Loco: An interactive Code (De)Obfuscation tool

Matias Madou, Ludo Van Put and Koen De Bosschere

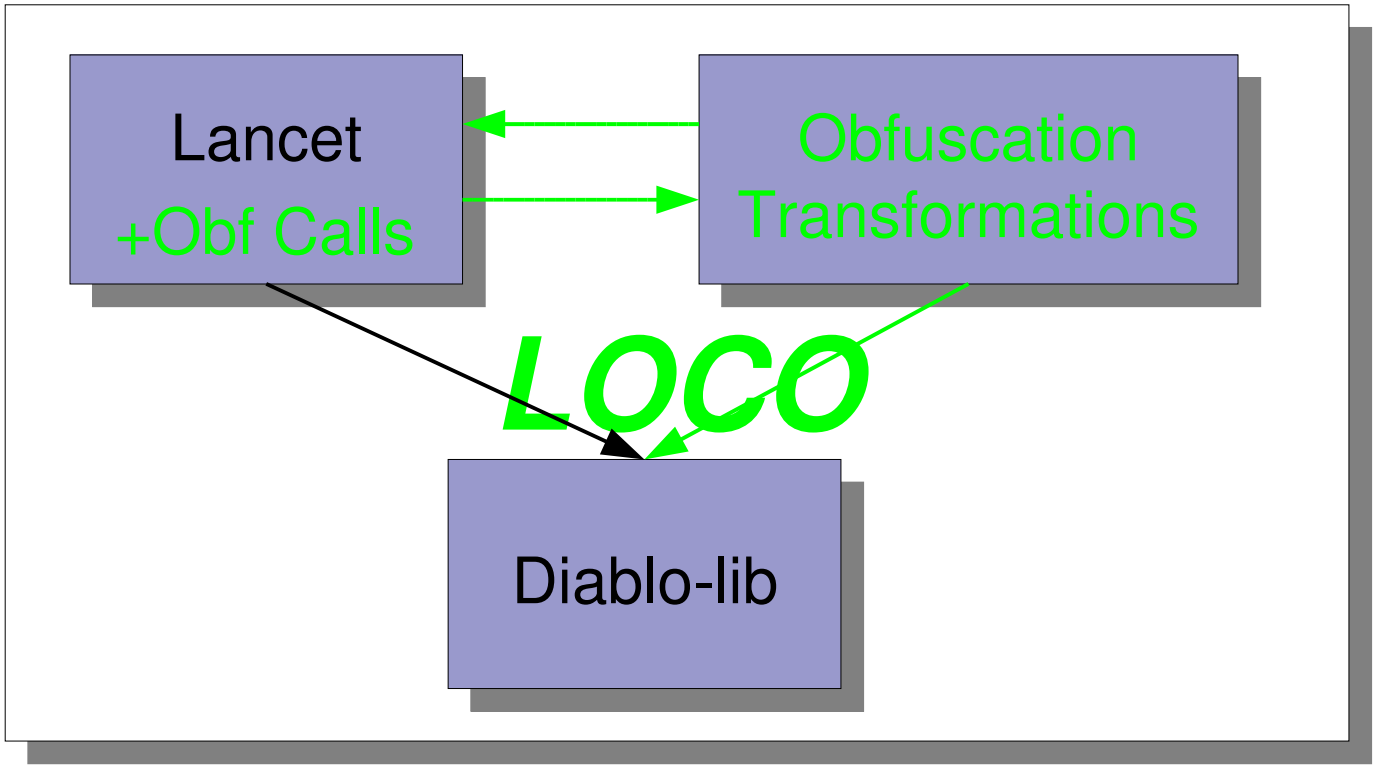
Motivation

- Software industry lost $31 billion of revenue due to software piracy
- Code obfuscation makes programs harder to understand
- Measuring the effort to undo a program transformation is very difficult

Goal

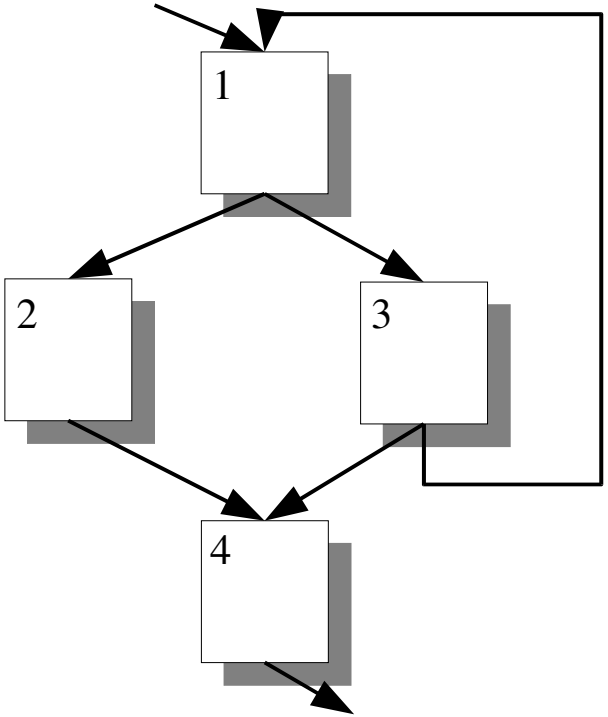
- A new experimental environment for obfuscation and deobfuscation
- The environment has to be graphical and interactive
- Transformations should be applied automatically, semi-automatically and by hand

