

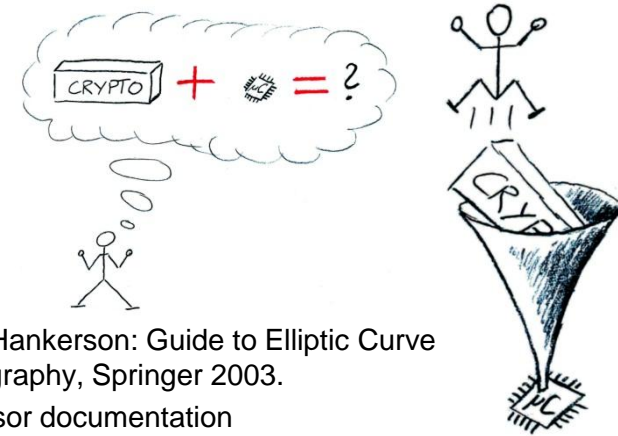
# Optimize Elliptic Curve Cryptography (ECC)

## Motivation

We are currently implementing an Elliptic Curve Library for embedded processors. It is written in C. It comes with test cases and already functioning algorithms. By implementing performance critical algorithms in assembler, the performance can be improved greatly.

## Project description

- Goals
  - Speed up the given C – Implementation
    - By replacing performance critical functions with assembler code,
    - Or rewriting the C – code.
- Tasks
  - Understand the structure of ECC
    - Learn about the difference of field and point operations.
    - Learn about big number and modular arithmetic.
  - Learn about the internal structure of embedded processors.
    - Get some insight in the instruction set and the register set of embedded processors.
  - Do some optimization.
- Possible platforms:
  - 32-bit AVR UC3
  - x86, x64, GPU
  - ...



## Literature

- Darrel Hankerson: Guide to Elliptic Curve Cryptography, Springer 2003.
- Processor documentation

## Deliverables

- Project files (.zip, cleaned)
- Documentation (inline)
- Readme (getting started)
- Presentation (10 .ppt slides)

## Project schedule

- Start Immediately
- Month 1 Reading, understanding ECC
- Month 2 Doing some optimizations
- Month 3 Final deliverables

## Master Project

Studies:  INF  SEW  TEL

## Prerequisites

- Basics in embedded processor programming.

## Gained know-how

- Elliptic Curve Cryptography
- Embedded C programming
- Processor architecture.

## Advisor / contact

[Erich.Wenger@iaik.tugraz.at](mailto:Erich.Wenger@iaik.tugraz.at)