Efficiency of AES implementations

Carsten Baum & Tyge Tiessen

1 Background

AES, the Advanced Encryption Standard, is one of the most used (if not the most used) current cryptographic algorithm. It has been designed from the ground up to be efficiently implementable in software. Nonetheless, due to it's widespread use it has become common for CPU to have dedicated hardware instructions (AES-NI) that allow for an even faster implementation of AES.

In this project, you will implement three versions of AES and determine their respective performance profiles.

2 Tasks

1. Find a description of AES-128 (e.g. "The Design of Rijndael") that you can use for your implementation. Implement a naive version of AES and test it test vectors as you can also find in "The Design of Rijndael").
2. Find an explanation of the optimization called "T-tables" for AES. Implement a second version of AES-128 using T-tables.
3. Determine how to utilize AES-NI instructions for an implementation of AES. Implement a third version of AES using AES-NI.
4. Benchmark your three implementations and compare them with each other. Also benchmark an implementation of AES from a reputable library and compare that to your implementations.

Note: To be able to utilized AES-NI instructions, it is important that you choose a programming language that is able to support it, for example C, C++ or Rust.

3 Deliverables

Your hand-in should consist of the following two deliverables:

1. A 10 page (maximum) report with a high-level description of AES, an explanation of the T-table optimization, a short introduction to AES-NI and a discussion of the results of your benchmarks.

2. A .zip-file containing the code which you wrote as well as the test cases. Your code must compile on a machine running Ubuntu 24.04.2 LTS. Please also deliver a README file that has the following 2 sections:

   Compilation and Installation: What are the prerequisites (installed libraries and versions etc.) to compile and run your code, which commands should be used to compile the code, how is the code installed?

   Running the test cases: Describe which commands should be used to re-run your test cases. How should the outputs of your tests be interpreted by a reader?

   Please test if the instructions in the README file work in the described environment before handing in. Non-functioning code will impact your overall grade.