Secure File Sharing in P2P (Peer-to-Peer) Networks

CS6415 - Network Security, Winter 2024

Mohammed Ghayasuddin
UNB
Fredericton, Canada
ghayasuddin.m@unb.ca

Rohith Kumar Saravanan
UNB
Fredericton, Canada
rohithkumar.s@unb.ca

Jayateja Alugolu
UNB
Fredericton, Canada
jayateja.alugolu@unb.ca

1 Abstract

This project aims to devise a secure file sharing mechanism tailored for Peer-to-Peer (P2P) networks, which face challenges in security due to decentralization and potential malicious activities. Leveraging cryptographic techniques and decentralized protocols, each user in the network is equipped with a unique cryptographic key pair for authentication and encryption. Additionally, a distributed trust model is implemented to ensure the integrity and authenticity of shared files. End-to-end encryption enhances privacy, while a hybrid architecture, combining centralized servers for authentication and decentralized file sharing among peers, strikes a balance between scalability, efficiency, and security. Extensive simulations and real-world testing evaluate the solution's effectiveness, focusing on metrics such as file transfer speed, resource utilization, and security robustness, aiming to contribute to the development of trustworthy data exchange in P2P networks.

keywords: *p2p networks, p2p file sharing, secure file sharing, confidentiality, integrity, availability, accountability, authenticity, encryption, hashing, blockchain*

2 Introduction

This argument is particularly applicable to P2P networks, where the pursuit of privacy (and the guarantee of file security) are central to avoid jeopardising the principles that follow from CIAAA:

(1) confidentiality
(2) integrity
(3) availability
(4) accountability
(5) authenticity

(the five key elements of caring for information). And most of those who think in this reductionist way nail all of that on decentralisation. Perhaps in that way they forget to think in systems, or that decentralisation could result in fewer vectors for malicious attacks than a centralised system. A transfer directly between both end users, with no need for an intermediate, has the advantage of reminding users that they are being watched. Likewise, a virtually infinite number of search options can be considered desirable due to the possibility for constant re-evaluation But decentralised P2P networks also bring enormous security challenges.for example to data integrity, malware proliferation and unintended 'outing' of identities. These expose vulnerabilities to confidentiality and integrity, which affect reliability and trust.

Another major concern is the threat of denial-of-service attacks (DoS), especially very elaborate Distributed Denial of Service (DDoS) attacks where certain networks and their processes are targeted to make them unavailable to users, reducing network stability. Issues concerning threats to integrity, availability and anonymity, as well as potential solutions, are part of the discourse in the project we launched to explore, on a theoretical level, how trustful and reliable the use and enjoyment of P2P networks can actually be and could become. Such discourse is important because, through the lens of **CIAAA**, it helps us to investigate viable security measures.

3 File Sharing in P2P Networks

Peer-to-peer (P2P) is a distributed model of computing that allows digital files to be shared through clients across the computers (nodes) of a computer network in a distributed manner, where clients of each participating node can act simultaneously as both servers and clients. P2P networks allow users to exchange the files directly (no need for a central server), that makes P2P more scalable, fault tolerant and anti-censoring.P2P file sharing consists of multiple parties with different roles and interaction processes.What they actually do is install P2P software (eg, **BitTorrent**, **eMule** or **Gnutella**), on each of which computer every other computer that installs such software has been similarly connected, as client to server, uploader to downloader.

The process is as follows: every time the end user wants to make a file available in a P2P sharing system, she has to choose the file she wants to share and add the name of the file to the directory of the P2P client software to share it. The P2P software then calculates a unique identifier – called 'hash' or 'magnet link' within the Torrentspeak – and copies it to share with other users to reassemble the shared file.If a user wants a copy of a given file, his P2P client sends out search queries, asking if its peers have the file (by its hash or metadata). Each search query might ask several other nodes in the P2P network the same question at once.

As soon as the file is located, the download begins. If the file is on an ordinary P2P network, its chunks or pieces are distributed across all the network's nodes. Each chunk is typically a few kilobytes to several megabytes in length. The downloading node sends requests to various nodes in the network (these nodes are called 'peers') for different chunks of the file they might have. As long as the chunks come from different peers, the download happens in parallel, making use of all the available bandwidth from the distributing nodes. For every byte of the file that the downloading node acquires from a peer, it checks its integrity - typically using cryptographic hashes or checksums - before filing it away. If a checksum fails, the client can ask for the chunk from a different peer.Meanwhile, the node that downloads the file will also upload the chunks it has obtained from the sources to other peers in the network. This activity of seeding (or sharing) the file increases its availability and reliability in the P2P network.

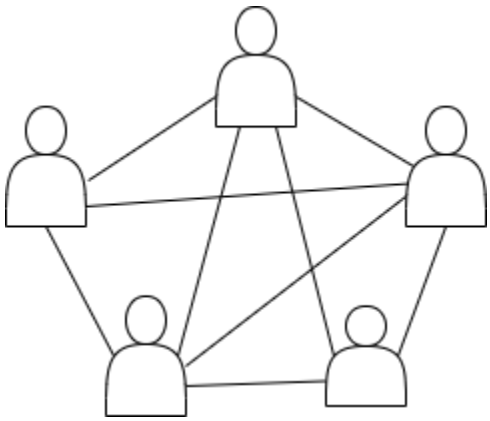

Figure 1: Peer-to-Peer Networks

When the downloading peer has received all the necessary chunks and stitched them together into a complete file, the download is finished. The peer is now free to share the file and can decide to seed the source to other peers on the network. But the network's resistance to both censorship and single points of failure is a primary benefit of file sharing. In a client-server architecture, there would be just one server storing the files and one authority governing access to those files; but in P2P, the files are distributed across multiple nodes, buffering against trouble. Yet for all its potential threats and perils – especially to copyright, a legal concept that's often been undermined by the unauthorised copying of intellectual property via P2P networks – P2P file sharing endures to this day. P2P applications themselves are not illegal, even though an illegal activity – sharing protected copyright material without permission – was closely linked to it. This has led to prolonged legal battles between host governments, copyright owners, and P2P file-sharing platforms and users over the fine line between the right to information and the right to property.

