Background

- You have a computer program(software) that you wish to sell to someone.
- However, you would not like the buyer to sell it to a third party without necessary authorization from our side.
- But, in case if such a sale happens you would like to prove the ownership of software by using watermarking.
- Towards that goal, you customize your program for each customer, by embedding some unique identifier (watermark) at different locations.
- Then you obfuscate the program to make the process of identifying the watermark difficult.

Goal of watermark

- Software watermark can be used to identify the owner of the software in the case of dispute.
- Process of software watermarking is to embed a **watermark** y into a program P to get P_y.
- It is required that y should be reliably located and extracted from P_y even after subjected to code transformation and obfuscation.

Definitions

A software watermarking system is comprised of two functions:

- $embed(P, y, key) \rightarrow P_y$
- $extract(P_y, key) \rightarrow y$

Collatz function

- Let $\mathbb{N}$ be the set of all positive integers and $y \in \mathbb{N}$ then Collatz Function is defined as:

$$\theta(y) = \begin{cases} y/2 & y \equiv 0 \mod 2 \\ 3y + 1 & y \equiv 1 \mod 2 \end{cases}$$

- For example let $y = 3$, the Collatz sequence generated is 3, 10, 5, 16, 8, 4, 2, 1
- The **watermarking** is done by replacing the **if** conditions in the source code using Collatz function (Control Flow Obfuscation).

Selecting Conditional Construct (if condition)

- The operator in the conditional construct should be the equal-to operator (==)
- The conditional construct should not followed by any else construct.
- The LHS of the conditional expression should be a variable name.
- The RHS of the conditional expression should be an integer constant.
- If the source code doesn't contain any conditional construct we add our own conditional constructs to program for embedding collatz function.

Watermark Embedding using Collatz function

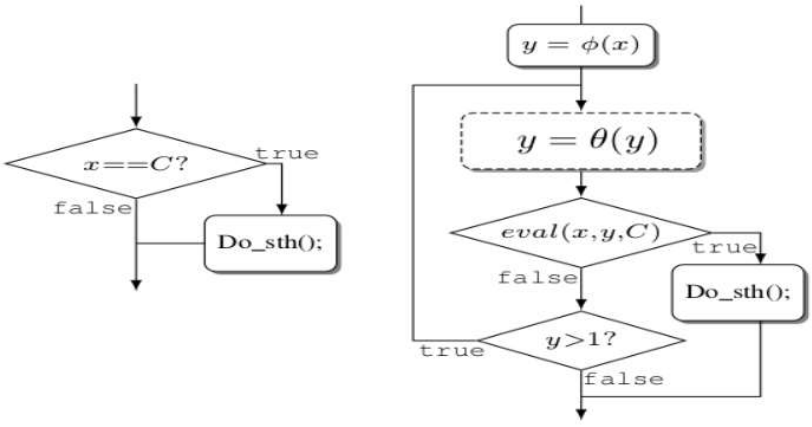


Figure: Watermark embedding

- eval(x,y,C) verifies if x + y < C + 2 and x - y >C - 2 are satisfied

Watermark Extraction using Collatz function

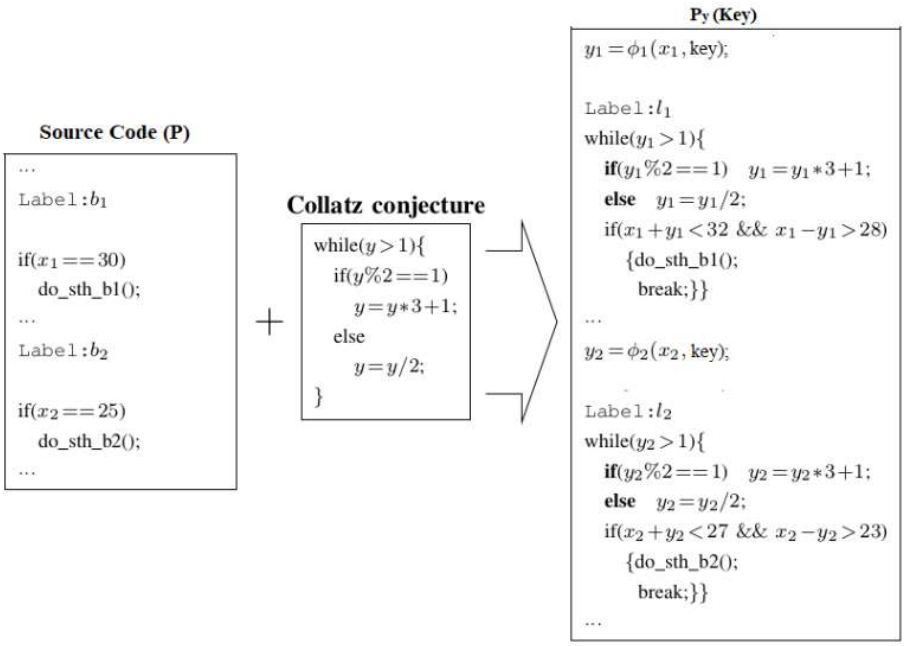
- Instrumentation - Two branches of Collatz function are mapped to 0s (y mod 2 ==1) and 1s (y mod 2 ==0).
- Traverse the trail from LSB to MSB.
- Obtain back the hailstone sequence.
- Recover initial integer of the sequence which is the watermark.

```
y = 3

3*3+1 = 10  ---> 0 --->  3
10/2  =  5  ---> 1 --->  10
3*5+1 = 16  ---> 0 --->  5
16/2  =  8  ---> 1 --->  16
8/2   =  4  ---> 1 --->  8
4/2   =  2  ---> 1 --->  4
2/2   =  1  ---> 1 --->  2

w = 3
```

Watermark Embedding - Example



Constraints for Selecting Watermark Values

- The complexity of each Collatz loop solely depends on the watermark value its based on.
- Starting with y = 12, one gets the sequence 12, 6, 3, 10, 5, 16, 8, 4, 2, 1. (Good Choice)
- The number y = 19 takes longer to reach 1: 19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. (Moderate Choice)
- The number y = 27, takes 111 steps (Bad Choice)
- To set a threshold we could take numbers that will generate orbit of length 20
  Eg. : 344064, 348160, 349184, 349440, 349504, 349520, 349524, 349525, 524288, 1048576, 2097152, ...