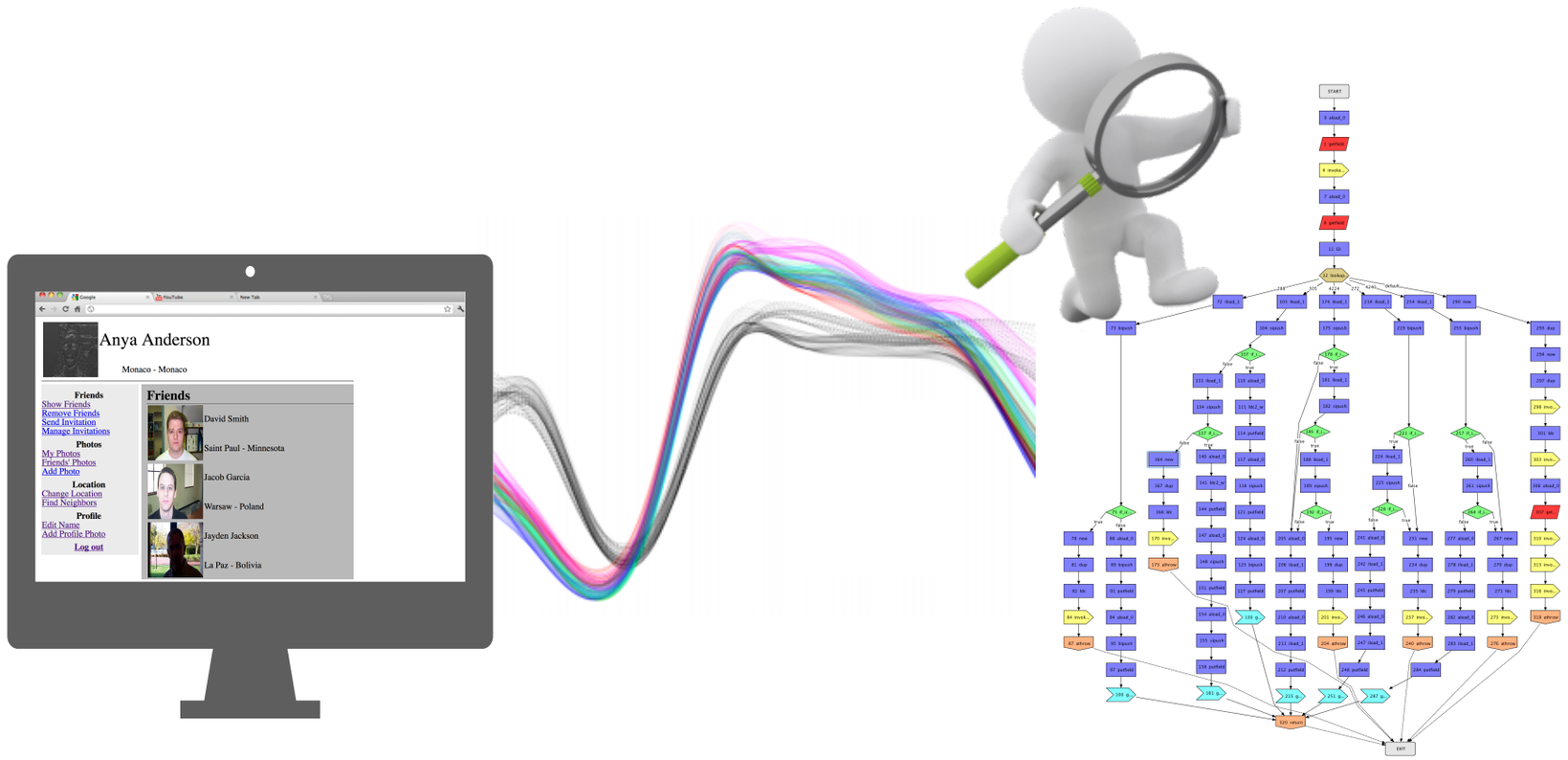
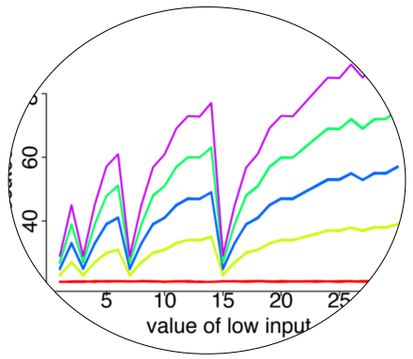


FUCHSIA: *Data-Driven Debugging for Functional Side Channels*

Saeid Tizpaz-Niari*, Pavol Cerny, Ashutosh Trivedi

*University of Colorado Boulder

Motivation



Functional
Side Channels

Case
Studies

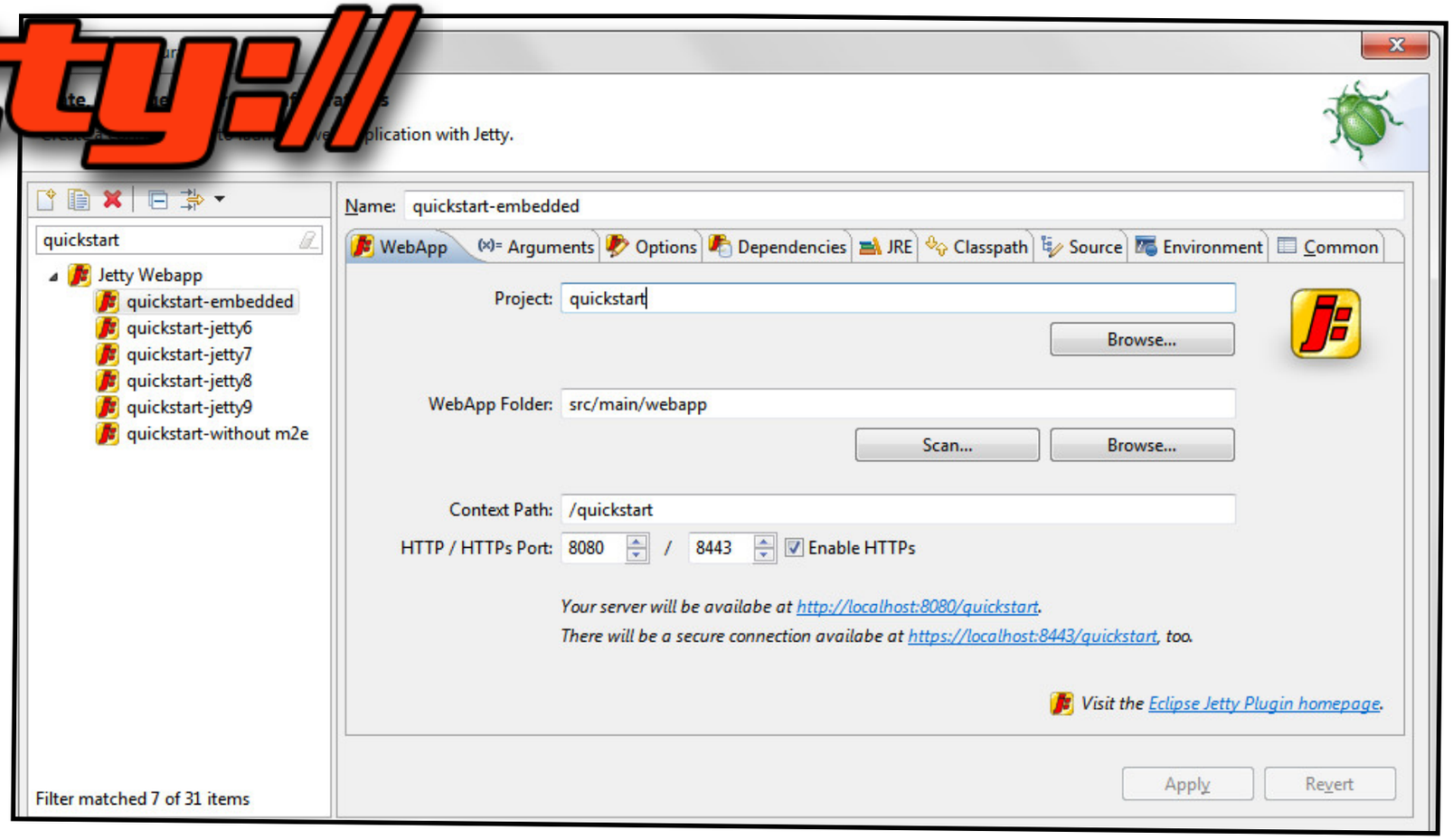
Motivation

Functional
Side Channels

Case
Studies

https://www.eclipse.org/jetty/

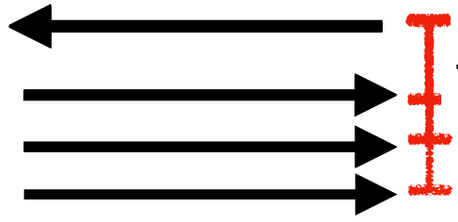
jetty://



maven

jetty:// V1

m y p a s s

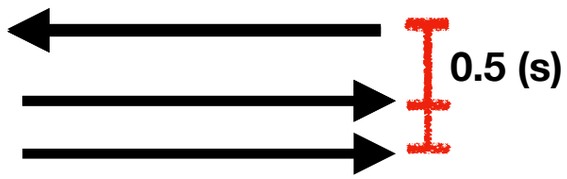


m y p a s s

Jia Chen, Yu Feng, Isil Dillig:
Precise Detection of Side-Channel Vulnerabilities using Quantitative Cartesian Hoare Logic. ACM Conference on Computer and Communications Security 2017: 875-890

