ISSN: 0971-1260
Vol-22-Issue-2-April-June-2019

Piracy Free Distributed Peer to Peer Content Sharing Using Distributed Ledger Technology

K Nimal Manickam
Department of CSE
Sri Sairam Engineering College
Chennai, India
nimalmanickam96@gmail.com

P Nishaanth
Department of CSE
Sri Sairam Engineering College
Chennai, India
nichuprabakaran@gmail.com

K V Pushparaj
Department of CSE
Sri Sairam Engineering College
Chennai, India
pushparajkvp@gmail.com

Mrs. B. Priya
Department of CSE
Sri Sairam Engineering College
Chennai, India
bpriya.cse@sairam.edu.in

Mr. Aravind Vasudevan
Member Technical Staff
Zoho Corp
Chennai, India
aravind.vt@zohocorp.com

***Abstract*—Distributed peer to peer file systems are the most reliable and stable means of transferring files. It doesn't require any cost of maintenance as there are no centralized servers to maintain. Anyone can share files easily without any subscription from cloud service providers and sometimes even without internet. But this technology has earned a bad name because of its misuse by people who share copyright content. The anonymity provided by the distributed peer to peer file systems like torrents are the reason for such piracy acts. Due to this content providers have completely ignored distributed peer to peer file sharing technology and rely on cloud service providers to deliver their content. We propose a piracy free distributed peer to peer content providing system that uses a distributed ledger to maintain the list of files that are uploaded to the swarm. The ledger will also contain payment information of the content provider and the price of the file. Only the devices that support screen capture prevention like android's secure layout will have content consumer applications. The file will be encrypted using password based encryption (PBE) and published to the swarm. Anyone can download the file present in the swarm but in order to get the decryption password payment has to be completed. Different types of file previewers will be programmed such a way that the file remains encrypted in the user's storage and decrypted for preview on the run. In order to prevent people from screen capturing content using virtual machines, the application will check the user's device and just close if it is a virtual machine. This provides extreme piracy protection to the content and also does not require any kind of investment.**

Keywords—PBE,P2P,Piracy,DFS

I. Introduction

Distributed peer to peer file systems are the most reliable and stable means of transferring files. It doesn't require any cost of maintenance as there are no centralized servers to maintain. Anyone can share files easily without any subscription from cloud service providers and sometimes even without internet. But this technology has earned a bad name because of its misuse by people who share copyright content. The anonymity provided by the distributed peer to peer file systems like torrents are the reason for such piracy acts. Due to this content providers have completely ignored distributed peer to peer file sharing technology and rely on cloud service providers to deliver their content. We propose a piracy free distributed peer to peer content providing system that uses a distributed ledger to maintain the list of

files that are uploaded to the swarm. The ledger will also contain payment information of the content provider and the price of the file. Only the devices that support screen capture prevention like android's secure layout will have content consumer applications. The file will be encrypted using password based encryption (PBE) and published to the swarm. Anyone can download the file present in the swarm but in order to get the decryption password payment has to be completed. Different types of file previewers will be programmed such a way that the file remains encrypted in the user's storage and decrypted for preview on the run. In order to prevent people from screen capturing content using virtual machines, the application will check the user's device and just close if it is a virtual machine. This provides extreme piracy protection to the content and also does not require any kind of investment.

II. Peet To Peer

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.

Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts.Peers are

ISSN: 0971-1260
Vol-22-Issue-2-April-June-2019

both suppliers and consumers of resources, in contrast to the traditional client-server model in which the consumption and supply of resources is divided. Emerging collaborative P2P systems are going beyond the era of peers doing similar things while sharing resources, and are looking for diverse peers that can bring in unique resources and capabilities to a virtual community thereby empowering it to engage in greater tasks beyond those that can be accomplished by individual peers, yet that are beneficial to all the peers.

While P2P systems had previously been used in many application domains,the architecture was popularized by the file sharing system Napster, originally released in 1999. The concept has inspired new structures and philosophies in many areas of human interaction. In such social contexts, peer-to-peer as a meme refers to the egalitarian social networking that has emerged throughout society, enabled by Internet technologies in general.

A peer-to-peer network is designed around the notion of equal peer nodes simultaneously functioning as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client–server model where communication is usually to and from a central server. A typical example of a file transfer that uses the client-server model is the File Transfer Protocol (FTP) service in which the client and server programs are distinct: the clients initiate the transfer, and the servers satisfy these requests.

IPFS is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. IPFS could be seen as a single BitTorrent swarm, exchanging objects within one Git repository. In other words, IPFS provides a high-throughput, content-addressed block storage model, with content-addressed hyperlinks.

III. Existing System

The existing system for a content delivery system make users buy the content but are not piracy free. Many of the existing systems support web platform to attract customers. This gives users the power to screen record and publish the content elsewhere.

