Fall-Through Semantics for Mitigating Timing-Based Side Channel Leaks

Aniket Mishra, Abhishek Bichhawat

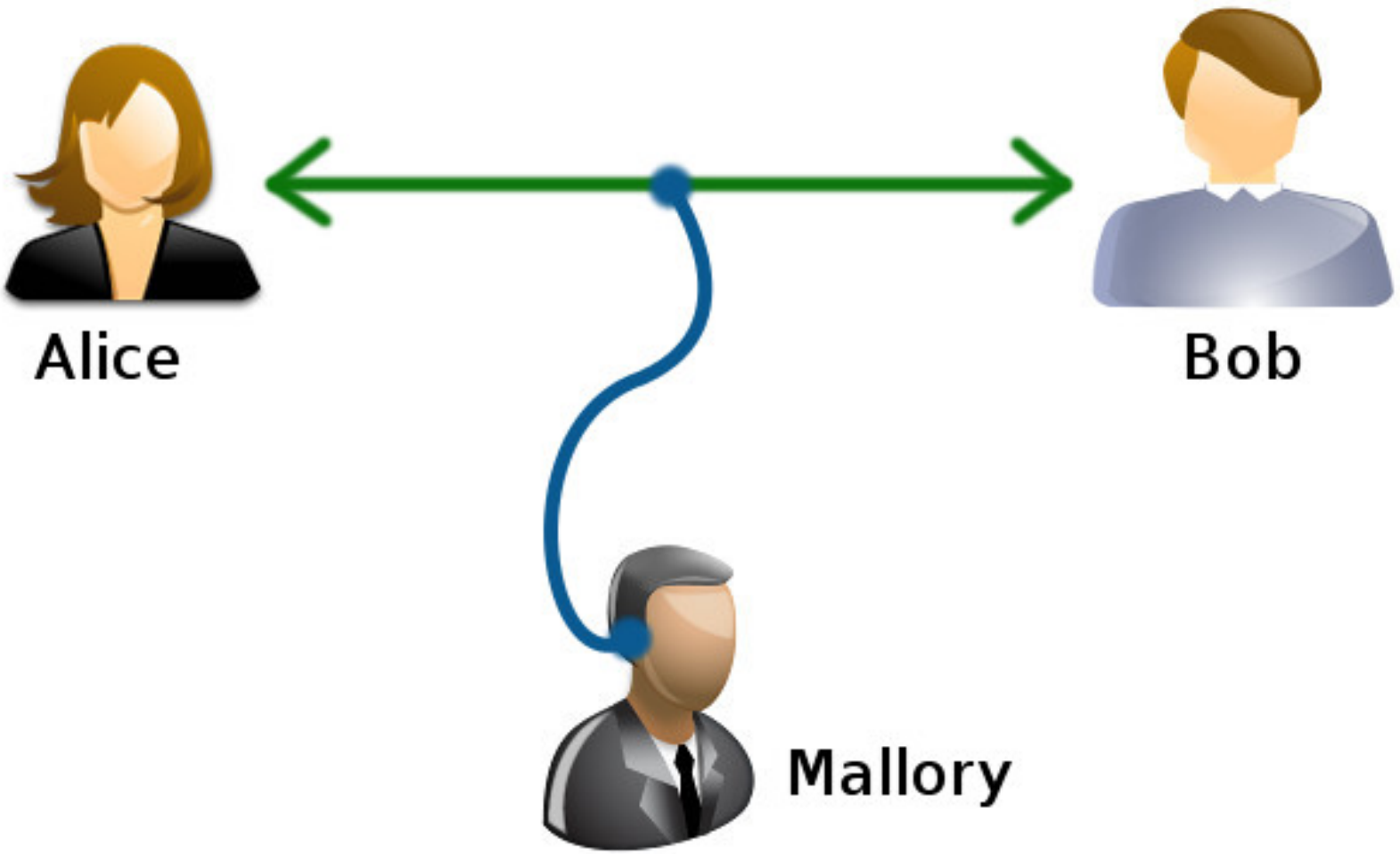
[2025-12-18 Thu]

Outline

1. Timing-Based Side Channel Leaks
2. Mitigating Timing-Based Side Channel Leaks
3. Fall-Through Semantics for Mitigating Timing-Based Side Channel leaks
4. (Formal) Fall-Through Semantics for Mitigating Timing-Based Side Channel Leaks
5. Methodology
6. Conclusions

Traditional Cybersecurity

Traditionally, cybersecurity has dealt with the analysis of overt channels.



What are Side Channels?

A *side* channel, on the other hand, involves no such direct access. Instead, vulnerabilities are introduced by side effects of a program's execution.

- Power used during the program's execution
- Sound generated by the machine running the program
- Time it takes for the program to execute

An Illustrative Example

Let us consider a C program that checks some input against a given password by comparing it by character by character.

```
bool matchpwd ( int * input , size_t n ) {
  if ( n != pwd_length ) return false;
  for ( int i = 0; i < n ; i ++) {
    if ( input [ i ] != pwd [ i ]) return false ;
  }
  return true ;
}
```

An Exploit

Let's say the password is 101010, an exploit may look like the following.

Example (An Attack Trace)

000000 ..... Rejected in 1st iteration
100000 ..... Rejected in 3rd iteration
110000 ..... Rejected in 2nd iteration
101000 ..... Rejected in 5th iteration
101100 ..... Rejected in 4th iteration
101010 ..... Password accepted!

The Reality of Timing Side Channels

Speculative execution and cache side channels are subtler ways that these vulnerabilities can be introduced.

A simple slide

This slide consists of some text with a number of bullet points:

- the first, very important, point!
- the previous point shows the use of bold emphasis which is translated to a `\alert{}` directive in LaTeX.

The above list could be numbered or any other type of list and may include sub-lists.

A more complex slide

This slide illustrates the use of Beamer blocks. The following text, with its own headline, is displayed in a block:

Theorem (Org mode increases productivity)

- *org mode means not having to remember LaTeXcommands.*
- *it is based on ASCII text which is inherently portable.*
- *Emacs!*

Two columns

- this slide consists of two columns
- the first (left) column has no heading and consists of text
- the second (right) column has an image and is enclosed in an example block

Example (A screenshot)

Babel

This slide shows some code and resulting output using Babel. Note the specification of `BEAMER_act` property for the second column.

Octave code

Babel

This slide shows some code and resulting output using Babel. Note the specification of `BEAMER_act` property for the second column.

Octave code

The output

Summary

- org is an incredible tool for time management
- but it is also excellent for writing and for preparing presentations
- Beamer is a very powerful LaTeX package for presentations
- the combination is unbeatable!