MysteryTwister C3

THE CRYPTO CHALLENGE CONTEST

Elliptic Boogaloo – Part 2

Author: newton

January 2024

Introduction

In this challenge, we take a closer look at elliptical curves. In the file `app.py`, a small program is given which creates 10 signatures on the curve NIST P-256 [1].
An explanation of how ECDSA works can be found on Wikipedia [2].
For this challenge, a cryptanalytic lattice attack [3] must be performed on ECDSA.

Challenge (1/2)

The attached file `signatures.txt` contains the 10 signatures that were created using `app.py`.
The program library `ecdsa` [4] required to run `app.py` can be installed using `pip install ecdsa`.

Challenge (2/2)

The challenge is to extract the private key `secret` and sign the following string from `plaintext.txt`:

```
Yay!  MysteryTwister Heureka!  Again!
```

The signature must be submitted in the form `"r,s"`, where `r` and `s` (separated only by commas) are each to be understood as decimal numbers.
In this second part of the Elliptic Boogaloo series, the signed messages were additionally tweaked with a random value, among other surprises. Have fun!

Resources

1. Mathematical analysis of the P-256 curve:
   neuromancer.sk/std/nist/P-256
2. Wikipedia article about ECDSA:
   en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm
3. Lattice Attacks on Digital Signature Schemes,
   Howgrave-Graham et al. (2001):
   doi.org/10.1023/A:1011214926272
4. ECDSA program library: github.com/tlsfuzzer/python-ecdsa

Additional Files

- → `app.py`: The source code used to create the signatures.
- → `signatures.txt`: The 10 signatures that were created with the program.
- → `plaintext.txt`: The text to sign.