DECENTRANET - AN ETHEREUM, PROXY RE-ENCRYPTION AND IPFS BASED DECENTRALIZED INTERNET

Koushik Bhargav Muthe
CSE Department
SRM University AP
Amaravati, India

Thiru Srinivasa Teja Vemuru
CSE Department
SRM University AP
Amaravati, India

Khushboo Sharma
CSE Department
SRM University AP
Amaravati, India

Nilofer Sultana Mohammad
CSE Department
SRM University AP
Amaravati, India

***Abstract*—The global daily data generation is estimated to be more than 2.5 quintillion bytes, and more than 90 percent of the total data produced is generated in the last two years. Most of this data is being generated and controlled by very few centralized agencies in the Web 2.0 internet architecture. This causes issues, including data manipulation, lack of privacy, and data leaks historically and is not just limited to the fields mentioned. This paper proposes a Web 3.0 based architecture which eliminates the centralized agencies and to promote a fully decentralized, secure, and transparent internet. It leverages IPFS, a peer to peer distributed hypermedia transfer protocol. Ethereum and smart contracts create a secure decentralized mechanism for initiating data-based payments. Furthermore, the proposed architecture uses zero-knowledge proofs and proxy re-encryption mechanisms to enhance the privacy of the nodes in the network.**

***Index Terms*—Decentralization, IPFS - Inter Planetary File System, Ethereum Blockchain, Proxy Re-encryption, Zero Knowledge Proofs.**

I. Introduction

The problem with the current Internet architecture is that very few entities control most of the data over the Internet. These entities are also some of the biggest corporations the world has ever seen with a valuation of trillions of dollars; they process and own quintillion bytes of data over the Internet [1]. Data that powers the Internet is regarded as the most valuable asset in the 21st century, in fact, costlier than oil [2].

The main reason that is empowering these entities lies within the core architecture of the Internet. The Internet is powered by the HTTP model, which is essentially a client and server-based architecture. A request from a client goes through multiple servers and gatekeepers over the Internet. These gatekeepers are third party services and are often the reasons for censorship and surveillance. In the modern internet architecture, these gatekeepers are extensive corporate services with their cloud services. Now Cloud has made the Internet cheaper and elastic while it gave more control and power to these giant gatekeepers. Technologies such as AI and ML have made Cloud efficient where users can get free personalized services for an exchange of their data. The data exchanged for these services are some of the most personal information about the users. These personalized services are creating a walled garden where users are comfortable with what they are receiving without acknowledging the harm in privacy breaches and control over one's personal information. This can lead to problems ranging from lack of privacy and resiliency, as seen in the case of censorships in various countries [3] to manipulation of the world politics, as seen in Facebook-Cambridge Analytica mishap.

A fully decentralized internet can solve this problem by eliminating the centralized entities and connecting the users directly with each other, genuinely satisfying the vision of the Internet. In a decentralized internet, there will be no government that has the power of censorship, and there will be no entity that can control the data of the other user without providing the appropriate monetary value for it. But an internet without a centralized agency can create specific issues such as DDoS, Privacy and Identity as seen in peer to peer distributed protocols [4]. This is the reason where additional protocols and security measures, that would secure these peer to peer distributed networks, are needed.

This paper focuses on utilizing technologies like Ethereum Blockchain, InterPlanetary File System (IPFS), and Proxy Re-Encryption to create a fully decentralized internet, thus making it free and equal to everyone.

II. Background and Related Works

Internet is the most advanced and innovative technology created by human beings in the modern world. It has transformed the world in a way that we have never seen and has a vision of making human beings a connected entity. Ever since the term "Internet" was coined to describe the TCP/IP protocol internet has gone through many changes and improvements. But most of these changes which were made were limited to improving its efficiency and security. There were not many innovative approaches to make the Internet free and equal to every user. This is one of the primary reasons for the introduction of Web 3.0 which is the third phase of the evolution of the Internet. In Web 1.0, architecture users were not able to produce any data, and producers have complete control over the data access. Web 2.0 architecture has improved this by giving users limited access to producing the data, which reduced the gap between producers and users. The Web 3.0 architecture aims to make producers and users indistinguishable in terms of access, sharing, and production rights [5]. The most important part of Web 3.0 stack consists of a Blockchain based distributed ledger and a distributed file transfer protocol. The work by Quanqing Xu and Zhiwen Song [6] considered only building a social network on top of Ethereum and IPFS. The limitations of the previous works include relying on on-chain computation for large files and lack of focus on security aspects of the network. The proposed

