, we only replace the input, while, in this type, the gate can implement any 2-input logic function. We use a look-up-table model as shown in figure 5. The strategy of an attacker is to make observations of a combinational circuit’s inputs and outputs and use those observations to eliminate programming vectors that are inconsistent with observations until only one possible circuit function remains consistent with the collected observations.

XOR

The goal of this camouflage mechanic is to prevent attacker from propagating forward or backward. We realize it by using camouflage cell connect to primary input or primary output (figure 6). The camouflage cell can be either a 2 inputs XOR gate or a 2 inputs XOR with an inverter which will provide an extra input, called “control bit”. The strategy for an attacker is to guess the correct control bit, otherwise the primary input will not be propagated in or the primary output will not be propagated out.