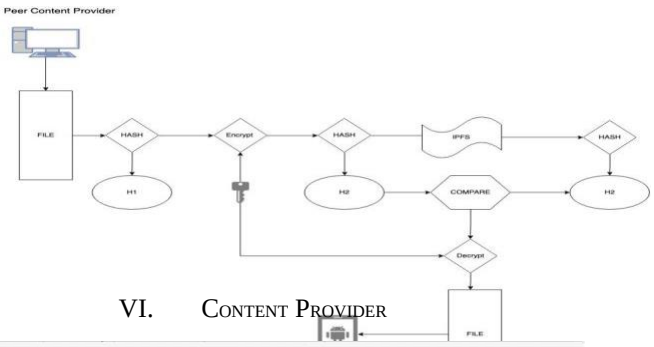
Distributed peer to peer file sharing systems like torrents are very optimized and allow content sharing at high speeds. But there is no concern for piracy in these networks. It is open for all and every file is treated differently. This disadvantage of bittorrent has earned it bad reputation and has become the home for pirates.

Existing content delivery systems are centralized and follow client server architecture.This makes the implementation costly and also involves maintenance cost. This also causes the problem of censorship and gives power to the company that hosts the servers. It is an existing example for single point of failure at the server.

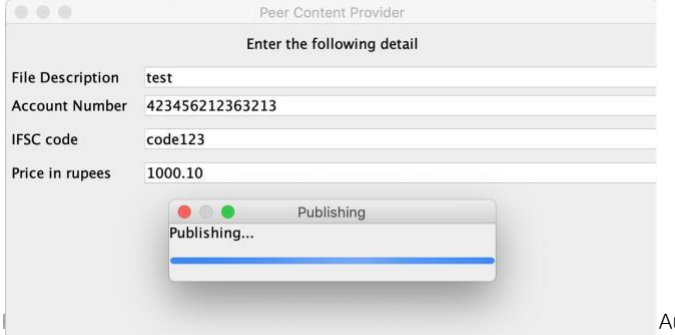
IV. Proposed Method

The proposed system is a piracy free distributed peer to peer content providing system that uses a distributed ledger to maintain the list of files that are uploaded to the swarm. The ledger will also contain payment information of the content provider and the price of the file. Only the devices that support screen capture prevention like android's secure layout will have content consumer applications. The file will be encrypted using password based encryption (PBE) and published to the swarm. Anyone can download the file present in the swarm but in order to get the decryption password payment has to be completed. Different types of file previewers will be programmed such a way that the file remains encrypted in the user's storage and decrypted for preview on the run. In order to prevent people from screen capturing content using virtual machines, the application will check the user's device and just close if it is a virtual machine. This provides extreme piracy protection to the content and also does not require any kind of investment. Duplicate publish will be prevented by comparing the both the hash of the files before encryption and after encryption with the ones in the swarm.

V. Architecture Diagram



VI. Content Provider



ISSN: 0971-1260
Vol-22-Issue-2-April-June-2019

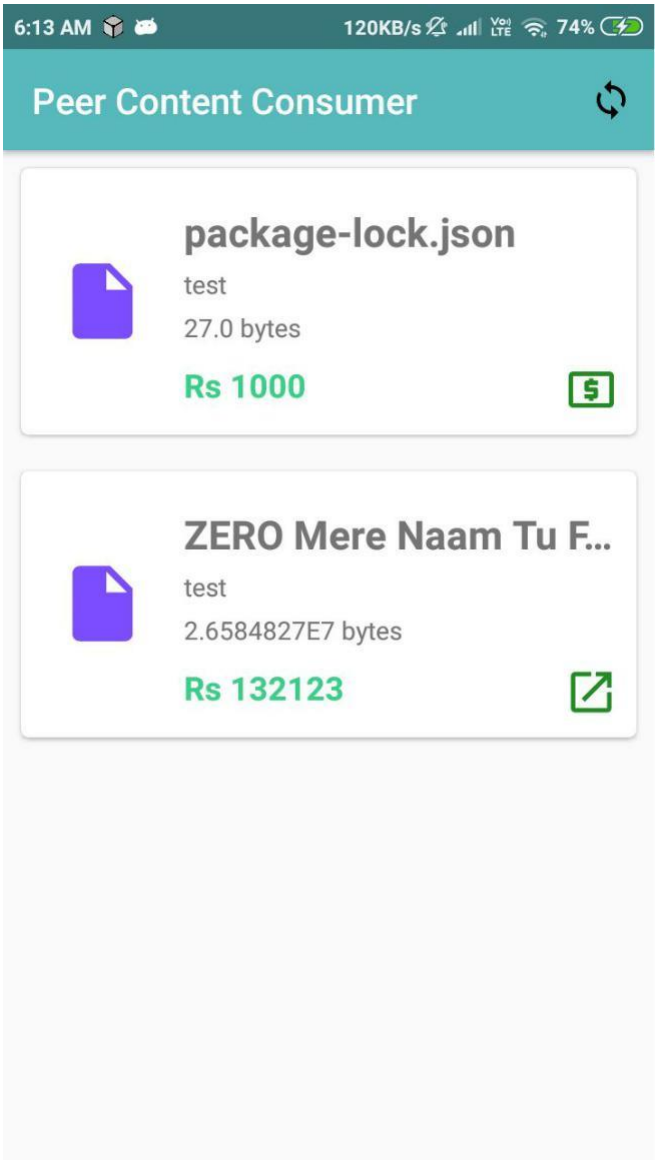
The content provider is a desktop application. It's purpose is to get the files from the user and publish it to IPFS as well as update the ledger. It also prompts the user for file metadata information so the buyer of the content knows what he is buying.

