# White Paper on Hashing Algorithms and IPFS

Created by: Chenna Kesav Daggubati
ICS-690 Blockchain

# Table of contents

# Cryptography concepts related to blockchain Technology

- Block: collections of data
- Chain: lists which are public databases of blocks
- Cryptography is also composed of two ancient Greek words: Kryptos means Hidden and Graphein means write
- Encryption: converting plaintext to cipher text and decryption
- Cipher: cryptographic algorithm used
    - Key is required to get the output of the algorithm

# Cryptographic algorithms

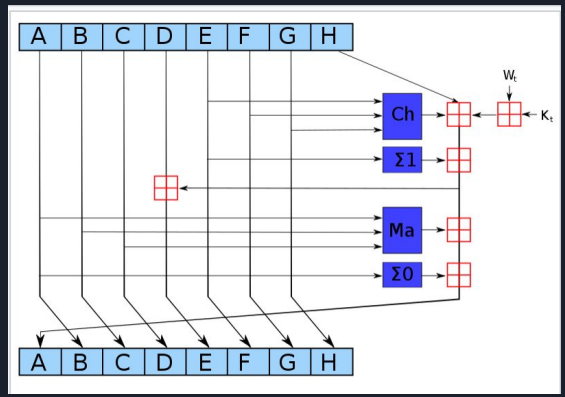
- Symmetric cryptography
  - Same key is used to encrypt and decrypt the data
- Asymmetric
  - Two different keys public and private keys are used
- Hash functions
  - Don’t require keys
  - Generates a fixed length has from given input
  - Properties: Avalanche effect(slight change gives different output), uniqueness, deterministic(same input have same outp, and quickness
  - Links blocks to one another
  - Helps maintain integrity stored in each block
- Blockchains uses asymmetric key algorithms and hash functions

# Additional Benefits of cryptographic concepts

- Digital signature
  - Provides integrity and authenticity validation of data
- Multi-signatures algorithm
  - Generates digital signatures where it requires multiple parties to be valid
- Zero knowledge proof
  - Helps user prove knowledge of a secret without revealing the secret itself
- Stealth addressing
  - Anonymizes the recipient side of a transaction by generating a one-time address and claiming the value
  - It's possible to know the value in the wallet
- Ring signature
  - Anonymizes the sender
  - Allows someone as a member of the group to sign data using set of public keys which generates a signature
  - Signer should have a private key which verifies he is member of group

# Hashing Algorithms

- Set of algorithms developed by the National Institutes of Standards and Technology(NIST) and other government and private parties
- Hashes are commonly shown in Hexadecimal format
- MD2, MD4
- 128-bit(32 hexadecimal characters) hash value is generated by MD5 or the Message-Digest algorithm : https://en.wikipedia.org/wiki/MD5
- NSA created SHA-0, SHA-1, SHA-2
- SHA-1(160 bit hash):Updtaed version of MD5 and not collusion resistant, Cracked by google in 2015 using distributed computing,
- SHA-2: made by National Security Agency
  - Consists of six hash functions with digests
  - Includes significant changes from SHA-1
  - SHA-256 creates 256 bit hashes, SHA-512 creates 512 bit hash
  - SHA-224 and SHA-384 created truncated versions of 256, 512