Decentranet protocol also has an edge over Web 2.0, as it is a Web 3.o based protocol which incorporates the existing Ethereum and IPFS technologies, and improves its security by integrating them with our Proxy Re-encryption mechanism. The work presented here is focused on the building a secure, fully decentralized and transparent internet architecture, which takes advantage of Proxy Re-encryption protocols.

III. Proposed Architecture

A. Ethereum Blockchain

Blockchain is an immutable digital ledger stored in a distributed network. The word Blockchain was first introduced in 2008 by Satoshi Nakamoto in his Bitcoin white paper [7]. Bitcoin gave the idea of Blockchain and solved some of the most complex problems, such as the Byzantine Fault Tolerance problem and Double Spending problem in electronic cash transfer mechanisms. The vision of Blockchain was later completely transformed by Ethereum, which was developed by Vitalik Buterin, extending Blockchain into various other areas [8]. Ethereum is a global open-source blockchain platform for building decentralized applications. It is also the world's first programmable Blockchain.

Fig. 1. Block Diagram of a Blockchain

The three main features of a Blockchain are Immutability, Transparency and Decentralization. A Blockchain is an immutable public digital ledger. Each block in a blockchain consists of information related to transactions such as the cryptographic signature of the users, timestamp, transaction details and hash of the previous block data. This block data is cryptographically hashed using algorithms such as SHA 512 and MD5. Blockchain achieves immutability by adding the cryptographic hash of the previous block data to the current block, any change in the previous blocks can be detected easily as the ledger is public. This makes the Blockchain immutable. The distributed ledger is made available to everyone in the network. This ensures the core of Blockchain transparency. The block data such as user signature, timestamp and previous block hash are all hashed using the SHA 512 and MD5 standards. This ensures the privacy of the nodes in the network and making this ledger public enables transparency at the same time. Blockchain is not controlled by any central athority. It achieves decentralization on multiple levels. Architectural decentralization ensures that Blockchain is a network that runs on all the systems of the network rather than a single set of servers. Logical decentralization ensures that all these systems are running the same Blockchain running with the same purpose. Political decentralization ensures that the Blockchain is not owned or governed by a single or set of authorities. It is instead controlled by the consensus or collective agreement of all the participants in the network.

B. Smart Contracts

Ethereum is a programmable Blockchain as it utilizes a feature called Smart Contracts. Smart Contracts were first introduced by Nick Szabo and are the real power of Ethereum. Smart Contracts are the rules and agreements of the transactions written in the form of code. The code from the smart contracts is converted into byte code by EVM (Ethereum Virtual Machine) and will be fed into the Blockchain for processing. Smart contracts can be programmed by using languages such as Solidity and Vyper.

C. Zero Knowledge Proofs

Zero-Knowledge Proofs are privacy-enhancing protocols which let the owner prove to the verifier, the possession of a secret without actually revealing the secret to the verifier. Zero-Knowledge Proofs are being used to authenticate, verify and validate blockchain transactions without revealing the identity of the nodes that are making the transaction. This adds up another layer of privacy and security over the blockchain network. The proposed architecture utilizes a specific type of Zero-Knowledge Proofs called noninteractiveZKP, which enables the verification process without any interaction between prover and verifier [9].

