Risk Management for Blockchain Networks

Max Wolter
0060218257
max@alvalor.org

June 2018

University of Luxembourg
Faculty of Science, Technology and Communication

This thesis is presented for the degree of
Master of Information Systems Security Management

Academic Year: 2017 - 2018
Supervisor: Dr Nicolas Mayer

Abstract

Abstract

This paper explores the feasability of a modern risk management approach in the context of blockchain consensus networks. A framework is created on the basis of a theoretical, technical and social analysis of vulnerabilities on blockchain networks. The result is then applied to a real-world blockchain project with a method similar to popular risk management approaches.

Declaration of honesty

I hereby declare that, except where specic reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualication in this or any other university. This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specically indicated in the text.

Max Wolter

Contents

1 Introduction

The concept of a blockchain network was introduced in the form of Bitcoin in 2008 by an anonymous author under the name of Satoshi Nakamoto [3]. Since then, the potential of the underlying blockchain technology to serve as a trustless mechanism to find consensus on the state of an underlying system has spread to many applications beyond the maintenance of a ledger of account balances.

The main value proposition of a blockchain network is its ability to ensure the security of the consensus state in a verifiable and transparent manner. While the software & business ecosystem developed around a blockchain project are susceptible to the same information security risks as typical software and business projects, the core platform has its own unique security properties.

This paper explores the applicability of a traditional information security risk management approach to the context of blockchain networks. It makes an attempt to integrate the current research litterature in the blockchain space with the more mature sphere of information security risk management.

In a first step, we explore current academic literature to establish a framework which applies the principles of modern risk management methodologies to the context of blockchain networks. We define the scope and conditions under which we examine the topic and use them as a basis to evaluate a number of relevant vulnerabilities and risks.

As the second step, a small case study applies the framework to a real-world blockchain network. We use the established risks to identify vulnerabilities and exploits for the project, assess the related probability & impact and finally recommend a number of measures to improve the security of the blockchain network underlying the consensus state.

2 Framework

2.1 Asset Identification

When identifying risks, the usual approach dictates that we identify assets which are vulnerable and use them as a basis for the analysis [1]. For a

blockchain network, this proposition is a difficult one; the network is composed of thousands of nodes which communicate in a peer-to-peer fashion. While each node can be compromised in its own right, there is no single tangible asset or actor that impacts the security of the consensus state as a whole. The same can be said for the periphery, consisting of infrastructure, tools and solutions built around the blockchain network.

Our approach posits that we can define the abstract concept of the consensus state itself as the asset which needs to be protected. In that context, we can focus on vulnerabilities that are unique to blockchain networks. We can then establish our own security criteria which should be fulfilled in order for the consensus state to be considered secure. At the end, can evaluate the potential risks to the consensus state in terms of impact on these criteria.

There are various algorithms used to achieved consensus in blockchain networks. Where the type of consensus algorithm is relevant, we will focus on the two most common algorithms: proof-of-work (PoW) and proof-of-stake (PoS). Other models, such as the directed acyclical graph (DAG) approach or derivatives of the PoS approach, have not been explored in sufficient depth in scientific literature to provide a meaningful analysis.

2.2 Security Criteria

As alluded to earlier, the usual security criteria of confidentiality, integrity and availability are not a good fit for the abstract concept of the consensus state in a blockchain network. We can, however, establish our own security criteria which make sense in the context of blockchain networks. The foundation for these criteria should be found in the basic process of consensus finding and on the security guarantees that a blockchain network tries to provide.

Fundamentally, a blockchain network are provides and open, (mostly) transparent & global platform for potentially malicious participants to find consensus on the state of the system in a trustless manner. It does so by employing cryptography to prove events that happened in the past and economic incentives to cause events to happen in the future [27]. This implies a number of properties which have to be conserved in order to provide the desired environment.

First, we need to conserve the integrity of the cryptography which pro-

tects the events in the past. Once the participants have agreed on a the history of the consensus state, it needs to be reliable. One important security criterion for a blockchain network is therefore the **immutability** of its consensus state.

Second, we need to make sure that all participants of the blockchain network can access the consensus state in the designed way. This implies being able to read the latest valid consensus state and being able to write to the consensus state according to the consensus rules. We call this criterion the **accessibility** of the consensus state.

Last but not least, we need to conserve the balance between the cryptography and the economics of the platform in order to conserve a viable theoretical model. Disrupting the economic equilibrium of a blockchain network will undermine its incentive structure and make the security guarantees weaker. We call this aspect the **stability** of the consensus state.

2.3 Vulnerability Categories

Blockchain networks can have flaws inherent in the theoretical models of the consensus state. In this respect, possible exploits which rely on work in the field of mathematics, more specifically game theory, and information science, in particular in regards to distributed systems, can be explored. This category of risks is confined to the internal model of the consensus state, so these intrinsic aspects will be defined as **theoretical** vulnerabilities.

It is also possible to exploit technical means outside of the immediate model to disrupt the consensus state. One prime target could be peer-to-peer (P2P) topology underlying the blockchain network, where disrupting the characteristics of the network can undermine assumptions of the overlayed theoretical model. We will describe these external aspects as **technical** vulnerabilities.