jetty:// V2

m y p a s s



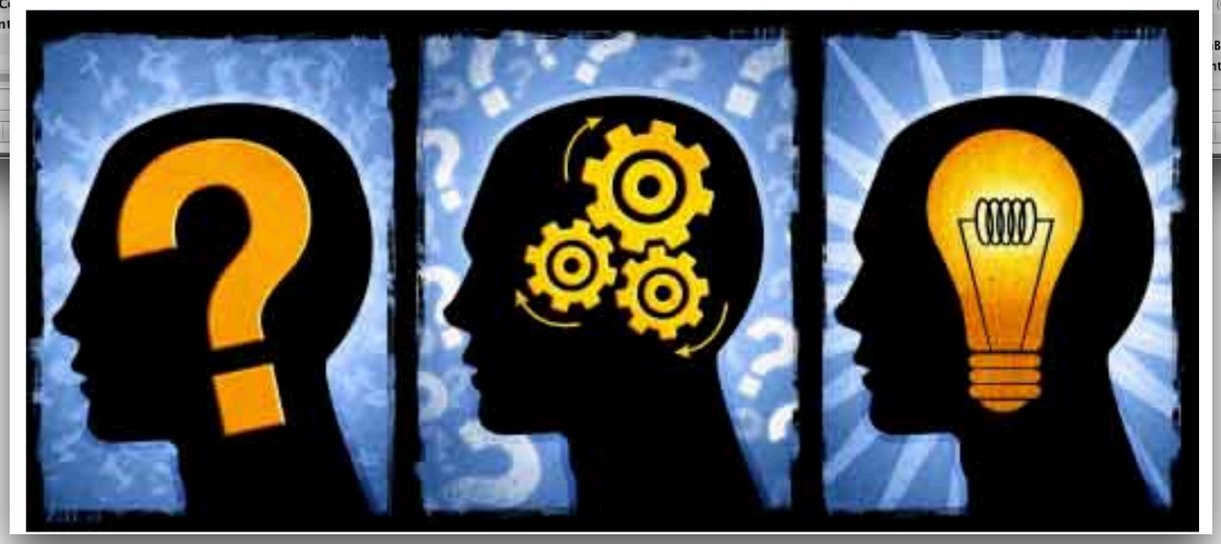
a b c d e f

jetty:// V3

?

Mind's Limit Found: 4 Things at Once

By Clara Moskowitz April 28, 2008 Health



- Time does not exist in the syntax or semantic
- Large applications with dynamic features

Data-Driven Differential Debugging: Program Analysis + ML

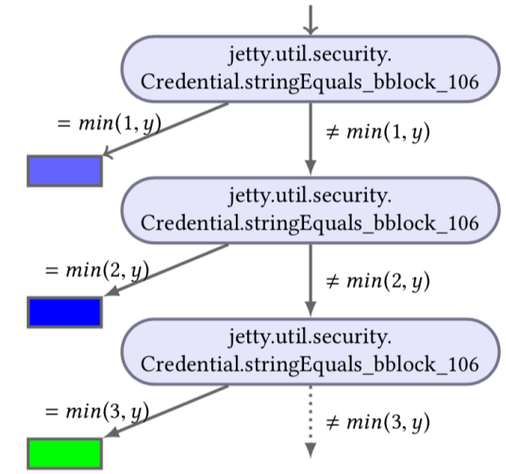
jetty:// V3

jetty.security…
StringEquals_bblock_5

jetty.security…
StringEquals_bblock_106

jetty.security…
StringEquals_bblock_5

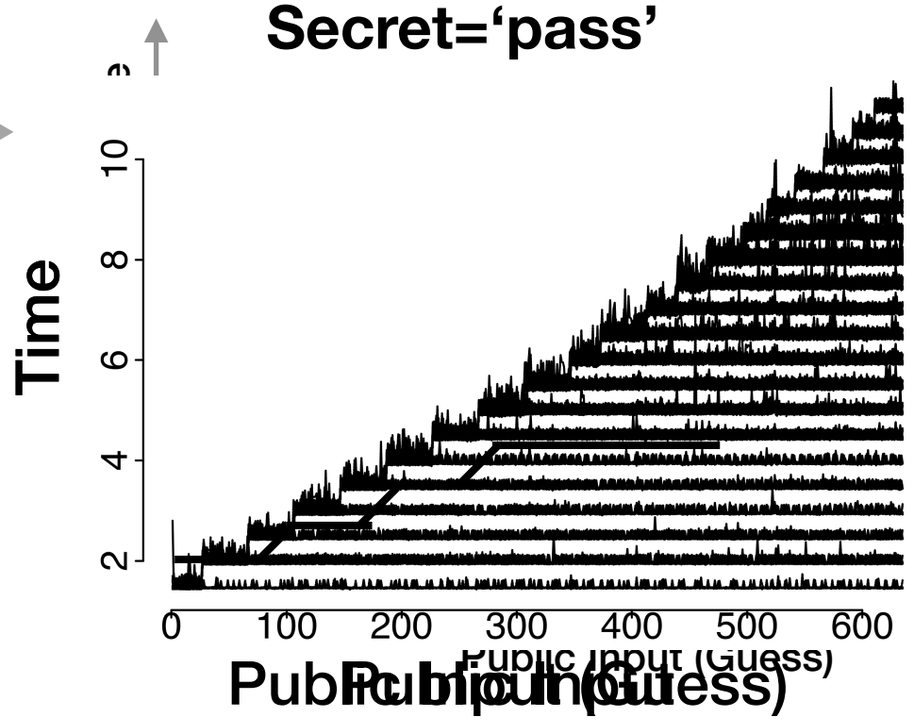
jetty.security…
StringEquals_bblock_106

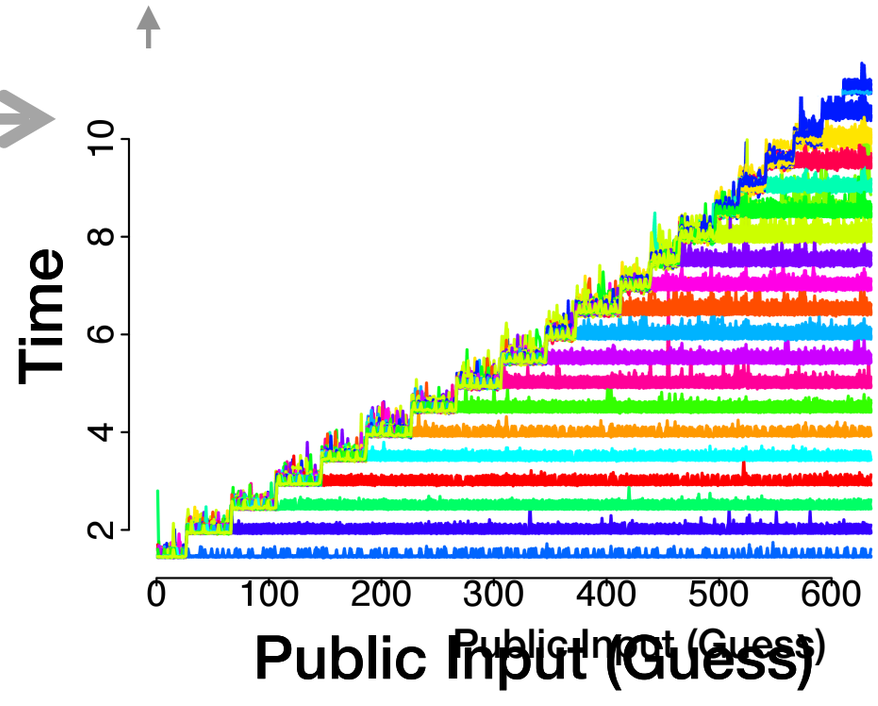


Secret	Guess
password	aa1234
password	pa12cd
password	…
mypass	a1b2c3
mypass	mypa
…	…

Instrumented Program
$\mathbb{P}'$

$\mathbb{P}$
Original Program





jetty:// V3

Secret	Guess
password	aa1234
password	pa12cd
password	…
mypass	a1b2c3
mypass	mypa
…	…