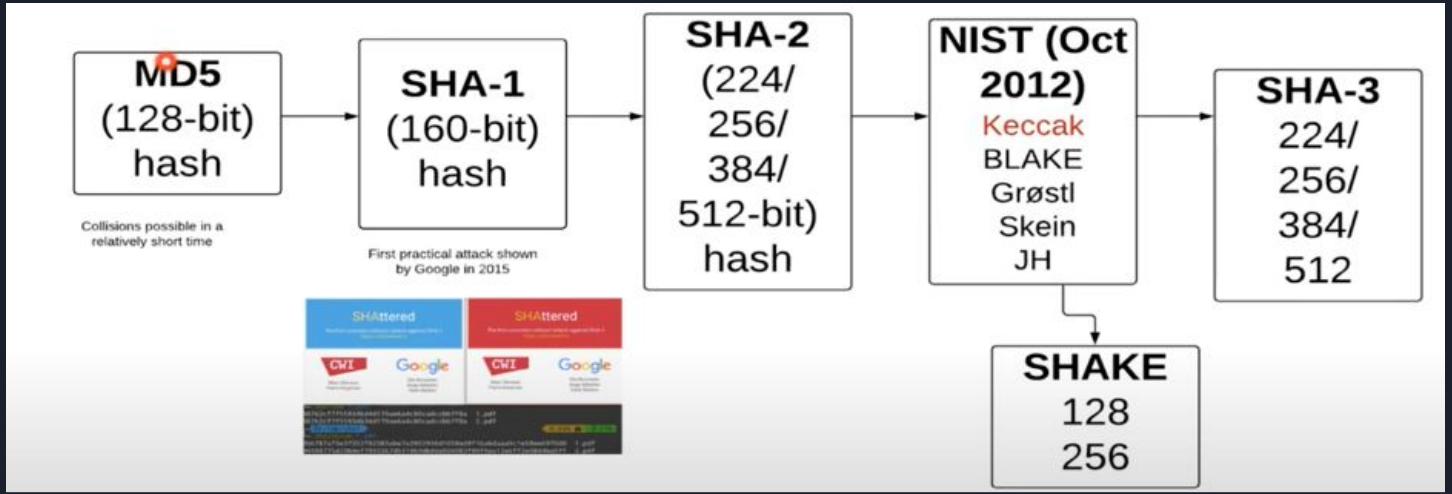


# Other Methods

- HMAC - Hash based Message authentication code
  - Fixed length string of bits similar to MD5 and SHA-1
  - Known as HMAC-MD5 and HMAC-SHA1
  - Uses shared secret key to add randomness to result
  - Only sender and receiver knows the secret key
- Key stretching
  - Salts the password with extra bits to make it complex
  - PBKDF2 uses salts of at least 64 bits
  - Uses pseudo random function such as HMAC to protect passwords

# Keccak

- SHA-3 was created outside of NSA, was selected in non-NSA public competition
- SHA-3 also known as Keccak won the NIST competition based on throughput and energy consumption by algorithm
- It can create hashes of same size as SHA-2.
- Became the official SHA3 but with a small difference  from keccak while padding
- Employed by Monero
    - But not for Proof-of-Work
- Used for
    - Random number generator(crypto zombies first module)
    - Block hashing
    - Transaction hashing
    - Stealth address private key image
    - Public address checksum
    - Authentication




```
pragma solidity ^0.6.0;

//input text, number and address
//output is a unique 32 byte hash
contract hashtest {
    function hash(string memory _text, uint _num, address _addr) public pure returns (bytes32) {
        return keccak256(abi.encode(_text, _num, _addr));
    }
}
```

# Sponge function

- Takes an input bit stream of any length
- Produce an output bit stream of any desired length
- Built from 3 components
  - State memory
    - Two sections: R and C
    - Two phases: absorbing phase + squeezing phase
  - Function that transforms the state memory
  - Padding function P

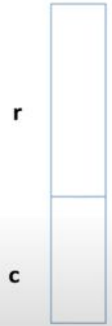
# Keccak process

- Keccak is more powerful because of its state size of 1600 bits
- Advanced Encryption standard(AES) operates on 4*4 column major order array of bytes known as state https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- Break data into r bits chunks
- EX-OR it with the rate part of the state
- Output feeds into the function (f)
- 24 rounds and is created with EX-OR, AND, and NOT functions
- Feed it into the next stage if there is more data
- Once all the message data is exhausted, we go into a squeezing function and produce an output (Y0)
- Output is truncated to the required hash size or processed until the required output size is produced