Last but not least, a blockchain network is run by software which encodes the consensus rules, and this software is written by humans working for projects governed by humans. This means that the direction of a blockchain project, including the evolution of the consensus rules and algorithm, is subject to the weaknesses of human psychology and online communities. We call these aspects the **social** vulnerabilities.

2.4 Risk Analysis

Our method for analysing risks is based on a systemic definition of the information security risk management domain [5]. This nomencloture defines a risk as an event, occurring with a certain probability, leading to a certain impact on the security criteria of our asset. An event, in turn, is a threat agent using an attack method to exploit a vulnerability.

In this paper, we propose to extend the framework slightly to make a clear distinction between the attack method and the attacks themselves. Indeed, one attack method applied to one vulnerability can open the door to the execution of a variety of attacks in the context of blockchain. In general, this means that most mitigations address the vulnerability directly, rather than handling each attack on its own.

We establish the following modified definition as the basis for our risk analysis: a risk is the event with certain probability of a threat agent using an attack method to exploit a vulnerability of the blockchain network in order to execute an attack, impacting the immutability, accessibility or stability of the consensus state.

In general, attacks require either access to block generators, or they can be executed by an outside party; however, the threat agent is not identified as part of our analysis. Identification of the threat agent adds little value to risk management in the context of blockchain networks, which attempt to provide global security properties, and is thus beyond the scope of this paper.

2.5 Mitigation Controls

The collapse of one security criterion often opens up the door for a number of attacks, regardless of the attack method used to exploit the vulnerability. This means that the mere exploit of a vulnerability already undermines the security of the blockchain network. At the point where a blockchain network's security is unreliable, it is broken even if no attack is executed against any victim. Often, the level of impact is relevant in this consideration.

For instance, a narrow impact on immutability of specific transactions in the consensus state allows the execution of double spend attacks against one victim, but doesn't affect the rest of the blockchain network. In contrast, a wide impact on immutability allows the reversal of any transaction and thus

makes the blockchain network unusable due to unacceptable risks.

Similarly, a narrow impact on accessibility can simply mean that one network participant can not execute any transactions or has a wrong local view of the consensus state. However, if the accessibility to the blockchain network is globally disrupted, network participants can no longer rely on the consensus state and the blockchain network in no longer useful.

It can therefore be said that the mere exploitation of a vulnerability can already lead to the collapse of the security criteria of the consensus state, thus making the blockchain network useless for its intended purposes. This means that we can have controls on all levels: vulnerability, attack method or the attack itself.

3 Theoretical Vulnerabilities

3.1 Hashrate Distribution

The security of PoW blockchains relies on the the miners, which compete in a lottery to generate the next block in order to obtain the related block reward. Through this design, all miners are incentivized to work on the latest valid version of the blockchain history, as they would otherwise run a higher risk of not obtaining said block reward.

However, this also creates a fundamental vulnerability of the blockchain network, as already noted in section 11 of the original Bitcoin white paper [3]. With a sufficiently high hashrate, an attacker can overwhelm the consensus forming of the rest of the blockchain network and disable the security guarantees of the consensus state either temporarily or permanently.

There are a number of possible mitigations for a hashrate exploit. On the social layer, participants on the network can run algorithms which *automatically balance* their hashrate between mining pools, which can eliminate the risk of any pool operator acquiring a majority hashpower. Unfortunately, this solution is hard to monitor.

An alternative solution would deploy a *two-phase PoW* model [14]. It would require a second sophisticated cryptographic layer using Markov chains to be built between mining pools, but would at least be provably diminish

the impact of a majority hashrate exploit.

A much simpler and more complete solution is to simply wait for a certain *number of confirmations*. Indeed, the deeper in the history a transaction is found, the harder it is to overwhelm the previously accomplished PoW even with a majority hashrate [9]. Unless an attacker permanently has a majorty hashrate, this strategy will always work at some point in time.

Finally, a more fundamental change to the consensus state could make it harder to reverse already written blockchain history. Instead of enforcing the consensus rule of always following the longest chain of valid blocks, transactions could be bound to a history of the blockchain and the *transaction weight*, using different algorithms, could be a deciding factor between conflicting histories [8].

3.1.1 Majority Hashrate

The most fundamental breach of security for a PoW blockchain network is when one participant manages to acquire a majority of the hashrate on the network. It allows him to overpower the combined hashrate of all other participants and thus gives him the power to define the history of the consensus state on his own.

Certain double spend. With the absolute majority of the hashrate, an attacker can always overpower the competing hashrate. This allows him to reverse his own transaction and replace it with another transaction spending the same funds on another version of the history with absolute certainty. It impacts the immutability of the consensus state.

History reversal. Every time a competing participant generates a block, the attacker can generate an alternative version of the history. Due to his overwhelming hashrate, he will overcome and override the competing history. This allows him to both monopolize block generation and rewards on the blockchain network. It impacts the immutability and the stability of the consensus state.