Instrumented Program $\mathbb{P}'$

$\mathbb{P}$ Original Program

jetty.security…
StringEquals_bblock_5

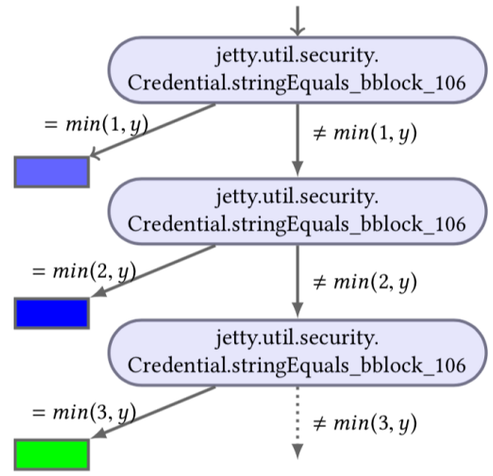
jetty.security…
StringEquals_bblock_106

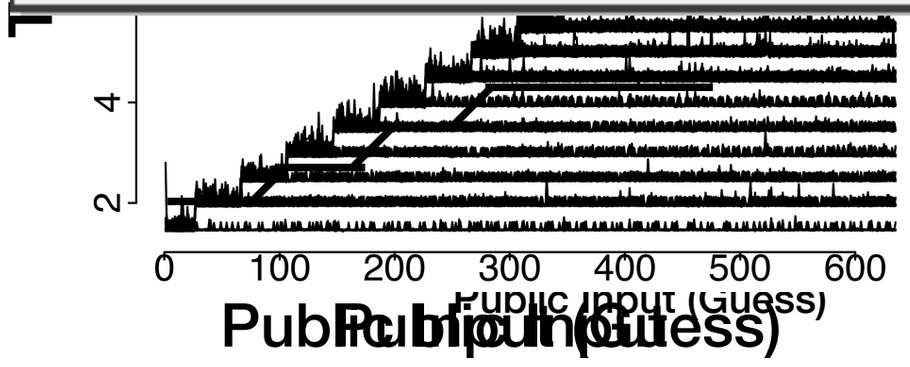
jetty.security…
StringEquals_bblock_5

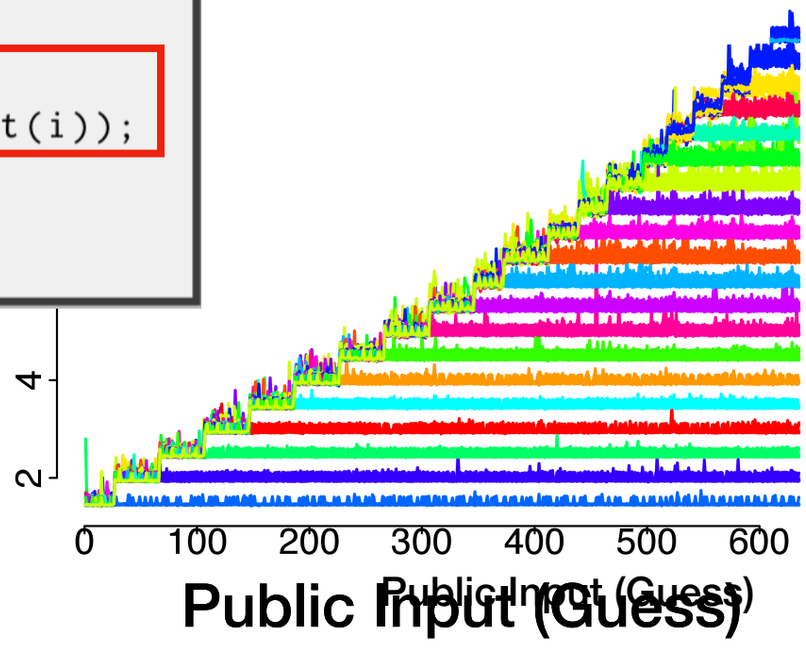
jetty.security…
StringEquals_bblock_106

stringEquals

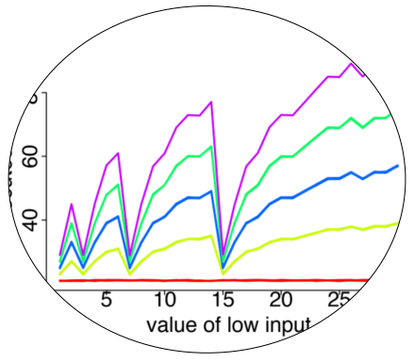
```
boolean stringEquals(String s1, String s2) {
  if (s1 == s2) return true;
  if (s1 == null || s2 == null) return false;
  boolean result = true;
  int l1 = s1.length(), l2 = s2.length();
  if (l1 != l2) result = false;
  int l = Math.min(l1, l2);
  for (int i = 0; i < l; ++i)
    result &= (s1.charAt(i) == s2.charAt(i));
  return result;
}
```