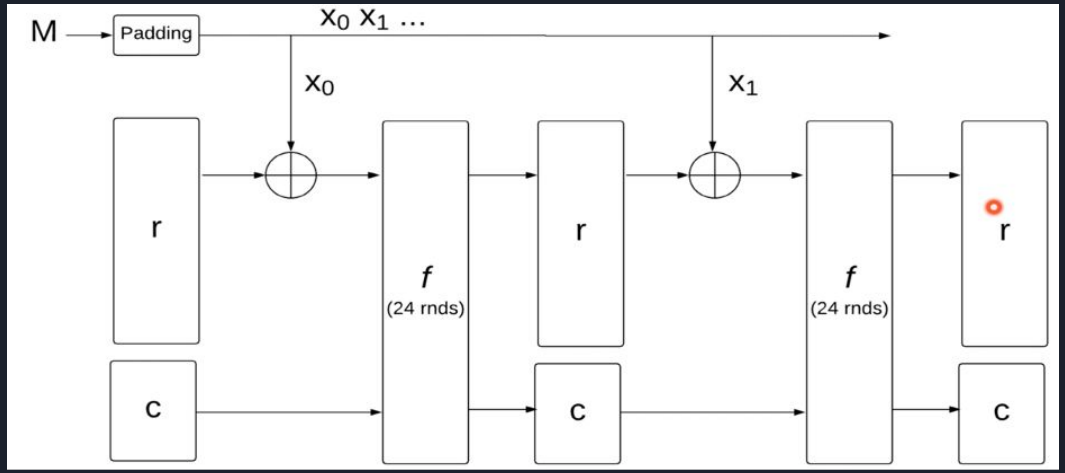
Keccak/SHA-3 (State)
"Ket-chak"

**State size (b) = 25 x $2^l$ = {25, 50, 100, 200, 400, 800, 1600}**

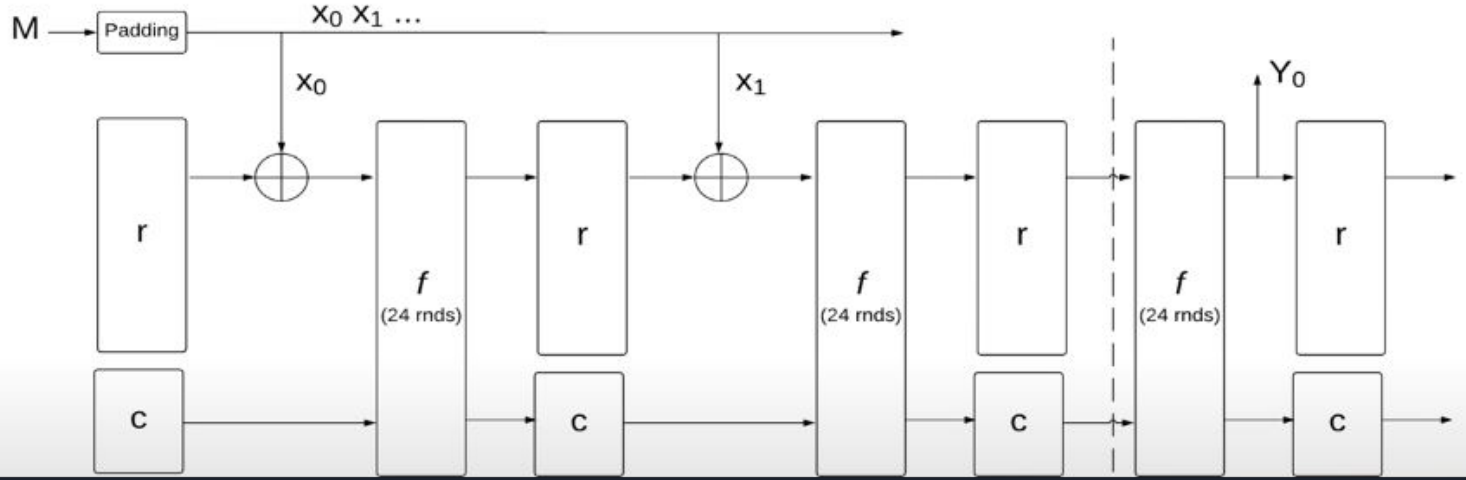
**Rounds = 12 + 2l**




| Type | Output (bits) | Rate (r) | Capacity (c) |
|---|---|---|---|
| SHA3-224 | 224 | 1152 | 448 |
| SHA3-224 | 256 | 1088 | 512 |
| SHA3-384 | 384 | 832 | 768 |
| SHA-512 | 512 | 576 | 1024 |
| SHAKE128 | d | 1344 | 256 |
| SHAKE256 | r | 1088 | 512 |