D. IPFS - Inter Planetary File System

Networking is regarded as the most game-changing innovation in the field of computer science, which led to the creation of the Internet. These initial networking protocols gave a strong foundation for the Internet, and many of these protocols are being implemented even now in the modern Internet architecture, e.g. HTTP. The World Wide Web (WWW) which utilizes the client and server architecture, has made the process of retrieval of the enormous amount of data transferred through the Internet easier [10]. This type of web architecture is called Web 2.0, the largest and most popular form of web architecture. In order to match the massive scale of the Web 2.0, the resources such as data storage, and computing power is enormously scaled and served to the end-user or clients based on demand [11]. This gave rise to Cloud computing, which currently powers most of the Internet. The increase in the efficiency of machine learning has made the cloud providers improve their services with personalized deliveries to end-users at a cheaper cost. This massively scaled up the amount of services provided by these cloud entities. Integration of self-driving cars, IoT and smart gadgets into the Internet also creates an exponential increase in data transactions over the Internet. It will become an integral part of a human being and a single glitch can create many consequences. The Internet is the present world is made into pieces and is controlled by several entities such as large cloud providers and governments. There are many cases where governments have blocked access to the Internet to suppress a protest or oppression. If the Internet is becoming an integral part of a human being, it should be free and democratic, just like everything about a human being.

IPFS is a peer to peer distributed file storage and hypermedia protocol that primarily aims to create a permanent web. IPFS can be a replacement to HTTP, which is one of the main

reasons for most of the issues in Web 2.0. HTTP, which is a client and server-based protocol, cause problems such as Centralization, Censorship, and Latency. IPFS, on the other hand, is a content-based distributed addressing protocol where there is no central authority that controls the Internet.

As IPFS is a content-based routing protocol, it searches for the nearest node that owns the data. This decreases the latency for routing. The files in the IPFS are encrypted before adding to the network using encryption standards such as the GNU Privacy Guard (GPG). The encrypted file is then uploaded using the ipfs add – command. Files which are uploaded can be retrieved by using the ipfs get command. This can be later decrypted using the gpg –decrypt command. The commands involving in adding and retrieving files:

–ipfs add myfile.pdf.gpg
–ipfs get QyYqSxWuZG8Cyo3MFAzqscC14ct4ybAayrx-
-clqzaJaFYTL

IPFS uses a Distributed Hash Table [12] to store data across multiple nodes. A DHT is a distributed key-value storage where the key is a cryptographic hash. Data transfer between the nodes is achieved using BitSwap [13], which is inspired by Bittorrent [14]. IPFS utilizes a Git inspired data structure called Merkle-DAG [15] to power the structure of the entire network. This makes DHT data processing and Version control default in an IPFS network. The security of IPFS is already enabled using immutable content hashing but is made efficient when integrated with a blockchain network such as Ethereum. It also in turn, helps to improve the speed of Ethereum.

E. Proxy Re-Encryption

Proxy re-encryption is a technique which permits an un-trusted intermediary or Proxy to change ciphertext from being encrypted under one key to another, without getting any information about the hidden plaintext. The public key encryption is made possible by using Elliptical Curve Cryptography (ECC) [16].

A proxy is, in essence, any provider. In general, the proxy or cloud provider doesn't have any rights to decrypt a secret message. Most of the encryption applications on the internet work on public- key cryptography. Hence the Proxy can only see encrypted messages and public keys. Private keys are given to individuals and are not revealed. Now the secret message can only be decrypted by the intended recipient, not the proxies. So, the Proxy can never see the information. Proxies can be Cloud Providers, Members in a Chat Group, Participants in a network, etc. For security reasons, proxies may not always be trusted. So data is encrypted with public keys and stored in an encrypted format, especially into the cloud. Here information is safe and theoretically, only the owner can access it since it is encrypted with the owners' public key. The problem arises if a third party needs to access the data. There are two ways this can be achieved. For example, Alice wants to send the stored encrypted data to Bob. Alice has to get Bob's public key, decrypt the stored data, re-encrypt it with Bob's public key and send it to Bob. This requires Alice to decrypt, then re-encrypt it for Bob and works only if there is one Bob. If Alice wants to send the stored data to any third parties, this method doesn't work as it takes a lot of re-encryptions for Alice every time.

