Project 3: Topic 1

Forging certificates

Luca Campa

Department of Computer Science
Universität Innsbruck

June 4, 2025

Topics

- Introduction
- Task Description
- Learning Objective
- Dates
- Evaluation

Introduction

In this project I'm asking you to generate valid certificates for your own website without having the private key.

Task description

Basically,

- I generated my own public-private key couple and I used it to generate (consecutively) two signed digital certificates for my two websites (fictional website, they don't exist).
- There will be three cases. More specifically, I'll provide you the code I used to generate the certificates for each case. **Your objective is to find the weaknesses and to forge a valid certificate for your website.**
- (OPTIONAL +2 points) Implement your own code for certificate generation (without using external tools, except for cryptographic functions).

"**Valid** certificates" means that I can verify them with my public key (provided in the folder).

Task description

Your project must be structures in the following way:

- it must contain a `src` folder where the tool will be located
- it must contain a `README.md` file that contains the detailed usage instructions with a detailed description of the attack. If you need to write equations, use Latex (even within Markdown).

Learning objective

- Understanding potential weaknesses in certificates generation.
- Understanding the reason under standard implementation choices.
- Exploit wrong implementations.
- Learn the well known derivation of ElGamal: DSA (https://en.wikipedia.org/wiki/Digital_Signature_Algorithm)

DSA

Environment

Public knowledge: p, q, g, H where p, q are prime numbers and $p = qz + 1$ for some $z \in \mathbb{N}$, g is a generator of F_p and H is a secure hash function.

Sign

- Choose $sk : x_a \in [1, ..., q-1]$ randomly
- Compute $pk : y_a = g^{x_a} \mod p$
- Send public key to the environment
- Choose message m and compute $H(m)$
- Choose $k \in [1, q-1]$ randomly
- Compute $r = (g^k \mod p) \mod q$
- Compute $s = k^{-1}(H(m) + x_a r) \mod q$
- Send (r, s)

Verify

- Receive (r, s) from Alice
- Retrieve y_a from the environment
- Compute $u_1 = H(m)s^{-1} \mod q$
- Compute $u_2 = rs^{-1} \mod q$
- Compute $v = ((g^{u_1} y_a^{u_2}) \mod p) \mod q$
- Check $v == r$

Dates

- **Deadline for submission**: 24 June at 23:59
- **Oral presentation**: 26 June 2025 (After the Quiz)

Evaluation

The project will count as the 20% of the final grade, in particular:

- 10% Implementation:
  - 2% if the code is running
  - 8% if the code returns correct results on all the test inputs (we will provide you with some of them, but not all)
- 10% Oral presentation: every member of the group must explain one of the parts of the project.

Bonus points (OPTIONAL items) will be taken into account for the final mark of the project. If you obtain more than 20 points, the additional points will be taken into account for the final mark of the course.
It is suggested, but not mandatory, to do the project in groups of **maximum 2** students.