Department of Computer Science & Engineering

M.Tech (CSE)

CS577: C-based VLSI

Group Name: Bits-n-Bytes

Harshil Sadharakiya	224101022
Kaushal Mistry	224101031
Utsav Bheda	224101054
Vatsal Vasoya	224101056
Chinmay Rathod	224101063

Guided By

Dr. Chandan Karfa (Course Instructor)

Rittick Mondal (TA Instructor)

Problem statement:

We are having the Oracle and the Obfuscated code that is locked with the keys. The keys are unknown. Using any method or solver, find the unknown keys.

The details about the inputs to the obfuscated function:

Inputs: int i1, int i2, int i3, int i4, int i5, int G1, int G2, int G3, int G4, int GG1, int GG2

Keys: int k1, int k2, int k3, int k4, bool k5, bool k6, bool k7

int → 32 bits, bool → 1 bit

4 int keys & 3 bool keys will require (4*32) + (3*1) bits = 131 bits

Using Brute-Force there would be $\mathbf{2^{131}}$ combinations of keys.

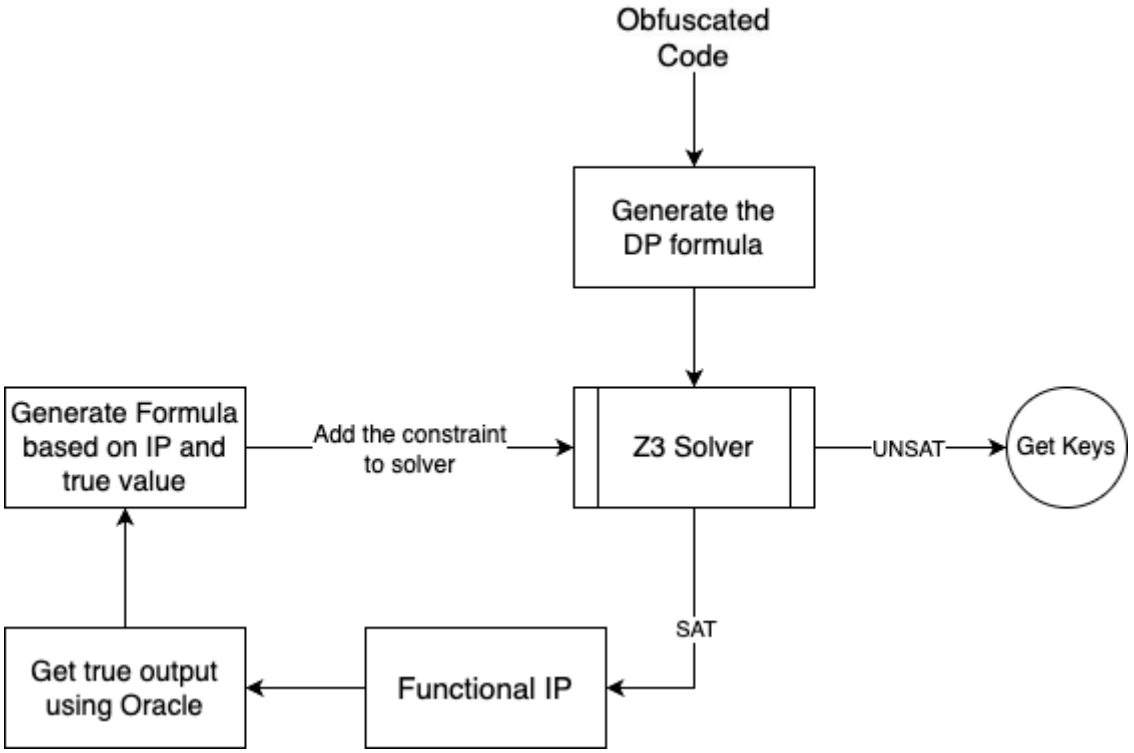
Details of tool and method:

- **Tool**: Z3 solver
- **Language used**: Python
- **Method**: SMT-based SAT Attack

Approach:

1. Initialize the inputs to the obfuscated function in the z3 variable format.
2. Initialize two copies of the keys using the z3 variable format.
3. Convert the Obfuscated code into the statement in the format required by z3.
4. Add the below constraint to the solver.
    a. output(1) != output(2) (for each output)
    Where,
    output(1) is using the first copy of the keys.
    output(2) is using the second copy of the keys.
    In both, the inputs would be the same.
5. Follow the steps below in the loop until z3 can find the unknowns given the constraints (Until UNSAT).
    a. Find a DIP given the constraints.
    b. Find the true output value using the inputs from the DIP using Oracle.
    c. Add new constraints to the solver with the original output from Oracle.
6. Once the solver could not find the DIP i.e. it gives UNSAT, change the constraints imposed in the 4th step as follows:
    a. output(1) == output(2) (for each output)
    b. key(1) == key(2) (for each key)
7. Run the solver again with the constraint and that should give the correct keys.

Flow chart showing the working of the Z3-based SAT attack:



Decoded values of the Keys	
Key	**Value**
Key 1	-89
Key 2	1243
Key 3	7
Key 4	-9
Key 5	False
Key 6	True
Key 7	True

---------- Thank You ----------