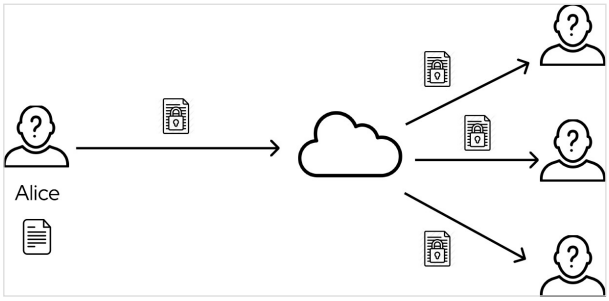

Fig. 2. Multiple encryption problem in cloud

This is where Proxy re-encryption can be used. Now, Alice can retrieve Bob's public key and issue a "re-encryption" key. This key represents the trusted relationship Alice has with Bob. Alice can send this data to the cloud provider, and the cloud provider can proceed to re-encrypt the already encrypted data that is stored in the cloud using the re-encryption key. Bob can download this re-encrypted data and decrypt it as well. This is possible because the initial data is encrypted with Bob's public key. In the second scenario, the decryption / re-encryption process is sidestepped, and Alice doesn't need to perform this operation on their own devices. Instead, both can generate a key that should be quick, and pass it to the cloud provider, which at no point can decrypt the original message, making this system very scalable and enables data-sharing apps in a cloud environment.

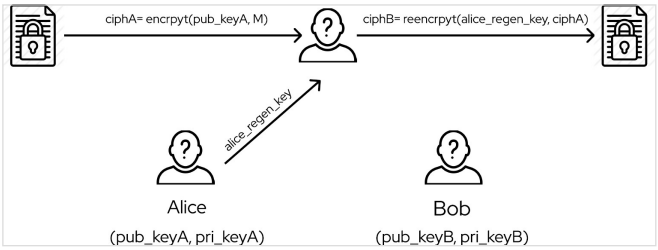

Fig. 3. Proxy Re-Encryption

In the traditional cloud mechanisms, there is no heavy necessity for proxy re-encryption techniques as most of the cloud services such as Whatsapp, Telegram and Dropbox encrypt user data before storing it in their servers. But the problem arises when we expand to distributed or decentralized networks. Particularly proxy re-encryption has the potential to revolutionise many aspects of the internet such as live streaming, multi-user chat rooms and services which provide premium content. This is not just limited to improving existing applications, but also this technique would solve some of the major problems such as universal data management systems in health care, communication media and banking.

When combined with decentralized applications such Ethereum and IPFS this technology can create a fully decentralized internet.

```
encrypted_message = encypted_msg

#Encrypted message can only be decrypted by Alice.

print(encrypted_message)
b'\x93\xda\x00"\x01\x03i\x1e\x1e\x08i\x0c\x9d\x83\xa2\xa6\xc6\xc7s\x86w\xb6
_\x08CZ\xb4V\xce\xd9gx0\xe9\x859\xa7\xe2\xda\x00"\x01\x02,\x03\x8b\x01~\x98
\xecXJ}?9\x88\x02\x9cf\xeb\xf5\x8e"[\x92\x0b!\xa2|\x81\x97u\xf7\x1e5\xda\x0
0"\x01\x02\xf1]\xfc|\xf5,\xe9\xb4\x8c\x96\xbeR\xc5a\xce\xca\x93\xc8\x10}Bl:
>\xc1Ap\xa7\x9cy\x8a>'

pre.decrypt(sk_a, encrypted_message)
b'This the text to be re-encrypted'
#Bob can't decrypt the message and can see only a hash.

pre.decrypt(sk_b, encrypted_message)
b'\x93\xdexj\xe5\xc2\xb3$3\xea\xeb2\xb3:Tg\x17O\x90\xcf\xd8\x19\xbe\xaf\x0c
H\xb8\x10'
```

Fig. 4. Proxy Re-Encryption - Decryption

F. Decentranet Architecture

The proposed protocol allows independent nodes to share data directly to desired destinations without leaking the contents to the remaining nodes in the network and without the need of any central agency. It integrates IPFS distributed hypermedia protocol and Ethereum to enable complete decentralization. IPFS provides a distributed Hash Table and a secure data exchange mechanism. Ethereum adds another layer of security through its immutable ledger, and Smart Contracts in Ethereum creates a reliable tool for initiating data-based payments. An additional layer of privacy is added to the Ethereum block data by integrating a secure Zero-Knowledge Proof mechanism. There are five critical elements in this architecture.

