Advisor	Bas Spitters & Diego Aranha
Students	Kasper S. Nielsen & Mathias Rud Laursen
Languages	English
Text tools	LaTeX
Other tools	Visual Studio Code, potentially a proof system (like Coq or Dafny)

Project Description

In this project, we will study the theory behind elliptic curve cryptography, through reading and understanding scientific papers. We will familiarize ourselves with Rust and write a reference implementation of a number of key cryptographic primitives in a safe subset of rust, using the Hacspec library. We will use the knowledge of our theoretical studies, to implement our primitives, and hereafter, using semi-automatic or interactive tools prove that the program satisfies the implementation.
After having implemented a simple implementation of the primitives, we will try and analyse this - optimizing the initial implementation while preserving the semantics - and afterwards proving that the semantics actually were preserved.

After this, we will compile our works into a report, summarizing the results of the work, comparing the works to other approaches, and concluding on the project.

Provisional Table of Contents

- Abstract (10-20 lines)
- Section 1: Introduction (1-2 pages)
- Section 2: Review of literature (4-8 pages)
- Section 3: Description of Task A (4-8 pages)
- Section 4: Description of Task B (4-8 pages)
- Section 5: Description of Task C (4-8 pages)
- Section 6: Comparison to other work and ideas for future work (2-4 pages)
- Section 7: Conclusion (1-2 pages)
- Acknowledgements (3-5 lines)
- References ($\frac{1}{2}$-1 page)
- Appendix with programming code, tables, full proofs, etc. (5-20 pages)

Review of literature: Most likely some BLS theories and potentially implementations.
Tasks: Implementing the BLS in Hacspec, proving correctness of the implementation, optimizing the code, while preserving semantics, and proving this.

Provisional Time Plan

First weeks of February (20 hours)
Planning of activities, including the production of the Bachelor's contract. Familiarizing ourselves with Rust programming language and the Hacspec library. Setting our workspace up.

Rest of February and first half of March (3×15 hours)
Read literature (one or more scientific papers) and make draft of Section 2 in Bachelor's report. Analyse the literature, and starting on an implementation of BLS

Rest of March and first week of April ($2 \times 15 + 2 \times 30$ hours)
Finishing the initial implementation and working towards proving this

Rest of April (3×30 hours)
Completion of task B and make draft of Section 4 in Bachelor's report.

First three weeks of May (3×30 hours)
Completion of task C and make draft of Section 2 in Bachelor's report.

Last week of April of first half of June (3×30 hours)
Write the missing parts, put drafts together, make things consistent, proof reading.

Note: This time table is heavily dependant on how far we actually get, and how complex the material and code work is. It may vary, if our predictions are far off!