4 Issues to address in P2P File Sharing

We have seen how the emergence of so-called peer-to-peer (P2P) file sharing networks has brought a new paradigm of file sharing, where content does not flow among users through centralised servers, but may be shared file-to-file, user-to-user. The decentralised nature of P2P networks and their implications raise questions about securing the confidentiality, integrity, availability, authenticity and accountability of files shared among network peers. Confidentiality protects the exchange of sensitive data from unwarranted third-party access; integrity ensures that files are not tampered with during transmission; ensuring availability means that shared files and other content are accessible whenever normal network usage is expected, while authenticity ensures that the origin, and hence the trustworthiness, of files, and peers sharing files, can be confirmed; meanwhile, accountability mechanisms aim to prevent certain kinds of bad behaviour against peers, or to foster desired behaviour for the benefit of the P2P network.

4.1 Confidentiality in P2P File Sharing

When discussing P2P file sharing networks, it is required to mention the topic of confidentiality because the information (e.g sensitive and private ones) transferred from a computer can be crawled by other computers while they are in transmission or stored by other parties. Encryption is an important technical means to provide more security guarantee to P2P file sharing networks. It can solve all the problems related with confidentiality because the content exclusively belongs to a receiver is encrypted but during transmission or rest and can not be understanding by others without the key only. There are different level of encryption techniques. Such as SSL/TLS is used to encrypt communication. File-level encryption can be used to encrypt files before they are stored. For example, AES can be used with transferring data. Such techniques can all be used in P2P file exchanging networks to protect your privacy from other computers. Unless your key can be cracked and other computers know what your shared files mean.

4.2 Integrity in P2P File Sharing

Integrity of files in peer-to-peer (P2P, or p2p) file sharing is essential. Integrity means ensuring that the files shared with you remain the same as they the files; integrity guarantees the correctness and reliability of the files throughout the whole process of sharing. Files should not be downloaded modified, or tainted by corruption.

Integrity isn't just a quality that's tied to user trust, but also has legal ramifications. Distributing copyrighted materials without authorisation is a crime, and it's entirely possible for users to modify files inadvertently (and thus expose themselves, as well as the P2P network, to legal liability). Checksums, cryptographic hashes and digital signatures maintain the integrity of shared files. Peer verification maintains integrity through comparing and contrasting file contents against various peers in a P2P network to reach a consensus on its integrity.

Since P2P networks are fully decentralised, these measures would enhance users' shared content safety, thus reducing the legal risk associated with P2P usage (including pirated software and movies), and, eventually, building users' confidence in what they share with their peers.

4.3 Availability in P2P File Sharing

This redundancy is crucial to keep files available to consumers even when P2P networks suffer from specific network connectivity issues, hacker-attacked nodes and moments when lots of people give up being seeders and become downloaders simultaneously. Redundancy in P2P networks can be achieved by means of file replication in other nodes of the network and of distributed storage architectures. This permits avoiding the vulnerability of relevant unique points of failure that can generate, under certain conditions, the network content inaccessibility. It's possible to create incentives such as gift tokens for seeding content for longer periods of time Or to implement reputation systems that reward users according to their average time spent seeding.

4.4 Authenticity in P2P File Sharing

To make sure that malicious or modified content isn't disseminated, and that peers aren't potentially collaborating with persons

or groups with malicious intent, the authenticity of shared files and the identity of peers in these P2P networks also need to be verified. To this purpose, peer authentication mechanisms, based on the issuance of digital certificates or public key infrastructure (PKI), can be used to create a trust framework between users. In addition, reputation systems or peer rating mechanisms based on past behaviour and community feedback help users identify the trustworthiness of peers and shared files. When the authenticity of files and peers is verified, the risks associated with cyber fraud and identity theft can be mitigated, thus ensuring P2P apps and social networks that buyers and sellers can confidently engage in business.

4.5 Accountability in P2P File Sharing

Building up the accountability for user actions is necessary to discourage malicious usage patterns and to ensure the integrity of P2P networks in the long run. As logging and auditing capabilities to monitor and analyse user behaviour in P2P networks are lacking, operators need to implement consistent mechanisms for logging and auditing user activity on the network. Additionally, some measures have to be taken to enforce accountability for user actions, for example, creating community-based mechanisms for setting up community norms and guidelines with appropriate sanctions for violators. For this purpose, an underlying logic of social and prudential norms for the group of users is critical. Similar to the Airbnb case discussed earlier, the reputation of users can play a crucial role in enforcing norms within P2P networks. Therefore, a culture of responsible sharing that holds users accountable for their actions by policing and deterring 'bad' behaviour can be promoted in P2P networks.