G. Decentralization (Ethereum Blockchain + IPFS)

Complete decentralization of the network is achieved through utilizing the Ethereum Blockchain and integrating it with the IPFS network. This enables an independent peer to peer information transfer using IPFS over the Ethereum Blockchain Network. Ethereum, which is a programmable platform along with the smart contracts, allows users to build better-decentralized applications by default.

H. Privacy (Proxy Re-Encryption and Zero-Knowledge Proofs)

The privacy of the nodes and data is achieved by utilizing the proxy re-encryption mechanism. A re-encryption key is created when a node initiates data transfer, and this encrypted data, along with re-encryption, is sent to the nearest available proxies for re-encryption. The proxies re-encrypt the data us- ing the key sent by the sender and transfer it to the destination over the IPFS network. As the nodes are connected through IPFS on top of Ethereum Network, their public keys, along with the content, can be used for routing. Zero-Knowledge proofs add another layer of privacy by minimizing the amount of information about the nodes that is shared over the network.

I. Security (Public Immutable Ledger)

The data transactions over the IPFS network are recorded on Ethereum Blockchain, which is an Immutable Public Ledger. This enables inbuilt security, where transactions cannot be modified or tampered. Also, considering Ethereum 2.0 is using the Proof-of-Stake algorithm, the security of Ethereum validation will be improved to a much greater extent.

J. Trustless protocol

The Internet becomes Trustless, where there is no need for a node to depend on third parties for data security and privacy. These standards are independently integrated into the network by default, thus making the Internet a free and democratic entity.

K. Incentives for Proxies

Proxies re-encrypt the information concerning the destination nodes, and in turn, they receive reward tokens for spending computational resources in order to encrypt the data. The reward mechanism can be deployed by leveraging the Smart Contracts in the Ethereum protocol. This gives an incentive to the proxies to participate in the network.

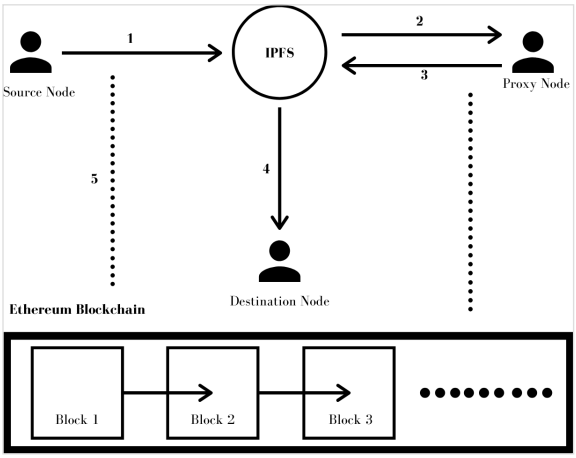

Fig. 5. Proposed Architecture

The steps involved in the architecture are:

1) The Source Node uploads the encrypted file to the IPFS network containing the targeted destination address list.
2) The Proxies in the network take up the encrypted file based on the Proof of Stake (POS) mechanism.
3) The Proxies re-encrypt the file to the prime destinations and upload it back to the IPFS.
4) The destination node receives the re-encrypted file and decrypts it using its secret key.

5) Every transaction in the IPFS network is recorded in the Ethereum Blockchain leveraging the power of Smart Contracts.

IV. Experimental Results

This section focuses on the evaluation of the proposed Web 3.0 based decentralized internet to its efficiency as a secure, private and faster internet architecture when compared to the existing Web 2.0 architecture. Utilizing the IPFS in the proposed architecture improved reliability, persistence and speed. The latency during file transfers has decreased by half when compared to using HTTPS. The below metric shows details about the comparison of average file retrieval rates of IPFS and HTTPS.

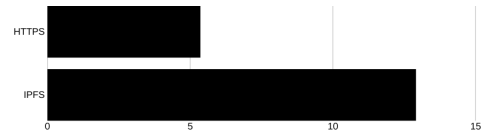

Fig. 6. HTTPS VS IPFS

Proxy re-encryption reduces the use of computation power for the individual node by transferring the computation load onto the proxies. Proxies enable the shift of the computation load from the user and re-encrypt the file using their computation power. The latency can be reduced when we deploy a number of proxies instead of one. Based on experimental observations utilizing proxies for re-encryption reduces the latency rates on an average by half.