# Keccak/SHA-3 (Squeeze)



# History of cryptocurrency mining

- Began with Bitcoin (BTC) SHA256 algorithm → Litecoin Scrypt algorithm (LTC) → X11 → Dash and Ethash for Ethereum (ETH)
- BTC, ETH, and LTC
  - 3 mineable coins
  - Operate within the proof-of-work
  - Use different hashing algorithms
- BTC uses SHA-256 hash function
- ETH uses Ethash Proof of Work hashing algorithm
  - In the near future, it will switch over to Proof of Stake
- LTC uses Scrypt algorithm

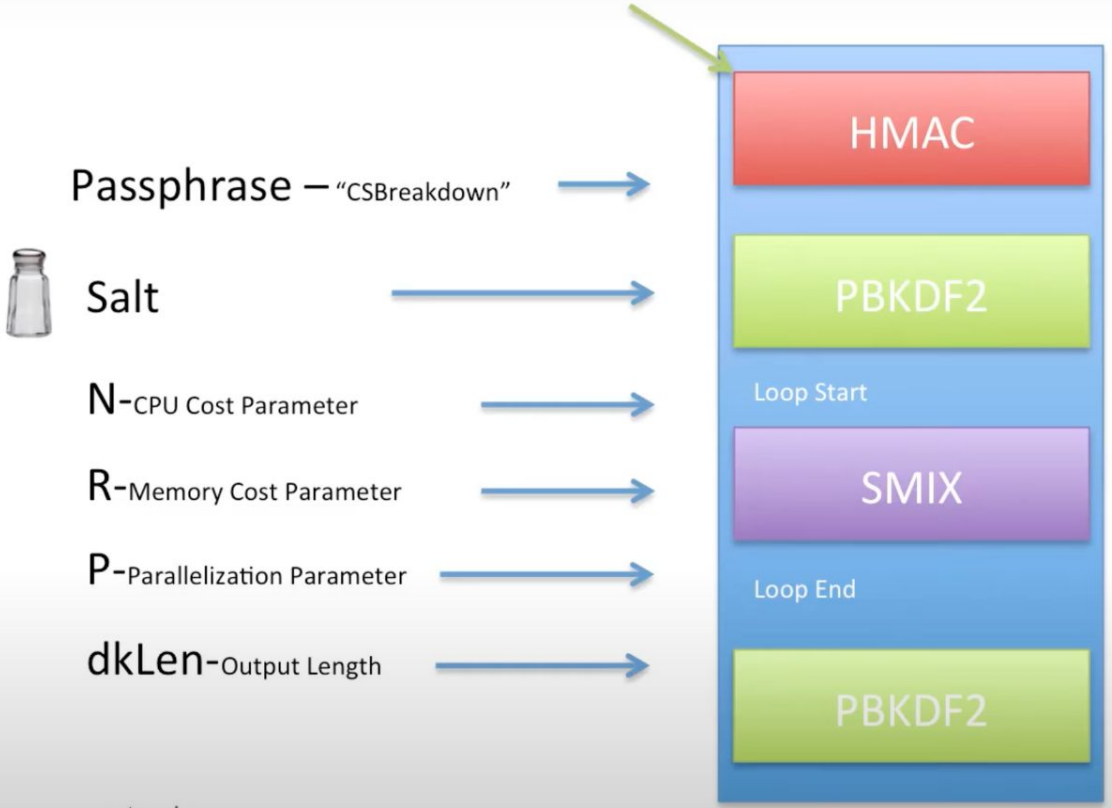
# Purpose of using different algorithms

- Resist purpose build hardware's like Application Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs)
- When cryptocurrency is ASIC resistant
  - No ASIC machine has been developed yet to mine the coin
  - The coin can only be mined using consumer grade hardware like CPU and GPUs
- It is only a matter of time for someone to develop an ASIC for that coin

