Projects for the Course on Proofs for Cryptography

Tamara Rezk

December 8, 2016

This document defines the projects and evaluation rules for the projects that you have to implement as part of the course "Proofs for Cryptography". In case of doubts, please contact me as soon as possible for clarifications.

1 Project 1 Description

1. You should implement a program problang that takes as input a program P written in the probabilistic programming language while given during the course. The program problang with input P should give as output the final output distribution of P with initial distribution as given in the tp.

2. Using problang, you should write a program ElGamalprover that takes as input a program P and tries to generate a proof of CPA when this program is ElGamal or a partial proof if the program is equal to ElGamal up to certain point.

The programs can be implemented in Java, C, javascript, or php+javaScript.

You should write a report of max 10 pages, Report 1, describing implementation details and giving examples.

The clarity and clearness of the report will be taken into account during evaluation.

1.1 Presentation

The group will be granted 10 minutes (strict) to present the project. After this presentation, all the members should answer questions during 20 minutes about the topic of the project.

1.2 Style

No extra points are given for style and simplicity of the code will be highly appreciated.‘

2 Project 2 Description: Study of Cryptographic Library

This project requires to do research in the topic of proofs of cryptography. You need to choose a well known library providing cryptographic functionalities. For each of them, you should look in scholar.google.com (articles can be usually obtained by free by visiting the authors'page) and find out if there exists a proof of a security property. You should write a report of max 30 pages, Report 2, describing for each cryptographic functionality in the library which are the properties that have been proved and the article where these proofs can be found.

3 Organization

3.1 Groups

Project 1 should be done by a group of 5 students. Project 2 should be done by a group of 7 students. The groups have to be fixed since 10/1 by sending me an email with complete names of the members of the group.

3.2 Originality

All the code in the project has to be written by the students of the group.

3.3 Timeline

- 7/2: Project 1 Presentation and questions. Provide code and Report 1 to teacher.
- 26/2: Project 2 Report 2
- 10/3: Corrections to Report 2 if need be.

4 Evaluation

4.1 Project 1

It is possible to obtain a total number of 12 points per member of the group with the project. Some of the points are granted to the group and some of them only to specific members (in particular, the questions are evaluated individually). The distribution of the points is as follows:

Positive Points You can obtain positive point:

- Implementation : 7 points max

- Presentation (group): 3 points max
- Questions (individual): 2 points max

Negative points It is also possible to obtain negative points.

- Unanswered questions during presentation (individual): -2 points max
- Delays: -12 points max, see detail below.

4.2 Project 2

You can obtain 10 points max.

5 Delays

In case you delay your submissions, at any stage, the following rules apply depending on the amount of the delay:

- 0-24 hours: 1 points less
- 1-3 days: 2 points less
- 3-7 days: 5 points less
- +7 days: mark is 0.

In case of absence (and absence of a formal justification), the day of the presentation, -5 points are individually withdrawn to the absent member.