Reverse engineering Java Card applets using power analysis

Workshop in Information Security Theory and Practices 2007

Dennis Vermoen, Marc Witteman and Georgi N. Gaydadjiev

May 10, 2007

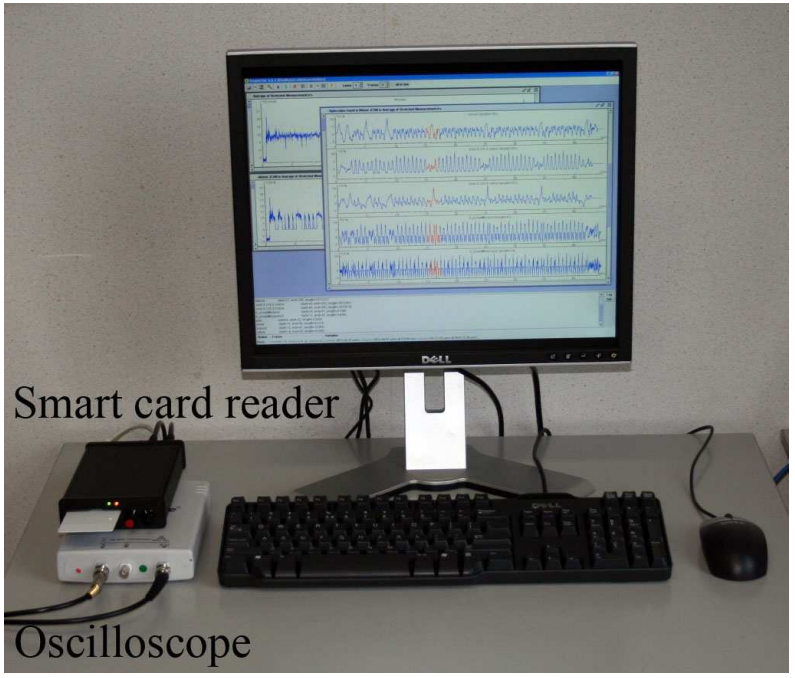
Outline

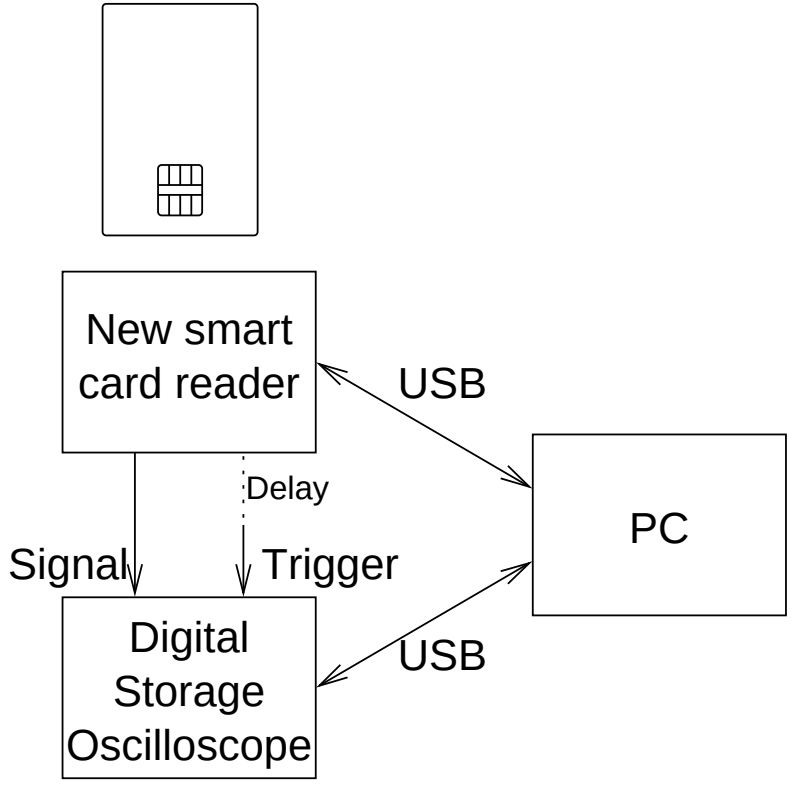
- Introduction
- Acquisition framework
- Methodology
- Experimental results
- Conclusions

Introduction

- Java Card technology
- Power analysis
- Motivation
- Experiments performed on several commercially available smart cards

