**Lesson Plan – Nonce and Proof of Work**

**Learning objectives -**

1. Revise concepts of checking if blockchain is being hacked from previous class
2. Learn about Nonce and Proof Of Work

**Materials required -**

repl.it login credentials

**Proof-of-Work**

Since participants on the blockchain network are anonymous users on their computers, we can’t trust them to verify transactions honestly. Proof-of-Work does nothing more than introduce an additional security constraint to verify transactions. This constraint takes the form of a computationally difficult math problem, which means to say that it takes a lot of time even for the computer to solve the problem.

Instead of randomly being chosen to broadcast their unconfirmed block, a special group of participants, also known as miners, now need to solve a problem in order to be eligible to broadcast their block. The problem, also known as Proof-of-Work, takes the form of a guessing game that involves the use of hashing.

Proof of work (PoW) is a consensus algorithm used in many blockchain networks to verify transactions and create new blocks. In PoW, miners compete to solve a mathematical puzzle by iterating through different nonces until the solution is found. The first miner to find the solution to the puzzle is awarded the block reward and any transaction fees associated with that block.

The first miner to produce a proof broadcasts their unconfirmed block together with the correct nonce value. The rest of the network then verifies the calculation. If the majority of the participants agree, the Proof-of-Work for the block is now complete and the block has now been confirmed! The network then moves on to work on the next block.

The hash function that’s most commonly used to create the hash for the block is the SHA-256. Miners first guess a *nonce* value, which is then combined with the contents of the block (i.e transactions, timestamp, hash, and previous hash). They repeat this process until the desired hash is generated.

Here’s an example of a simple problem — find a number which, when combined with the unconfirmed block’s contents, produces a hash whose first four digits equals 0000. Every participant uses their computer and a hash function (typically SHA-256) to find a number that generates a correct hash. Since this is a random guessing game, everyone usually starts out with 0 and increases their guesses until they produce an acceptable hash.

**Security of POW:**

The puzzle that miners must solve in PoW is designed to be computationally difficult but easy to verify. The difficulty is adjusted over time to maintain a steady rate of block creation and prevent the network from being overwhelmed by too much mining power.

The security of PoW comes from the fact that the cost of mining is directly related to the computational power required to solve the puzzle. This makes it difficult for any one entity to control the network or carry out a 51% attack.

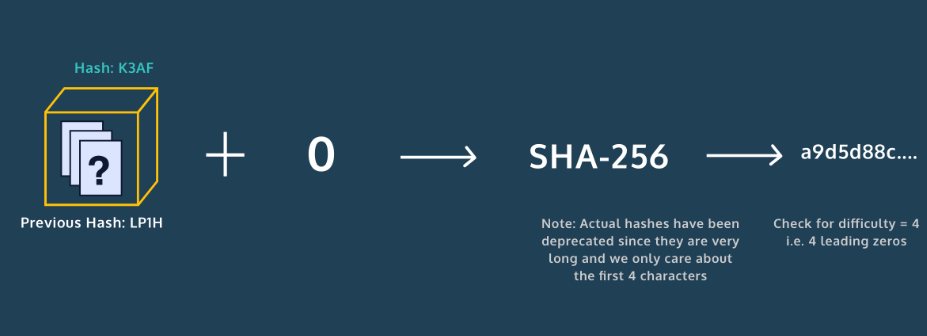
Limitation of POW:

However, PoW has several limitations, including its high energy consumption and the centralization of mining power in the hands of a few large mining pools. To address these issues, alternative consensus algorithms, such as proof of stake (PoS) and delegated proof of stake (DPoS), have been developed.

Despite its limitations, PoW remains a popular consensus algorithm and is used in many prominent blockchain networks, including Bitcoin, Ethereum, and Litecoin.

**Key Terms:**

* **Miners:** Special participants who calculate the Proof-of-Work to mine new blocks.
* **Nonce:** A number to be guessed by miners which when combined with the block produces an acceptable hash.



Let’s review the concepts of nonce and proof of work. In this exercise, we will implement an example that demonstrates the difficulty of the math problem that helps protect the blockchain from potential attackers.

Introducing the lesson/project including the concepts (Time - 45 min)

Link to Repl it Project :

<https://replit.com/@PriyankaJetLea1/Lesson9NonceAndProofOfWork#main.py>

**Homework –**

# Write a program to find the sum of all items in a dictionary

<https://www.geeksforgeeks.org/python-program-to-find-the-sum-of-all-items-in-a-dictionary/>

Write a program to

* [Get Current Timestamp](https://pynative.com/python-timestamp/#h-get-current-timestamp) using
  + [Datetime to Timestamp](https://pynative.com/python-timestamp/#h-datetime-to-timestamp)
  + [Get Timestamp Using time Module](https://pynative.com/python-timestamp/#h-get-timestamp-using-time-module)
  + [Get Timestamp Using calendar Module](https://pynative.com/python-timestamp/#h-get-timestamp-using-calendar-module)

Also convert timestamp to string and datetime (format)

https://pynative.com/python-timestamp/