5 Ensuring Confidentiality in P2P File Sharing Networks

The ways often proposed in maintaining confidentiality in file-sharing in P2P networks focus on File Encryption and maintaining Anonymity of the senders and the receivers. The first approach is based on two fundamental components, namely, the Peer-to-Peer Node Discovery Protocol (PPNDP) and the implementation of Diffie-Hellman algorithm for secure key exchange [1], which enable secure file transfer system aiming to assure the confidentiality of the P2P file sharing.

The **Peer-to-Peer Node Discovery Protocol (PPNDP)** oversees discovering other nodes on the local area network (LAN) without having to explicitly advertise one's identities. PPNDP uses the **Internet Group Management Protocol (IGMP)** to allow nodes to query and respond to each other without exposing more than necessary. A crucial step toward creating peer-to-peer security is ensuring that nodes can discover each other without compromising anonymity and confidentiality.
Secondly, **Diffie-Hellman**. Diffie-Hellman was an algorithm for key exchange to ensure secure links between nodes. This allows a secret key as input to the encryption. For example, to exchange a secret key between two parties who have no secure channel on which to communicate the key itself the Diffie-Hellman algorithm generates and shares intermediate values such that the two end-points

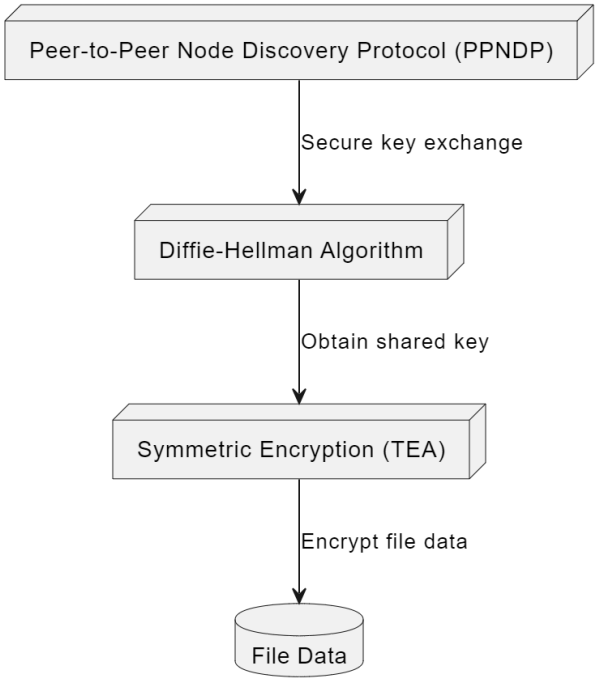

Figure 2: File Encryption with Diffie-Hellman Algorithm [1]

can calculate the key independently from the other intermediate information. As an RSA algorithm, Diffie-Hellman enables a secure handshake between nodes for file-transfer [1]. Only a node that has proven to possess the secret key will have permission to download a file.
Besides, as another means of protecting the file contents from being eavesdropped during transmission, it adopts symmetric encryption. Symmetric encryption is performed on file data using the Tiny Encryption Algorithm (TEA) after a connection is securely established with the shared key obtained by Diffie-Hellman negotiation between nodes [1]. File data is encrypted before being transmitted so that only machines with the shared key can decrypt the file contents. Since TEA provides a good level of encryption, all file contents can be guaranteed to remain confidential.

To conclude, the system ensures safety by applying various secured protocol in P2P file sharing network with end-to-end encryption to establish secure connection without central authorities required and presents opportunistic routing and secure communication over the World Wide Web. The file transfer process works on a model of secure-by-design paradigm including **PPNDP**, **Diffie-Hellman** algorithm, and **TEA encryption** for secure data communication where files transfer between nodes with full confidentiality ensuring the privacy. Hence, it significantly reduces the possibilities of unauthorized access, hijacking, theft of data, and deterioration of privacy. This system will enhance the security and privacy in P2P file sharing network where the decentralized model for ensuring security and privacy is suggested along with secured quantitative

approach. Hence it significantly reduces the possibilities of unauthorized access, hijacking, theft of data and deterioration of privacy. This kind of communication model is successfully implemented in local area network where software clients install in peers to ensure the security of file-sharing by using TEA encryption and AES encryption for anonymous and secure communication. This approach can be generalized for all the systems of P2P file sharing over the World Wide Web [1].
Important work has also gone into ensuring confidentiality in P2P file-sharing systems by describing and analysing some popular anonymous peer-to-peer file-sharing systems in detail. Here we propose you to become briefly acquainted with some fundamental abstractions regarding the ideal properties of confidentiality, such as the subtle and important differences between sender anonymity, responder anonymity and sender-responder unlinkability, before delving into the next paragraph. Anonymity will help to provide some background on the different mechanisms that anonymous P2P systems use to hide user identity and protect confidentiality.

Furthermore, we already have numerous anonymization-based anonymous P2P file-sharing systems: from ant P2P networks that use the Ants protocol, or even networks based on Onion routing and Mixes [2]. We evaluate each one to verify how good it is at protecting the anonymity of users, that is to say, what kind of protection it guarantees against attacks: sender anonymity, responder anonymity, etc. There are also real-world attacks that threaten the confidentiality of file-sharing in P2P networks, like time-to-live attacks, statistical attacks, and denial of service attacks, each of which is a specific attack that endangers the anonymity guarantees of these and similar systems. Identifying these vulnerabilities means having a wealth of knowledge about what to strengthen in an anonymous P2P system in the face of possible attacks [2]. The insights and findings provide a roadmap for the development of more robust and resilient anonymous P2P systems, ultimately enhancing the privacy and confidentiality of users engaged in file sharing activities.

