MysteryTwister C3

THE CRYPTO CHALLENGE CONTEST

Elliptic Boogaloo – Part 3

Author: newton

March 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 5 signatures on the curve brainpoolP320r1 [1, 6].
An explanation of how ECDSA works can be found on Wikipedia and in *Cryptography and Network Security*, chapter 13.5 [2, 7].
For this challenge, an **advanced** cryptanalytic lattice attack [3] must be performed on ECDSA.

Challenge (1/2)

The attached file `signatures.txt` contains the 5 signatures that were created using `app.py`.
The program libraries `ecdsa` and `pycryptodome` [4, 5] are required to run `app.py`. They can be installed using `pip install ecdsa pycryptodome`.

Challenge (2/2)

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

```
Solved! MysteryTwister! The Boogaloo Is Broken!
```

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 third part of the Elliptic Boogaloo series, the signed messages were even secured with two nonces.
Is it still possible to extract the private key?

Resources (1/2)

1. Mathematical overview of the brainpoolP320r1 curve: neuromancer.sk/std/brainpool/brainpoolP320r1
2. Wikipedia article about ECDSA: en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm
3. Howgrave-Graham, N.A., Smart, N.P. Lattice Attacks on Digital Signature Schemes. Designs, Codes and Cryptography 23, 283-290 (2001): doi.org/10.1023/A:1011214926272
4. ECDSA program library: github.com/tlsfuzzer/python-ecdsa

Resources (2/2)

5. PyCryptodome program library: pycryptodome.readthedocs.io
6. RFC 5639: Elliptic Curve Cryptography (ECC) Brainpool Standard – Curves and Curve Generation: datatracker.ietf.org/doc/html/rfc5639
7. W. Stallings, Cryptography and Network Security: Principles and Practice, 6th ed. USA: Prentice Hall Press, 2013: dl.acm.org/doi/10.5555/2523199

Additional Files

- → `app.py`: The source code used to create the signatures.
- → `signatures.txt`: The 5 signatures that were created with the program.
- → `plaintext.txt`: The text to sign.