WE JUST CORRELATE

Timing Side-Channel Attack

Using linear correlation to reveal secrets

A. Anselmo, S.A. Chiaberto, F. Chiatante, G. Roggero

Supervisor: *Prof. Renaud Pacalet*

21st June, 2019

RSA

Outline

Introduction

- in several algorithms used for security purposes some optimizations are introduced
- these optimizations lead to a linear dependency between time and the data encrypted
- knowing information regarding the time-data pair, it is possible to find a correlation
- this correlation can be used to unveal part of the secret

Hypothesis

Tools needed

In order to successfully extract the secret through the correlation, we have to make a list of assumptions:

- timing for a sufficiently large number of cyphertexts is known
- cyphertexts are known
- secret is the same for all cyphertexts
- the HW/SW implementation is known to the attacker
- a timing model can be built

Titlepage settings

- by changing settings in

  `header_footer.sty`

  you can choose whether and where you want a second logo to be positioned on the titlepage:
  - small logo can be placed on the bottom right
  - big logo can be placed on the top right
- spaces and graphics dimensions will have to be adjusted depending on your logo

Outline

- divide the presentation, using the command `section` (as it is usually done in LaTeX)
- other divisions, just as chapter or part are not supported
- the sections are are listed on the top of each slide, the section the recent slide belongs to is highlighted
- you can automatically receive an outline out of this section by the command

  `\tableofcontents`

Itemize

- black circle is the default; other possibilities are:
  - ball
    - triangle
- the color of the items can also be changed
- all this settings have to be done in the preamble of the `presentation.tex` file

Overlays

Overlays

- its possible to build slides succesively

Overlays

- its possible to build slides succesively
- to do so use the command `onslide`

Overlays

- its possible to build slides succesively
- to do so use the command `onslide`
- other useful commands are `uncover` and `only`

Overlays

- its possible to build slides succesively
- to do so use the command `onslide`
- other useful commands are `uncover` and `only`
- this works also very nice to "develop" formulas:

Overlays

- its possible to build slides succesively
- to do so use the command `onslide`
- other useful commands are `uncover` and `only`
- this works also very nice to "develop" formulas:

$$f(x \mid \mu, \sigma^2) \; =$$

Overlays

- its possible to build slides succesively
- to do so use the command `onslide`
- other useful commands are `uncover` and `only`
- this works also very nice to "develop" formulas:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}$$

Overlays

- its possible to build slides succesively
- to do so use the command `onslide`
- other useful commands are `uncover` and `only`
- this works also very nice to "develop" formulas:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left\{ \right\}$$

Overlays

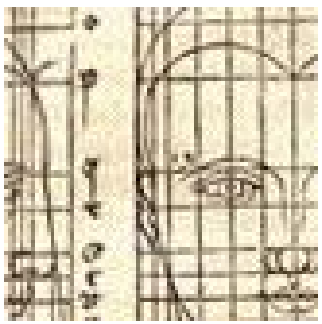
- its possible to build slides succesively
- to do so use the command `onslide`
- other useful commands are `uncover` and `only`
- this works also very nice to "develop" formulas:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}$$

Pimp up your presentation

- an easy way to include pictures is by using

  `\includegraphics[width=...,height=...]{file}`
- in connection with `pdflatex` this supports a wider range of graphic formats, including GIF, PNG, JPG



Useful hints

- if you use a verbatim environment on a slide, declare that slide `fragile`:

  `\begin{frame}[fragile]`
- bibliography actually works as usual, just keep in mind that not all bibliography styles are supported by the *beamer* package, maybe you have to include some other packages to get your preferred style working

Possible solution

Blinding

Test

linding the messages before encryption.

References

Bansal, M., Kumar, A., Devrari, A., Bhat, A., UTU, D., and Dehradun, U. (2015). Implementation of modular exponentiation using montgomery algorithms. *International Journal of Scientific & Engineering Research*, 6(11):1272–1277.

Crockett, L. H., Elliot, R. A., Enderwitz, M. A., and Stewart, R. W. (2014). *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. Strathclyde Academic Media.

Kocher, P. C. (1996). Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Annual International Cryptology Conference*, pages 104–113. Springer.

Walter, C. D. (1999). Montgomery exponentiation needs no final subtractions. *Electronics letters*, 35(21):1831–1832.

Xilinx (2015). *Zynq-7000 All Programmable SoC Software Developers Guide*. Xilinx.