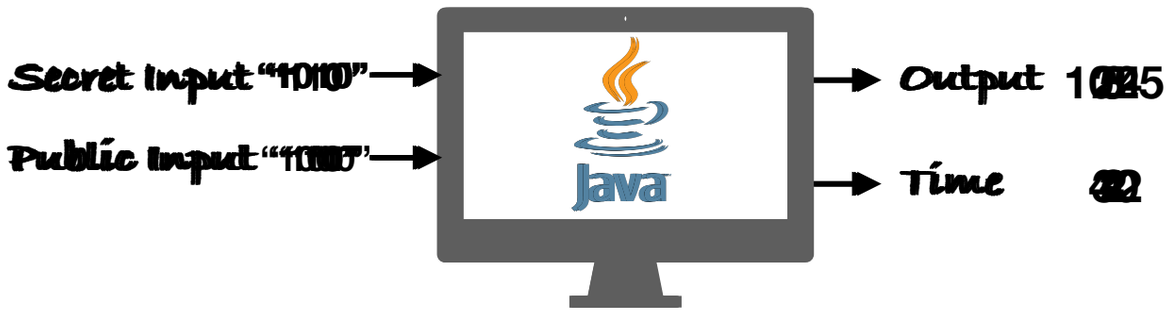


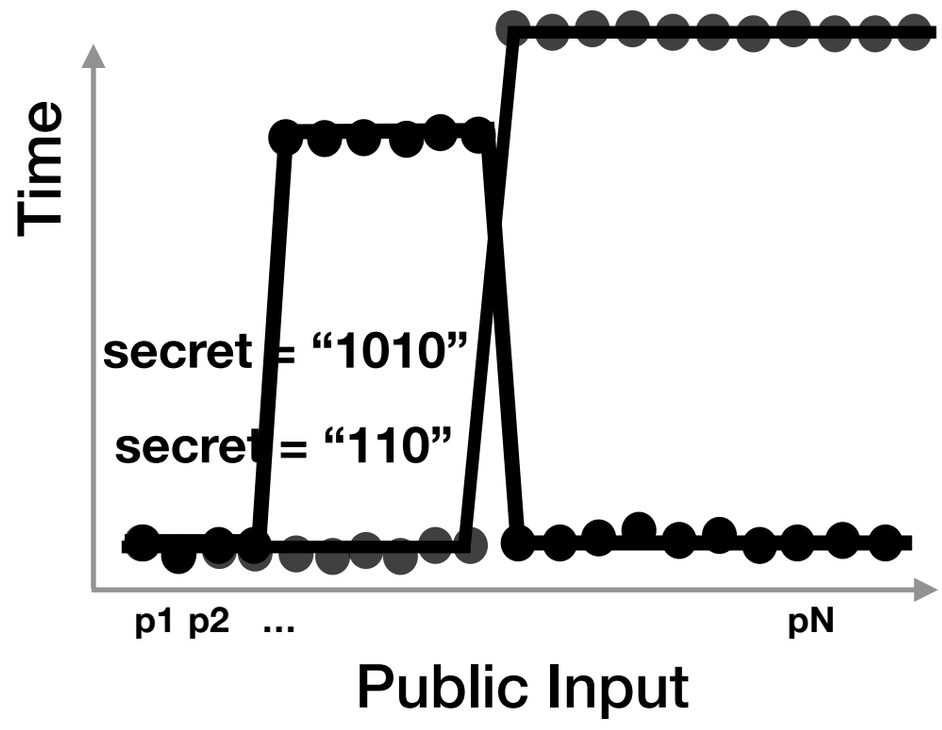
Motivation



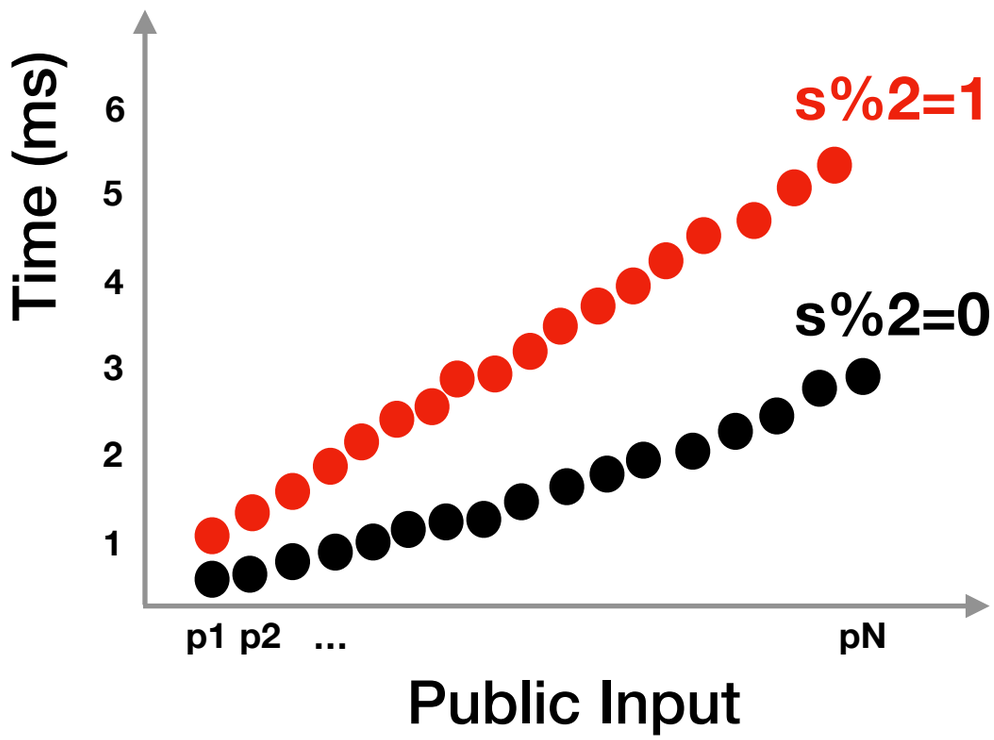
Functional
Side Channels

Case
Studies

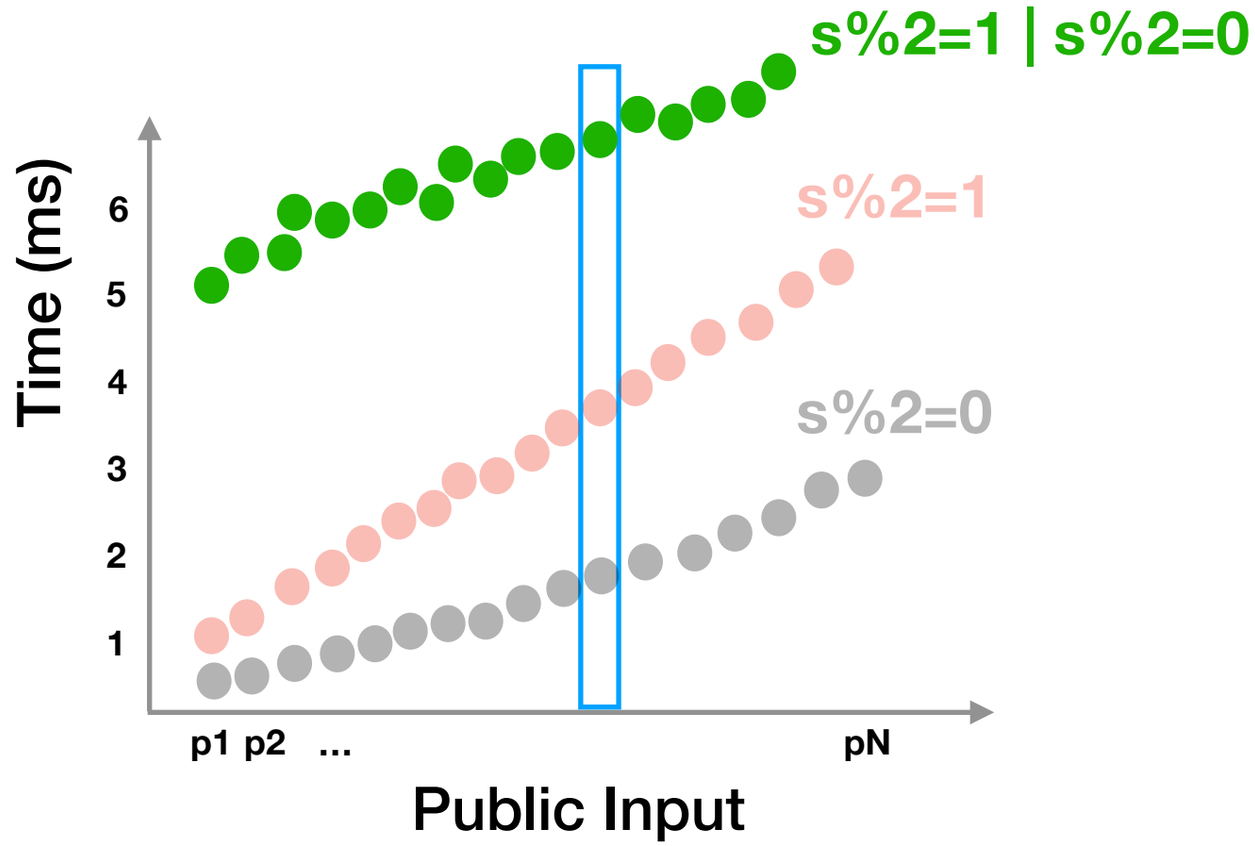

Secret Input "1010"
Public Input "110"
Java
Output
Time

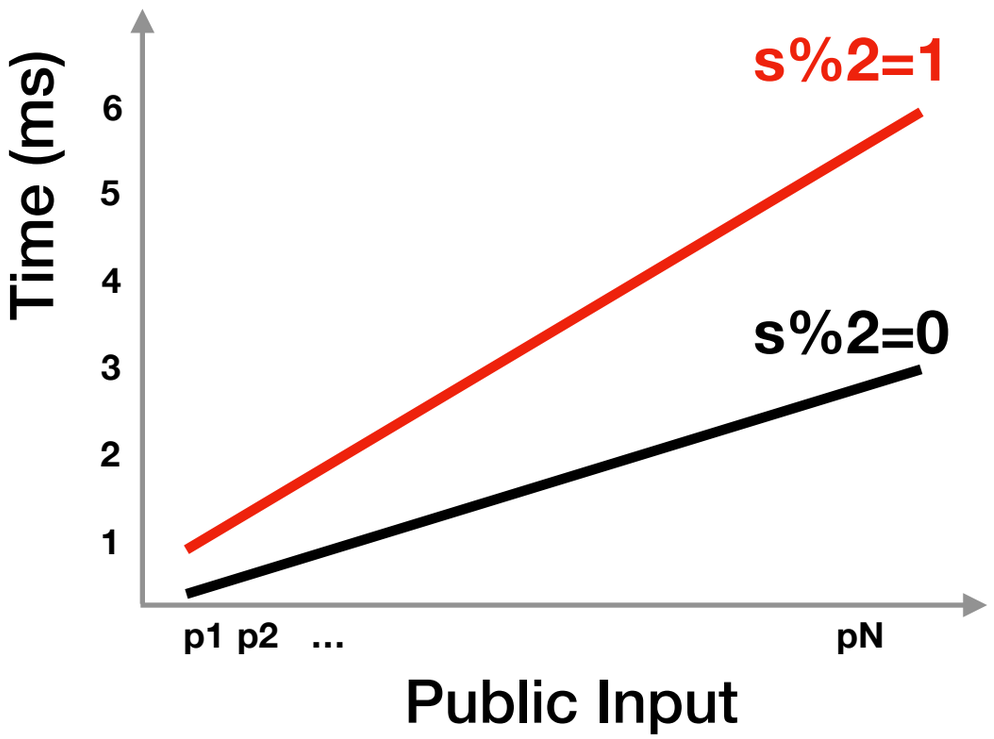

Time
secret = "1010"
secret = "110"