Anonymous File Sharing Software and Underlying Protocols [2]:

Name	**Based on**
Ants	Ants
AP3	Crowds
APFS	Onion routing
Entropy	Freenet
GNUnet	MIXes
Free Haven	Secret sharing and MIXes
I2P	Onion routing
Mantis	Ants and UDP spoofing
Waste	Friend-to-Friend
SSMP	Secret sharing and Onion routing
Tor	Onion Routing
Napshare	Ants
Nodezilla	Freenet
Mute	Ants

The proposed schemes for confidential sharing of files through the development and analysis of the **AnonymP2P** protocol help address one of the most important concerns in Peer-to-Peer (P2P) communication, namely ensuring the anonymity of both the requesters and the suppliers of resources in P2P networks, and thus keeping the sensitive information confidential throughout the process of sharing. First, its basic protocol design features applicant anonymity. In contrast to approaches used in traditional protocols where requests are transmitted to all neighbour nodes, AnonymP2P distributes requests with Mobile Agents that move randomly from node to node to look for available resources [3]. The random distribution makes derivation of the requester's source node less likely for an attacker, while anonymity is achieved through a set of deterministic algorithms maintaining a series of agents going out from the source node. Furthermore, the protocol uses TTL-based obfuscation to disrupt the mapping of TTLs against a background model of inferences about the sources of data. Additionally, supply chain anonymity, provided by the use of AnonymP2P and vital for protecting the identities of nodes sharing resources using AnonymP2P, counters timing attacks, by artificially delaying the responses for queries received from peers they do not trust, thus preventing attackers from inferring the distance between data sources and themselves by measuring response times [3]. Moreover, the protocol's dynamic network topology makes collusion attacks impractical, as the topology changes render it unlikely for attackers and accomplices to be directly connected to targeted nodes.

In terms of data exchange, AnonymP2P ensures both confidentiality and integrity through encrypted responses and Diffie-Hellman key exchange [1][3]. Despite the potential risk of Man-in-the-Middle attacks, the protocol provides a mechanism for the requester to verify the integrity of received responses through multiple key exchange paths. This robust approach to data exchange safeguards against unauthorized access and tampering, thereby enhancing confidentiality throughout the file-sharing process.

Overall, there are significant strides in addressing the confidentiality challenges inherent in file sharing within P2P networks. By proposing and analyzing the AnonymP2P protocol, we can achieve practical solutions to maintain anonymity for both requesters and suppliers while ensuring the confidentiality and integrity of shared data. These contributions have implications for enhancing privacy and security in P2P file-sharing environments, making them more resilient to potential threats and attacks.

6 Ensuring Integrity n P2P File Sharing Networks

In a peer-to-peer (P2P) file-sharing network, integrity is the basic constituent that governs the way data is exchanged across decentralised processing nodes without it, computing couldn't be secure, let alone trusted. When digital information has multiplied by three orders of magnitude on the internet over the past five years, this represents a commensurate exponential increase in data and its organisation, and in the requirement to ensure its security and integrity. The protocols that we presently have at our disposal face considerable challenges in managing this growth with the integrity we've come to expect [4].

With more and more content passed around in digital-first exchange formats in this moment of P2P, there's a growing need for more robust integrity controls to operate in those systems. Many approaches now in use are based on older centralised models for data management and forms of verification that are not well suited to the sorts of exposures and complexities emergent in distributed systems. The need for new approaches that will allow users and networks to actually prevent, detect or correct risks and maintain data integrity across distributed environments is a growing niche in cybersecurity, distributed computing and network engineering [4].

Among the most promising possibilities for this work is the intersection of various existing technologies such as InterPlanetary File System (IPFS) and Blockchain, a distributed network that can be used to permanently and publicly store data without central authority. IPFS has a distributed architecture where file storage and retrieval is disseminated through a network of file-sharing nodes: it shifts away from the traditional centralised client-server model to a peer-to-peer model which allows for increased efficiency and scalability, as well as new possibilities towards redundancy, fault tolerance, and resiliency against single points of failures [4].

On the other hand, Blockchain technology can help ensure the integrity of the data in P2P networks as an immutable (unalterable) ledger that cryptographically guarantees the integrity of transactions and data records through a consensus-based, decentralised bookkeeping mechanism. If the file metadata is chained to a Blockchain, the data stored in P2P networks is immutable and verifiable [5]. This way, whenever data is tampered with, unauthorised access occurs, or the data is erased, modified or corrupted, one can prove the illegal or unauthorised nature of the act. However, the 'decentralised' nature of many P2P networks contains numerous challenges and vulnerabilities, which require novel solutions and robust integrity checks. Malicious actors, network bugs and systemic flaws can easily corrupt the integrity of shared files, compromising their authenticity, verifiability and trustworthiness. Standard approaches to data integrity, such as digital signatures and error-correcting codes, don't work well for these challenges. What's needed is novel approaches that can address these issues. In this respect, Blockchain technology brings a justified hope: a distributed model of trust that is based on a consensus-building, distributed integrity-checking and encryption paradigm [5]. The idea that verification of the integrity of files is distributed and no part of the trust is with any single player reduces the risk of single points of failure, thus ensuring robustness and resilience in the P2P ecosystem. At the heart of this solution is the use of cryptographic primitives and consensus algorithms combined with economic incentives to encourage participation, validate transactions and transactions, and create a trust system through a decentralised blockchain. A consensus algorithm based on a distributed ledger on the blockchain and smart-contract management creates a trustless verification system without any central authorities or mediators.

