HAW
HAMBURG

Bachelor Thesis

Gustavo Andrade

Technical Analysis of Reputation Systems based on Blockchain Technologies

Fakultät Technik und Informatik
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Gustavo Andrade

Technical Analysis of Reputation Systems based on Blockchain Technologies

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of European Computer Science
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter: Prof. Dr. Philipp Jenke

Eingereicht am: 05. Januar 2019

Gustavo Andrade

Thema der Arbeit

Technical Analysis of Reputation Systems based on Blockchain Technologies

Stichworte

Blockchain, Reputation, Peer-to-Peer

Kurzzusammenfassung

Das Ziel dieser Arbeit ist die Analyse eines Reputationssystems, das in einer Blockchain implementiert werden soll.

Blockchain ist aktuell ein heißes Thema. Es gibt viel Forschung zur Weiterentwicklung der Technologie und es gibt diverse Projekte die Blockchains in verschiedene Szenarien nutzbar machen. Blockchain als dezentrales System ist sehr leistungsfähig und kann die Organisation von Entitäten und Unternehmen ändern. Es ist ein dezentrales System, welches alle Datensätze registriert, die als „distributed ledger“ verfügbar sind.

In dieser Thesis werden Reputationssysteme in Peer-to-Peer-Netzwerken und in Blockchain-Technologien detailliert analysiert. Zunächst wird eine technische Analyse beider Systeme durchgeführt, damit der Leser die Entwicklung und die Anwendung von Reputationssystemen verstehen und nachvollziehen kann.

Da Blockchain selbst ist ein Peer-to-Peer-System ist, werden die Vorteile dieser Technologie aufgezeigt. Bisher wurden nur wenige Ansätze in der Forschung zur Implementierung von Reputationen in einem solchen Szenario gemacht, daher gibt es einige wichtige zu beantwortende Fragen wie z.B. Reputationsquantifizierung und Sicherheitsmaßnahmen.

Eine Quantifizierung der Reputation ist wichtig, wenn ein Reputationssystem implementiert werden soll. Die Art und Weise, wie Ratings vergeben werden, kann zu einer subjektiven Vorstellung der Beteiligten führen. Ein binäres System ist hilfreich, um unlautere Bewertungen zu vermeiden, unterscheidet jedoch eine zufriedenstellende Transaktion nicht von einer hervorragenden.

Am Ende werden Sicherheitsmaßnahmen aufgrund möglicher Angriffe und bösartiger Aktivitäten gegen Blockchain diskutiert. Die Art und Weise, wie das System designt wird, ist entscheidend dafür, ob es zu einer sichereren oder anfälligeren Umgebung führt.

Gustavo Andrade

Title of Thesis

Technical Analysis of Reputation Systems based on Blockchain Technologies

Keywords

Blockchain, Reputation, Peer-to-Peer

Abstract

The aim of this thesis is to analyze a reputation system to be implemented in the blockchain.

Blockchain is a hot topic and there are plenty of researches and projects to develop it and to apply it to different scenarios. Blockchain as a decentralized system is very powerful and it can change how entities and companies organize themselves. It is a decentralized system that registers all the records, available as a distributed ledger.

In this paper reputation systems in Peer-to-peer networks and in Blockchain technologies, with more detail, will be analyzed. First, a technical analysis of both systems will be made so the reader can understand and follow the paper development and the application of reputation systems.

Blockchain itself is a Peer-to-peer system, so the advantages of using this technology will be described. Few approaches were made by researchers to implement reputation in such a scenario but there are some questions to answer (reputation quantification and security measures).

Quantifying reputation is important when it comes to implementing a reputation system. The way how ratings are attributed can lead to a subjective idea of the involved parties. A binary system is helpful to avoid unfair ratings but at the same time, it does not distinguish a satisfactory transaction from an excellent one.

In the end, security measures will be discussed due to possible attacks and malicious activities against Blockchain. The way how the system is designed is crucial leading to a more secure or more vulnerable environment.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to take this opportunity to thank my advisor, Prof. Dr. Klaus-Peter Kossakowski, for helping and motivating me to write this Bachelor Thesis. My first contacts with IT security courses happened in Germany through Capture The Flag project, by Prof. Andrej Zieger, and IT Security Management, by Prof. Kossakowski. I would like to express my gratitude to them for motivating and supporting me. Thanks to Prof. Dr. Philipp Jenke for accepting to be my second supervisor.

Thanks also to all Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg) / Hamburg University of Applied Sciences professors for being helpful in my tough but gratifying journey in Hamburg. A special word for all the members of Instituto Superior de Engenharia de Coimbra (ISEC) / Coimbra Engineering Academy for being kind and important in my academic course.

I would like to thank my Portuguese colleagues who studied with me in HAW Hamburg, Kevin, Miguel and Ruben. Without their support and friendship, it would not be possible to successfully conclude my studies in Portugal and Germany.

Special thanks to my family, especially my brother João, my sister Lui, my grandmother Cecília and my parents (Fernando and Lurdes). Their family spirit has been fundamental in all my life, supporting me to achieve my goals. Despite all the difficulties, they have been heroic in helping all members of our family. Thank you!

1 Introduction

1.1 Problem Definition

Recently several technologies have grown faster than expected. The blockchain is one of it and it is actually a hot topic thanks to cryptocurrencies. Researchers and companies are focused in leveraging Blockchain technologies to different applications in Entertainment, Retail, Health care and Insurance industries. In the commercial world, most properly in product reviewing, costumers review a product or service they previously bought or used. Bringing the reputation concept to Peer-to-Peer systems gives the possibility to motivate a straightaway usage by the participants. Motivating them to participate in the activity and giving them reputation score for their contributions in the community builds trust in the environment and allows applying technologies as Blockchain to more applications than cryptocurrency.

Applying a reputation system in a Blockchain is quite interesting because there is no central entity to manage the reputation score distributed between peers. The challenge is to provide a way that can deal with the transactions in the chain, monitor and attribute the score, as well as provide a secure reputation system to avoid manipulation and collusion.

Considering a real scenario, it is known that several companies use artificial intelligence to create fake profiles and fake reviews that create an opinion that they really want to make customers believe in. Having a platform that can offer a reliable information source that cannot be changed or manipulated for any part providing a trusted environment between all the participating parts and brings responsibility so the costumers can genuinely review products and services and the companies show the real reviews without manipulating it.

1.2 Goals

The aim of this thesis is to analyze reputation systems in Peer-to-Peer networks, as providing a technical analysis will lead the reader to understand the value of building trust more consistent trust in Blockchain and understand whether it is possible to apply this technology to a wider range of situations and which conditions apply.

In this analysis an overview of general reputation systems in Peer-to-Peer networks will be provided, comparing and explaining improvements.

In this paper reputation systems approaches will be discussed and analyzed with detail suggesting implementations to Blockchain networks, explaining its advantages. This will inform the reader why this type of model is ideal for different scenarios not only for the cryptocurrency, most properly Bitcoin.

1.3 Structure of the Paper

This thesis is divided into six main chapters

1. Introduction
2. Online Reputation - Concept and History
3. Peer-to-Peer Systems - Basics and Reputation Systems
4. Reputation in Blockchain
5. Security for Reputation Systems in Blockchain
6. Summary and Conclusion

The first chapter is the Introduction of the work. It introduces the structure of the paper, the motivation and introductory concepts that motivate the research of the topic.

The second chapter details the concept of Reputation in online communities, its importance and value.

The third chapter approaches Reputation Systems in Peer-to-Peer networks in general. Blockchain itself is a Peer-to-Peer network so it is important to make an initial and consistent approach to the general type of system.

Reputation in Blockchain is the fourth chapter. After approaching Reputation in general Peer-to-Peer networks, the reader is now more prepared to take a deeper look into Reputation in Blockchain, studying its design, advantages and features.

In the fifth chapter possible failures in a Reputation system and countermeasures and how to mitigate different attacks will be described.

The sixth chapter will have a summary of the work and conclusion of the research, emphasizing the importance of Reputation systems, most properly its advantages when applied in a Blockchain.

1.4 Thesis Delimitations

This thesis will discuss technical aspects of reputation systems, peer-to-peer networks and Blockchain technologies regarding scientific methods. Discussing how markets, companies and other entities use these systems and how such systems impact the economy is not the aim of this paper.

Another topic that it will not be focused on is the anonymity in reputation systems. Blockchain and the majority of P2P networks rely on anonymity, but on the other hand, reputation systems goal is to improve reliability by predicting network state. Designing reputation systems for anonymous networks while ensuring anonymity will not be discussed.

2 Online Reputation - Concept and Basics

The aim of this chapter is to provide an understanding of the concept of Reputation and its concrete framework in online communities. Afterwards, a detailed analysis of Reputation Systems in Peer-to-Peer networks is provided.

2.1 Definition

Reputation is the amount of trust and credibility that the community attributes to someone considering his past actions. It summarizes the opinion that group of individuals has about another specific individual or entity. Taking into account the reputation level, the individual or entity can have certain benefits within the organization or system that they are involved in.

"Reputation is a tool to predict behaviour based on past actions and characteristics. We use reputation regularly in our daily lives — reputations of individuals, as when we choose a physician; groups, as when we decide that individuals above a certain age can be trusted to purchase alcohol; and collective entities, as when we decide whether Ford is a company that will sell us good cars." Dingledine et al. [10]

2.2 Reputation Systems Importance

Online services that provide transactions between parties have a big issue. These types of services, known as e-Commerce, connect two parties that have a great probability they have not met each other before. The consumer assumes a certain risk, i.e. has to pay for services or goods before receiving it and before seeing it in person. So the consumer is in a vulnerable situation. There is no equal position for both consumers and sellers because the sellers know what they will get, the money. These unbalanced positions from both parties raised the necessity of using the concept of reputation and trust in online communities [40].

With a reputation and trust system the consumer still has the issue of not seeing the product or experiencing the service, but he has access to the reputation level of the seller. The buyer can expect a certain result based on the feedback given by precedent buyers. On the other hand, the seller will also be interested in being supported by a reputation system because it will put him in a better position to sell services or goods. Although, a reputation system does not solve completely the problem and there is still a certain level of subjectivity [31].

2.3 Reputation System Goals

The main challenge of reputation systems in online communities is to make them the most similar as possible to the traditional and physical world. This obviously includes encouraging a community to behave properly regarding the main goal of that community. If the community is related to e-commerce, the goal is to have credible sellers selling products that correspond to the buyers' expectations without frauds and the same thing to the buyers' actions. This automatically builds trust between the system users.

First, when designing and researching about reputation systems, the flaws that an online system has, compared to the physical world, have to be identified. The electronic and online world has the advantage of leveraging the communication to a global scale so everybody can share information and communicate with almost everybody.

According to Jøsang et al. [20] the goals in reputations systems must answer two needs. The first is to find ways to apply the advantages of reputation in the physical world to the online world. And the second, doing so, online reputation systems can be a true alternative to traditional ones leading to an important development of global communication, especially in the fields of exchanging goods and services.

Following Resnick et al. [30], Reputation systems must follow these three principles:

- Long-lived entities that inspire entities to make decisions and interactions
- Spread the feedback about current and past interactions
- Use feedback to guide trust decisions

Following these principles, a system will guarantee better services. Service providers that allow being rated are more likely to be trustworthy and to offer higher-quality services. As Rennicks et al. [30] mention, users who use real names and avoid pseudonyms and Web sites that indicate physical stores or companies with physical addresses give offer quality indicators.

The implementation of a reputation system gives to an organization a better control over the social dynamics within the community. According to Oracle [27], the main goals of a reputation system are:

- Promoting high-quality content used and exchanged within the community
- Building trust within the network members
- Motivating members to participate in recognizing their contributions

With a reputation system implementation and management a company has the chance to control its community social dynamics. It motivates community members to participate, avoiding free-loaders (users that take advantage of the community but do not contribute). This participation must involve many members for a long period of time. The reputation system is helpful to ensure that participation is positive, safe and respect system rules [27].

