Cryptography

Lab 7 – 17 V

Problem 1 (10 points) Implement a command-line Helios voting [1] client.

Your implementation should output:

- an encryption of a ballot,
- (together with) zero-knowledge proof of knowledge of exponent (randomness used).
- commitment to randomness.

Follow technical description which you can find on the website [2]. The output of your program needs to be in json format and be compatible with the Helios implementation *i.e.*, one should be able to use Helios verifier (`https://github.com/benadida/helios-server/blob/master/heliosverifier/verify.html`) to audit correctness of your ballot encryption.

Problem 2 (10 points) Implement a command-line Helios' verifier which takes as an input an encrypted ballot (in json format) and checks correctness of proofs attached to it.

References

[1] Ben Adida. Helios: Web-based open-audit voting. In *USENIX Security Symposium*, volume 17, pages 335–348, 2008.

[2] Ben Adida. http://documentation.heliosvoting.org/verification-specs/helios-v4, 2012.