# Scrypt Algorithm

- Scrypt: password-based key derivation function (KDF)
- KDFs: designed to be computationally intensive
  - Efficient at preventing brute attacks and Rainbow table attack
- Scrypt offers
  - High level  security
  - Improves network security by resisting large scale custom hardware attacks
  - SMIX : Hashing value in memory intensive way
- Its ability to hinder ASIC mining machines it is considered to be the most effective alternative to Bitcoin’s SHA-256 hashing algorithm
- Used by Litecoin and Dogecoin

# Scrypt Algorithm




## What makes scrypt different?

Parallelization Factor – fine tunes the relative CPU-Cost

dkLen – The user has the ability to define the output size

N & R – User set CPU & Memory cost

The inner workings have a lot of differences as well!

→ Password Key

6B CA AD 1E 3D FE 79 1F 3B F5 EE F5 D5 3A 43 D8 B0
12 9C 5A 3E 1F 23 17 9B 0D 23 AC EC 0C A4 D0 48 00 9F
C4 97 C3 69 E3 B8 1D 82 58 D8 D8 8C 26 E5 CD 6B 8D
2F 27 1F 29 1F C4 4A C0 74 B0 4D 4B

# X11

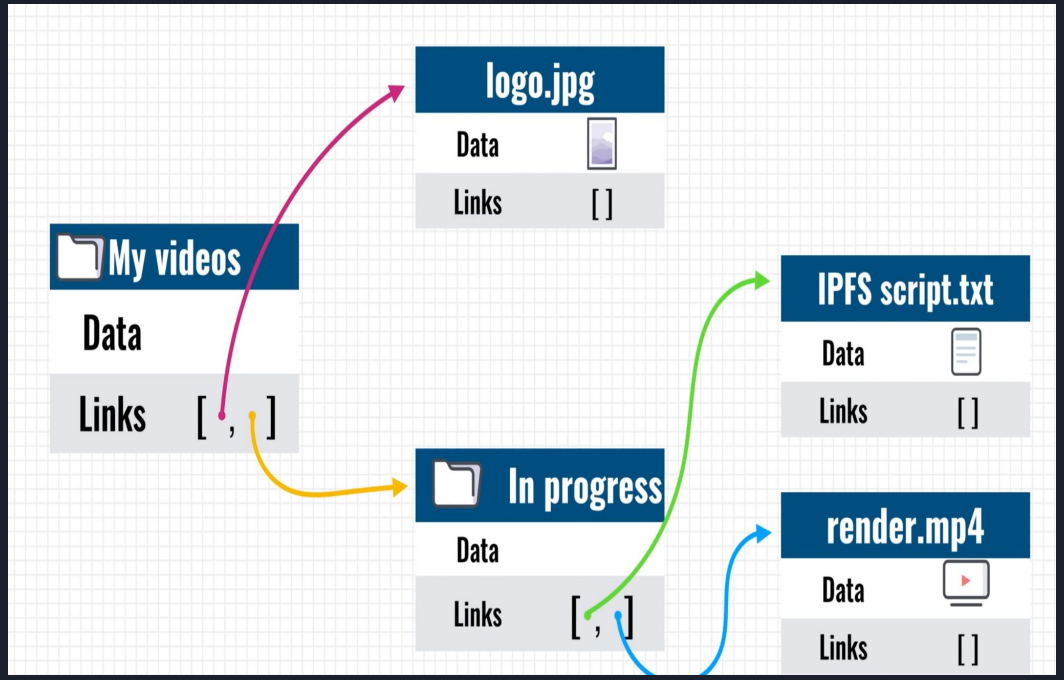
- Designed for cyrptocyrrencies, Darkcoin protocol in 2014
- More secure than SHA-256
- Not used by ASICs
- Most famous: Dash.X11
  - Uses 11 different hash algorithms such as BLAKE, BMW which qualified in NIST Competition
  - Probability of it failing is close to zero
- How it works
  - When a value is submitted to the BLAKE function → produces a hash value
  - Value is submitted to the BMW function → produces another value
  - Process repeats until the last function
- BLAKE made into the final five in the NIST open competition

