Project 1
Lightweight AES

Project Due: Wednesday, October 21, 23:59

Compulsory Project

The goal of this project is to develop a deep understanding of constraints persisting in real-world embedded systems and of techniques available in lightweight cryptography to cope with those. As a prominent example from cryptographic practice, we take the U.S. Advanced Encryption Standard, more specifically, AES-128 (with 128-bit block and 128-bit key) — referred to simply as AES in the following. AES is arguably the most widely used cryptographic algorithm today. AES consists of 10 rounds of data transform and a key expansion from a 128-bit master key to 11 128-bit round subkeys. Note that your AES implementation in Project 1 should include the key expansion.

As a standard embedded platform, we choose AVR ATmega16 by Atmel featuring a RISC architecture. ATmega microcontrollers are widely used in the field, with applications ranging from access control over sensor networks to medical devices. Moreover, their architecture is similar to microcontrollers used in some high-security smart cards. ATmega16 has a RAM (SRAM) of 1 KByte and a code memory (flash) of 16 KBytes. As a development environment, AVR Studio 6 is recommended. Note that it is based on MS Visual Studio 2010. So your possible pre-knowledge of Visual Studio may come in handy.

Three optimization goals we have are 1) low RAM consumption, 2) small code, and 3) low execution time. Note that these are *distinct* and sometimes *conflicting* goals! So one has to perform a separate implementation for each of these optimization goals since one is likely to take different approaches in the three different optimizations. You are free to choose your own approaches and tricks to optimize your lightweight AES implementation for each of the goals.

Along with your code, you are also asked to hand in a report briefly documenting your code and implementation choices made.

To summarize, your hand-in should include:

1. C source code for lightweight AES optimized for low RAM consumption on ATmega16,
2. C source code for lightweight AES optimized for small code size on ATmega16,
3. C source code for lightweight AES optimized for execution time on ATmega16, and
4. Short report in PDF format containing a brief documentation of your three pieces of code listed above and the implementation choices made. Your report should also include the code size (program memory in AVR Studio), RAM consumption (data memory in AVR Studio), and the number of needed clock cycles (in AVR Studio, use the simulator for counting cycles) for each of your AES optimizations.

Group work is strongly encouraged in Project 1 but please make sure to hand in individually. Note that we will have time slots in class dedicated to your work on the project where you can get consultation.

Reference material: Chapter 3 from "The Block Cipher Companion" by Lars Knudsen and Matt Robshaw and pages from "The Design of Rijndael" by Joan Daemen and Vincent Rijmen available on CampusNet. Please also refer to the AVR Studio help and Atmel documents.