2.4 Categorizing Reputation

The semantic features of ratings, reputation scores and trust measures are crucial to the participating community of the system. Participants have to be able to interpret the rules. According to Jøsang [20], the semantic measures are divided into two main groups: specificity-generality and subjectivity-objectivity.

- Specific measure is related to a statement given by a user to rate a concrete service, e.g. saying if an item arrived in the expected time.
- Subjectivity measure is related to feedback based on a judgment that a user does to determine the quality of a product or service. This is something subjective because it depends on user to user.

2.5 The Reputation Circle

Farmer et al [14] say that reputation can be divided into positive or negative, considering a system with negative reputation scores instead of 0 as a minimum score. It is understandable the meaning of this division. When visiting a website with a reputation system implemented, users get access to the reputation of an entity or user. If the entity or user's reputation is positive the visitor will take into consideration the amount of reputation and act in consequence to it. If the entity or user has a negative reputation the content shared by him will be automatically designed as undesirable. Positive reputation represents the relative value of an entity or user and is used to feature the best-shared content. Negative reputation identifies undesirable content and users for further actions.

It is very interesting how negative reputation can help a company to reduce costs maintaining its system. When a user is visiting a website he is first presented with the higher reputation users and content. The content with lower reputation, or even negative, is put in the bottom, helping the system to be more accurate. Some content marked as negative due to abuse, spam or any kind of violation is automatically deleted by some systems, presenting a blanking content instead of that negative content. If this task was done by a human team it would cost thousands of euros to be completed.

One of the greatest features of a reputation system is creating incentives for users to create good content and good interactions. This adds value to the system and a company gains many benefits. According to Farmer et al. [14] creating the best content leads to a virtuous circle. A virtuous circle, as shown in figure 2.1, is created by consumers of content visit a site and link to it because it has the best content and the creators of the content share their best resources on that site because all the consumers go there.

2.6 Reputation Systems Architectures

Reputation Systems Architectures are divided in two groups: centralized reputation systems and distributed reputations systems.

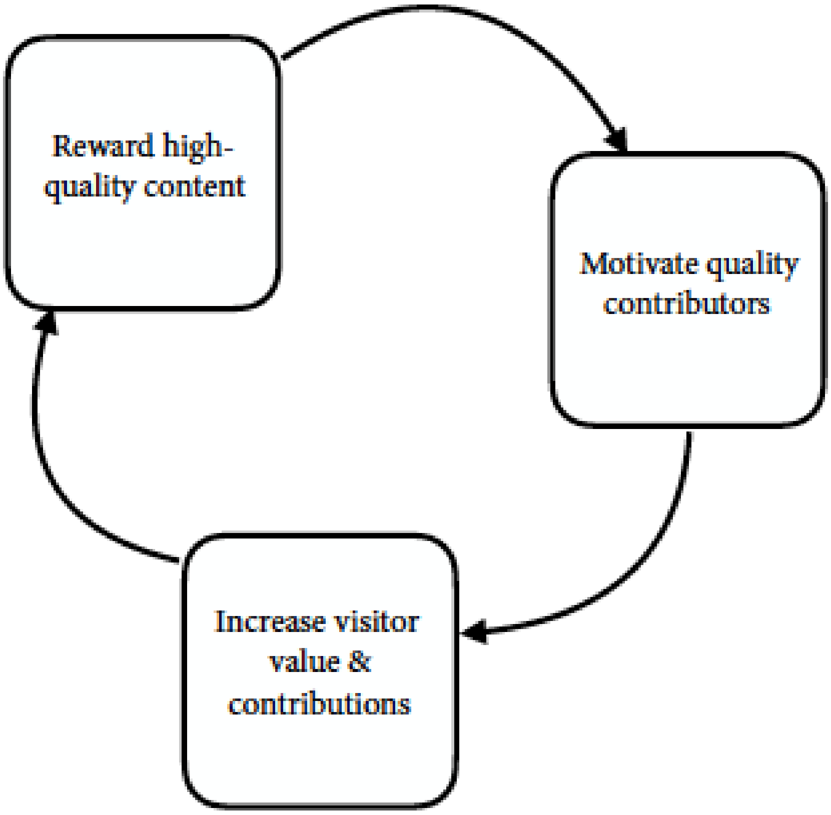
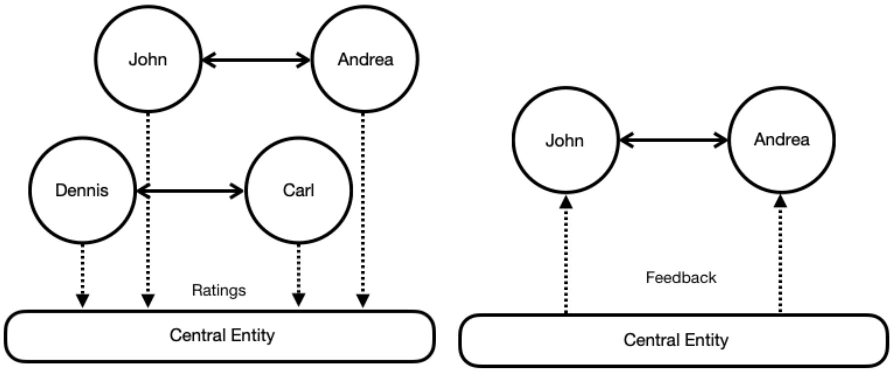

Reward high-quality content
Motivate quality contributors
Increase visitor value & contributions

Figure 2.1: Online Reputation Virtuous Cycle - by Farmer et al. [14]

2.6.1 Centralized reputation systems

In centralized reputation systems there is a central entity that manages the system [20]. An entity that receives feedback from members calculates the number of transactions and the score and it is responsible to display the data to all parts. The central authority communicates with all participants. With this protocol, a participant can then decide if he wants to interact with certain members. It is more convenient to have a network with reputable members than with disreputable ones. A trust paradigm is built.

In figure 2.2 (a) transactions are shown between John and Andrea and between Dennis and Carl. Both communicate with each other and rate their experiences to the central entity. In (b) it is possible to verify the reputation manager communicating of previous feedback and reputation scores to the members [20].



(a) Transaction and feedback

(b) Asking feedback

Figure 2.2: Centralized Reputation System - [20]

Central entity is also responsible for collecting ratings from members, counting the number of transactions, updating in real time the reputation scores and displaying it so all agents can see it.

According to Jøsang et al. [20] centralized reputation systems have two main fundamental aspects: centralized communication protocols and reputation computation engines.

- Centralized communication protocols allow members to rate other members after having transactions in common with them and to obtain reputation scores from certain members given by the central entity.
- Reputation computation engine is used by the reputation manager to calculate rating and the number of transactions for each member, according to received feedback and other possible information.

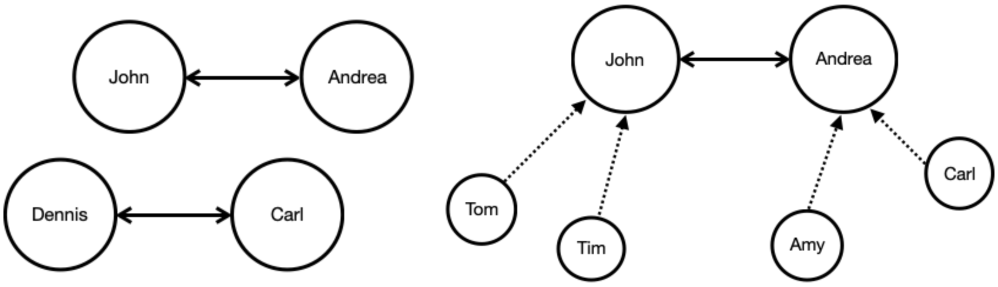
2.6.2 Distributed Reputation Systems

Distributed reputation systems follow a totally different paradigm than centralized reputation systems. There is no central entity managing the systems which makes the analysis of this architecture quite interesting. The fundamental aim of this paper is to analyze a decentralized technology, Blockchain, to manage a reputation system.

The figure of a central entity to secure the system operation and its rules (managing, displaying scores) does not exist. Instead, the scores are distributed when rating can be submitted. Each member can store the feedback of each transaction with third parties and then provides requested information to others.

A decentralized reputation system brings more trust in the network and a more honest environment. A centralized reputation system, like eBay, has, can be manipulated by the owner in order to reach the company goals.

In the decentralized system a member must find the distributed stores in order to evaluate a member who he wants to make transactions with. The member may also search ratings from the other member in third nodes. Those thirds nodes may have had past experiences with the node the member wants to communicate with, so they can provide feedback on their experiences, as illustrated in figure 2.3.



(a) Transaction (b) Giving feedback

Figure 2.3: Decentralized Reputation System [20]

The two fundamentals of decentralized reputation systems are distributed communication protocol and reputation computation method.

- Distributed communication protocols allow members to get a rating from different nodes of the network.
- Central entity engine is used by all the members to calculate manager the rating and the number of transactions for each member, according to received feedback and other possible information.

The next chapters will discuss with more detailed this type of architecture, starting in Peer-to-Peer Network analysis and then to Blockchain as the main topic.

2.7 Platforms Using Reputation Systems

There are hundreds of platforms and services using one or more reputation systems. It is quite interesting to notice that services from different fields use these systems. Programming communities, web searching, e-commerce, social media, wikis, internet security and much more. In this section it will be described as a significant number of them, some are well-known and others are not so recognized but with very importance online.

Regardless the type of service these systems offer their goals are the same, instigate interaction between users based on trust. If these systems want to maintain or grow their businesses or services they must have reliable users who make reliable transactions.

2.7.1 Centralized Systems

As detailed before, centralized systems have a central entity managing the system responsible for calculating and storing reputation scores, verifying transactions and creating measures to mitigate attacks (central entity is a crucial target because manages the reputation system).

The following examples are applications that have centralized reputation systems.

Ebay

eBay feedback system is probably the most known reputation system in the world. The system is based on the number of transactions completed by a user (seller or buyer). The system is centralized because the reputation score is calculated by the website.

Each buyer rates a seller after making a transaction. The rate can be one to five.

TrustedSource

TrustedSource is an online reputation system owned by created by CipherTrust and actually owned by Intel Security that analyzes IP addresses, domains or URLs. This analysis observes in real-time content categories, global email and network traffic patterns. The resulted scores are combined with local filtering policies of devices and services to make an accept/shape/traffic shape types of decisions on the network connections associated with those Internet identities [43].

Reddit

Reddit is an American social news aggregation and discussion website. Members create text posts, content such as links, images, videos, streams and are then voted by other registered members.

The votes can be up-vote or down-vote. Users with higher rates have privileges such as creating their own subreddits on a specific topic of their interest. Other users can then add it to their front pages.

Users can earn 'post karma' or 'comment karma'. Post karma refers to karma points received from text and link posts. Comment karma refers to karma points received from comments.

Reddit awards users who get valued comments or posts due to humorous or high-quality content. This process is called 'gilding' and users are gifted with Reddit gold that gives them several features like ad-blocking and unlocking special content, locked to regular users.

Stack Overflow

In Stack Overflow platform, a type of forum for IT community (students, professionals and enthusiasts), the reputation system is taken seriously. Users gain or lose reputation concerning the quality of their posts (questions or answers).

When users are voted up/useful to a question they get five points. If it is an answer they get ten points and if the answers get accepted, the user earns fifteen points. The losing reputation protocol is not so punishing in terms of points. A vote down to a question equals to minus two and the same happens to a post that was previously well succeeded and then the user deleted it. This action is punished with minus two points. A simple vote down to an answer it is just subtracted with two points.

Stack Overflow is an important platform for the IT community. Nowadays there are several employers that look for potential employees with a good reputation in the platform.

2.7.2 Decentralized Applications and Algorithms

There are no pure decentralized reputation systems in the market at the moment. There are algorithms (Google PageRank or EigenTrust) that can be applied in any system. This fact can be explained by the wish of companies wanting to control every single component of their businesses. A company that

The following examples are composed by algorithms and infrastructures ready to be applied in different scenarios.