p1 p2 ...
pN
Public Input

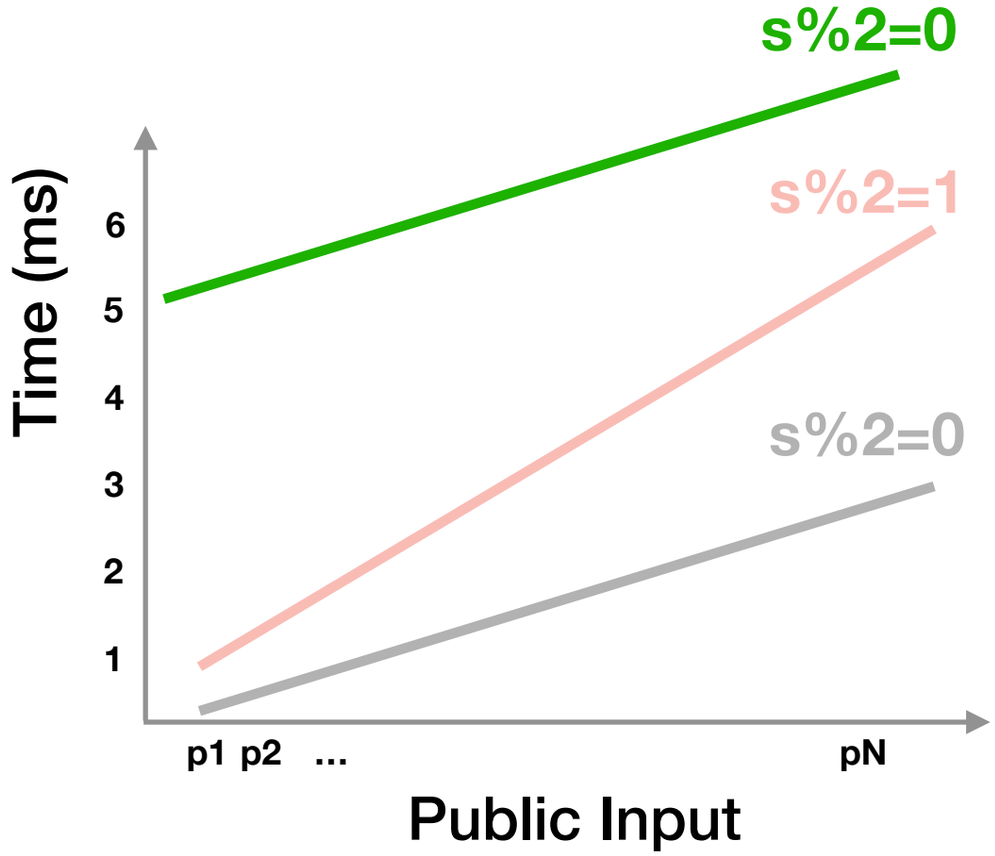
Attacker's Local Observations



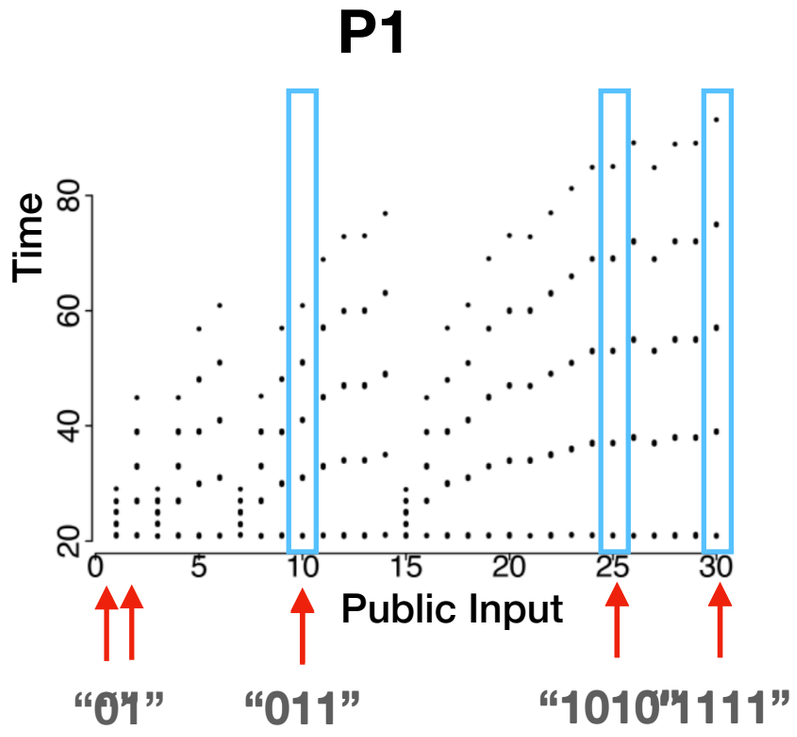
Attacker's Remote Observations

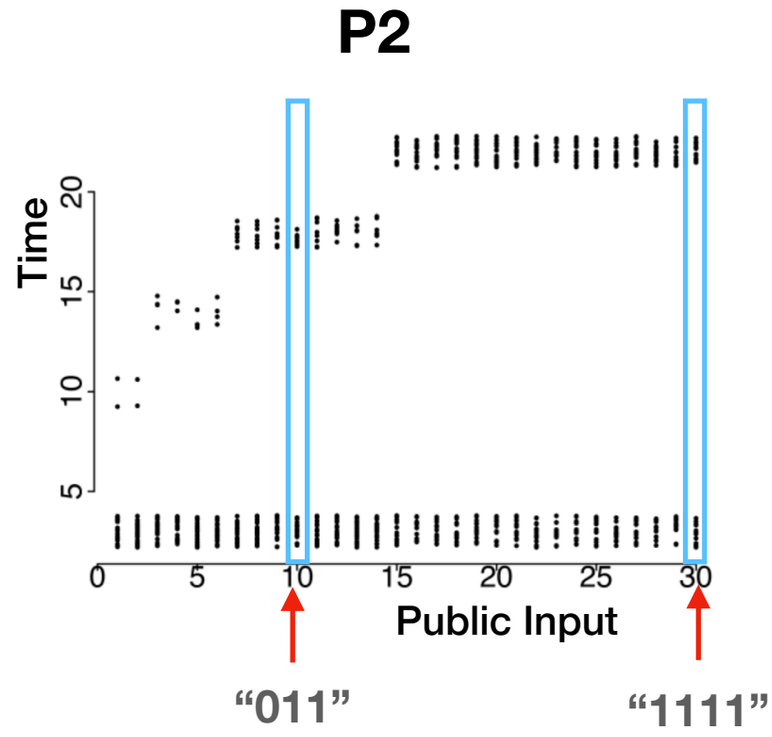




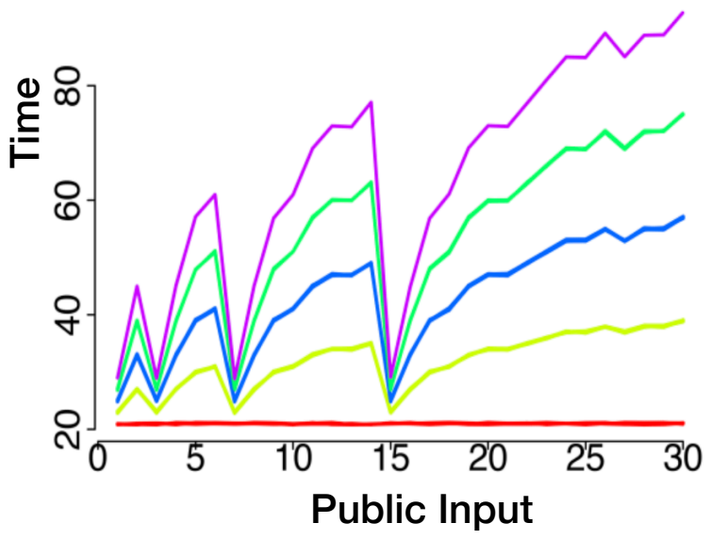


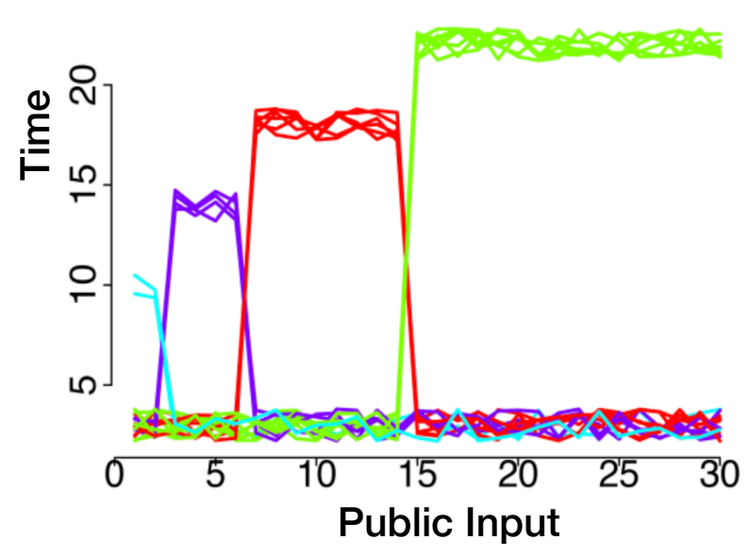
Point-wise Noninterference: Nilizadeh et al., ICSE'19