Furthermore, the alignment of economic incentives (such as tokenisation or incentivised validation schemes) creates an alignment of interest for nodes to act according to the rules of the protocol, to validate transactions, and to protect the integrity of the network. This feedback loop between technology and economics creates a self-sustaining loop of integrity, where collective action, consensus and economic incentives maintain system integrity. When combined, it's possible to share P2P files over IPFS in a decentralised, tamper-proof and fault-tolerant way. With IPFS, the decentralised storage and retrieval of files can be done at a lower cost and in a more equitable way; with Blockchain, its integrity can be guaranteed: the data is immutable and verifiable. More importantly, the combination of IPFS and Blockchain could signal a profound change in our approach to cybersecurity, distributed computing and network engineering. Overturning entrenched paradigms and establishing trust as the basis for digital interactions could lead to a future in which integrity, transparency and accountability underpin life online [5].

Overall, seeking integrity in P2P file-sharing networks is a three-pronged effort that combines technological wizardry with cryptographic principles and economic incentives [4][5]. IPFS and Blockchain have the power to decenter the web and secure data integrity, fostering trust in our emerging digital jungle. As the digital society makes strides forward, ensuring integrity in P2P file-sharing networks will become a guiding light for the future of humanity, hoping to make the digital world a safe, transparent and trustable place.

7 Ensuring Availability in P2P File Sharing Networks

If sharing is a public good in such a P2P file sharing system then we are interested in understanding how to maintain availability of content. As is the case with most public goods, the marginal cost for each peer of contribute to the shared pool of files is positive. Common economic models tell us that maximisation of social welfare is achieved when we impose fixed contribution schemes – in other words, it is best when everyone contributes in the same way. However, we soon see that fixed contribution schemes are painful to satisfy - particularly in real world P2P networks that have no concept of a central authority that can force contributions. The response to this, is principled, in the sense that it suggests an enforceable set of fixed contribution schemes that do not impose any reliance on memories, nor any centralised entities as other fall-back mechanisms do. Rather, such a scheme is based on the fact that time spent by peers consuming resources can be used as a guide for contribution.

The proposed scheme adds a functionality to enforce contribution but does so without relying on a centralised controller, eliminating the punitive incentive issues that arise in such a distributed environment. It also presents an economic model to estimate the economic distortion of the proposed mechanism when compared with theoretical optimal mechanisms based on the amount of information about other users that is available. The preliminary results indicate that the proposed mechanism is not theoretically optimal, but nevertheless may be the best trade-off between economic efficiency and practical feasibility, at least in the context of content availability in P2P systems [6].

Essentially, the scheme proposed in the paper provides a new measure to encourage sharing of content by treating peer-resource consumption (as a result of bandwidth resource) as the quantification for contribution; addressing the relevant factors regarding feasibility and economic considerations, it presents a realistic scheme

of sharing in P2P networks, and hence contributes to the efficacy of P2P file sharing systems in terms of ensuring universal sharing.

7.1 Availability Using SeAl:

The SeAl (Selfishness-Awareness) architecture employs new ways to maximise availability, making sure requested files are actually available, based on the participants behaviour. SeAl monitors and accounts for selfish behaviour while putting five different mechanisms in operation. SAL, which stands for Selfishness and Accounting Mechanisms, monitors peers for selfish behaviour, and keeps track of who your friends are and who you don't want to do business with. If you try to be a free-rider, the SAL will make the whole world know about it, thus encouraging you to cooperate with the others. After all, if you don't cooperate, you will not have available your resources yourself. It's a good idea to use the same rules you give to others on yourself [7].

Another essential part of the SeAl is the Verification and Auditing Mechanisms (SVL).

- SVL is a collection of cryptographic-based mechanisms that ensure trust among peers and against malicious actors influencing the network by corrupting accounts, modifying resources, or altering content.
- SVL works as a defence mechanism against availability attacks, protecting the integrity and reliability of content and the network itself.
- In detail SVL uploads receipts as proof of transaction delivery to later check that all peers receive the requested delivery SVL employs blacklisting of malicious peers in the resource-allocation protocol

Ad-hoc networks can provide a platform for sharing content, but they also offer the potential for otherwise unaccountable individuals or registrars to hijack transaction routing and corrupt content. The more complex the scenario and the more peer-to-peer participants, the greater the danger this presents. SVL is essential to assure trust, accountability and, most importantly, high efficiency and reliability of content-sharing [7].

Additionally, SeAl tries to promote 'peer-to-peer altruism' requests and get prioritised according to the reputation score of the peers. If the reputation score is high, the peer will get treated well. This creates an incentive for participation in the larger co-operative system, giving her more resources to copy and share with others. SeAl uses shaming penalties to deter selfishness, and reward co-operation The whole process benefits everyone by making sure resources are still available. They conducted tests that showed that SeAl improved efficiency and promoted co-operation, thus increasing availability overall. From SeAl's origins, More players were added, policing became less and less effective and about 10 per cent of the time available to the computer was spent dealing with BitTorrent adoption requests that repeated on a bi-hourly and bi-daily basis. The biggest problem was that, besides requesting bits, policing the system involved small amounts of processing frequent time. Finding the resources to police was becoming increasingly difficult.