DREP

DREP is a reputation protocol based on blockchain technology that quantifies and tokenizes online reputation for trading, investment and data sharing purposes. DREP aims to empower internet platforms to solve their pain points, restructure their value ecosystem and facilitate their transition and acceleration via reputation-centred tokenomics and blockchain technology [15].

PageRank

PageRank is an algorithm used by Google search engine to rank websites in its search results. PageRank measures the importance of a website counting the quality of links that point to it. It is an algorithm to classify websites.

If a website from a certain entity has links considered important or many links with lower importance pointing to it, the website will be more trustworthy and will have more probability of appearing in higher positions in Google search results.

EigenTrust

EigenTrust is an algorithm for reputation systems based on P2P networks. Each peer in the network has a unique value based on its past contributions.

EigenTrust uses transitive trust (if A trusts in B and B trusts in C, even without knowing C, A will trust C because B also does). In EigenTust, the global reputation of each peer i is given by the local trust values assigned to peer i by other peers, weighted by the global reputations of the assigning peers [21].

3 Peer-to-Peer Systems - Basics and Reputation

The aim of this chapter is to make a detailed analysis of reputation systems in decentralized networks. It is quite important to do it before heading to Blockchain analysis, so it can be easier for the reader to understand the advantages and features of applying a reputation system in a Blockchain technology and to understand its analysis, advantages and how can Blockchain leverage reputation systems to higher levels.

3.1 Peer-to-Peer Networks

When it comes to mentioning decentralized systems, one of the first types of architectures coming to mind are Peer-to-Peer (P2P) networks.

There is no central entity, so every node of the network is considered as client and server at the same time. Peers communicate directly with each other exchanging information, data and making transactions. Processes that are part of a peer-to-peer network are all similar. They all have the same functions that are part of a distributed system. As Tanenbaum et al. [37] mention, much of the interaction between peers is symmetric, each node works as a client and server, also referred to as a servant. P2P networks can be classified by centralization and structure. In terms of overlay network centralization, P2P can be centralized or decentralized. And can also be classified as structured or unstructured because nodes may not be able to communicate directly with another arbitrary node.

3.1.1 Centralized Peer-to-Peer Networks

In centralized P2P there is a central entity responsible for managing the downloaded content between peers, maintaining directories of meta-data and routing information, file requests. The post office service perfectly illustrates a P2P centralized architecture. A sender A wants to send a letter to receiver R. So A will write R address in the envelope and then sends the letter to the post office. The post office will validate by checking R's address and sends the letter to him.

Napster was the pioneering P2P file sharing service. Napster had a central manager that received search, browse or file transfer requests sent by peers. The central manager did not participate in download action. This central entity limits network scalability, robustness and can be a point of failure.

3.1.2 Decentralized Peer-to-Peer Networks

In decentralized P2P networks there is no central manager. All the actions are handled by the peers, such as transactions. Each peer has the same features, so it is hoped that they behave similarly and can act as a client (when requesting a service) and server (when providing a service). As Xiong et al. [42] say, in decentralized architectures, peers with better performance can become 'super peers'. Better performance means having a better computing ability and network bandwidth so then they can play a more important role in the infrastructure. It is fairer to providing chances of promotion to peers who make good and several contributions.

Partial decentralized P2P network is a concept that has to be taken into account. Gnutella is the best known and the first platform adopting this type of architecture. In Gnutella there are nodes with more privileges, called super peers (ultra-peers) and each normal peer (leaf peer) is connected to an ultra-peer. The ultra-peer has a table of the hash value of files that are available in its local leaf peers. Super peers forward requests, acting as servers.

3.1.3 Structured Peer-to-Peer Networks

In structured P2P systems, nodes are organized in a well-defined topology (ring or binary tree) so it can be easier to look up for data. In this type of organization, each data item maintained by the system has a unique key associated, that is used as an index. The system is responsible for storing pairs (*key,value*). It is essential to use the following hash function:

$$key(dataItem) = hash(dataItemValue) \tag{3.1}$$

Each node has an identifier from the set of all possible hash values, so it can be responsible for storing all the data that belongs to a specific set of keys [37].

This system topology provides an efficient implementation of a function lookup that maps to an existing node:

$$existingNode = lookup(key) \tag{3.2}$$

Each node can be asked to look up a given key which leads to routing the request to the node who is responsible for storing data associated with the given key, representing a distributed hash table [37].

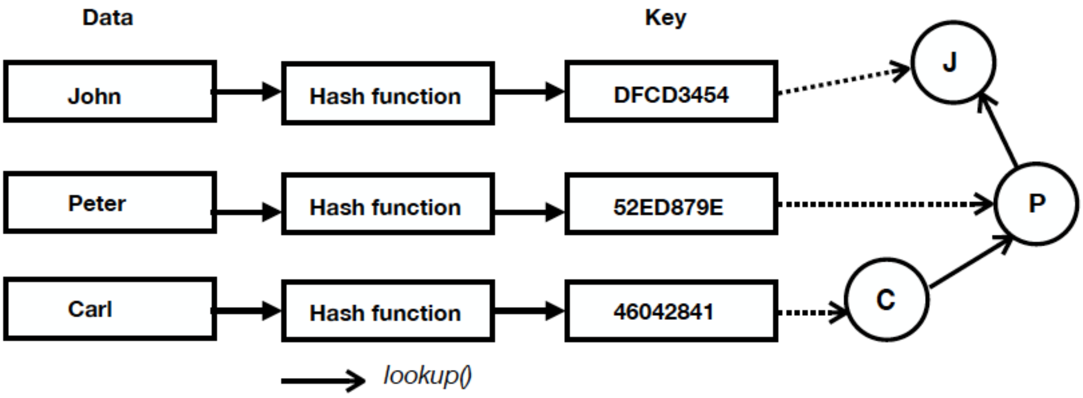


Figure 3.1: Distributed Hash Table - based on wiki [44]

3.1.4 Unstructured Peer-to-Peer Systems

In unstructured P2P networks each node has an ad-hoc list of neighbours. Looking up data is not possible just following a determined route because the lists of neighbours are build in an ad-hoc fashion. It is needed to implement new ways to search for data. The most common methods are *flooding* and *random walks*. In *flooding*, a node simply makes a request for a data item to every single node in the system. The other nodes search locally for the requested data item and if they have something requested by the requester, they answer directly to it or answer back to the node that forwarded the request. If the node does not have anything related to the requested data, it simply forwards the request to its next neighbours. It can be considered an expensive method because there is a probability of the requested items just being found in the last nodes. It is not accurate and can consume resources. To reduce the costs it can be implemented a time-to-live (TTL) mechanism, to define the maximum attempts that each node has. In *random walks*, a node makes a request to a random node. This node will search locally for the requested data and if does not have anything related makes another arbitrary request call to another node in the network and so on. It can take much longer than flooding but it has the advantage of not consuming so much traffic. To reduce the search time there are some rules that can be defined, e.g. defining a limit n of random searches for each node. To stop the random walk it can also be configured a TTL [37].

3.1.5 Reputation in Peer-to-Peer Networks

Building a trust system in a P2P network is a challenging task. The network has different peers, so rules and procedures have to be defined to avoid malicious behaviour and to establish a protocol to manage the system. The malicious behaviour can be providing fake feedback or a certain type of collusion or other attacks in order to change another peer's feedback.

It is also important to think about the context where the system will be built and its goals. Most of the reputation systems used to build trust are applied in a centralized method. It is a big challenge to apply in a decentralized way so it can be scalable, secure and applied to different scenarios and efficient.

3.1.6 Flaws in Decentralized P2P Reputation Systems

Decentralized P2P reputation systems have some considered flaws, features that have to be improved to provide a better service to its users. The main challenging concern developing this type of systems is to design organized ways to reduce any malicious activity of nodes [46]. In e-commerce communities and in file-sharing services, the known systems have the following issues:

- the system does not distinguish dishonest feedback: users interpret a certain buyer reputation by the number of transactions and its' feedback. The system does not provide a structured way to detect which previous feedbacks are trustworthy or not;

- the system does not have a structure to detect different types of transactions and contributions, e.g. if peer A contributes with 500 MB of good content it will receive the same type or amount of score than peer B that only contributed with 50 MB; This can be done by having parametric levels of contributions, avoiding a peer A building reputation with small transactions and then taking advantage of bigger transactions;

- most systems do not provide an incentive system, so there is a lack of feedback and contribution in many systems. This is related to the free-riding phenomenon that is based on the lack of contribution by several peers. These peers deny requests because they do not receive an incentive for sharing files (no reputation system) and there is no punishment for not answering queries from other peers [47]. Excessive free riding in P2P networks leads to network congestion and degradation of system performance.

- it is quite easy to build a reputation and then starting to make dishonest activity. The most flagrant case is the collusion attack, also known as shilling attack. In this type of attack, a group of peers act as a team making fake transactions between them to raise their reputation. They also can act in a group to take down other peers' reputation so they can lose score. This is the definition of collusion.

The existing threats in general distributed systems, obviously also existent in P2P networks, have to be taken in to account when designing a P2P reputation system. As Xiong et al. [46] list, the following security threats that have to be mitigated when structuring a P2P reputation community:

- Spreading of tampered information. Peer P can search for a resource called 'Name.ex' and peer V can provide a malicious file with the same name. The file can be a random virus that the requester will download and then realize that the content is not appropriate;

- Man in the middle attack. A malicious peer can intercept a content shared between two other peers and rewrite it with its IP address and port of the provider so it can infect the requester peer with changed content;

- Peers are easily compromised without the existence of a central authority. A malicious peer can spread malicious content through the network by simply making a copy in the program directory. This leads to security issues as confidentiality and data integrity, which is hard to solve, according to Kwok [22];

- Free-loaders are non-cooperative peers who do not contribute within the network and just request and download content. P2P networks usually fail to have a reputation system that punishes these peers.

3.2 Attacks in Peer-to-Peer Networks

P2P networks have an unstable structure. It allows unreliable nodes to constantly access or leave the system. P2P networks have specific aspects that differ from a typical client-server network. Each peer acts as server and client at the same time. In this section the following attacks to P2P networks and countermeasures will be discussed:

- DoS and DDoS attacks
  - TCP syn flooding
  - Query Flooding attack
- File Poisoning attacks
- Sybil attack
- Eclypse attack

3.2.1 DoS and DDoS Attacks

Denial of Service (DoS) attack is based on attackers exhausting the system with a considerable amount of requests. Consequently, the target host cannot provide its service to users [41]. Distributed Denial of Service (DDoS) is developed from DoS, both have the same concepts. Although, in DDoS, attackers exploit considerable amount os distributed hosts to launch the attacks to the target. It is a larger scale DoS.

In P2P networks, the most common DoS is network flooding in a fake package. Sinking the victim account so that it cannot respond to queries is also very productive attack [34].

TCP Syn Flooding Attack

The classic TCP Syn flooding attack can also occur in a P2P network. As Wang [41] states, this attack can happen when the attacker is downloading content from the victim host. This connection is made by TCP so the vulnerability of three-way handshakes of TCP make the peers could be a target of DDoS attack. In a normal TCP three-way handshake [19], the exchange is made by the following steps:

1. Client requests connection by sending a SYN (synchronize) message to the server.
2. Server acknowledges by sending SYN-ACK (synchronize-acknowledge) message back to the client.
3. Client responds with an ACK (acknowledge) message, and the connection is established.

File downloading in a P2P network is performed via a TCP connection, three-way handshakes of TCP make this attack happen. The attacker sends the SYN request to the victim node using a forged IP address, the victim node responses with SYN-ACK message. The point is that the IP address of the attacker is forged, so the attacker will not get this SYN-ACK message and the third handshake will not be completed. To perform Syn flooding attack, the attacker will continuously send the SYN request to victim node and the node will exhaust and will not be able to provide services to legitimate nodes [3].

Countermeasure

Filtering techniques are good practices for packet filtering based on IP addresses. With the absence of an army of controlled hosts, attackers have the ability to send packets with forged source IP addresses [12]. Taking out this ability without modifying TCP connections is a good solution to mitigate spoofed IP addresses. These filtering techniques are described by RFC (Request for Common) [1] [RFC2827][RFC3013][RFC3704].