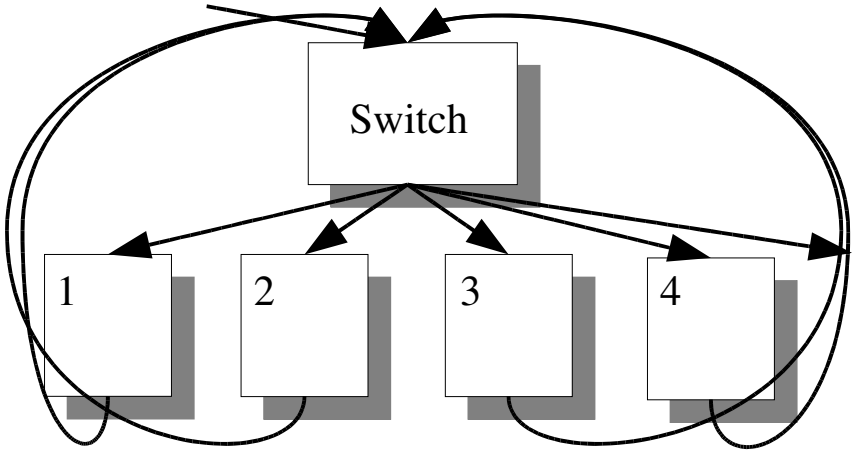
Background


Lancet
+Obf Calls
Obfuscation
Transformations
LOCO
Diablo-lib

Obfuscation Transformations

(1) Control Flow Flattening

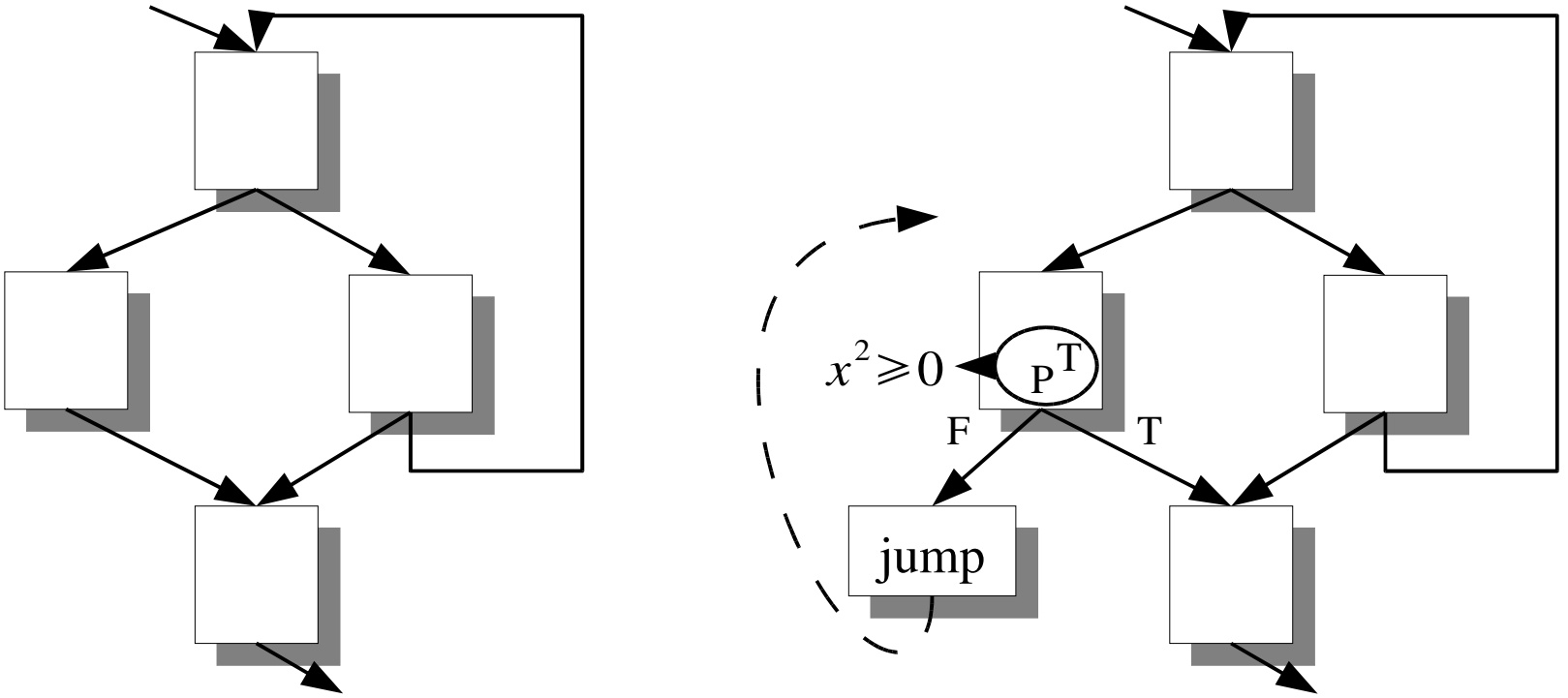




(1) A Security Architecture for Survivability Mechanisms, PhD thesis, C. Wang
(2) Central Part of an obfuscation tool by Cloakware Inc.

Obfuscation Transformations

(2) Opaque Predicates



Introduced by Collberg et al.

Example: Key code validation algorithm

Key codes:

Valid: 0003 0006

Invalid: 0003 0007

```
bool check(int key_part1, int key_part2)
{
    if(factorial(key_part1)==key_part2)
        return true;
    return false;
}
```

```
int factorial(int key)
{
    int a=1;
    if (key<1)
        a=1;
    else
        do{
            a *= key--;
        }while (key>1);
    return a;
}
```

Questions?

Information:

- Loco: http://www.elis.ugent.be/diablo/obfuscation
- Presentation: http://www.madou.net
- ACM SIGPLAN 2006 Workshop on Partial Evaluation and Program Manipulation (PEPM '06)