# ICR 2014-2015

# Practical Work #2

# *Designing and Implementing a Secure Container for Medical Data*

## Rules of the Game

This practical work can be performed by *groups of two*, or *individually.* A written report must be delivered, that will contain your answers to the questions, all the code you wrote (that must be commented), an introduction, a conclusion, and if necessary, screenshots. The report can be written in French, German, or English, and it must be delivered as a PDF file named

`lab02_report_Lastname(s).pdf`.

The code must be delivered as a ZIP file named

`lab02_code_Lastname(s).zip`.

The two files must be sent by electronic mail to

`pascal.junod@heig-vd.ch`

before

**Monday December 15th 2014, 18h00 CET.**

Being up to one day late will cost you one grade point, from one day to two days will cost two grade points, etc. Please do not forget to cite all your sources in a clear and precise manner!

# 1 Preliminaries

A university hospital has mandated you to design and implement a secure archival software system for documents containing medical data. Legal regulations require that those data are kept in a highly secure way. In the following, we will focus on confidentiality, authenticity and integrity of the archived data, and we will assume that their availability is handled by separate mechanisms.

In order to simplify the scenario, a further assumption that can be taken is that these documents containing sensitive medical data are stored in flat (binary) files, that can however have an arbitrary bytes length. Furthermore, one can assume that the unprotected files have been compressed.

To implement this secure archival system, you can freely select the cryptographic library (e.g., Botan, OpenSSL, PolarSSL, etc.) and the programming language.

# 2 Security Requirements

The first step consists in defining the security requirements of the system, as well as identifying the proper cryptographic primitives that will be used to enforce these security requirements.

**Question 1.**

1. What is the overall security level that you are targeting?
2. How can you ensure that the stored data will be kept in a confidential way? What about the file metadata?
3. How can you ensure that the stored data are authentic? What is the risk addressed?
4. How can you ensure that the stored data keep their integrity?
5. What are the cryptographic keys required to be managed in your system?

**Task 1.**

Define formally the data format of a secure container for sensitive data allowing to fulfill all the previously defined security requirements.

## 3 Implementation and Tests

This part is dedicated to the implementation of the secure container.

**Task 2.**

Define and implement an API taking a file in input and returning a secure container stored in a flat file.

**Task 3.**

Define and implement an API taking a secure container in input, checking its integrity and authenticity and returning the original data, if the previous checks are successful.

**Task 4.**

Define and implement a minimal test suite ensuring that your API is working properly.

**Question 2.**

1. What are the critical parts in your code in terms of security?
2. How did you ensure that these critical parts are properly implemented?
3. What are the remaining weak points in your implementation?
4. How do you propose to address the remaining weak points?