[1] RFC - a type of publication from the technology community. An RFC is authored by engineers and computer scientists in the form of a memorandum describing methods, behaviours, research, or innovations applicable to the working of the Internet and Internet-connected systems

Query Flooding Attack

Query Flooding attack happens mainly application layer in pure P2P networks like Gnutella. The query node broadcasts the queries to its neighbours. A malicious node will constantly generate a maximum number of queries in order to flood the network [34].

It is also known that some attackers can attack a victim by sending queries to ask popular files using the victim's IP address. The destination nodes that contain the pretended files can be another set of distributed peers. Each destination will send its reply to the source address, that is forged. This means that the victim has to accept a considerable amount of replies from different servers and may be overloaded by that traffic [41].

Countermeasure

Nodes can accept at most the maximum queries from a requesting peer. After getting the maximum number of queries from a request node, it just drops the rest requests from that incoming link. This mechanism can effectively decrease the harm of the query flooding attack but can not totally avoid it [36].

Wang [41] also states that the countermeasure to non-distributed query flooding attack cannot solve this problem successfully. Because the number of queries from each attack is less than the maximum value, the victim does not do any reaction to deny the incoming traffic. But the total amount of queries from attackers are a disaster to the victim node.

3.2.2 File Poisoning

A file poisoning attack consists in using false information such as fake file indexes, forged IP addresses, fake routing tables, to violate a P2P network.

Centralized P2P networks have a central server (e.g. Napster)to store the index of files. An attacker can try to poison these indexes in order to add adulterated index information to the index server. The most important thing is all bogus information point the target address of the popular files to one target victim host. When other peers download these files, they will get false information from the index server allowing the attacker to establish a TCP connection with the victim.

Countermeasure

Index poisoning attack is hard to find. A TCP connection is established between the victim node and requesters. It is also hard for servers to recognize their potential client nodes in TCP connection phase [41].

3.2.3 Sybil Attack

The Sybil attack is based on an attacker creating or gaining access to multiple identities to act in the network. The main goal of the attackers to develop a Sybil attack is to gain access to the majority of the network.

Most P2P networks use the virtual addressing scheme based on the logical identifiers to manage and organize the network. Identity should be unique and can form a one-one mapping pair to actual entry [41]. Although, if the relation of one-one mapping of an entity to identity is destroyed by a malicious peer, in other words, a malicious entity acts as a number of multiple identities.

Countermeasure

A measure to mitigate the Sybil attack in P2P is establishing a trusted certificate authority to make distinct entity has distinct identities. There are several existing methods, for example, the explicit certification agency, VeriSign; CFS cooperative storage system that identifies node by a hash of network address; EMBASSY, utilizes cryptographic key in hardware peers to identification.

3.2.4 Eclipse Attack

Eclipse attack is a general attack in overlay networks. An attacker controls a group of neighbours of a considered good node. Then, the malicious nodes' group work together to trick the good node writing their addresses in its neighbour table. Doing so, an attacker can control a considerable part of the overlay network. A large-scale malicious node can get even make bigger troubles. Nodes can not forward messages properly and the network cannot be managed correctly.

Countermeasure

Countermeasure used for Sybil attack can be also applied in eclipse attack. Wang [41] proposes and 'identity registration procedure'. Node hashes its An IP address and port as ID to calculate its identifier. Then it registers the ID at already registered nodes. The registration process is accomplished and the node can join the P2P networks and other nodes have the ability to verify the ID and distinguish whether it is a forged ID.

To deploy an eclipse attack the attacker must have a higher indegree (number of direct routes coming into a node) than the average level of the indegree of nodes in the network. Every single node, in P2P networks, maintains a list of its neighbours. So it is possible to query the neighbour peer list periodically. If the items on the replied neighbour list are greater than the indegree bound, or that node is not on his neighbours' list or the size of returned neighbour is greater than the outdegree (number of direct routes going out of a node) bound.

3.2.5 Attacks Comparison

Attacks	Centr. P2P	Decentr. P2P	Centr. Rep.	Decentr. Rep.
TCP SYN Flooding	✓	✓	✓	✓
Query Flooding	✓	✓	✓	✓
File Poisoning	✓	x	✓	x
Sybil	x	✓	x	✓
Eclipse	x	✓	x	✓

Table 3.1: P2P Attacks Comparison

- Centr. P2P - Centralized peer-to-peer network
- Decentr. P2P - Decentralized peer-to-peer network
- Centr. Rep. - Centralized Reputation
- Decentr. Rep. - Decentralized Reputation

Overview to P2P Attacks

Reputation systems based on distributed architectures such as P2P networks need to answer to threats like fake transactions, dishonest feedback and strategic malicious behaviour from peers. In chapter 5, countermeasures about these threats will be discussed.

A peer can get good trust values and build a good reputation in the environment and then have malicious transactions. Systems consider the average of transactions. So it is quite important to define a strategy to study the oscillation of behaviour and monitor the transactions and of the nodes. Srivatsa et al. [35] defend a dependable trust model is a right way to study behavioural patterns. These patterns must monitor fluctuations in peers' behaviours, tolerate unintentional malicious transactions and reflecting consistent behaviour. A way to do this is to save current feedback reports about a peer, historical reputation and oscillation in peer's behaviour.

Building a decentralized reputation system creates the need to suppress fake transactions. Having a central entity responsible for managing the system, it could do that task. But when it takes to distributed systems this problem rises. One of the possible solutions is to consider filling transaction proofs after peers develop a transaction. Transaction proofs must only be valid if both peers accept to make the transaction with each other and should also be exchanged by both. A transaction proof from peer v is only valid if peer u also sends its transaction proof to avoid malicious peers from obtaining proofs without conceding one. This avoids malicious peers filling feedback after an unfair exchange or a non-existent transaction. Transaction proofs use public key cryptography.

Detecting fake transaction does not solve another big issue in distributed reputation systems, dishonest feedbacks. This action is well-known through collusion when a group of malicious peers act together to fake transactions between them and get feedback for it, or just simple act to give dishonest feedbacks to third peers in order to take down these peers' reputations and raise theirs.

3.3 Reputation Models

One of this thesis goals is to study distributed trust models in decentralized systems. Studying a model suitable for such type of system must answer questions concerning reputation quantification, transaction processes between peers, rules to be respected by participants and designing patterns to avoid possible threats.

In this section reputation, score calculations are described as well as trust patterns in decentralized systems.

Reputation model is a group of rules and strategies used to design a reputation system. All users know and respect this group of rules so they can participate according to it. Violating a reputation model can lead to a user expulsion or loss of reputation score.

Establishing a reputation model demands a deep analysis of several parameters. Li et al. [23] mention that trust management has three issues:

- Trust needs to be quantified so peers can compute and interpret other peers' trust;
- Trust scores have to be maintained and stored somewhere. The location has to be clarified to the community;
- A communication protocol for exchanging reputation scores among the peers has to be well defined, so each peer can improve the accuracy of its local trust values.

Abdul-Rahman et al. [1] propose the concept of *conditional transitivity of trust.* This idea is interesting and brings some restrictions to what is called *transitive trust.* Transitivity trust allows a peer A, that trusts peer B, to automatically trust peer C because B trusts C. This can be a dangerous assumption because it may not be true in many scenarios. Conditional transitivity trust establishes three conditions so peer A can trust peer C having B as an 'intermediary':

- B recommends its trust in C to A;
- A trust B as a recommender;
- A can decide to take into account B's recommendation.

Trust models in online communities have to represent as much as they can to take the same scenarios of the human real world to online systems. This is the only way to convince users to rely trust on the system. In distributed systems peers, have the same care as a human has to trust or ask the trustworthy of third individuals. So peers can rely on a recommendation. This interest of this study relies on the uncertainty of trusting on second-hand information to measure the trust of third entities. In this way, Abdul-Rahman et al. [1] defined two types of trust relationships: *direct trust value* and *recommender trust value.* Direct trust is trivial, is relevant to direct trust relationships. Recommender trust is more complex and has a defined protocol. Let's suppose that A recognizes B trustworthiness. A asks B opinion about D and B does not know D either. B will forward A's request to C, that has knows D, and C will reply D's trust value back to A. So the path A x B x C x D is the *recommendation path.* Rahman and Hailes present the following formula to compute the trust value of a target in a recommendation path:

$$tvT = [rtv(1)/4] \times [rtv(2)/4] \times .. \times [rtv(n)/4] \times tv(T) \tag{3.3}$$

- *rtv(n)* is the recommender trust value of recommenders in the return path including the first and the last recommender;
- *tv(T)* is the recommended trust value of the target T;
- *tvT* is the target T trust value.

It can happen, in some scenarios, that there more than one recommendation path. So in the case of multiple recommendation paths between requester and target, the target's trust value is the average of the calculated values in different paths:

$$tv(T) = Average(tv1(T), tv2(T)) \tag{3.4}$$

Although this model has some pitfalls that have to be taken into account. The model does not consider fake recommendations by the peers and assumes that a peer with good recommender trust value always makes reliable judgments. The second issue that it does not provide a system to monitor and reevaluating trust.

Xiong and Liu (2004) considered an interesting trust metric computing the trust value of a peer u by the average of the amount of satisfaction peer u receives for each transaction.

$$T(u,t) = \frac{\sum_{v \in P, v \neq u} S(u,v,t) \times Cr(v,t)}{\sum_{v \in P, v \neq u} N(u,v,t)} \tag{3.5}$$

- P is a set of peers
- u and v are peers that belong to P
- *S(u,v,t)* is the metric of satisfaction that v has about u in t number of transactions

- *Cr(v,t)* is the balance factor for filtering feedback from *v*
- *N(u,v,t)* is the number of interactions that *u* has with *v* up to the *t* number of transactions
- *T(u,t)* is *u*'s trust value evaluated by other peers until a number *t* of transactions

T(u,t) is the amount of satisfaction received regarding a certain number of transactions in a P2P system.

Comparing the approaches of Rhaman and Hailes 1997 and Xiong and Liu 2004, Xiong and Liu have a more accurate design. They consider both positive and negative feedbacks.

Although Xiong and Liu have some issues. According to the balance factor used in the equation, at page 28, the system will assume that peers with higher trust value will give more reliable feedbacks, which is not always true. The other identified issue is that this approach gives more value to recent feedbacks. A peer behaviour can change from the past to the present, so an evaluation considering later feedbacks will lose accuracy.

Some questions related to the quantification of reputation and how to avoid unfair ratings remained unanswered. In the 4th chapter, Reputation in Blockchain, details about secure transactions with digital signatures and receipts, transactions broadcast will be detailed and explained, answering why Blockchain can leverage reputation systems to higher levels. Later in chapter 5, Security for Reputation Systems in Blockchain, countermeasures for unfair ratings and other malicious behaviours will be analyzed.

3.3.1 Models Comparison

Attacks	Rhaman and Hailes	Xiong and Liu
Type of Feedback	x	✓
Feedback reliability	✓	x
Feedback equality	✓	x
Number of transactions	x	✓
Transivity trust	✓	x
Fake transactions	x	x
Reputation quantification	x	x

Table 3.2: Reputation Models Comparison

Parameters used in the comparison

- Type of feedback - consider if feedback is positive or negative
- Feedback reliability - considering that a peer with higher reputation score will give more reliable feedback may not be true. Malicious peers can develop good actions and behaviour to create a good reputation and then make collusion.
- Feedback equality - some systems consider recent feedback more reliable than an older one which may not be true. Xiong and Liu do not offer feedback equality because they consider recent feedbacks more valuable.
- Number of transactions - the system considers a number of transactions made by a peer to calculate its reputation
- Transivity trust - peers rely on their trusted peers to evaluate and trust in third peers that they do not know directly
- Fake transactions - system capacity to detect fake transactions
- Unfair ratings - system capacity to detect unfair ratings
- Reputation quantification - reputation score range and definition of good reputation and bad reputation to clarify the system

4 Reputation in Blockchain

After making an analysis to reputation systems in P2P networks what would take anyone to study or develop reputation systems based on Blockchains? The answer can be resumed in one word, security. The blockchain is a technology that allows preventing attacks with more accuracy, as it will be detailed in chapter 5. On the other hand, P2P reputation systems have limitations such as being single-dimensional systems, each peer has one bit of data related to the previous transactions which limit efficiency and reduces the load on the network [9]. Enrollment in P2P reputation systems is not mandatory, leading to privacy loss issues.

