CAA 23-24

Lab #2

01-11-2023

Introduction

In this lab we are going to study three wrong Python implementations of EdDSA.[1] The basic structure of the code follows the Python implementation in RFC8033.[2] Note that this RFC implementation should **never** be used in production. Obviously, some modification were made to make them vulnerable. You will have to submit your code and a small report on cyberlearn. To access the servers hosting the challenges, you will need the school's VPN.

To simplify your task, the vulnerabilities are never in the mathematical part of the implementation. Hence, they never occur above line 319.

1 Challenge 1

A colleague of yours (not sure it's an IL this time given how they perform in SLH) decided to make his own EdDSA implementation, following the RFC. You will find the code in the `chall1` folder. You will also find in this folder a public key as well as the signature of the message "Grade of Alexandre Duc at CAA = 6.0".

1.1. What is wrong in the implementation?

1.2. Forge a signature of the message "My grade in CAA is 6.0".

1.3. Explain your attack in your report.

2 Challenge 2

After this first attack, your colleague decided to reimplement from scratch EdDSA (still following the RFC). This time, a website (`http://10.190.133.22:9005`) is available to you on which you can sign any message except the message "My grade in CAA is 6.0". You will find the new version of the code in the `chall2` folder. You will also find in this folder the public key used by the system.

2.1. What is wrong in the implementation?

2.2. Forge a signature of the message "My grade in CAA is 6.0".

2.3. Explain your attack in your report.

[1] We don't do it in Sage as Sage does not support twisted Edwards Curves.

[2] https://datatracker.ietf.org/doc/html/rfc8032

3 Challenge 3

You colleague decided now to enhance the EdDSA algorithm to add a timestamp to it. The idea is that when you sign a message, the date at which the message was signed is also authenticated (and returned). When you verify the signature, you also have to provide the signing date. Again, a website (`http://10.190.133.22:9006`) is available to you on which you can sign any message except the message "My grade in CAA is 6.0". You will find the new version of the code in the `chall3` folder. You will also find in this folder the public key used by the system.

3.1. What is wrong in the implementation?

3.2. Forge a signature of the message "My grade in CAA is 6.0".

3.3. Explain your attack in your report.

3.4. Fix the implementation so that the timestamping works but so that the code is not vulnerable. Explain your fix in your report.