Acquisition framework





Methodology

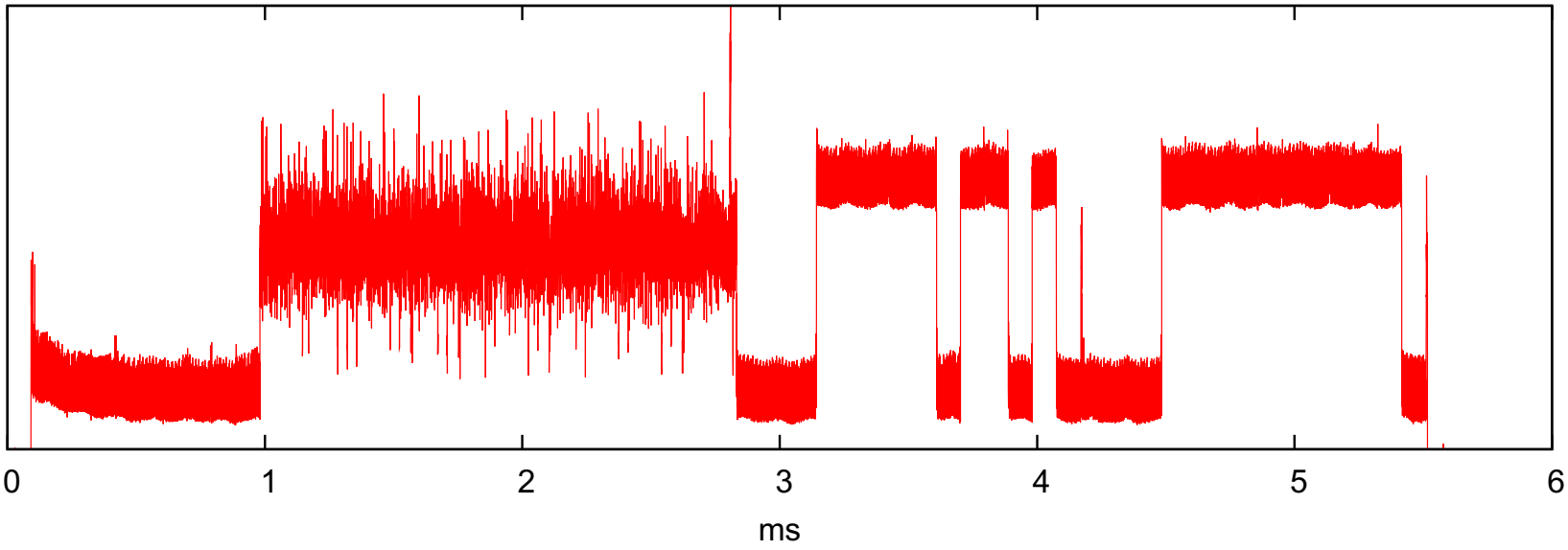
- Trace set preprocessing
  - Alignment
  - Resampling
  - Averaging
- Template determination (known applet)
- Template recognition (unknown applet)
- Loop rerolling

Template determination

```
1       public void process(APDU apdu) {
2          byte a = (byte) 0x03, b, c;
3
4          if (selectingApplet())
5             return;
6
7          byte buffer [] = apdu.getBuffer();
8          short len = apdu.setIncomingAndReceive();
9
10         b = buffer [(short)(ISO7816.OFFSET_CDATA)];
11
12         c = (byte)(a + b); // sload_2, sload_3, sadd, s2b, sstore_4
13         c = (byte)(a * b); // sload_2, sload_3, smul, s2b, sstore_4
14         c = (byte)(a * b); // sload_2, sload_3, smul, s2b, sstore_4
15         c = (byte)(a + b); // sload_2, sload_3, sadd, s2b, sstore_4
16         c = (byte)(a * b); // sload_2, sload_3, smul, s2b, sstore_4
17         c = (byte)(a + b); // sload_2, sload_3, sadd, s2b, sstore_4
18         c = (byte)(a + b); // sload_2, sload_3, sadd, s2b, sstore_4
19         c = (byte)(a * b); // sload_2, sload_3, smul, s2b, sstore_4
20      }
```