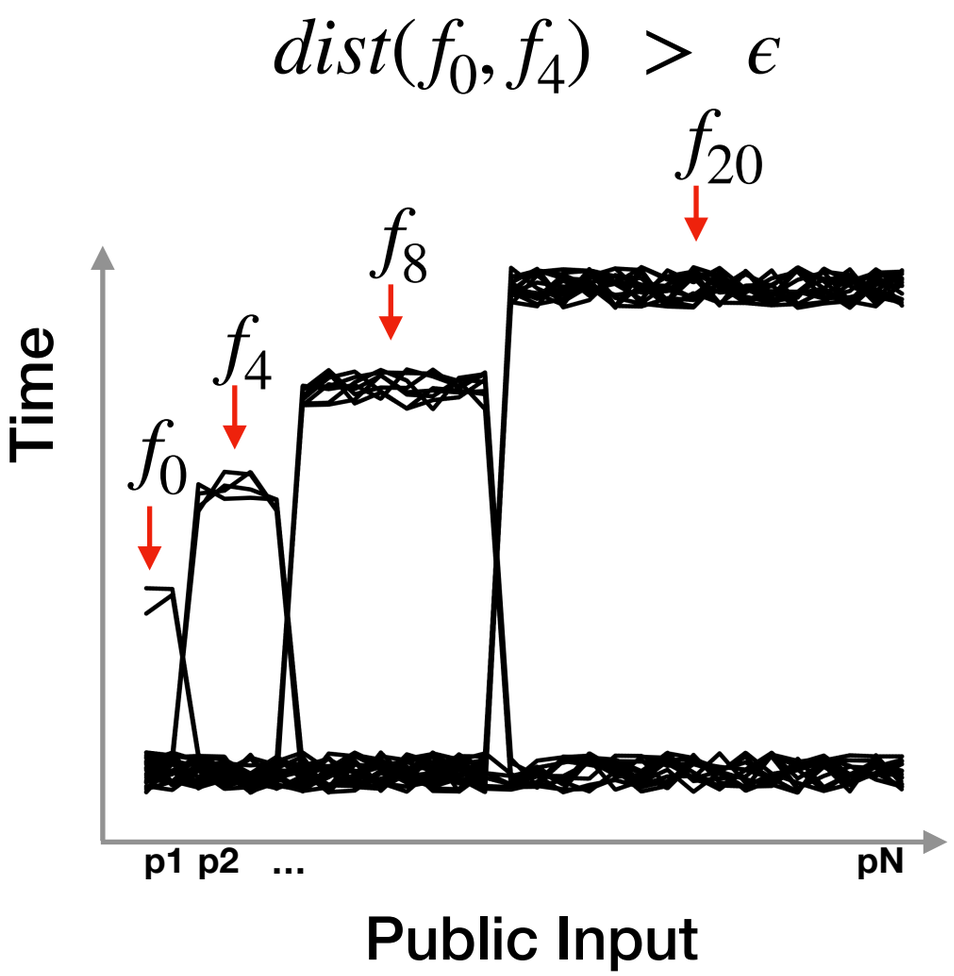


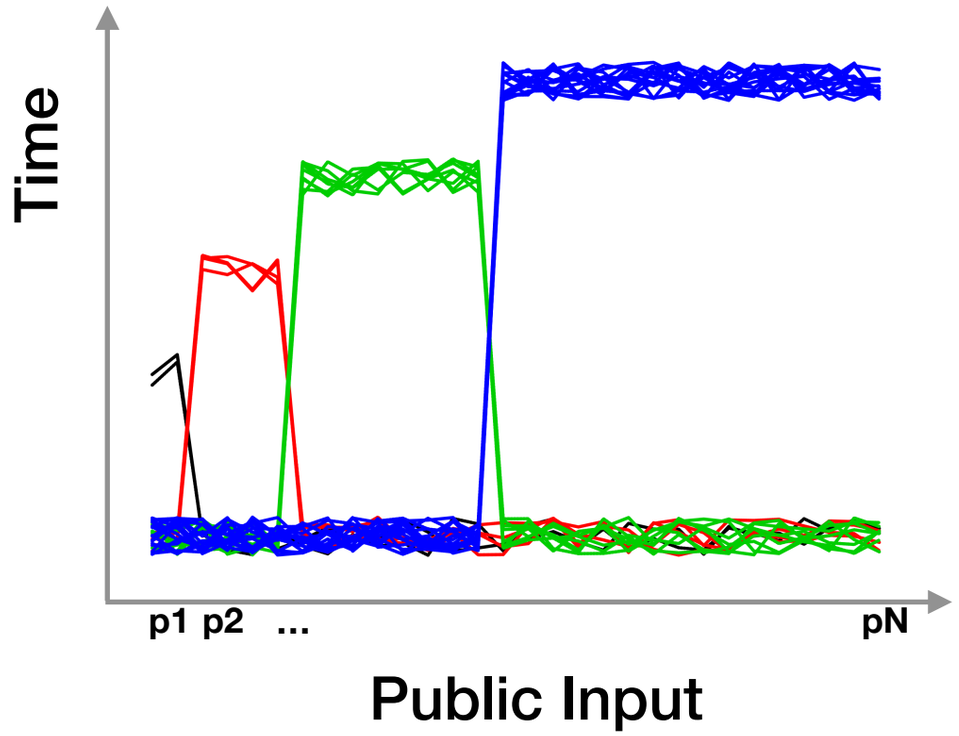
Functional Noninterference: Tizpaz-Niari et al., NDSS'20





Clustering: Distinguishable Functional Observations

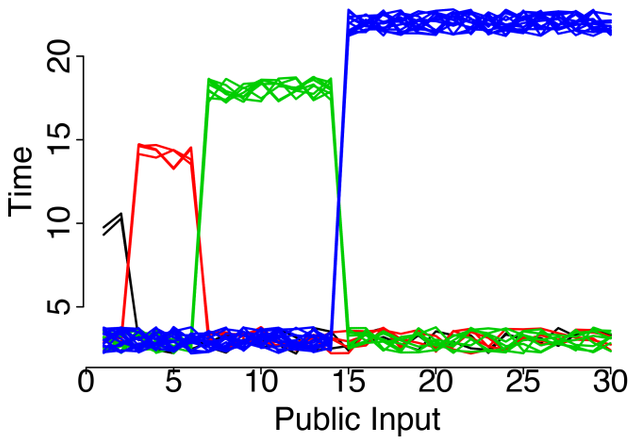


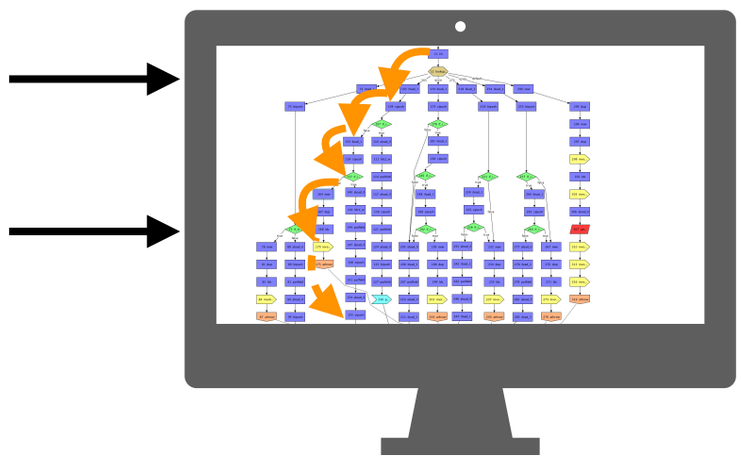


(f_0, f_{20}) in the same cluster!

Classification: Root Cause of Timing Side Channels

Secret	Public
“110”	“0”
“110”	“1”
“110”	“00”
…	…
“0110”	“0”
“0110”	“1”
“0110”	“00”
…	…





BasicBlock_13 ⋯▶ Basic_Block_18

BasicBlock_13 ⋯▶ Basic_Block_18

●●●

Secret =“110” → **Basic_Block_18**

Secret = “0110” → **Basic_Block_18**

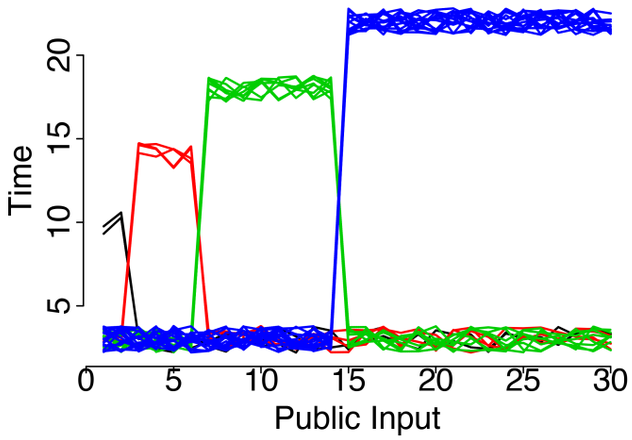
Secret = “110” → **BasicBlock_13**

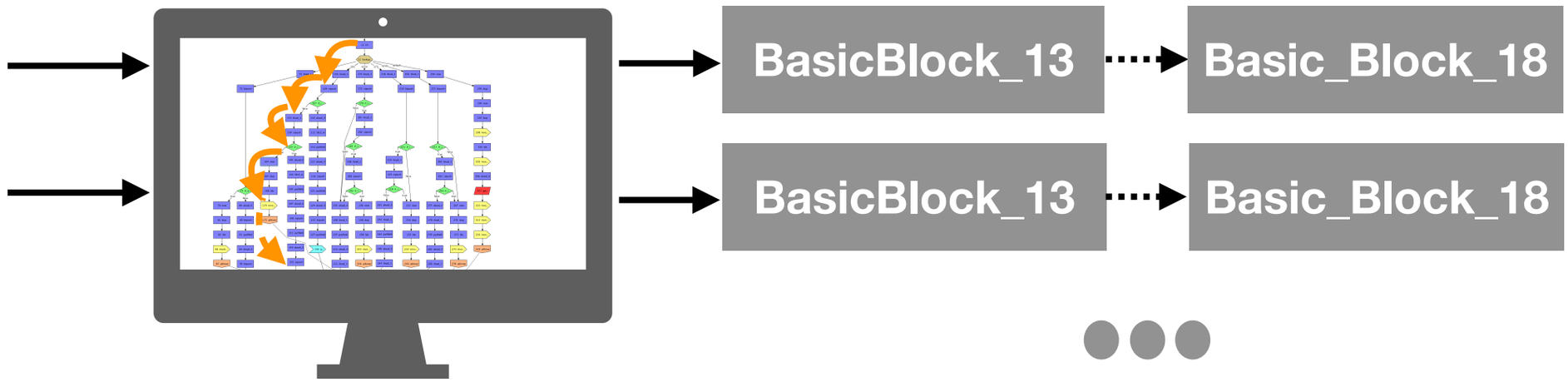
Secret = “0110” → **BasicBlock_13**

Public

	Secret =“110”	Secret = “0110”	Secret = “110”	Secret = “0110”
“01”	1	1	2	2
“111”	3	1	3	3
“1101”	1	4	4	4
	…	…	…	…
	min(3,y)	min(4,y)	1 * y	1 * y

Secret	Public
“110”	“0”
“110”	“1”
“110”	“00”
…	…
“0110”	“0”
“0110”	“1”
“0110”	“00”
…	…





Secre “0110”

Basic ock_13

Public

“01”
“111”
“1101”

Secret	Basic_Block_18	BasicBlock_13	…	Label
“1”	min(1,y)	y	…	
“10”	min(2,y)	y	…	
“110”	min(3,y)	y	…	
“1101”	min(4,y)	y	…	
“0110”	min(4,y)	y	…	
….	…	…	…	