The desktop application is made with Swing application. It consists of a frame with initially a single button with "Open" text. On clicking the button is programmed to open the file picker. When the user selects the needed file, the frame is navigated to the frame where file details are gathered. The details that are gathered are file name,file description,payment Details and price.

Other metadata about the files like size, encoding etc are gathered using kotlin File class methods.

Once all the details are collected, the application tries to publish the file to IPFS if it is succeeded then the file hash is calculated. The hash and other file details are added to the ledger using the publish() method in PubSub class.

VII. Content Consumer



ISSN: 0971-1260
Vol-22-Issue-2-April-June-2019

The content consumer is an android application. That has ipfs binaries packed into it. It's purpose is to display the files in the ledger, preview the file,accept payments and get decryption key.

he android application carries the IPFS binaries as a resource file. When the app is ran for the first time. This binary is planted inside the devices storage and the absolute path is stored. Using the absolute path all the IPFS commands can be ran.

During each launch of the application, the ipfs daemon is started. Once it is started the ledger is fetched using the getLedger command in the pubsub topic. After getting the ledger details. It is first deserialized and converted to objects. This list of objects is then used to populate the listview in home screen.

User can click the listview item to download the file. When the download is clicked, the hash of the file is used to get the file from IPFS from other peers. Once the file is retrieved it is pinned by the application to serve other application users.

User can now make payment to get the decryption password and keys. Once the payment is confirmed, the application will use the publish method to get the key from the provider or other authentic peers. Once the key is received, user will be shown option to open the file. This
takes the application to a new activity called FileViewer activity. This is where the encrypted file is decrypted on the run and displayed.

VIII. Distributed Ledger

A distributed ledger is a ledger whose copy is held by all the peers in the network. The validation and integrity of the ledger is maintained by a trust mechanism. This is similar to crypto currencies that store transaction information in the ledger. This is basically similar to blockchain. In the proposed concept we will be storing the file metadata and transaction details in the ledger.

A blockchain is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block,a timestamp, and transaction data..

By design, a blockchain is resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way".For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority. Although blockchain records are not unalterable, blockchains may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been claimed with a blockchain.

IX. Encryption

The core strategy of the proposed system is that the file remains encrypted all the time after publishing. Only the previewers have the capability to decrypt the file on the run provided the key or passphrase is known. This eliminates a wide range of piracy concerns.

The encryption used is Password Based Encryption (PBE) with MD5 and DES. In password based encryption a salt is added to the file before encryption. Then a key is generated. The key is used to encrypt the file. The salt is used so that new output is produced for the same file every time.

X. Screen Capture Prevention

Screen capture prevention is complicated because it can be done at an hardware level where the application may not have access. In desktop that supports multiple monitors, we can't guarantee to prevent screen capture. This is the reason why it is better to only support mobile devices.

Mobile devices have control over the display being shown to the display. Hence it is possible to prevent screen capture in mobile devices. Hence the consumer applications can only be made for mobile devices and other devices that support screen capture prevention.

There is still one other way through which screen capture of content can be done. Pirates can run a virtual mobile device, install the application and capture from the desktop where the virtual machine is running. To prevent this we will have to check for the device specifications when the application is opened and prevent application usage if it is detected that the device is a virtual device.

XI. Conclusion

Content piracy control can be established in distributed peer to peer file sharing systems as well. Content providers can monetize their work without relying on any third party or pay a third party.

ISSN: 0971-1260
Vol-22-Issue-2-April-June-2019

References

[1] A Blockchain Based Truthful Incentive Mechanism For Distributed P2p Applications,Qingzhao Zhang , Yijun Leng , and Lei Fan1 in Shanghai Jiao Tong University, 2018.

[2] IPFS - Content Addressed, Versioned, P2p File System Juan Benet, creator of IPFS, 2017.

[3] Omniledger: A Secure, Scale-out, Decentralized Ledger Via Sharding by Eleftherios Kokoris-Kogias,Philipp Jovanovic,Linus Gasser,Nicolas Gailly,Ewa Syta and Bryan Ford at 2018 IEEE Symposium on Security and Privacy (SP).

[4] Cost-effective Index Poisoning Scheme For P2p File Sharing Systems by Yuusuke Ookita, Satoshi Fujita at 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS).

[5] Password-based Encryption Analyzed by Mart´ın Abadi1 and Bogdan Warinschi2 at Computer Science Department, University of California, Santa Cruz and Computer Science Department, Stanford University