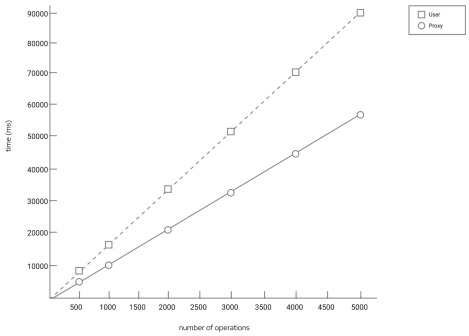

Fig. 7. Proxy vs User

V. Conclusion

This vision of this paper is to create a Decentralized Internet using the Web 3.0 architecture. A Decentralized Internet can power human beings to become a single connected entity. This paper discussed the problems with the current web architecture and proposed a protocol for its complete decentralization. The proposed Decentranet protocol is in a proof of concept stage and has to be used for research purposes. Scalability is an issue in terms of production due to the restrictions posed by the current Internet architecture. Incentivising the proxies to participate in the network plays a crucial role in scaling up the architecture. The current protocol relies on Ethereum 1.0, which runs on a Proof of Work-based consensus mechanism, which is not ideal for achieving complete decentralization. Ethereum 2.0 has a proof of stake which can improve the architecture, but its release is awaited.

References

[1] B. Marr, "How much data do we create every day? the mind-blowing stats everyone should read," Sep 2019. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#3860788c60ba

[2] T. Fauerbach, "Data reigns in todays data economy," Nov 2019. [Online]. Available: https://www.northridgegroup.com/blog/more-valuable-than-oil-data-reigns-in-todays-data-economy/

[3] R. Ensafi, P. Winter, A. Mueen, and J. R. Crandall, "Analyzing the great firewall of china over space and time," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 1, p. 61–76, Jan 2015.

[4] J. Li. [Online]. Available: https://www.cse.wustl.edu/ jain/cse571- 07/ftp/p2p/

[5] M. M. I. Pattal, Y. Li, and J. Zeng, "Web 3.0: A real personal web! more opportunities and more threats," *2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies*, 2009.

[6] Q. Xu, Z. Song, R. S. M. Goh, and Y. Li, "Building an ethereum and ipfs-based decentralized social network system," in 2018 IEEE 24th International Conference on Parallel and Distributed Systems(ICPADS), 2018.

[7] Satoshi Nakamoto, "Bitcoin: A Peer-To-Peer Electronic Cash System," 31 October 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[8] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," Dec 2014. [Online]. Available: https://ethereum.org/whitepaper/

[9] "Validating confidential blockchain transactions with zero-knowledge proof," Hyperledger INDY, Apr 2019. [Online]. Available: https://www.hyperledger.org/blog/2019/04/25/research-paper-validating-confidential-blockchain-transactions-with-zero-knowledge-proof

[10] T. Berners-Lee, "The world-wide web," Computer Networks and ISDN Systems, vol. 25, no. 4-5, p. 454–459, 1992.

[11] B. White, "The implications of web 2.0 on web information systems," *Lecture Notes in Business Information Processing Web Information Systems and Technologies*, p. 3–7, 2007.

[12] W. Galuba, "Distributed hash table," *Encyclopedia of Database Systems*, p. 1190–1191, 2018.

[13] Kingma, F. H., Abbeel, Pieter, and Jonathan, "Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables," Oct 2019. [Online]. Available: https://arxiv.org/abs/1905.06845

[14] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The bittorrent p2p file-sharing system: Measurements and analysis," *Peer-to-Peer Systems IV Lecture Notes in Computer Science*, p. 205–216, 2005.

[15] A. Auvolat and F. Taiani, "Merkle search trees: Efficient state-based crdts in open networks," *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, 2019.

[16] L. Mo and G. Yao, "Multi-use conditional proxy re-encryption," *2013 International Conference on Information Science and Cloud Computing Companion*, 2013.