2
3
4
…
y

Motivation

Functional
Side Channels

Case
Studies

Regular Expressions in Java

(#Methods: 620)

OpenJDK

Regex Library

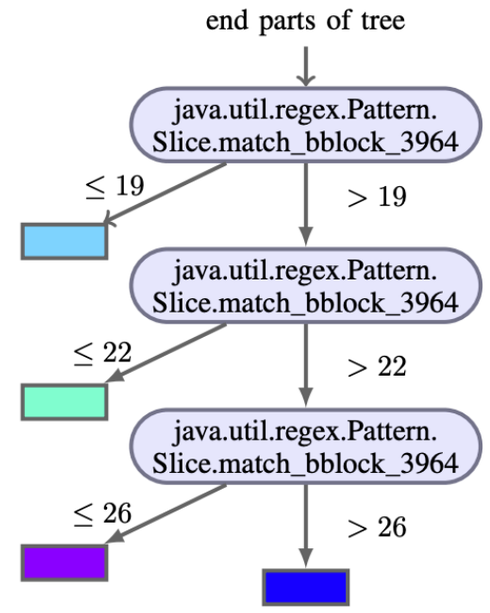
Secret	Guess
“abc123”	“aa123”
“abc123”	“mypa"
“abc123”	…
“mypass”	“aa123”
“mypass”	“mypa”
…	…

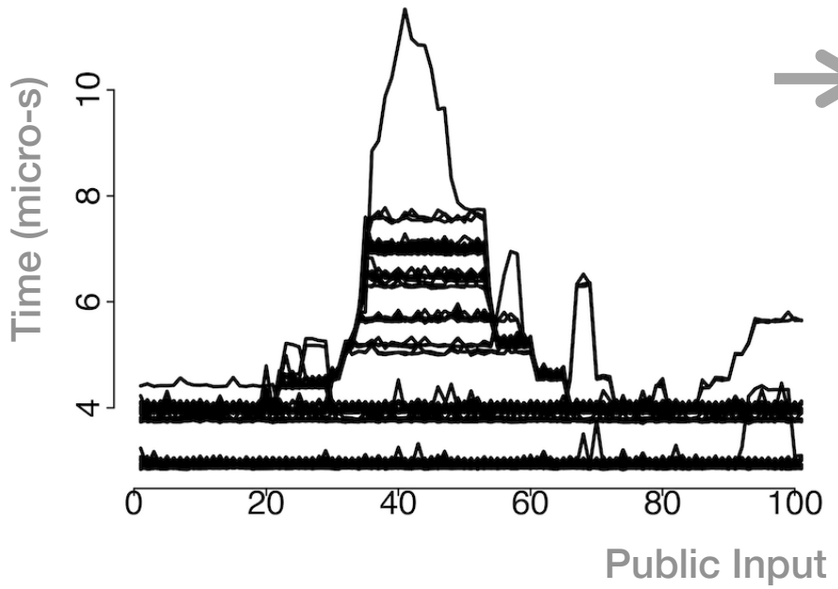
Instrumented Program $\mathbb{P}'$

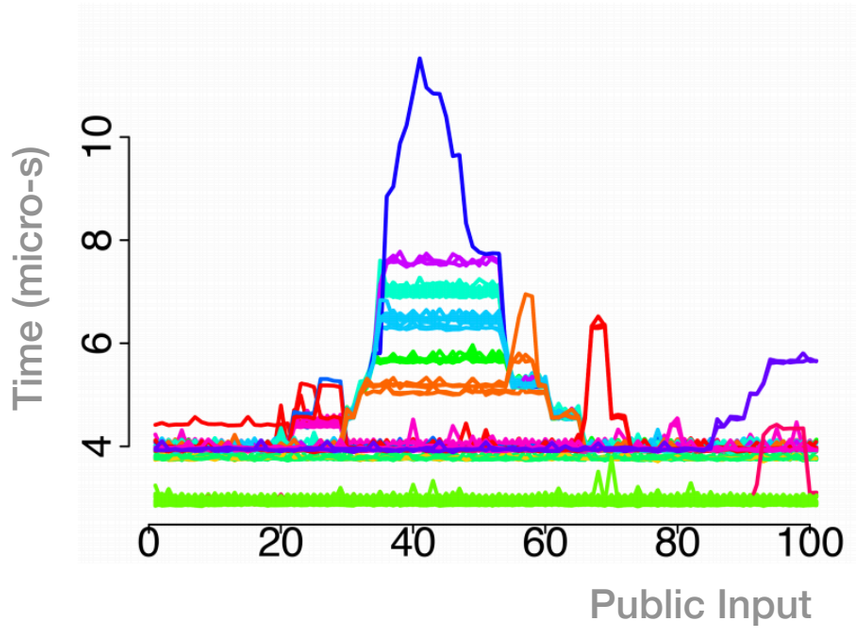
$\mathbb{P}$ Original Program

java.util.regex.Pattern.call_bblock_10 ⇢ java.util.regex.Pattern. Slice_bblock_3964

java.util.regex.Pattern.call_bblock_10 ⇢ java.util.regex.Pattern. Slice_bblock_3964







OpenJDK

Regex Library

java.util.regex.Pattern.
call_bblock_10

java.util.regex.Pa
n. Slice_bblock_3

java.util.regex.Pattern.
call_bblock_10

java.util.regex.Pa
n. Slice_bblock_3

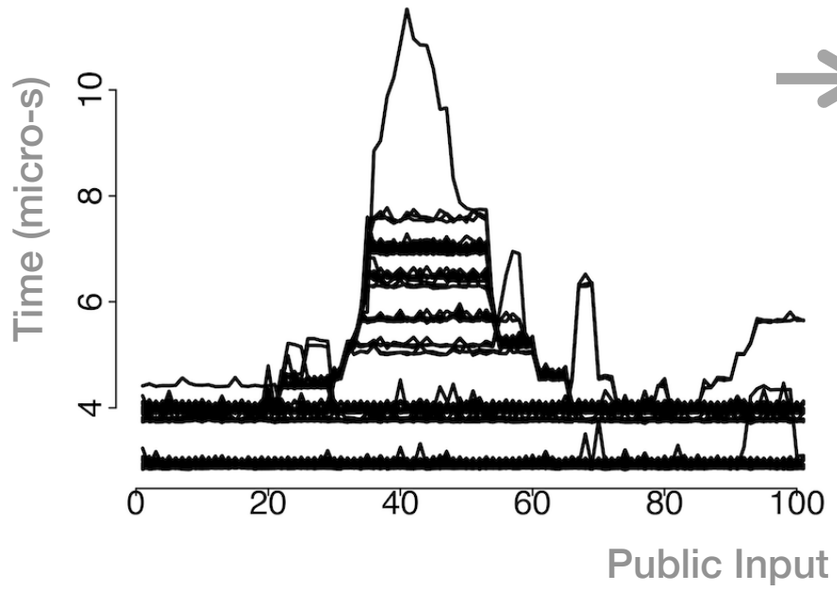
Regex.Pattern.Slice

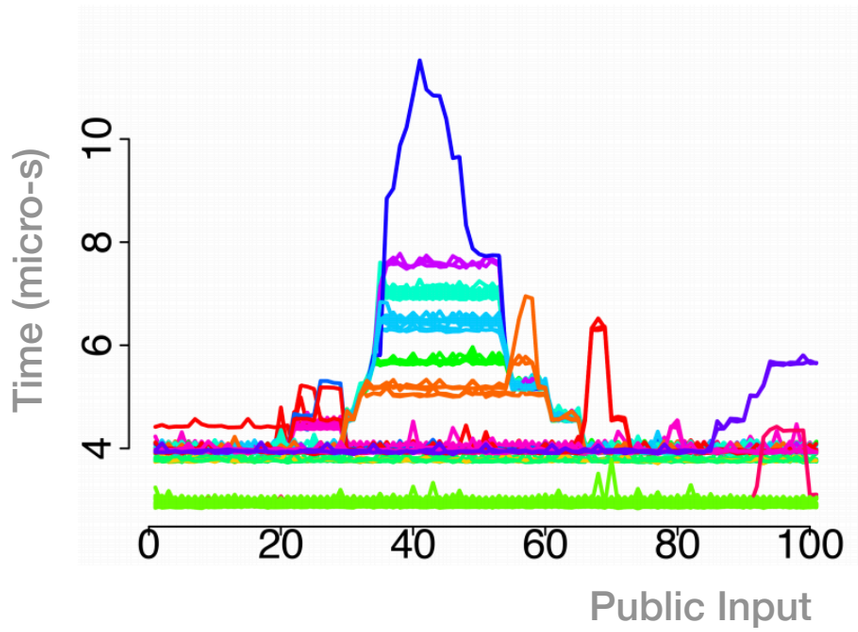
```
boolean match(Matcher matcher, int i, ...
  CharSequence seq){
    int[] buf = buffer;
    int len = buf.length;
    for (int j=0; j<len; j++) {    (line.3964)
      if ((i+j) >= matcher.to){
        matcher.hitEnd = true;
        return false;
      }
      if (buf[j] != seq.charAt(i+j))
        return false;
    }
    return next.match(matcher, i+len, seq);
  }
```

Secret	Guess
“abc123”	“aa123”
“abc123”	“mypa"
“abc123”	…
“mypass”	“aa123”
“mypass”	“mypa”
…	…