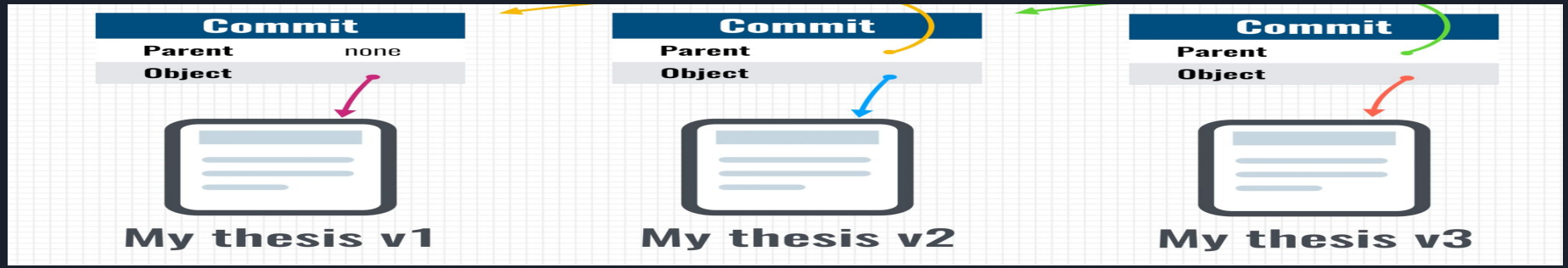
# Ethash

- Proof-of-work function in Ethereum-based blockchain currencies
- Hash function belonging to the Keccak family
  - Not a SHA-3 function
- Developed as an upgrade of Dagger-Hashimoto to remove computational overhead
  - Dagger: uses Directed Acyclic Graph (DAG) https://en.wikipedia.org/wiki/Directed_acyclic_graph
  - Hashimoto: Random Access Memory (RAM)
  - ASIC resistant
  - Light Client Verifiability : a block should be relatively efficiently verifiable by a light client
- List of a few mineable coins uses this algorithm based on popularity
  - ETH
  - Ethereum Classic
  - Metaverse ETP
  - Expanse
  - Musicoin

# InterPlanetary File system(IPFS)

- Decentralized internet similar to bittorrent
- SHA-256 hashing algorithm by default
- Issues with centralization
  - Centralized servers Ex: Youtube, Wikipedia, google
  - Censorship : government can block access  Ex: Turkey 2017 Blocked Wikipedia
- Why do people still use it
  - Fast and high quality
  - No good and fast alternative
- Location based addressing vs Content based addressing(where to find vs what you want)
- Based on the hash of the file
- Data is stored in IPFS objects which can store upto 256kb and objects also contains links of another objects
- Larger files are stored using empty IPFS object which has links of the ipfs objects related to file
- IPFS also has version controlling


logo.jpg
Data
Links [ ]
My videos
Data
Links [ , ]
IPFS script.txt
Data
Links [ ]
In progress
Data
Links [ , ]
render.mp4
Data
Links [ ]

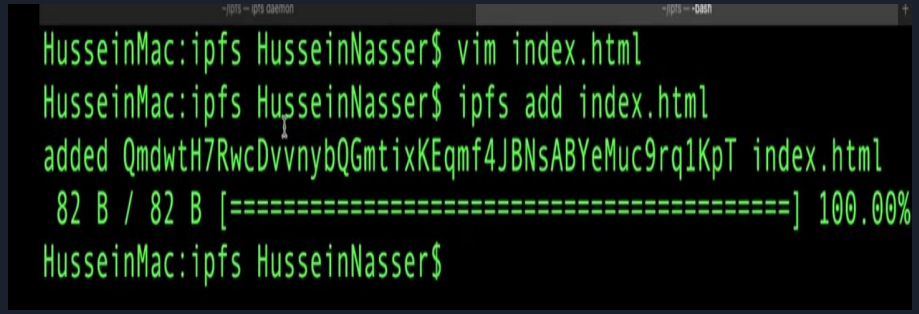

Commit
Parent none
Object
My thesis v1
Commit
Parent
Object
My thesis v2
Commit
Parent
Object
My thesis v3