In this chapter basic concepts and structure of Blockchain will be explained so the reader can follow then the discussion of reputation systems based on Blockchain technologies.

4.1 Blockchain - Basic Principles

> *"Blockchain technology is not a company, nor is it an app, but rather an entirely new way of documenting data on the internet. The technology can be used to develop blockchain applications, such as social networks, messengers, games, exchanges, storage platforms, voting systems, prediction markets, online shops and much more. In this sense, it is similar to the internet, which is why some have dubbed it 'The Internet 3.0'."*
>
> Lisk [24]

Blockchain is a public ledger in which all transactions are stored in blocks that are chained to each other. As time goes by and more transactions are done, they are appended to the blocks. None of the past transactions is deleted, everything is stored and can be seen by all the members. One of its main features is being decentralized integrating core technologies as digital signatures, cryptographic hash and distributed consensus mechanism. Transactions occur in a decentralized environment which allows to save the cost and improving efficiency, as Zheng et al. [49] says. It provides interesting features for users such as anonymity and auditability. Auditability in a way that the participants can check transactions because each machine in the network has an entire copy of the chain.

Blockchain is globally known by its first and major application, Bitcoin. If this technology is capable of assuring payments and money transactions between the network, it will be also capable of being adaptable to other scenarios. The blockchain is starting to leverage the Internet of Things, smart contracts, public services, security services and **reputation systems**.

Although, Blockchain is facing challenges concerning to scalability. As mentioned before, the network is made of blocks that contain records and these records grow because new transactions are being constantly executed. At the beginning of Blockchain development, each block was limited to 1 MB size with the aim of making it more secure but it brings challenges to take the technology to higher levels. With a transaction comes data and with just 1 MB per block, just three or four transactions will be processed per second. This is a small amount. When it comes to comparing Bitcoin blockchain technology with other payment platforms, PayPal is able to manage 193 transactions per second while VISA manages 1667. *Ethereum*, the most popular platform using Blockchain technology, just processes 20 [38].

4.2 Blockchain Architecture

Blockchain is an ordered list of blocks that have all the transaction records (TX in 4.1) as a public ledger. The transactions are executed by the participants and are broadcast to the entire network. Each block points to the previous block keeping a hash value of it.

4.2.1 Block Structure

A block is made by a block header a block body as it is shown in 4.1.

A block body is basically composed by the transaction list. The number of stored transactions depends on the block size and the size of each transaction. The Merkle tree is

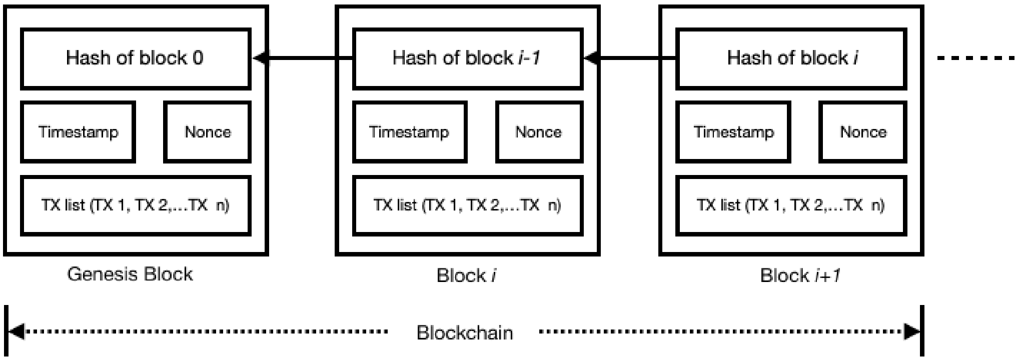


Figure 4.1: Blockchain - blocks sequence - based on Prabhakar [28]

Block Header	
Header Component	Description
Block Version	Describes the structure of data the block has to indicate the rules to be followed
Previous Block Hash	A 256 bit hash value that identifies the previous block
Merkle Tree Root Hash	All the transactions that belong to a block are hashed together to form a line of text
Timestamp	Current time
Target	Current hashing target
Nonce	Each block has a 4 byte field used to every hash calculation

Table 4.1: Block Header Components

the capital block body component. It summarizes all the transactions in the block by getting a fingerprint of the entire set of transactions.

Merkle trees are created constantly by hashing pairs of the nodes that are making a transaction until there is one hash left. This hash is called Merkle root. The Merkle tree is built from the bottom with hashes from each transaction (*Transaction IDs*). The image 4.2 shows a Merkle tree example.

Merkle trees are binary and require an even number of leaf nodes. If the number of transactions is odd, the last hash is duplicated to create an even number of lead nodes.

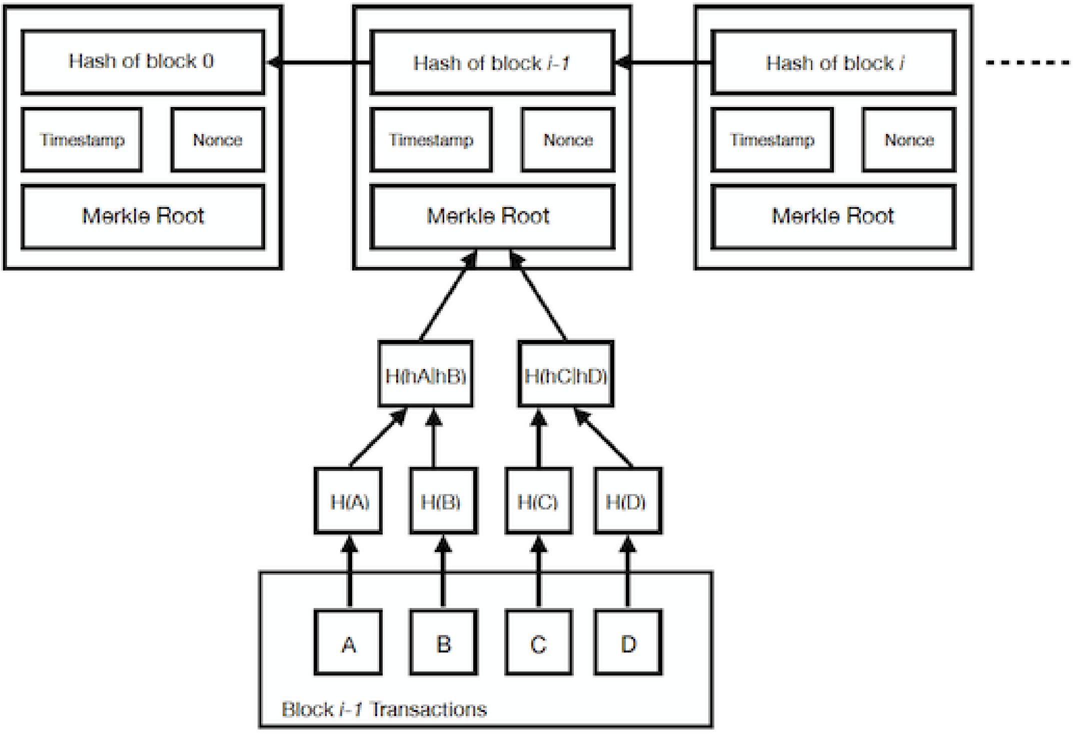


Figure 4.2: Blockchain Merkle Tree, based on D. Harding [18]

In figure 4.2, *i-1* block has four transactions (A, B, C and D). All transactions are hashed and the hash is stored in each leaf node. Then pairs of leaf nodes are summarized in parent nodes. Hash A and Hash B result in Hash AB, while Hash C and Hash D result in Hash CD. In the end, AB is hashed with CD to create the Root Hash, also known as The Merkle Root.

The Merkle Root combines all the data in its related transactions which are stored in the block header (the Merkle tree itself is stored in the block body). This ensures data integrity. If there is a change in any transaction of the block, the Merkle root will automatically get changed too.

According to S. Ray [29], Merkle trees have the following advantages:

- Provide means to prove data integrity and validity
- Require low memory because the proofs are computationally fast

- Merkle Tree proofs and management only require a small amount of information to broadcast to the network

The hashing process is assured by SHA-2 cryptographic hash function. As explained above, hashing is crucial in Blockchain technology. SHA, Secure Hash Algorithm, is a set of algorithms used to ensure better digital security. SHA has a different version (SHA-0, SHA-1, SHA-2 or SHA-256). The task of SHA is taking a piece of data, compacting it and creating a unique output hard to emulate with a different piece of data. The output has always a fixed length while the input has a random size. The greatest advantage of hashing is that it is possible to hash an input, but it is not possible to use the hash function in the output in order to reconstruct it. Hash functions are 'one-way'.

4.2.2 Mining

Mining is the mechanism that allows Blockchain to be decentralized because is the process of generating blocks without the need of central authority management or supervision. Miners are responsible for validating transactions and record them in the chain. The aim of the miners is also to compete to solve a mathematical problem of a cryptographic hash algorithm. The block who first answers the problem is automatically elected as a miner and can then broadcast to the whole network. The solution to the mathematical question is called *Proof-Of-Work (PoW)*. PoW helps a miner to prove that it spent a certain amount of time and resources to solve the problem. When a block is solved, its transactions are automatically confirmed and the resultant resources of a certain transaction can be spent. According to Cosset [7], a block is mined every 10 minutes. Miners receive incentives for generating blocks, in Bitcoin scenario they receive bitcoins and transaction fees related to transactions that belong to the generated blocks.

4.3 Reputation Systems Based on Blockchain Technologies

Implementing a reputation system in a P2P network has advantages and it is interesting due to a central authority absence. When designing and developing such systems it cannot be assumed that all the ratings given by peers are reliable and its user is not given a chance to have a view of the entire system. Some of them take into account that there are no peers with malicious goals and does not prevent strategical behaviours. This means a peer will act honestly to gain some reputation within the systems and then take advantage of it to deploy dishonest activity. One of the most critical problems in these approaches is that they do not have a strategy to avoid multiple identities by a peer leading to collusion. The Sybil attack [11] is another concern in P2P networks. This attack is based on a peer gaining access to multiple trustworthy peers (see table 3.1). Later it will be explained why a reputation system in Blockchain can avoid Sybil attacks. Re-entry attack [36] is also usual in P2P and consists in a user entering in the network and develops malicious activity. When its activity leads to low or negative reputation they leave and re-enter. This happens because the entry cost is not high and the system attributes a zero score to a new peer when there are peers in the network that have negative reputation scores. So automatically the new member has a higher reputation than other with negative scores.

Designing a reputation system based on Blockchain suppresses the previous vulnerabilities and also prevents attacks to a system. Reputation management in the chain can then be applied in the most known cases: e-commerce websites or services who demand classical rating exchanges.

Although, Blockchain also has vulnerabilities and there are identified attacks to it. The previous attacks to P2P networks and reputation systems, identified above such as the Sybil attack and unfair ratings, can also happen in the Blockchain. Blockchain specific attacks like 51% Attack and Re-entry Attack can also occur and compromise the reputation system. In chapter 5 these attacks will be discussed as well as its strategies and countermeasures and how Blockchain can resist with more accuracy comparing to P2P networks.

4.3.1 Transacting

A good measure to implement is creating receipts of the transactions. A peer after receiving required service or file sends back a transaction with the feedback and timestamp. Dennis and Owen [9] made this approach to suppress unfair ratings.

Miners can do the task of a central authority simply communicating with both sender and receiver, asking a proof to each one of them. In a traditional Blockchain application, miners also have this task but in this scenario validating transactions involve validating a receipt from all entities involved. Another issue with the existent P2P reputation approaches is the unfair ratings. Quantifying reputation can be a solution simply implementing a binary scheme, 1 for satisfactory feedback and 0 for unsatisfactory transactions and it is possible in a Blockchain case.

