Final projects

1. Administrative stuff

Projects are done individually, and each topic can be done by only one person. Topics are assigned on a first come first serve basis. You should select a topic, email me at `domagojvrgoc@gmail.com` to confirm that the topic is still available.

The outcome of the project will be:

- A written report on what you did. This can include the code in case you implement something. The report needs to be written on a computer, and hand in .pdf format.
- A 30 minute presentation.

The deadline for handing in your work is on the day of the presentation, which will be held on February 22, at 11:00.

2. FAQ

Do I need to hand in the work we do?

Yes, you need to write a short report detailing your findings. If you did some implementation, a link to the repository containing it should be provided.

What should be the structure of my presentation?

You have 30 minutes, so there is sufficient time to quickly explain the setting and background, and then detail the specific problem that is tackled, and how is this resolved.

I have a size 9 font in our presentation so that all the information fits on the slides. Is this OK?

No! An important part of the evaluation is your ability to summarize the work you did in a short presentation. This is no easy task. In particular, a good short presentation requires better understanding of the topic than a bad long presentation. In this course you will need to be able to isolate the important details of your work and present it in an accessible way.

I don't like any of the proposed topics.

Propose a new topic. For this, send me an email (domagojvrgoc@gmail.com).

3. Projects

Bellow follows a list of proposed projects.

3.1. Math behind elliptic curve cryptography

In class we presented the bare bones of elliptic curve digital signature algorithms, however, we did not dwell deep into the math of elliptic curves. In this project you are asked to do precisely that.

There is a number of subjects to cover here: from why the point at infinity has a natural interpretation in projective geometry, to bounding the number of points on an elliptic curve over a finite field (also known as Hasse's theorem). Similarly, you might dwell into how to use elliptic curves to test whether a number is prime, or how to factorize numbers using Pollard's rho method, and Lenstra's algorithms.

All of these topics are covered in chapter 6 of the book: *A Course in Number Theory and Cryptography* by Neal Koblitz. In case you decide to take the topic we will make the book available to you (also, you can try entering the book's name followed by 'pdf' into a search engine and see what pops up).

3.2. Breaking the Discrete Log problem

When describing elliptic curve cryptography, we said that the security is based on the assumption that solving the discrete log problem is an exponential time algorithm, which basically checks all the possible solutions. However, there are some options that can significantly reduce the search space. A good list of such algorithms can be found (in the section called Algorithms) at: `https://en.wikipedia.org/wiki/Discrete_logarithm`.

In this project you should select at least three of these algorithms, explain in detail how, why, and when they work, and, if possible, do some experimentation on breaking the problem in some simple group.

3.3. Proof of stake

In Bitcoin one uses a proof-of-work based protocol to reach a distributed consensus. The other often proclaimed option is proof of stake, in which participants guarantee the correctness at the danger of losing money they have (note that this is a gross simplification).

For this project you should understand what proof of stake is, what are its advantages, and what are its main weaknesses. You should also check out some concrete proposals of how a proof of stake protocol could be implemented. You can start reference hunting at the following links:

- https://en.wikipedia.org/wiki/Proof_of_stake
- https://en.bitcoinwiki.org/wiki/Proof-of-stake

3.4. Lightning Network

Bitcoin transactions are painfully slow. In this project you should explore the basics of the Lightning network, which makes them faster. For this, you will need to explore concepts such as timelocks, payment channels, revokable commits, etc. A good starting point is the Chatper 12 of the "Mastering Bitcoin" book, available as open source at: https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch12.asciidoc

3.5. Ethereum Modified Merkle Patricia Trees

Ethereum uses a more efficient data structure than a Merkle tree to store the state data. In fact, each Ethereum node stores the entire global state. The secret how this is done are Modified Merkle Patricia Trees. In this project you are asked to study this data structure, explain how it can be implemented, and why it works so efficiently (in particular how the state gets updated from one block to another). Furthermore, you are asked to implement this data structure, and do some basic experimentation to show how it works. Good references to start exploring this are:

- Non technical overview: https://blog.ethereum.org/2015/11/15/merkling-in-ethereum/
- High level description: https://easythereentropy.wordpress.com/2014/06/04/understanding-the-ethereum-trie/
- Formal(ish) specification: https://github.com/ethereum/wiki/wiki/Patricia-Tree
- A nice discussion: https://ethereum.stackexchange.com/questions/6415/eli5-how-does-a-merkle-patricia-trie-tree-work
- Ethereum Yellow Paper Appendix D: https://ethereum.github.io/yellowpaper/paper.pdf#appendix.D

3.6. Create your own user application for Ethereum

In this project you are required to propose a good use case scenario for implementing on the Ethereum blockchain. You should familiarize yourself with the Ethereum ecosystem, and the scripting language used in Ethereum to define smart contracts called Solidity. Then propose a smart contract, explain why it makes sense to implement it on the Ethereum blockchain, and implement/test it in Solidity. Lots of exploration to be done here.

3.7. Propose your own topic

Quite simple: send me an email (domagojvrgoc@gmail.com) to confirm the topic you would like to explore.