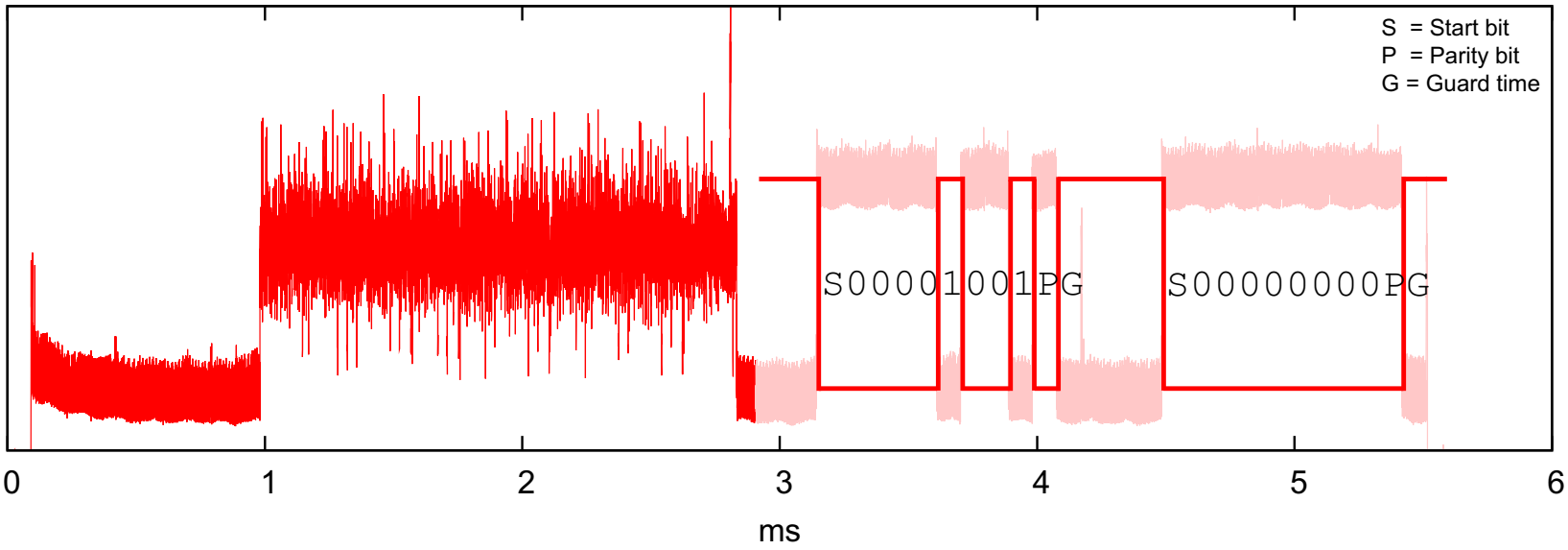
Template determination (2)

- **Execute the known applet**
- Determine part of interest
- Acquire lots of traces and average them



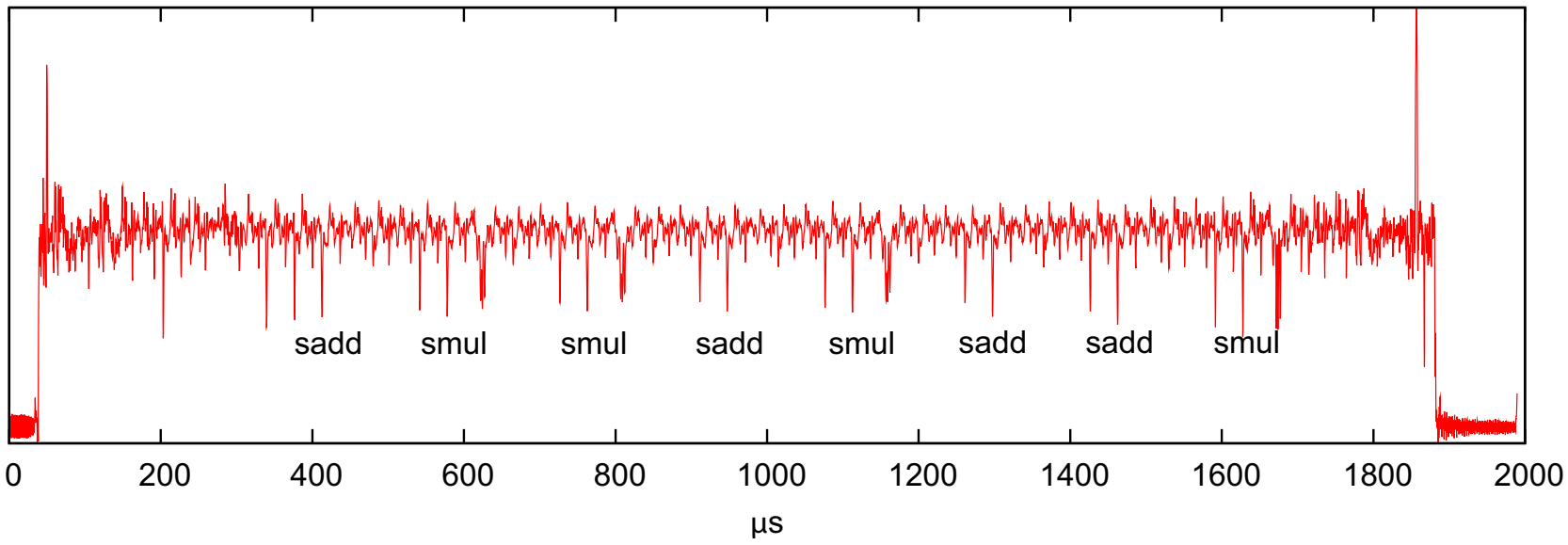
Template determination (2)