The transaction process is assured by the public-key cryptography, that is an essential part of the Blockchain protocol. This avoids attackers to steal and get access to nodes' records, as mentioned by Sharma [33]. Creating transaction integrity is crucial mainly when it comes to reputation. Elliptic Curve Digital Signature Algorithm (ECDSA) is the algorithm used to create a set of private and belonging public key. The public key, with the help of a hash function, creates a public address that the nodes use to send and receive transactions. Each node has a private key that has to be kept secretly. Private keys sign transactions proving that the node who did it, is the true owner.

The sender uses its private key to sign a transaction to the requester. After receiving the requested data, the requester sends a receipt of the transaction to the miner (4.3). The miner acts as a manager and intermediates the transactions between sender and receiver.

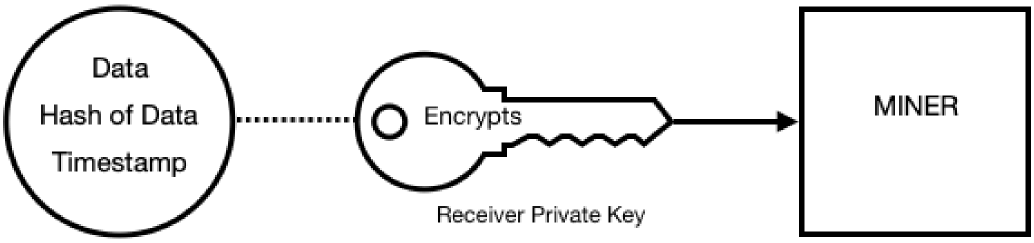


Figure 4.3: Receipt process to be sent to miner

The receipt, an approach made by Dennis et al. [9], is a proof containing the reputation score, timestamp and a hash of the received file. This proof avoids fake reputations given based on fake transactions, as unfair ratings are given by users that received the data required but claim the opposite. The transaction receipt must include:

- Sender part
  - Data requested
  - Hash of requested data
  - Timestamp

  - Private key
- Receiver part
  - Data requested
  - Hash of requested data
  - Timestamp
  - Private key

The following step is made by the miner responsible for verifying the transactions detailed with the received receipt.

Miners ask both users for the file hash and a nonce to prove that they sent and received the required data. The verified transactions are then grouped in a block waiting for confirmation. In the request made by the miners to the transaction participants, it is included a hash of the transacted file and a nonce of each user. This is important to the miner so all transaction can be validated.

4.3.2 Quantifying and Calculating Reputation

Most of the existent approaches fail in proposing strategies to reduce the subjectivity of human judgment. In some scenarios, the subjectivity is understandable, for example in e-commerce platforms. The range score is wide and a buyer can be satisfied with a product or service but does not give the maximum score for some reason. This brings imprecision to the reputation system. A precision strategy is missing. There are no proposals that bring ideal solutions for reputation quantification. The existing approaches (binary metric or more precise calculations) have pitfalls. A solution that is not so straightforward as the binary system (1 for positive or 0 for negative) but at the same time mitigates the unfair rating problem is missing.

4.3.3 Proof-of-Reputation

Consensus algorithms consist in the rules that a group of users have to respect while they interact in the system. It is a method to decide within the group.

These algorithms establish the rules concerning the majority of the votes but it also does it in order to benefit the network. The aim is to create equality and fairness in the environments. Without a central authority, users need to agree to the network can be expanded and the system can work by itself without an entity managing it. Consensus algorithms also prevent malicious activity by miners (faking transactions and blocks' creation). According to Anwar [2], the main goals are coming to an agreement, develop collaboration, co-operation between users, promote participation and activity. Taking the example of Bitcoin, its consensus rules are avoiding double spending, correct format of blocks, a certain amount of reward for miners. Blocks who do not respect these rules will be rejected.

Proof-of-Reputation (PoR) is a consensus algorithm that shows the contribution of each peer to the network and at the same time the growth, security and stability. A PoR approach can avoid the usage of miners in the network, avoiding also competition between nodes to get the right to publish blocks. The consensus algorithm allows dismissing miners because they work as intermediates and as small central authorities. In a reputation system scenario, the block responsible for generating blocks is the one with higher reputation score. Without the need for competing, there are no mathematical problems to be solved and this is crucial to have a cheaper environment. In PoR it is necessary to define rules such as the registration process and ensuring secure transactions (details in chapter 5).

In this section a possible PoR approach is detailed based on Gai et al. [16] Reputation-Based Consensus Protocol. Gai et al. [16] defined a recruitment process where each candidate has a pair of a cryptographical key for authentication and digital signature. This process is made on a control layer built into the blockchain nodes. The public key of the candidate is submitted to the registry as the ID, more specifically the hash, and then it is broadcasted to all participants in the network so they can authenticate it.

In the system itself, each node has a copy of all public keys of other participants. This allows participants to authenticate potential candidates and verify transactions. In a transaction between b_i and b_j, where $i, j \in N$, is signed by b_i's private key $Sig(n_i^j)$, where n is the number attributed to represent the transaction between b_i and b_j.

Maintaining Consensus

In the classic Bitcoin, PoW promotes competition between blocks to solve the mathematical question and gain access to publish blocks, as an incentive. In PoR the block with higher trust value can organize transactions and stores them in a block. As it happens in PoW, the remaining participants validate the process [48].

Motivating Publication

The reward for publishing blocks is getting reputation scores, which it will incentivize blocks to do so. Block publication attributes trust score and participants with higher trust scores which helps them to be abler do it [16].

Transaction Broadcast

PoR approach has a similar process as detailed in 4.3.1, but without using miners as intermediates. The sender sends the requested service or file signing it with its private key and the hash of the service or file and timestamp. The receiver verifies the data recalculating the hash and produces a receipt with the timestamp, hash of the received file or service digital signature and **reputation score** [9][16]. The transaction is represented by $Tr(b_A - > b_D = r)$, where r is the reputation score attributed by block A to block D.

In the previous approach [9], the miner intervention was based on intermediating the transaction between sender and receiver and then asking for a proof to both to validate the transaction. In PoR case, the receiver broadcasts the transaction to the entire network (figure 4.4), remembering that miners do not exist.

Then the quantification of reputation has to be defined. The solution can be binary, as Dennis et al. [9] proposed 1 for satisfactory and 0 for unsatisfactory transactions, making n_i^j, where $n \in \{0, 1\}$. Other simple solution, avoiding a binary rating, is to quantify reputation in a range from 0 to 1, where $n \in [0, 1]$. As detailed before, the binary solution mitigates unfair ratings and *bad-mouthing* situations and makes the reputation system more straightforward. On the other hand, the binary rating quantification does not allow a rater to distinguish a good transaction from an excellent one, it just attributes good or bad, positive or negative.

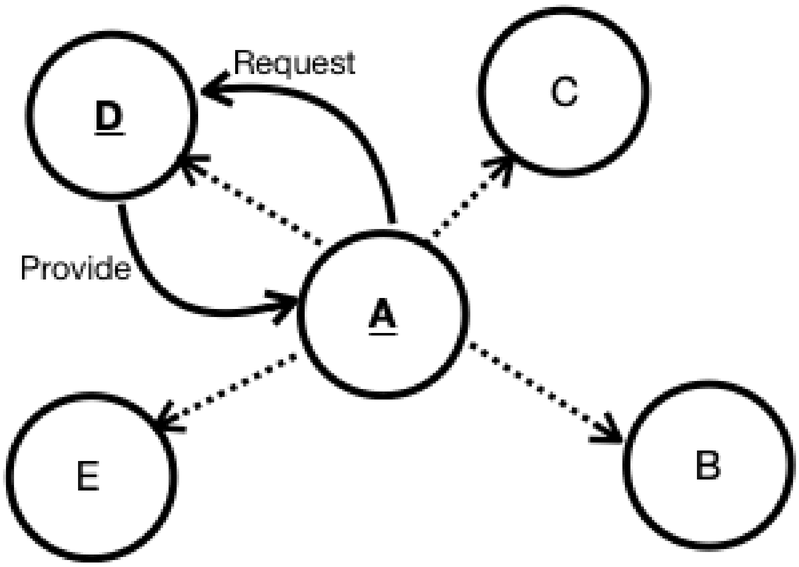


Figure 4.4: Broadcast Transaction and Reputation Score [16]

4.3.4 Limitations

Blockchain based reputation systems have limitations. While P2P networks have no limitation concerning the number of peers, Blockchain networks have a limited number of transactions per second, usually 7 to 10. Some cryptocurrencies implementations, like Litecoin, support around 55 [26]. Consequently, reputation systems in Blockchain will not be able to support so many transactions that the same systems implemented in a P2P network. A solution to this problem is based on removing the maximum size of each block. Having a 5MB block would allow making 50 transactions per second. Decreasing the time a block needs to be mined, gives the chance to increase the transaction number per second. In average, a block is mined in every 10 minutes [13]. It could be reduced to 5 minutes.

Dennis et al. [9] mention that when miners receive more than 7/10 transactions per second they will get forced to queue reputation scores. This will create problems for users because it will lead to delays and it can also create the possibility do denial of service (DoS) attacks. In such a scenario, malicious nodes can force miners sending them a huge number of transactions leading to an expensive computational verification of all transactions. Legit transactions from benign users will be delayed.

Another problem implementing reputation system in Blockchain is the network growth turning it expensive and demanding heavier resources. Each block has 1MB size. Considering the usual Bitcoin network growth, blockchain can increase 144MB a day, which means 53GB a year [4].

These are some issues that need to be solved in order to turn reputation systems in Blockchain a sustainable solution. In the next chapter, security measures and Blockchain attacks will be detailed.

Resume

	Blockchain	
Features / Issues	Solved	Unsolved
Security measures	✓	
Consensus	✓	
Transactions / second		✓
Scalability		✓
Broadcasting protocols	✓	
Transaction receipts	✓	
51% Attack	✓	
Re-entry Attack	✓	
Collusion	✓	
Unfair ratings (bad-mouthing attack)	✓	
Sybil attack	✓	

Table 4.2: Blockchain Solved / Unsolved Issues

- Security measures - effective measures to mitigate attacks and strategies to prevent malicious activity
- Consensus - consensus algorithm to define rules in the Blockchain so all participants agree and act accordingly
- Transactions/second - the number of limited transactions per second
- Scalability - system growth capacity without losing consistency
- Broadcasting protocols - defined protocols to broadcast transactions the entire chain
- Transactions receipts - receipts creation for participants after making transactions

- 51% Attack - getting more than half of Blockchain's computational power enables entire network control
- Re-entry Attack - participants with low reputation due to malicious activity reenter to clear its reputation
- Collusion - a group acts together to develop malicious activity
- Unfair ratings - provide dishonest feedback due to a legit transaction
- Sybil Attack - the creation of multiple identities by the same participant

5 Security for Reputation Systems in Blockchain

In this chapter, threats to the Blockchain and reputation systems (Section 5.1) will be analyzed as well as countermeasures to mitigate security issues (Section 5.2).

Security in reputation systems is a serious topic. It is not hard to find which reasons take an attacker to execute attacks in such a system. The most popular application of reputation is implemented in e-commerce websites or services exchange community where benefits are on the line, e.g. discounts and premium access to services for highly reputed users [35].

Without a decentralized entity managing the whole reputation system, where the security measures have to be addressed and executed, in a decentralized system fighting vulnerabilities and implementing defensive strategies is challenging.

5.1 Blockchain and Reputation Systems Attacks

In this section, the following attacks in Blockchain technologies and in reputation systems will be discussed:

- 51% Attack (Blockchain attack)
- Sybil Attack (Blockchain and reputation system attack)
- Unfair Ratings - bad mouthing attack (reputation system attack)
- Re-entry Attack (Reputation system attack)

These attacks can be performed separately or at the same time. An attacker can perform a Sybil attack (multiple entities creation) to provide unfair ratings against certain participants. 51% attack can also be developed so then attackers can give dishonest feedback to others.

