Week 1 Study Guide

For folks who are interested in study groups, we're providing a few resources you can walk through as a group. **These are just suggestions to get you started and you don't have to follow them.**

Please note: Some of these exercises may take more than one session! Don't feel like you have to finish them all in one sitting. It might be helpful to start working on them separately, then come together when you're all at a similar stopping point. You can share what you've gone through, some challenges you had, and help anyone still working through the exercises.

We'll also be going through these examples / tutorials in the weekly sessions. We'll be encouraging people to share their work there as well. Be kind to those in your group!

1. Beginning: Ethereum 101 Workshop

See a recording of the first exercise here

The first is a few basic exercises showing what a private key looks like and how it functions. We'll be walking through these slides, but there are exercises located in the middle of the presentation that you might find interesting, especially in a group.

Scroll through the slides using your arrow keys until you see "Exercise 1" on the slide, about twenty slides in. At the bottom of the left column, click the light red "Generate Ethereum Address" button to see the "Ethereum Accounts" and "Private Key" sections populate.

Please note: Private keys generated here are NOT safe and should not be used outside of educational purposes!

Click the "Private Key" field to drag and drop the private key into the "Private Key" empty slot. Write a message in the "Message" field and click "Sign Message"

In the right column, drag the “Message Text,” and the “Message Signature” from the right column and the “Ethereum Address” from the left column. Note: Do not use the private key!

Check the validity of the message by clicking “Verify Signature.” Trying tweaking any of the inputs (Text, Signature or Account) to see the verification fail. (Please note this is all happening locally on your machine using a cryptographic library, not checking some server or third party.)

If you click “Download Message Data”, it will generate a .txt file you can send to other folks in your group to test their messages as well. Be careful: You have to copy and paste this perfectly and you have to open their file in a way that won’t change the .txt encoding. If you open it using Microsoft Word, for example, or Google Docs, even that basic encoding will tamper the message and render it invalid. This is a great example of how decentralized identity is very powerful but also very brittle: Even the smallest of issues breaks the system.

If you’re interested, there’s another exercise after this “Encode / Decode” exercise. It’s similar to the Anders Blockchain exercises, that shows how changing one piece of information in a hash chain breaks all the subsequent blocks.

There’s also a toy Solidity editor that allows you to deploy code with a click if you have MetaMask installed with enough ether for whatever network you’re on!

We’ll be going through these during the Wednesday live session

2. Technical: Bitcoin hashing exercises

First, clone this repo of Bitcoin Whitepaper Exercises

Second, follow the steps in this README to do the first exercise, Hashing.

Please note, it doesn’t matter which hashing algorithm you use, as long as it is consistent.

A more advanced tutorial you can follow is this one in Python. This is a from-scratch implementation of Bitcoin with **zero dependencies** (🤯 🤯 🤯). For those of you who following Machine Learning, it's from Andrej Karpathy, who's written great stuff on Neural Networks). You can follow along with the Python as he builds it or you could run the Jupyter notebook here. **This is for intermediate-to-advanced Python users.** If you're not familiar with Python, this is not the place to learn.

3. Technical: Build a P2P Network Workshop

So far, this study guide has introductory material and material about cryptographic primitives. The thing we're missing is a distributed system / distributed consensus tutorial!

Here are two good tutorials from WebTorrent. The first tutorial (P2P Workshop) walks through the process of building a toy server on your local machine, builds a chat service, then builds out the networking processes to discover peers in your network.

The second tutorial (P2P Filesharing Workshop) walks through the technical specifics of streaming files, discovering services, hashing files, chunking files (a feature of torrent protocol) and then sharing files with multiple participants.

These tutorials are in Node.js and are good because they break down these complicated projects into discrete steps. However, it is fairly advanced and, if you're having trouble, try joining a group that's working through it.