7.2 P2P File Sharing DoS Resilience

A vulnerability present in P2P file-sharing networks consisting of a potentially large number of hosts is their susceptibility to denial-of-service (DoS) attacks. Such an attack would be targeted to impair the network's ability to serve users, for example by flooding the network with irrelevant and false information.
DoS attacks could take two forms:

(1) File-targeted
(2) Network-targeted

- In the first case, the goal is to flood the network with unrequested additional copies of a given file or files.
- The second involves flooding the network with fake nodes that, upon query, reply with misleading information.

The file-targeted attack could work if clients refuse to share files that are being flooded, or even so crippling that clients stop downloading while the attack is in execution. While this attack can work in the short term, in cooperative P2P systems it is unscalable. However, starting at a minimum level of nodes participating in the system, a network-targeted attack can easily impair performance, and even shut down the system altogether. Network-targeted attacks require orders of magnitude fewer attackers to achieve similar effects than do file-targeted attacks.

However, in response to these outrages, other experts have suggested counter-measures, including reply selection at random, multiple copies of redundant/parallel downloads of the same content, and reputation systems. Sadly, the result is of the usual hardware versus software kind, the implementation of any of the suggested counter-measures would result in a fuzzier, therefore more robust, system with an attacker-dependent randomisation introduced. unfortunately, this is likely to result in a partial or full degradation of the system's performance when other non-stochastic stresses are present, and in conjunction with such stresses, reputation systems are likely to be ineffective unless they are, 100 per cent of the time, flawless. Disruptive events are then a problem of system fragility and of the hardest 'hard-versus-easy' trade-off of them all, identifying the best hardness [8].

To sum up, the discussion provides us with five main conclusions about DoS resilience in a P2P file-sharing system, file-targeted attacks are ineffective in collaborative settings, network-targeted attacks are more powerful because of their greater scalability, effective counter-strategies are available but come at a cost, overall resilience necessitates a holistic understanding of the whole system, network topology and clients behaviour, as well as the effect of actively implemented countermeasures in host systems.

8 Ensuring Accountability in P2P File Sharing Networks

Since there are always weaknesses and limitations present in Centralised File-Sharing System, this **Distributed File Sharing System (dFSS)** which leverages **Blockchain**, **Interplanetary File System (IPFS)**, and **Public Key Infrastructure (PKI)** technologies to maintain accountability [9]. Accountability is as important as transparency in P2P networks, as it helps users recognise when others behave dishonestly, and it is also one of the key components of reducing inherent security issues. This can be mitigated

through decentralisation as it eliminates the need for third-party regulators and promotes transparency throughout the file-sharing process. Using Blockchain to authenticate files through immutable record-keeping and the public and visible validation process helps in holding network users accountable.

The system's operational accountability models are also anchored in a system of smart contracts on the Blockchain network. Using smart contracts means that contract terms are automated and enforced - for example, only exclusive or standard music files can be shared on the system and only the people with access (i.e., the ones that paid for them) can access them. These access parameters are encoded into the smart contracts, and the system cannot share files outside of these rules. Thus, the smart contracts embed access control mechanisms directly in contracts, making it much harder for others to share files outside the rules. In summary, an integrated blockchain system for P2P file sharing could increase accountability by enforcing financial accountability measures through Monet. It also recovers value for artists. And, most importantly, it reduces the risk of privacy breaches for users, with encrypted transactions and communications, and smart contracts encrypted into the Blockchain.

Furthermore, cryptographic authentication mechanisms, i.e. **MetaMask** and others, in improving accountability on P2P networks [9]. Cryptographic digital signatures and encryption can be used to make sure that only legitimate users with known identities are sharing files on the network. Cryptographic digital signatures ensure that file-sharing actions can be attributed to a specific user [9]. Users' private keys sign running programs, so other users in the P2P network can verify that the program has not been modified. Strong encryption algorithms, i.e. **AES**, are used to protect the confidentiality and integrity of the shared files. This helps accountability by preventing access to the files by unauthorised persons or tampering with them.

Beyond the technological methods of accountability there is a wider question of the ethical dilemmas posed by P2P networks for the public good, including issues of trust and security. P2P relies on the principle of open and honest transactions. By paying attention to the surveillance issues raised by BitTorrent users, we can see not only what file-sharers should do, but what kind of robust system we need to build if these networks are going to be sustainable.

By utilizing the feasibility and efficacy of decentralized file-sharing systems, we can achieve accountability and transparency aligning with broader discussions surrounding digital governance and cybersecurity, making it relevant to policymakers seeking to foster trust and security in digital ecosystems [9]. By leveraging advanced technologies and innovative approaches, the proposed system can lay the groundwork for a more transparent, efficient, and accountable file-sharing ecosystem, with far-reaching implications for trust, security, and collaboration in the digital age.