5.1.1 51% Attack

In Blockchain, preventing fake blocks and transactions is automatically mitigated by the features of the technology. An attacker to do so has to gain control of 51% of the computational power of the entire chain.

Gaining control of an important part of the network gives the opportunity to the attacker to interfere in the process of recording new blocks. Under this scenario attackers can also prevent miners to complete other blocks, getting rewards and blocking other blocks' transactions, improving by manipulation specific peers' reputation and violating others.

It is hard to perform this attack. As said before, attackers have to gain access to more than half power of the network. According to Blockchain.com [5], in November Bitcoin computation power reached around 57,500,500 TH/s (TeraHash). Physically saying, an attacker would need 28,862 times the combination of worlds' 500 most powerful computers [6], making Blockchain very secure. The robustness of the system mitigates itself this threat.

What if the attack is done in the first times of the network? In this case, it would not be difficult to perform 51% attack because there are not many machines supporting it. Merging the mining process is a possible solution. It will be discussed in section 5.2.1.

5.1.2 Sybil Attack

Sybil attack consists on a node creating multiple identities to manipulate unreal transactions and manipulated feedbacks without contributing to the system and contributing to fool the reputation system. Danezis et al. [8] say that the Sybil attack is possible because node creation is a cheap process, new nodes are equal (have the same 'benefits' as older nodes) and new nodes are treated being less relevant. Blockchain has alternatives to fight this attack but with some drawbacks. Increasing the cost of node creation is helpful but it has to be carefully thought because the aim is not avoiding potential honest users to join the network. According to Garner [17] mining can be a protection because in PoW, if a user wants do mine to create a new identity, it has to use a new computer with processing power to contribute.

Implementing a chain of trust where just highly reputed users can invite new members to join the network, it also prevents Sybil. The system would turn more selective. A probationary system can be an advantage where new users do not have all the privileges and just gain score after contributing and after a certain amount of time. Both Garner [17] and Dennis et al. [9] propose connecting identity to the IP address of the user. Nowadays IPs are expensive to buy and the search of IP addresses is bigger than the offer. Systems such as eBay address an account to an e-mail address, but it is easy to pass over. Identity-based encryption systems can generate public keys based on the e-mail address but this measure would require a centralized entity to generate the keys.

Reputation systems help to reduce Sybil attacks weighting user score. Older users with proved activity and contributions have more power, reducing the benefits of new users (possible identities from a user deploying a Sybil attack). Potential malicious users would have irrelevant action effect in more reputed nodes. This is really important because it also reduces the risk of collusion. It is a classical attack in reputation systems. It is the main point of Sybil, having different identities so they can collude between them and perform a malicious activity, in reputation system scenario, raise their own reputations and lowering other nodes score.

Basically it is almost impossible to completely avoid Sybil attacks, but there are several countermeasures.

5.1.3 Unfair Ratings

Also known as *Bad-mouthing attack* consists on providing dishonest feedback when the part that is being rated respected the protocol and did not cheat or performed an unexpected action. It is the easiest attack to perform because any participant with no technical skills (IT security, hacking) can perform it. It just needs to provide a dishonest rating. The aim of this attack is to decrease some node reputation. In a reputation system based on Blockchain having a miner, as detailed in chapter 4, asking both users for the file hash and a random nonce to prove the transaction parameters between sender and receiver. With this procedure is no longer possible to provide unfair ratings because miners hold a cryptographic proof that the sender sent the requested data. Schaub et al. [32] propose the use of tokens. Users can only submit feedback about a sender for a transaction that really happened and the user was involved in. Anyway, this does not prevent unfair scores, just ensures that both parties were involved in the transaction.

The unfair rating problem can be also mitigated by implementing a quantification of the reputation system. In a feedback protocol where a user can only submit a score of 1 or 0, there will not be such problems with unfair comments. 1 for a satisfactory transaction (service or good received as requested) or 0 for non-satisfactory transfer (data does not coincide to what was really asked). It is a binary system used to decrease the rating subjectivity [9].

5.1.4 Re-entry Attack

Re-entry attack consists in a user developing a malicious activity and after building a negative reputation, losing privileges in the system, tries to generate a new account and develop again malicious activities with it.

The success of this attack is also connected to the cost of entry.

Nojoumian et al. [25] propose an interesting rational trust model to deal with the re-entry attack. They propose two trust functions that consider the lifetime of a node in the system and its reputation evolution concerning their lifetime in the system.

$$f_1 = (t_i^{p-1}, \alpha_i) \tag{5.1}$$

The inputs of f_1, t_i^{p-1} is the trust value of a node in a period $p-1$. α_i means the contribution from an user. The function calculates the contribution of a node in a certain range time.

$$f_2 = (t_i^{p-1}, \alpha_i, l_i) \tag{5.2}$$

f_2 has one more parameter, l_i related to the lifetime of a user in the reputation system. The function computes the number of interactions users have in a certain range of time.

While using f_2 it is possible to implement rules concerning users' lifetime. Imagining a scenario with U_i and U_j with equal trust values, $t_i^{p-1} = t_j^{p-1}$, and contributions, $\alpha_i = \alpha_j$, but U_i having a longer lifetime in the system, $l_i > l_j$, U_i can receive higher reputation score than U_j. This rule will lead to fairness in the system persuade users to develop malicious activity and then returning to the system under another identity.

Applying the above scenario to a real e-commerce situation, as Nojoumian et al. [25] refers, a seller can fall into temptation of selling defected products rather than selling good condition ones if f_1 is implemented. The seller could then disappear with that identity and building business under another one. Deploying f_2 would persuade the seller to sell the good condition product because after losing reputation, re-entering would demand much effort to establish his position as a reputed seller.

5.2 Measures to Reduce Malicious Activity

Implementing a reputation system is itself a measure to make a system more secure. It is important to mitigate the previous attacks and malicious activity in a Blockchain system.

The most common attacks are identified and countermeasures too. Anyway, there are policies and strategies to mitigate possible different attacks and vulnerabilities that systems have.

5.2.1 Merge Mining

Merge mining is a process that allows miners mining more than one block concurrently. The hash power a miner needs to build two blocks is the same as the combination of hash power needed for two miners build one block each.

The miner builds a block for both hash chains using same hash calculation, securing both blocks. Work units based on this block are then assigned to miners. When a miner solves a block, the block is re-assembled with the completed PoW and submitted to the correct block in the chain. And it does not reduce the hashing power of the Blockchain. It increases the global hashing power of the reputation system.

Merge mining is one of the strategies to mitigate the 51% attack, essentially in the initial time of the chain when there are fewer blocks and it is easier to gain access to 51% of the computational power.

Dennis et. al. [9] states that Bitcoin network miners can use their hashing power on other networks' reputations systems. This process does not affect the Bitcoin network, as it will not lose hashing power, but on the other hand, increases the total hashing power and the security of the reputation. In such scenario, attackers would need to gain control of both Bitcoin network and the specific reputation system.

5.2.2 Transaction Filtering

Transaction filtering is a method used to avoid possible dishonest ratings. Gai et al. [16] propose an algorithm consisting of testing the signature of the sender and verifying if the participant is authenticated in the system. Then check if both sender and receiver appear in the current block, replace the existent transaction (stored in the block) for a more recent one. This update aims to keep just one transaction record for each pair of participants in order to reduce unfair rating attack. When the number of filtered transactions reaches the limit, the remaining transactions are packaged in an alternative block.

5.3 Attacks and Malicious Activity Countermeasures Resume

In this section, it is made a resume of Blockchain and reputation system attacks, as well as the measures used to mitigate malicious activity.

- 51% Attack
    - Problem: if attackers gain access to the network major machines, they get control of the system.
    - Solution: Merge mining using Bitcoin network miners to use their hashing power, increasing the hashing power of the reputation system. In this situation, an attacker has to gain access to both the Bitcoin network and the reputation system.
- Sybil Attack

  - Problem: an attacker creates multiple identities to manipulate unreal transactions and manipulate feedbacks.
  - Solution: increasing the cost of node creation. Mining can also be a protection because in PoW if a user wants to mine to create a new identity, it has to use a new computer with processing power to contribute [17]. Connection IP identity to IP addresses of a user also mitigates this attack.

- Unfair Ratings
  - Problem: attacker provide dishonest feedback in order to decrease the victim's reputation.
  - Solution: miners hold a cryptographic proof that the sender sent the requested data. Users can only submit feedback about a sender for a transaction that really happened and the user was involved in. Creating receipts of transactions also mitigates *bad-mouthing* attacks.
- Re-entry
  - Problem: an attacker develops malicious activity and after building a negative reputation, losing privileges in the system, tries to generate a new account and develop again malicious activities with it.
  - Solution: calculate reputation according to previous reputation scores and with the lifetime of a user in the system. Older users get higher reputation scores making the same transactions as a more recent user.
- Merge mining: miners mine more that one block concurrently. It does not reduce Blockchain hashing power. Contrariwise, it increases the global hashing power. This is important to mitigate 51% attack.
- Transaction filtering: algorithm created to verify if participants are authenticated and test the signature of senders and receivers. This algorithm mitigates unfair ratings, verifying the transactions.

6 Conclusion

In this paper details concerning reputation in Blockchain systems were examined. It was provided a structured analysis of P2P networks and reputation systems applied in such scenario so the reader can follow the Blockchain analysis and the goals of a reputation system based on it. It was also discussed how attacks to the Blockchain can compromise the reputation system.

Online Reputation is fundamental to take the internet to higher stages. Nowadays most services in our communities are used on the internet, such as governmental services or e-commerce world. People started to shop online and physical commerce is decreasing. Trust and reputation are important parts in the process so people can rely on online services instead of the traditional methods. The implementation of a reputation system helps to promote high-quality content to be exchanged within a community (reducing free-loaders). The reputation system helps to build trust between participants and motivates members to give feedback.

Decentralizing reputation and trust make reputation systems more reliable to the community. Amazon and eBay provide systems that allow sellers and buyers to rate each other so the rest of the community can make an analysis before making decisions. But these companies use centralized systems. They have the control of the rating platform so it is not guaranteed that they manipulate feedbacks and reputation to achieve their goals (e.g. having just users with high reputation scores in their systems to show that their services are good and reliable). For this reason, several researchers studied and proposed decentralizing reputation mainly in P2P networks.

In the third chapter, P2P networks and its design were analyzed before an examination to reputation systems in these networks. Several approaches, among their advantages and pitfalls, were detailed. The main challenge is to develop a system without a central authority managing it. It is important to design strategies to avoid malicious activity and to mitigate several threats such as avoiding unfair ratings and collusion. Implementing a reputation system in a scalable P2P network is also important to the system can grow to be applied in real scenarios (e.g. eBay) that have millions of users. Reputation model is a group of rules used to design a reputation system to answer the previous questions. The participants know the rules and act accordingly.

In the fourth chapter, Blockchain design and features were analyzed before the reputation discussion. Security is a valid reason to implement a reputation system based on a Blockchain technology. Miners have an important role when it comes to applying reputation systems in the Blockchain. They intermediate transactions between senders and receivers. Then miners collect proof of transaction from senders and receivers and their reputation scores. An approach without miners it was also discussed a Proof-of-Reputation protocol. PoR is a consensus algorithm that defines rules so a group of users respect while they interact in the system. It is a method to decide within the group. Consensus algorithm allows dismissing miners because they work as intermediates and as small central authorities. In a reputation system scenario, the block responsible for generating blocks is the one with a higher reputation score. To broadcast a transaction, the sender sends the requested service or file signing it with its private key and the hash of the service or file and timestamp. The receiver verifies the data recalculating the hash and produces a receipt with the timestamp, hash of the received file or service digital signature and reputation score.

In both scenarios reputation quantification is important. It is crucial to define a method to calculate the reputation and its attribution. A binary system is a good option, where 1 is satisfactory and 0 is non-satisfactory. This approach does not distinguish an excellent transaction from a satisfactory. Although it does not allow receivers to attribute unfair ratings. Another solution is to define a wider calculation that accepts different values. The problem with this method is the subjectivity and it is easier to provide unfair ratings. Although, Blockchain provides a receipt system. Receivers have to provide a receipt proving they received the requested transaction.