- Execute the known applet
- **Determine part of interest**
- Acquire lots of traces and average them

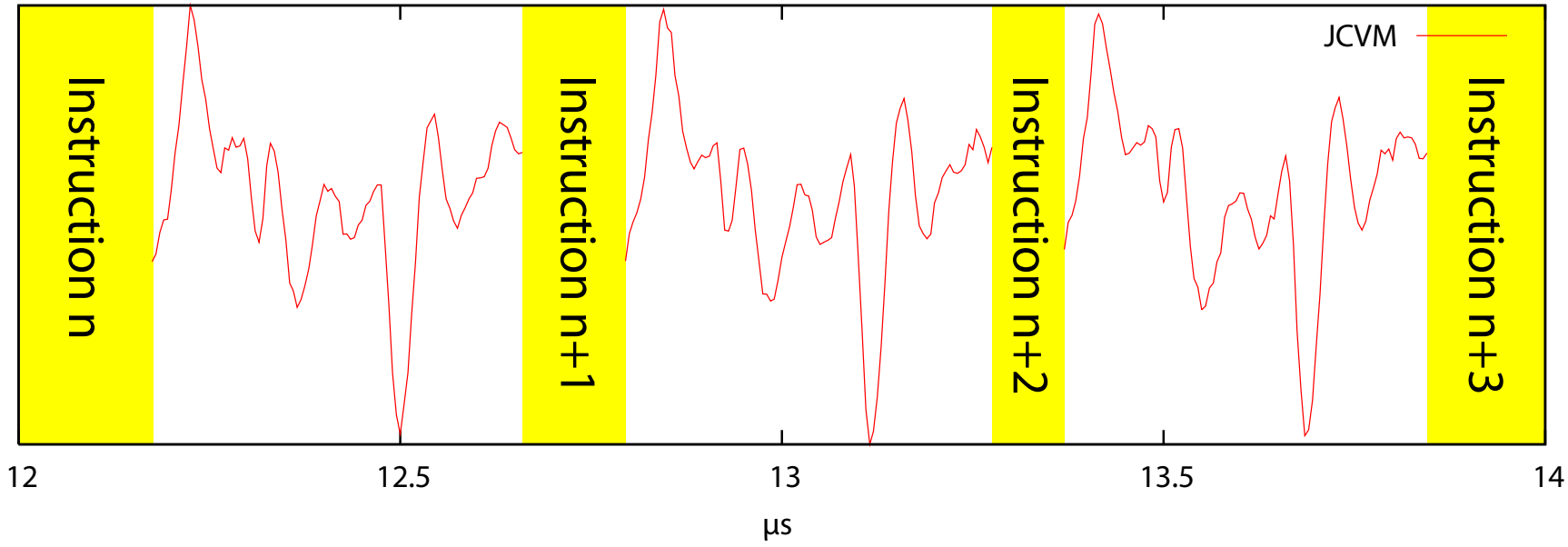


Template determination (2)