Another way of maintaining accountability is by using Consortium Blockchain Technology and designing a decentralized peer-to-peer file storage and sharing system [10]. Starting with the problem that makes centralised storage solutions (encompassing everything from data centres to the cloud) difficult not only technically but also from a data-rights point of view as regards trust, it is possible to gauge some limits of current approaches to decentralised

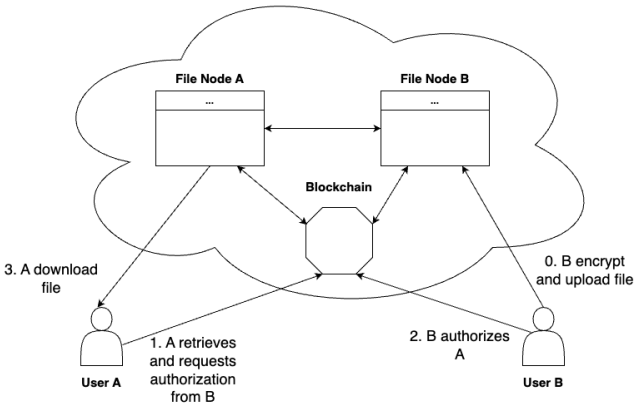

Figure 3: Secure P2P File Sharing using Blockchain [9]

blockchain-based storage and to propose an alternative based on consortium blockchain technology.

The proposed system is a peer-to-peer file storage architecture in which organisations work together to create a consortium blockchain network. Each organisation offers peer file nodes which are responsible for the following functions: blockchain data interface, file storage, identity authentication, access control and file retrieval [9][10]. To identify individual nodes, the system supports a strong authentication layer (identity access into the Storage Nodes) based on a Certificate Authority (CA) and a Member Identity Provider (MSP). At the certificate level, a digital certificate is issued by an outside body, such as the CA, and is then verified against a digital signature by the MSP.

Other access control mechanisms complete these accountability measures by checking if an organisation is given permission to access a file. Permission grantors and verifiers are required in granting access permissions and verifying authorisation requests, respectively. According to these access-control rules, correct behaviours of a user should follow carefully to ensure accountability in these types of activities [10].

This peer-to-peer file storage process features steps to ensure equitable distribution of storage work between organisations, and to prevent access or modification of files by potentially unauthorised nodes within the network. It helps maintain the principle of accountability. Moreover, a fully fledged system that tackles retrieval, authorisation and sharing issues to allow easy and responsible file sharing among organisations can be created. Using a complete full-text retrieval strategy that systematically indexes file names, along with the recognised authorisation mechanisms, allows the users in the system to retrieve and share the files securely, while also keeping track of all the file sharing activities.

The proposed system may be implemented by testing the write and reading performance on file. This would help to check the feasibility and viability of our proposed system in terms of performance by showing its viability for file activity, accountability, and data integrity [10]. Overall, leveraging **Consortium Blockchain** technology and implementing robust authentication, access control,

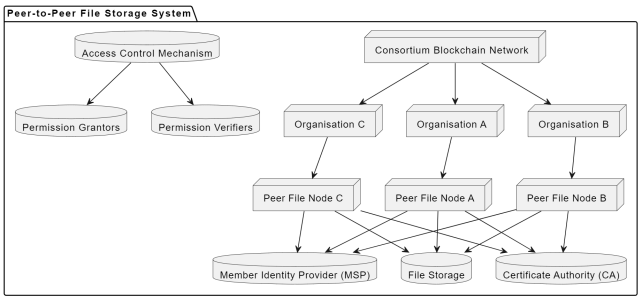

Figure 4: Identity Authentication and Accountability in a Consortium Blockchain [10]

and file storage mechanisms contribute significantly to maintaining accountability in file sharing within P2P networks [9][10]. This comprehensive approach ensures that organizations can securely share data while adhering to accountability standards, thereby addressing the challenges associated with contemporary data sharing paradigms.

9 Ensuring Authenticity in P2P File Sharing Networks

In the context of peer-to-peer (P2P) file sharing to deal with this complex challenge of integrity and authenticity security, a special attention to dynamic and decentralised nature of mobile environments and absence of restriction to P2P networks where nodes are transient, unplanned, or lack any centralised authority. The protocol may be described by three main subprotocols **Join**, **Content Authentication**, and **Content Access** [11].

The entrance of new nodes into the P2P network relies on running the Join subprotocol which issues authorisation certificates for sharing content with them. The Content Authentication subprotocol focuses on verifying the authenticity and integrity of shared content. Content owners generate content certificates containing essential information such as the originator of the content, a hash of the content for integrity verification, and a list of signers. These certificates are recursively signed by a subset of trusted peers to establish their authenticity. Additionally, cryptographic puzzles are employed to protect the content from unauthorized decryption, ensuring that only users with the proper authorization can access the decryption key and retrieve the content.

Authorized users obtain access to the shared content by obtaining the necessary authorization certificates and solving cryptographic puzzles to access the decryption key [11]. The users already possess valid authorization certificates and those where they need to initiate the Join subprotocol to obtain clearance. It emphasizes the importance of verifying the authenticity and integrity of the content after decryption to ensure that it has not been altered or tampered with during transmission.

Furthermore, a thorough security analysis, considering potential attack scenarios and how the protocol mitigates them is done. Measures to prevent eavesdropping, message modification attacks, message replay attacks, and other forms of malicious behavior

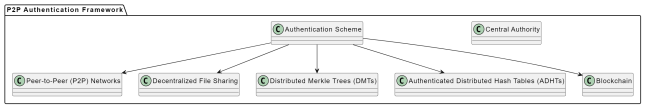

Figure 5: Maintaining Authenticity in P2P Networks [11][12]

are considered. Additionally, an efficiency analysis is conducted to assess the computational and communication overheads of the protocol, ensuring that it remains practical for use in real-world P2P environments.