Blockchain is a secure technology. Anyway, attacks can occur and affect the system itself and the reputation system. Attacks such as Sybil attack, unfair ratings, 51% attack and re-entry. Sybil attack is the same attack that happens in P2P networks. An attacker gets multiple identities to create unreal transactions. In Blockchain is possible to mitigate this attack by creating enrollment policies or increasing the cost of node creation. 51% attack consists in a group of attackers getting access to the computational power of the network majority. In Blockchain is almost impossible to develop this attack because the attackers would need to get access to approximately 500 most powerful computers in the world. Unfair ratings consist of giving dishonest feedback. In the chain, it is possible to create a receipt mechanism so the receivers have to provide a receipt after receiving the requested services. This receipt is the proof of transaction, so there is no chance to give cheated ratings. Re-entry attack is possible to mitigate by creating enrollment policies and attributing a real IP address to each identity.

Developing measures and policies to reduce malicious activity are also fundamental. Mechanisms such as merged mining or transaction filtering help the system to be more secure. Merged mining allows mines to mine more than one block at the same time. The hash power a miner needs to build two blocks is the same as the combination of hash power needed for two miners to build one block each. This strategy is important to mitigate the 51% attack in the initial time of the chain when the block amount is less. Transaction filtering is used to avoid dishonest feedback. An algorithm can solve this issue by testing the signature of both the sender and receiver and verifying if both are authenticated in the system.

To ensure the whole system security and robustness, attacks on operating systems (OS), IP networks, decentralized P2P / Blockchain and reputation systems must be mitigated.

Security measures to protect the OS from threats, viruses, worms or malware are crucial to secure any computer assets. An insecure OS can compromise participants operations in the network. At the same time, IP network protection has to be ensured because almost every network uses TCP/IP protocols. Users need to isolate their networks and at the same time send and receive traffic over the Internet. The authentication and privacy mechanisms of secure IP provides the basis for a security strategy for all users.

6.1 Future Work

Privacy is a critical factor in creating an approach that can deal with reputation systems and anonymity at the same time will be needed. The biggest challenge is to discover methods to improve privacy in reputation systems without compromising the anonymity in Blockchain technologies to mitigate attacks. Designing reputation systems for anonymous networks while ensuring anonymity will be challenging.

The crucial improvement that reputation systems based on Blockchain technologies need is to reach an approximate number of transactions per second as systems, as eBay, do. As detailed in chapter 4, Blockchain handles 7 to 10 transactions per second. Some specific Blockchain systems like Litecoin support around 55. eBay system handles thousands per second. To apply these reputation system scenarios in real situations, this is the first needed improvement.

An approach that does not need miners, as the Proof-of-Reputation discussed in section 4.3.3, will be capable to increase the transaction amount per second, skipping the utility of using PoW in reputation systems.

Zilliqa, a new Blockchain platform based on the technology of sharding [1], solves the scalability issues in Blockchain platforms. Sharding enables transactions, and smart contracts to be confirmed in parallel, without the risk of double-spending. Zilliqa reached 2,488 transactions per second using 3,600 Amazon EC2 [2] instances. Zilliqa's high throughput Blockchain will perform up to 1,000 times faster than Ethereum [50]. In the future, applying a reputation system on such a Blockchain platform will help to implement such a system in real-world scenarios, especially when it comes to supporting a considerable amount of transactions per second.

[1] Sharding - a setup that is based on multiple partitions that create many pieces of a system (e.g. database) that are then referred to as shards. This practice can help with server hosting and can also contribute to faster query times by diversifying the responsibilities of a database structure [39].

[2] Amazon Elastic Compute Cloud 2 - forms a central part of Amazon.com's cloud-computing platform, Amazon Web Services (AWS), by allowing users to rent virtual computers on which to run their own computer applications [45].

Glossary

TERM	DESCRIPTION
P2P	Abbreviation for Peer-to-Peer Network
DoS	Abbreviation for Denial of Service
DDoS	Abbreviation for Distributed Denial of Service
SYN	Abbreviation for synchronize. SYN is a TCP packet sent to another computer requesting that a connection be established between them
ACK	Abbreviation for acknowledgment. ACK is an answer given by another computer or network device indicating to another computer that it acknowledged the SYN/ACK or other request sent to it
RFC	Abbreviation for Request for Comments.
TTL	Abbreviation for Time-to-Live
Tx	Abbreviation for transaction
SHA	Abbreviation for Secure Hash Algorithm
PoW	Abbreviation for Proof-of-Work
PoR	Abbreviation for Proof-of-Reputation
TH	Abbreviation for TeraHash, hashpower measure
OS	Abbreviation for Operating System

Table 6.1: Glossary Items

Bibliography

[1] Abdul-Rahman, Alfarez ; Hailes, Stephen: in Distributed Trust Model. (1997)

[2] Anwar, Hasib: Consensus Algorithms: The Root Of The Blockchain Technology. (2018). – URL https://101blockchains.com/consensus-algorithms-blockchain/

[3] Avinash Chaudhari, Pradeep G.: Analysis of various attacks on P2P networks. (2014)

[4] BitcoinNews.com: The Implications of Bitcoin's Blockchain Growth. (2018). – URL https://bitcoinnews.com/the-implications-of-bitcoins-blockchain-growth/

[5] Blockchain: Hash Rate. (2018). – URL https://www.blockchain.com/charts/hash-rate

[6] Cohen, Reuven: Global Bitcoin Computing Power Now 256 Times Faster Than Top 500 Supercomputers, Combined!: https://www.forbes.com/sites/reuvencohen/2013/11/28/global-bitcoin-computing-power-now-256-times-faster-than-top-500-supercomputers-combined/. (2013)

[7] Cosset, Damien: Blockchain: What is Mining? (2018). – URL https://dev.to/damcosset/blockchain-what-is-mining-2eod

[8] Danezis, George ; Schiffner, Stefan: On Network formation (Sybil attacks and Reputation systems). (2007)

[9] Dennis, Richard ; Owen, Gareth: Rep on the block: A next generation reputation system based on the blockchain. (2015)

[10] Dingledine, Roger ; Freedman, Michael J. ; Molnar, David ; Parkes, David ; Syverson, Paul: Reputation. (2003)

[11] Douceur, John R.: The Sybil Attack. (2002)

[12] Eddy, W.: TCP SYN Flooding Attacks and Common Mitigations. (2007). – URL https://www.ipa.go.jp/security/rfc/RFC4987EN.html#3

[13] Eremenko, Kirill: How does Bitcoin / Blockchain Mining work? (2018). – URL https://medium.com/swlh/how-does-bitcoin-blockchain-mining-work-36db1c5cb55d

[14] Farmer, F. R. ; Glass, Bryce: Web Reputation Systems. (2010)

[15] foundation, DREP: DREP Chain - A Novel Sharding Infrastructure Technical Whitepaer. (2018)

[16] Gai, Fangyu ; Wang, Baosheng ; Deng, Wenping ; Peng, Wei: Proof of Reputation: A Reputation-Based Consensus Protocol for Peer-to-Peer Network. (2018)

[17] Garner, Bennett: What's a Sybil Attack and How Do Blockchains Mitigate Them? (2018). – URL https://coincentral.com/sybil-attack-blockchain/

[18] Harding, David A.: Blockchain pruning problems and solutions. (2015). – URL https://dtrt.org/posts/blockchain-pruning-problems-and-solutions/

[19] Incapsula, Imperva: TCP Syn Flood. (2018)

[20] Jøsang, Audun ; Ismail, Roslan ; Boyd, Colin: A survey of trust and reputation systems for online service provision. (2006)

[21] Kamvar, Sepandar D. ; Schlosser, Mario T. ; Garcia-Molina, Hector: The EigenTrust Algorithm for Reputation Management in P2P Networks. (2003)

[22] Kwok, Yu-Kwong R.: Peer-to-Peer Computing: Applications, Architecture, Protocols, and Challenges. (2012)

[23] Li, Huaizhi ; Singhal, Mukesh: Trust Management in Distributed Systems. (2007)

[24] Lisk, Academy: Blockchain Basics. (2018). – URL https://lisk.io/academy/blockchain-basics

[25] Nojoumian, M. ; Golchubian, A. ; Njilla, L. ; Kwiat, K. ; Kamhoua, C.: Incentivizing Blockchain Miners to Avoid Dishonest Mining Strategies By a Reputation-Based Paradigm. (2018)

[26] O'Keeffe, Daniel: Understanding Cryptocurrency Transaction Speeds. (2018). – URL https://medium.com/coinmonks/understanding-cryptocurrency-transaction-speeds-f9731fd93cb3

[27] Oracle: Best Practices for Reputation Management. (2012)

[28] Prabhakar, Ajith V.: BlockChain Fundamentals Part 1. (2018). – URL https://ajithp.com/2018/02/01/block-chain-fundamentals/

[29] Ray, Shaan: Merkle Trees. (2017). – URL https://hackernoon.com/merkle-trees-181cb4bc30b4

[30] Resnick, Paul ; Kuwabara, Ko ; Zeckhauser, Richard ; Friedman, Eric: Reputation Systems - Magazine Communications of the ACM. (2000)

[31] Resnick, Paul ; Zeckhauser, Richard: Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. (2001)

[32] Schaub, Alexander ; Bazin, Rémi ; Hasan, Omar ; Brunie, Lionel: A trustless privacy-preserving reputation system. (2016)

[33] Sharma, Toshendra: How does Blockchain use public key cryptography? (2018). – URL https://www.blockchain-council.org/blockchain/how-does-blockchain-use-public-key-cryptography/

[34] Shredeh, Khalied: Analysis of Attacks and Security Issues on the Peer-toPeer Networks. (2016)

[35] Srivatsa, Mudhakar ; Xiong, Li ; Liu, Ling: TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks. (2005)

[36] Such, Jose M.: Attacks and vulnerabilities of trust and reputation models. (2013)

[37] Tanenbaum, Andrew S. ; Steen, Maarten van: Distributed Systems. (2017)

[38] Team, Blockgeeks: Blockchain Scalability: When, Where, How? (2017). – URL https://blockgeeks.com/guides/blockchain-scalability/

[39] Techopedia: Sharding. (2018). – URL https://www.techopedia.com/definition/22041/sharding

[40] Vavilis, Sokratis ; Zannone, Nicola: A Reference Model for Reputation Systems. (2014)

[41] Wang, Lin: Attacks Against Peer-to-peer Networks and Countermeasures. (2006)

[42] Wang, Ping ; Aslam, Baber ; Zou, Cliff C.: Peer-to-Peer Botnets. (2010)

[43] Wiki: TrustedSource. (2016). – URL https://en.wikipedia.org/wiki/TrustedSource

[44] Wiki: Distributed Hash Table. (2018). – URL https://en.wikipedia.org/wiki/Distributed_hash_table

[45] Wikipedia: Amazon Elastic Compute Cloud. (2018). – URL https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud

[46] Xiong, Li ; Liu, Ling: PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. (2004)

[47] Yu, Bin ; Singh, Munindar P.: Incentive Mechanisms for Peer-to-Peer Systems. (2003)

[48] Yu, Jiangshan ; Kozhaya, David ; Decouchant, Jeremie ; Esteves-Verissimo, Paulo: RepuCoin: Your Reputation is Your Power. (2018)

[49] Zheng, Zibin ; Xie, Shaoan ; Dai, Hong-Ning ; Chen, Xiangping ; Wang, Huaimin: Blockchain Challenges and Opportunities: A Survey. (2017)

[50] Zilliqa: Zilliqa tackles scalability with Sharding blockchain. (2018). – URL https://www.finextra.com/pressarticle/73493/zilliqa-tackles-scalability-with-sharding-blockchain

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: ______________________________

Vorname: ______________________________

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Technical Analysis of Reputation Systems based on Blockchain Technologies

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

________________	________________	__________________________
Ort	Datum	Unterschrift im Original