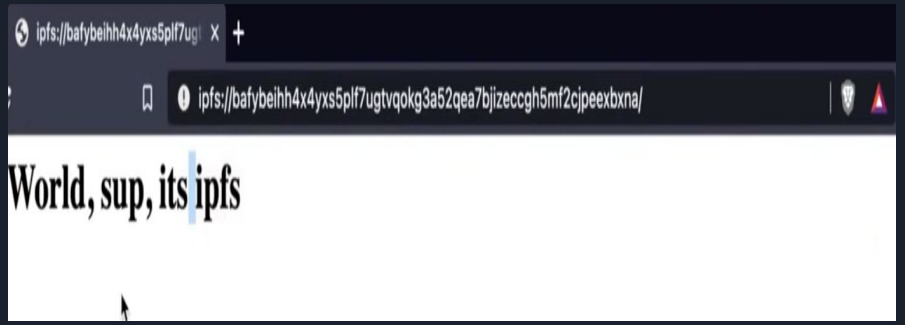
# Problems with IPFS and Possible Solutions

- Problem
  - Nodes being  offline
- Solutions
  - Incentivize nodes
  - Proactively distribute files
- Filecoin is created by same people as IPFS: Decentralized market for storage
- Provide incentives for keeping file online and it also help replicate files for better availability
- Applications of IPFS
  - Hosted copy of Wikipedia on IPFS for people to access from Turkey in 2017
  - Dtube basically like youtube
- Can this really support interplanetary ??

# Routing

- Hash which is known as content id
- Distributed hash tables maps content id to peer address
- DHT server(IPFS server) hosts DHT's
- DHT client (IPFS client) connects to DHT server
- IPFS clients ask local DHT for Cid and retrieves collection of ip address
- If nothing returns it asks its peers



# Learnings from this Class

- Fundamentals of blockchain : https://github.com/Maghribi-Foundation/chennakesavdaggubati/blob/main/Blockchain%20Notes.md
- Crypto zombies
  - How to write solidity code
- Chitfund Project
  - Metamask, deployment of existing project, capitalization
- In-depth analysis of Cryptographic concepts
  - Hashing Algorithms
- IPFS
  - How to use it in my thesis

# References


1. https://github.com/ethereum/wiki/wiki/Dagger-Hashimoto
2. https://eth.wiki/en/concepts/ethash/ethash
3. https://asecuritysite.com/hash/s3
4. https://asecuritysite.com/hash/gokang
5. https://asecuritysite.com/hash/goshake
6. https://zerocrypted.com/what-is-ethash/
7. https://coinguides.org/asic-resistance-explained/
8. https://medium.com/asecuritysite-when-bob-met-alice/one-of-the-greatest-advancements-in-cybersecurity-the-sponge-function-keccak-and-shake-6e6c8e298682
9. https://www.youtube.com/watch?v=bTOJ9An9wpE&t=2s
10. https://en.wikipedia.org/wiki/NIST_hash_function_competition
11. https://www.mycryptopedia.com/x11-algorithm-explained/
12. https://monerodocs.org/cryptography/keccak-256/
13. https://cryptoadventure.com/blockhain-hashing-algorithms-explained-all-you-need-to-know/
14. https://coinguides.org/scrypt-coins/
15. https://www.youtube.com/watch?v=PlvMGpQnqOM&t=6s
16. https://www.youtube.com/watch?v=TkWAgeSYL_Q

# Thank you for listening!

Questions??????????????????????????????????????