In summary, the protocol proposed offers a robust and comprehensive solution for maintaining authenticity in file sharing within P2P networks, particularly in mobile and decentralized environments. By incorporating mechanisms for authorization, authentication, and content integrity verification, the protocol ensures that only authorized users can access and share content while protecting it from unauthorized access or tampering. This contributes significantly to the security and reliability of file sharing in dynamic and decentralized network environments.

On top of that, to maintain authenticity within peer-to-peer (P2P) networks, where decentralized file sharing poses unique challenges due to the absence of a central authority to oversee data integrity, a sophisticated authentication scheme centered around two key components: **Distributed Merkle Trees (DMTs)** and **Authenticated Distributed Hash Tables (ADHTs)** can be implemented [12].

Exploiting key properties of distributed hash tables (DHTs). DMTs is an adaptive logarithmic protocol for storage and query complexity of authentication based on hash chains. DHTs exploit the structured randomness of hash functions to distribute hash values and authentication verification paths across DMT network nodes in a balanced manner to do authentication quickly and minimise the storage and communication overhead. The two research papers have a technical presentation of DMT construction. The paper represents the ongoing research, and draws on network theory, modelling, and stability for understanding the construction of DMTs using weight-balanced trees and route distribution, enabling optimised operations of verification path retrieval and update.

Built on top of the DMTs, the authenticated data types (ADHTs) carry over authenticated operations of put, get, and remove to data sets, which could be of dynamic nature. The ADHTs leverage the signature amortisation techniques: the source node would compute and hold a cryptographic digest of the data set and sign the digest along with the query or update request. Inside a DMTs, we can efficiently verify if the data are still authentic, in addition to reducing the storage and communication overheads. All the ADHTs protocols for granting and enforcing authenticated operations demonstrate that the scheme is provably secure against a broad range of attacks.

By integrating DMTs and ADHTs into existing P2P file sharing protocols and platforms, such as BitTorrent or IPFS (Interplanetary File System), the envisioned authentication framework can bolster the security infrastructure of these systems without imposing significant overhead on network resources or user experience [12]. Also, the extensibility of the proposed scheme makes it a viable

candidate for further research and development, facilitating the exploration of new authentication schemes and optimisations for specific use cases or network architectures. We can also conduct a performance evaluation and simulation studies to determine the performance and scalability of the proposed solution and provide empirical evidence of its effectiveness and contribution to authenticity preservation in P2P networks.

In conclusion, Blockchain, along with authentication structures and protocols, such as DMTs and ADHTs, offer a scalable and secure framework for verifying data authenticity in decentralized environments [11][12].

References

[1] Jun Liu, Yuying Zhu, Hao Wang, Jia Guo, and Mingjie Guan. Design and implementation of a secure peer-to-peer file transfer system based on ip multicast and diffie-hellman algorithm. In *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, pages 520–524. IEEE, 2022.

[2] C TOM. A survey of anonymous peer-to-peer file-sharing. In *IFIP International Symposium on Network-Centric Ubiquitous System, 2005*, 2005.

[3] Mohamed Amine Riahla, Karim Tamine, and Philipe Gaborit. A protocol for file sharing, anonymous and confidential, adapted to p2p networks. In *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pages 549–557. IEEE, 2012.

[4] Chaitanya Rahalkar and Dhaval Gujar. Content addressed p2p file system for the web with blockchain-based meta-data integrity. In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pages 1–4. IEEE, 2019.

[5] Dongdong Yue, Ruixuan Li, Yan Zhang, Wenlong Tian, and Chengyi Peng. Blockchain based data integrity verification in p2p cloud storage. In *2018 IEEE 24th international conference on parallel and distributed systems (ICPADS)*, pages 561–568. IEEE, 2018.

[6] Panayotis Antoniadis, Costas Courcoubetis, and Ben Strulo. Incentives for content availability in memory-less peer-to-peer file sharing systems. *ACM SIGecom Exchanges*, 5(4):11–20, 2005.

[7] Nikos Ntarmos and Peter Triantafillou. Seal: Managing accesses and data in peer-to-peer sharing networks. In *Proceedings. Fourth International Conference on Peer-to-Peer Computing, 2004. Proceedings.*, pages 116–123. IEEE, 2004.

[8] Dan Dumitriu, E Knightly, Aleksandar Kuzmanovic, Ion Stoica, and Willy Zwaenepoel. Denial-of-service resilience in peer-to-peer file sharing systems. *ACM SIGMETRICS Performance Evaluation Review*, 33(1):38–49, 2005.

[9] Md Nasim Uddin, Abu Hayat Mohammed Abul Hasnat, Shamima Nasrin, Md Shahinur Alam, and Mohammad Abu Yousuf. Secure file sharing system using blockchain, ipfs and pki technologies. In *2021 5th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–5. IEEE, 2021.

[10] Shaoliang Peng, Wenxuan Bao, Hao Liu, Xia Xiao, Jiandong Shang, Lin Han, Shan Wang, Xiaolan Xie, and Yang Xu. A peer-to-peer file storage and sharing system based on consortium blockchain. *Future Generation Computer Systems*, 141:197–204, 2023.

[11] Esther Palomar, Juan ME Tapiador, Julio C Hernandez-Castro, and Arturo Ribagorda. Secure content access and replication in pure p2p networks. *Computer Communications*, 31(2):266–279, 2008.

[12] Roberto Tamassia and Nikolaos Triandopoulos. Efficient content authentication in peer-to-peer networks, July 5 2011. US Patent 7,974,221.