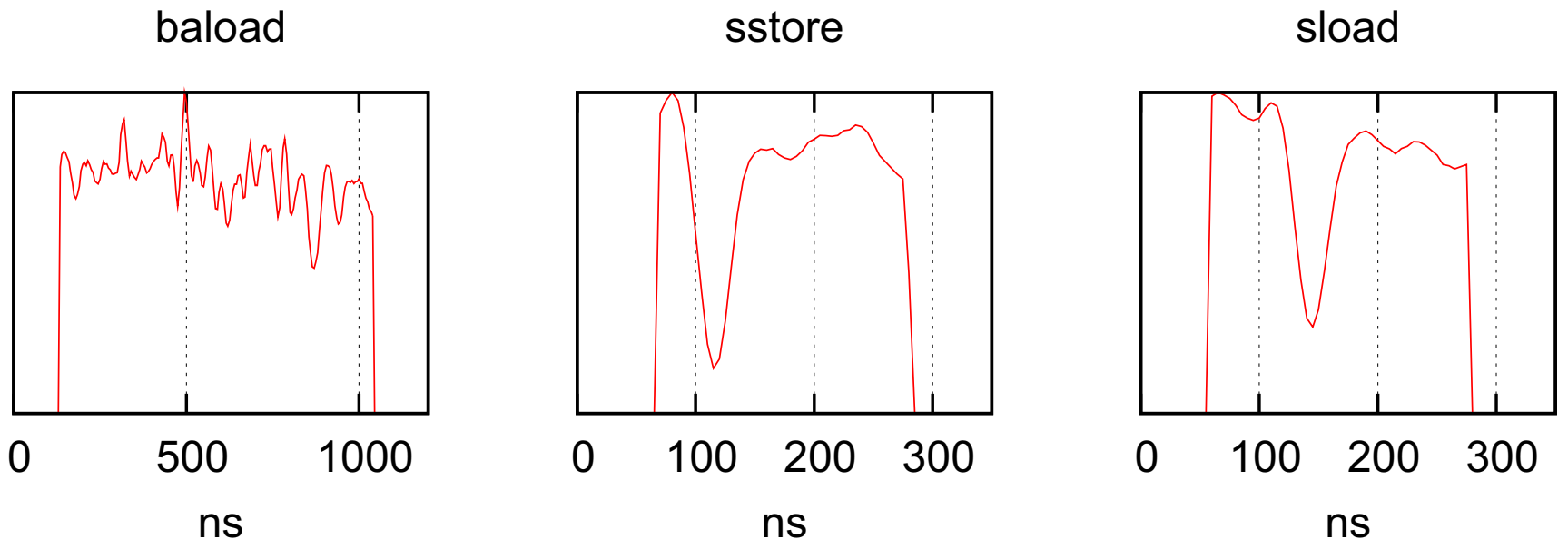
- Execute the known applet
- Determine part of interest
- **Acquire lots of traces and average them**



Template determination (3)



