Re-Encryption Mix-Net Module

Vitalis Salis

2017

Zeus

- Web-based open-audit e-voting system.
- Open source.[1]
- Derived from Helios[2].
- Uses the Sako-Kilian re-encryption mix-net for anonymity.
- Already used by various institutions for elections.

[1] https://github.com/grnet/zeus

[2] https://github.com/benadida/helios

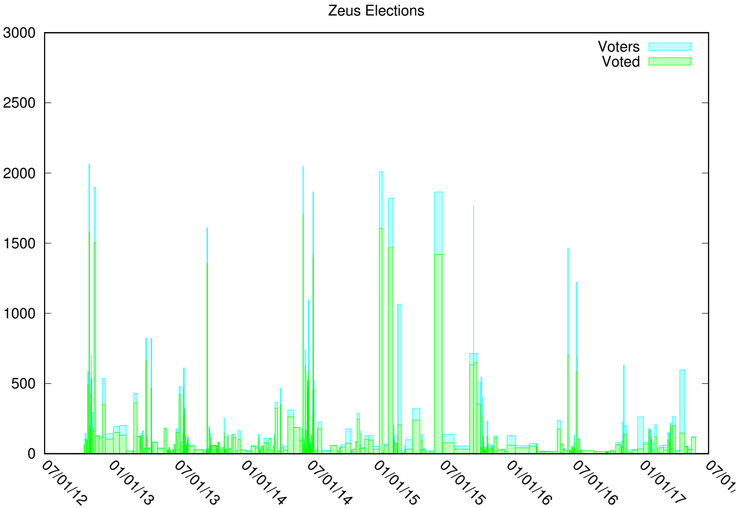

Figure: Registered and actual voters on Zeus.

The Issue

- The re-encryption mix-net used by Zeus is impractical.
- It requires a lot of costly, performance wise, cryptographic. operations, leading to longer times to get the election results.
- I.e for 10,000 votes the mixnet might take up to 8 hours!
- Our goal is to create an open source Python module that implements a faster re-encryption mix-net for applications requiring anonymity.

Faster Mix-Nets

- In order to overcome this issue, we've been looking on new research about mix-nets that guarantee faster performance.
- The best candidate we identified is proposed by Fauzi et al, from the University of Tartu.[3].
- The mix-net is based on elliptic curves.

[3] https://eprint.iacr.org/2016/866

Existing Prototypes

- A Python prototype that implements the mix-net proposed by Fauzi et al, was developed by GRNET[4].
- Still, the prototype wasn't satisfying.
- The main issue we identified was that multiplications on the elliptic curve structure are slow.
- The library implementing those multiplications is OpenSSL.
- A good replacement for OpenSSL is a similar library, libff.[5]

[4] https://github.com/grnet/ac16/

[5] https://github.com/scipr-lab/libff

Metrics

- In order to compare these libraries we have defined specific metrics.
- Our profiling involved a test case where we performed thousands of multiplications from C on both libraries: g^ρ where g is the generator of the elliptic curve group and ρ is a 256 bit number.
- libff yielded up to 6 times better performance than OpenSSL.
- So, we moved forward with the implementation of a libff wrapper for Python.

Wrapping libff With Cython

- libff is implemented in C++.
- So it needs to be wrapped by Python in order to be used as a Python module.
- No such wrapper exists, so we set out to create one.
- We identified that Cython is the best candidate for wrapping libff.
- The wrapper exists as a separate open source module so it can be used by other Python projects that need to use libff.

Comparing Wrappers

- After creating the Cython wrapper for libff, in order to verify that it is indeed better than the Python wrapper for OpenSSL, we defined specific metrics.
- Our profiling involved a test case where we performed thousands of multiplications from Python on both wrappers.
- The results validated our hypothesis, so we'll use the Cython wrapper for the implementation of the re-encryption mix-net module.

Future Work

- Python Module
- Integration with Zeus
- Testing

https://github.com/eellak/gsoc17module-zeus