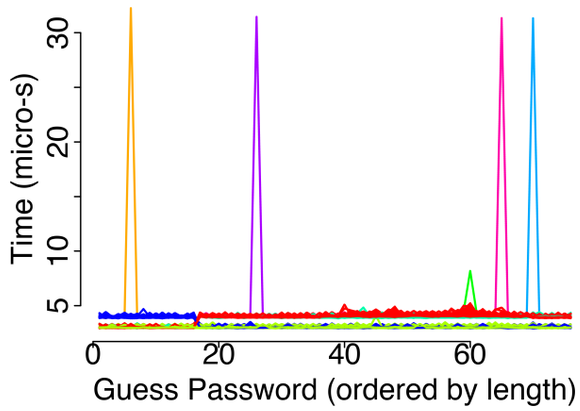
Instrumented Program
$\mathbb{P}'$

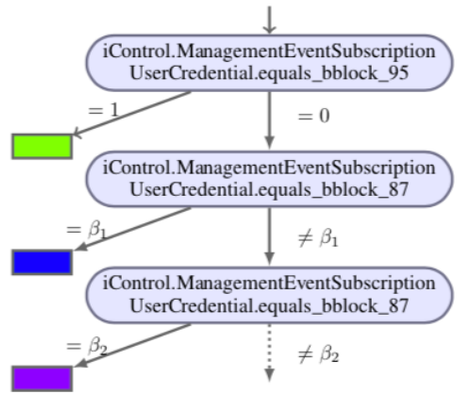
$\mathbb{P}$
Original Program



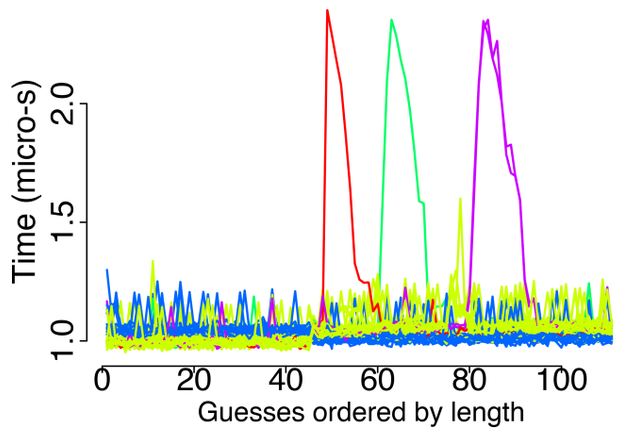


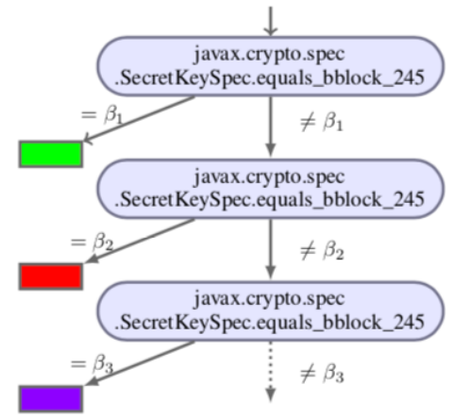
iControl-SOAP
(User Credential)
#Method: 41,541



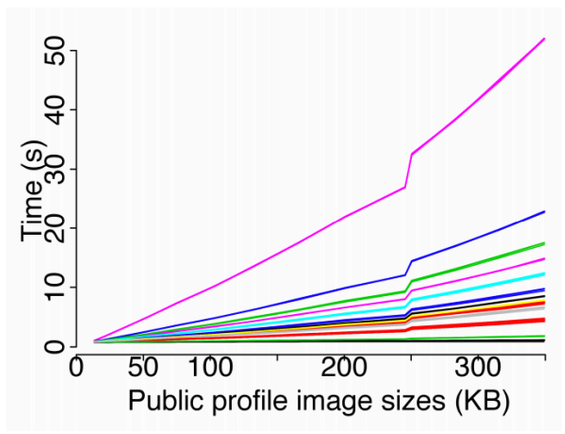


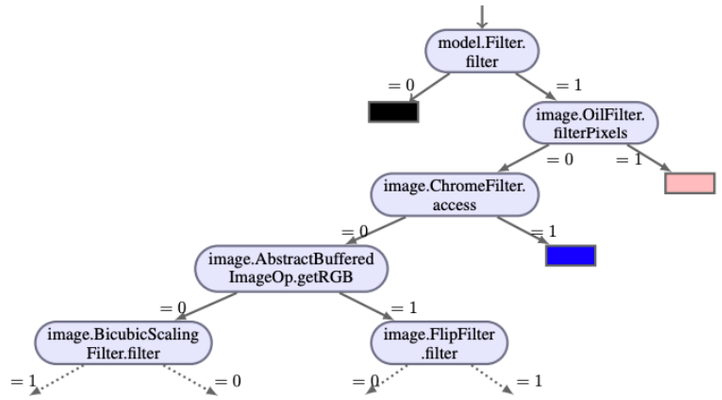
Java X
(Crypto)
#Method: 63



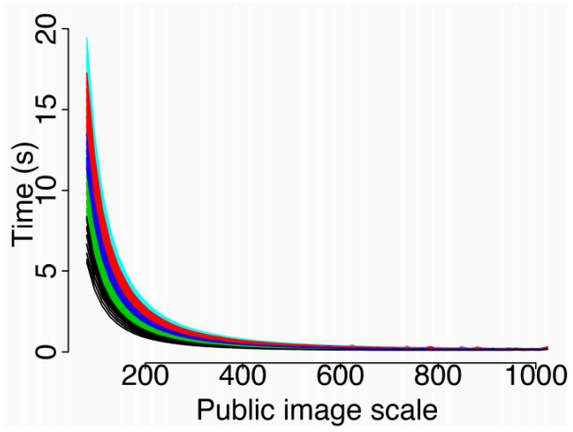


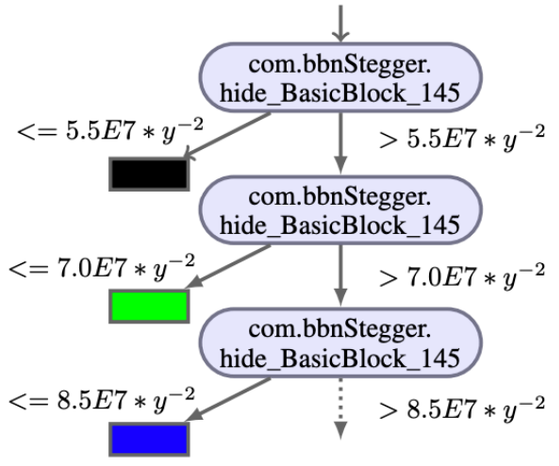
SnapBuddy
(Social Network)
#Method: 3,071





Stegosaurus
(Message Service)
#Method: 273

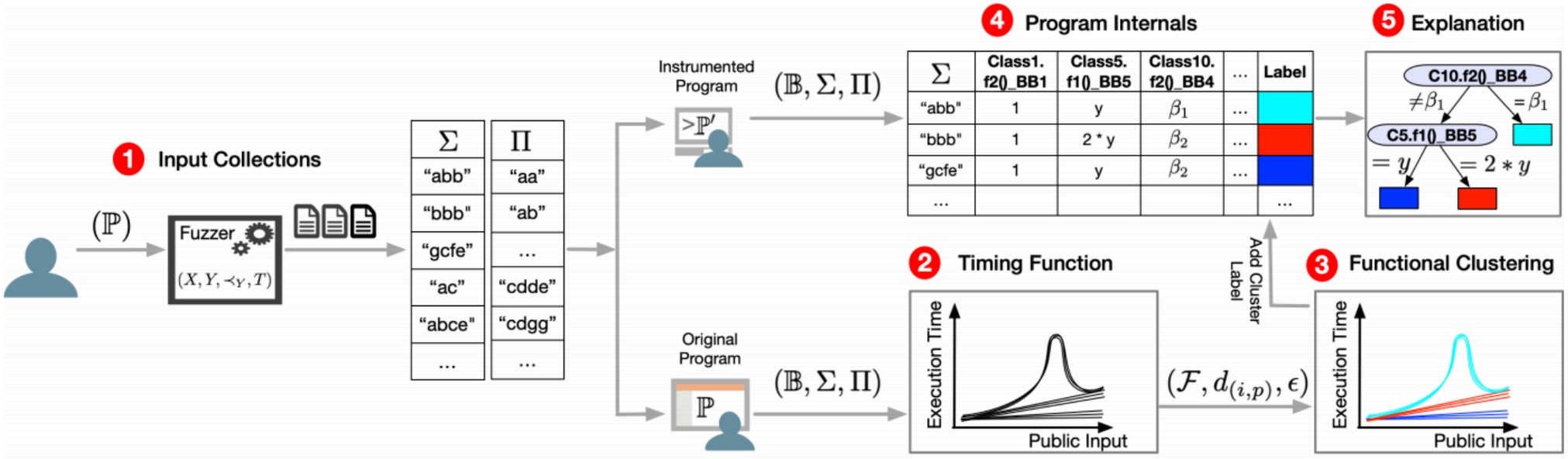




University of Colorado Boulder

CUPLV

Thank you for your attention!



Saeid.Tizpazniari@colorado.edu