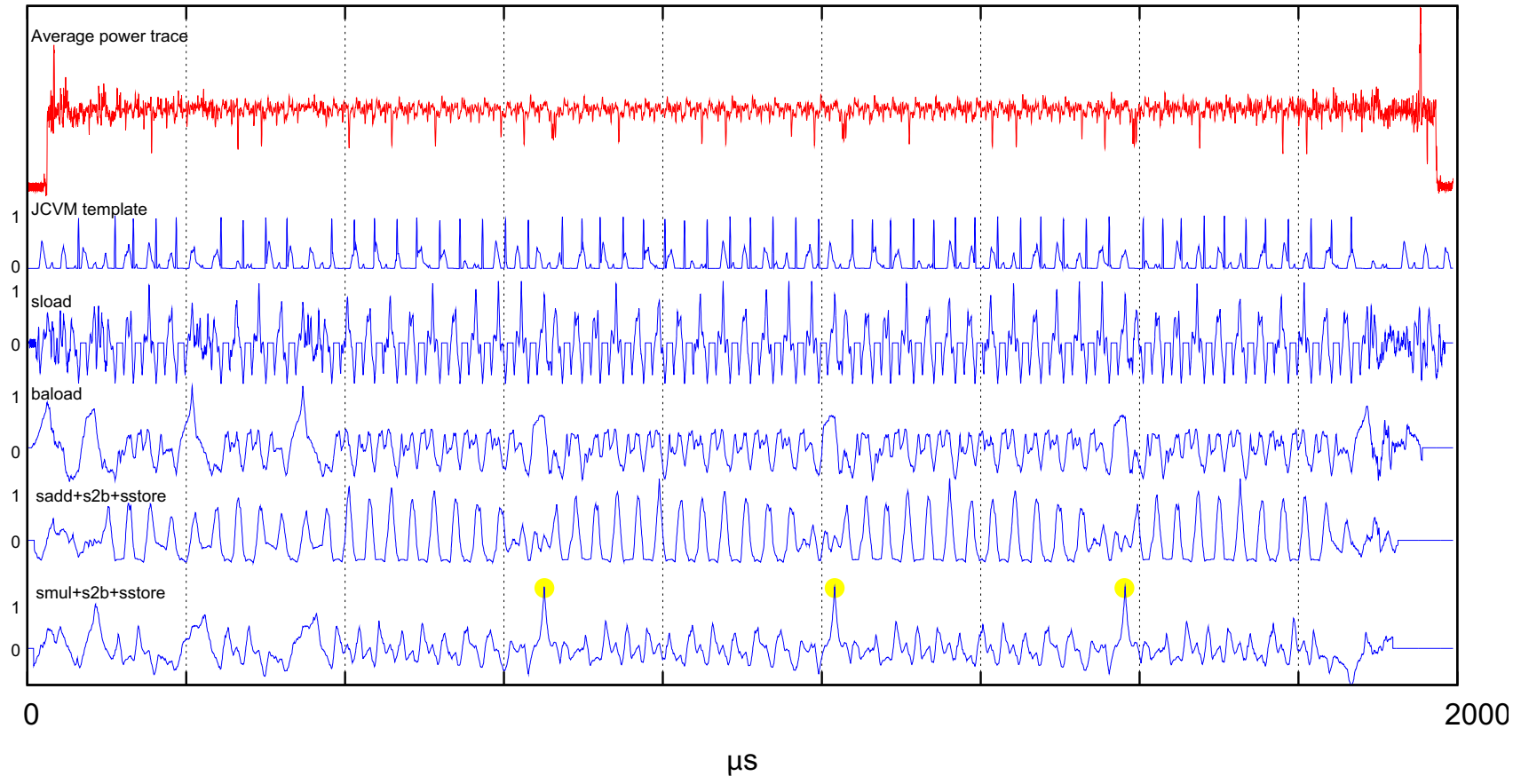
Template determination (4)



Template recognition

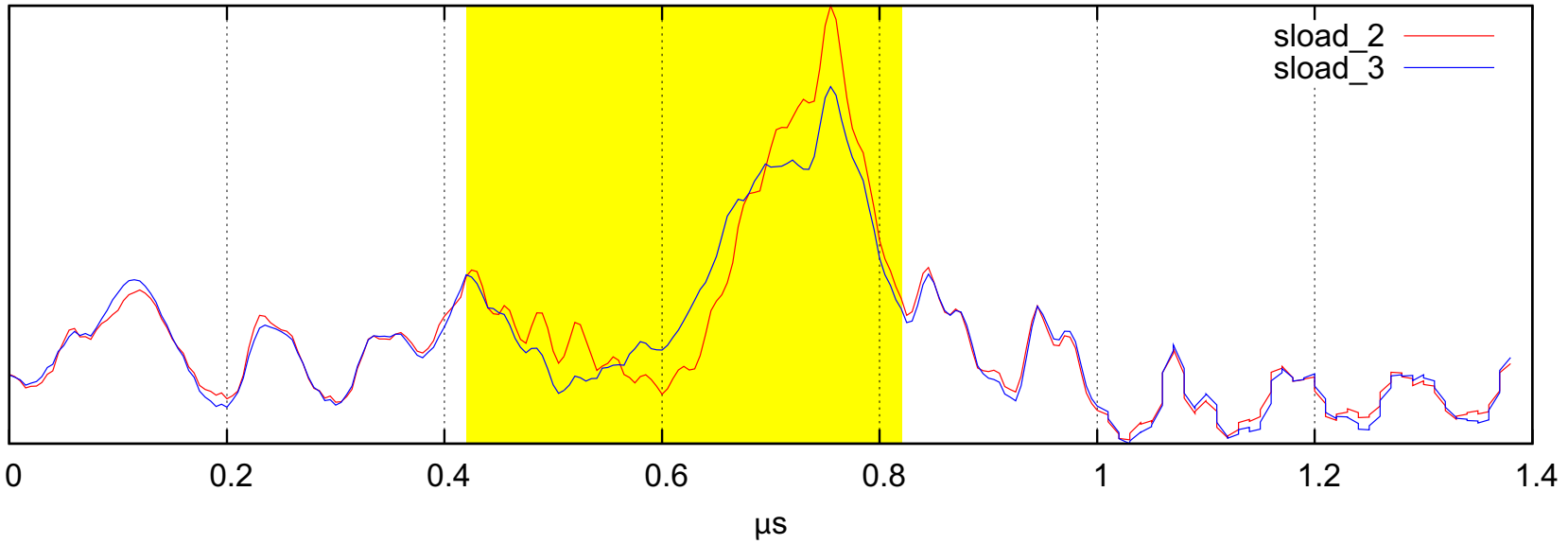
- Execute the (un)known applet
- Obtain power traces
- Preprocess power traces
- Correlate the power trace with all templates

