Software Design for Cryptography

Francesco Regazzoni

How do you develop software

(Simplified!)

- Specification (Platform selection)
- Writing using a programming language
- Compile
- Test

Platform Selection

- Know your platform **very** well
- How many bits (8, 32, ...)
- Which instructions do you have?

The Shift Operation

- How many cycles for a shift?
- How many cycles for a rotate?
- With a shifter?

Computations vs Look-up-tables

- S-box as look-up-table
- All the round functions (except key addition) as look-up-tables

Data Storage

- Transposed state
- Change Mix-Column, higher performance

At which level?

- High level “portable”, fast developing time
- Low level, higher control, more difficult to develop

Benchmarking?

- Fair comparison
- Reproducible results

What do you measure?

- Clock Cycles
- Memory Occupation
- Power
- Energy

The role of the compiler

- Compilers are **NOT** designed for security
- Compilers can introduce security hazards
- Compilers are a nice place to automatically increase the security

Instruction Set Extension

- Add new instruction to the instruction set
- More flexible than hardware accelerators
- AES-NI

Which instructions?

- Profile
- Trade off with area/critical path

Designing the algorithm on the architecture

Questions?