Template recognition (2)

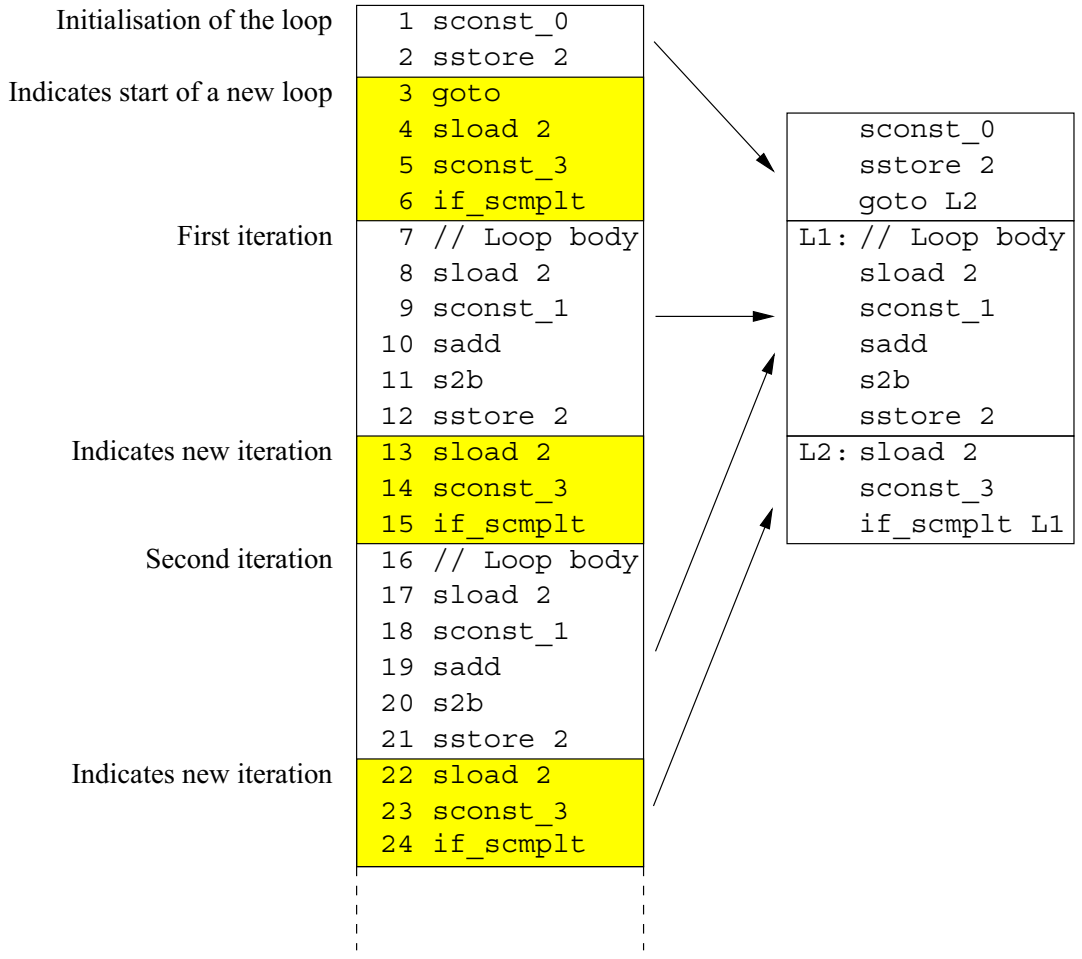


Distinguishing similar instructions

- 25,000 measurements at 200 MS/s:
  - 12,500 × `sload_2`
  - 12,500 × `sload_3`



Loop rerolling



Experimental results

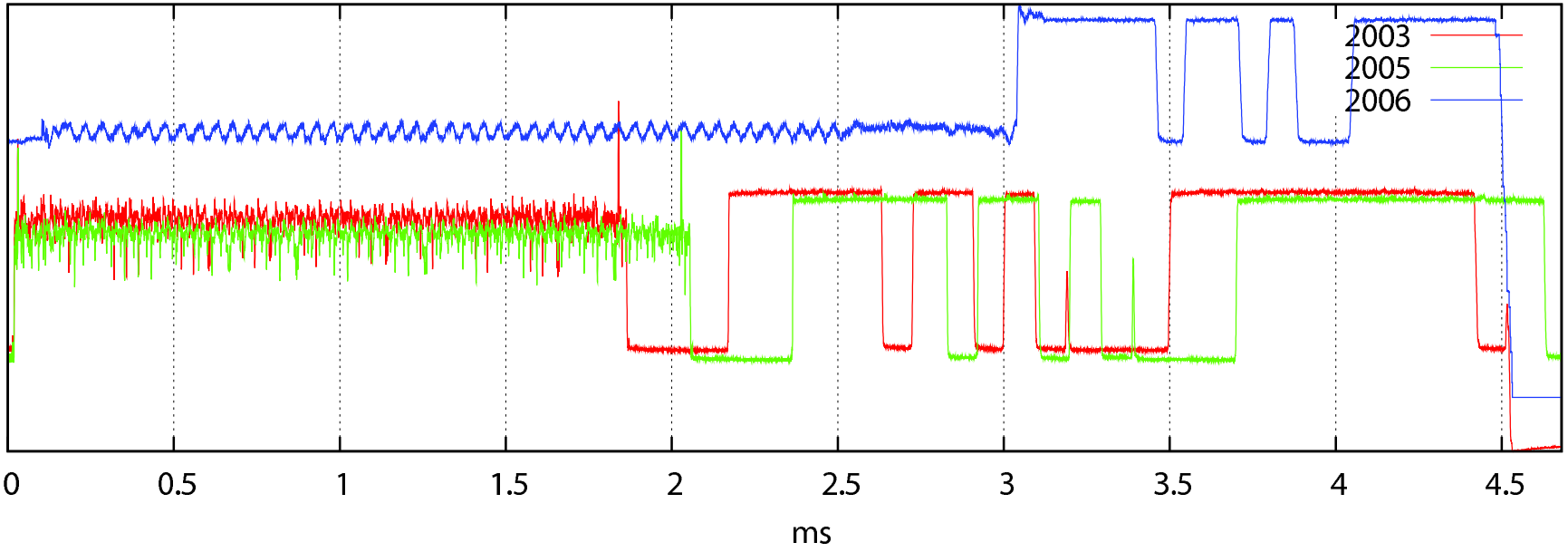
Expected	Recognised	Alternatives
`sload`	`sload` (93%) *JCVM*	`aload` (89%)
`sload`	`sload` (92%) *JCVM*	`aload` (91%), `sconst` & `sstore` (57%)
`sadd`	`sadd` (91%) *JCVM*	`sload` (55%), `aload` (51%)
`s2b` & `sstore`	`s2b` & `sstore` (91%) *JCVM*	`sload` (51%)
`sload`	`sload` (92%) *JCVM*	`aload` (78%), `sconst` & `sstore` (54%)
`sload`	`aload` (92%) *JCVM*	`sload` (91%)
`sadd`	`sadd` (90%) *JCVM*	`sload` (54%), `aload` (53%)
`s2b` & `sstore`	`s2b` & `sstore` (90%)	`sload` (53%)

Methodology refinements

- Impossible instruction sequences
  - `sload`, `aload`, `sadd`, `sstore`
- Unlikely instruction sequences
  - `sconst_0`, `sdiv`
  - `sload_2`, `sstore_2`
- Instruction duration
  - Branches
- Instruction statistics

Different Java Card smart cards

- Same brand/type
- Analysis focussed on: 2003, 2005
- Partial analysis: 2006



Conclusions

- Determining and recognizing individual instructions is possible
- Additional information sources reduce errors
- Reconstructing structured bytecode can be difficult
- Future work

Questions?

For more information:

- vermoen@riscure.com
- http://www.riscure.com