In order to mitigate history reversals, *checkpoints* can be introduced at certain depths of the blockchain history. They can be broadcasted through the P2P network or exchanged through a trusted out-of-band system.

Transaction censorship. Due to his ability to reverse blocks at will, the attacker can decide to censor certain transactions by not including them in his blocks and reversing any block that contains them. This impacts the accessibility of the consensus state.

Targeted censorship can be avoided on blockchains that offer *anonymous transactions*. Due to the fact that nobody knows who is transacting, it's not possible to target any specific victims and only random or complete censorship of transactions is possible.

3.1.2 Minority Hashrate

Due to the nature of the PoW algorithm, which is a memoryless process, there is also significant time variance during block generation. This means that a number of attacks remain possible with a non-majority share of the hashrate on the network, allowing a malicious minority miner to exploit the vulnerability with a non-zero probability of success.

Probabilistic double spend. With a minority hashrate, an attacker still has a probability at generating one or several block at any point in time. This allows him to attempt generating a sufficient amount of blocks in time to reverse his own unconfirmed or even confirmed transaction at a non-zero probability. It impacts the immutability of the consensus state.

Transaction throttling. Being able to generate a portion of the blocks for the blockchain history, an attacker with a relevant amount of hashrate can slow down the confirmation of transactions he wants excluded from the consensus state; he even has a chance to reverse confirmation with a non-zero probability. It impacts the accessibility of the consensus state.

3.2 Block Propagation

Every miner participating in a proof-of-work (PoW) consensus finding algorithm relies on the propagation of valid blocks through the network in order to generate a state transition for the latest valid consensus state. The theoretical model assumes that all miners will freely share newly found blocks in order to obtain their block rewards.

However, the option to withhold a discovered block and thus hide it from the network is a theoretical possibility. Depending on the intentions of an attacker, this opens up a vulnerability on the consensus state. An attacker can withhold a block to gain profit in some other way, while still conserving a significant probability of obtaining the block reward.

One mitigation of the vulnerability itself is to make sure blocks not on the recent history are somehow punished. One implementation suggests suggests that blocks need to be received within a maximum acceptable time interval [21]. Another similar one relies on publishing time rather than generation time for blocks [26].

3.2.1 Finney Attack

The Finney attack is a more sophisticated version of a double spend that does not rely on having a significant amount of hashrate on the network [28]. Instead, the miner waits to generate a block during an arbitrary interval and only starts the attack subsequently. Once the block is found, it's withheld from the network and thus hidden. It includes a transaction that sends certain funds from the attacker to the victim.

The attacker now propagates a conflicting transaction that sends the funds to the victim as part of the transaction. As soon as the desired exchange has occurred in both directions, the attacker propagates his hidden block and reclaims the funds. It should be noted that the attacker risks another block being found before the attack was executed, thus risking to lose the block reward.

Unconfirmed double spend. As the block with the transaction is already mined, the attacker can execute a double spend on an unconfirmed transaction instantly. However, the attacker risks losing the block reward if another network participant generates another valid block during the execution of the attack. It impacts the immutability of the consensus state.

A previously unmentioned method to counter double spends in the case of the unconfirmed kind, which is the easiest to execute, is to *monitor the memory pool* of nodes on the network, either by placing observers or by relaying conflicting transactions.

3.2.2 Block Discarding

When a miner finds a block, he can simply discard it [12]. A lot of mining pools reward miners who contribute their work to the mining pool in proportion to the work completed. In order to get rewards, miners submit proofs for completed work to the mining pool. These so-called shares are significantly more frequent than finding a valid block, which makes the incured loss negligible.

Unfair economic loss. A malicious participant of the mining pool can submit his shares and receive most of his block reward. When he finds a valid block he discards it instead of submitting it to the mining pool. This will cause a significant loss to the pool operator, who will award the same rewards but see lower returns. It impacts the stability of the consensus state.

Some pool operators mitigate the risk by using a reward structure that attributes extra *rewards for blocks*, thus making the loss non-negligible for the malicious miner, while at the same time making it less costly for the mining pool operator, as he will pay out less rewards overall.

3.2.3 Selfish Mining

The most sophisticated block withholding attack is the selfish mining strategy [11]. A malicious miner who finds a block can withhold it from the network in order to gain a headstart for next block. Once the network finds a block, he can still release his block to contend for the block reward. However, he will significantly increase his own likelihood of finding the next valid block.

Unfair economic gain. An attacking miner can increase his profitability over other participants on the network by optimizing his mining strategy according to the selfish mining principle [15]. He will thus gain an unfair economic advantage that was not accounted for in the original model. It impacts the stability of the consensus state.

The authors of the original paper suggest accepting the new optimal mining strategy as the default for the network. In order to reduce the advantage for well-connected miners, they suggest propagating all competing block histories and *randomizing the choice* of parent. This would create a level playing field once more.

3.3 Cheap Validation

In the context of a proof-of-stake consensus algorithm, validators invest a negligible amount of resources into the generation of new blocks. As the generation is extremely cheap, any validator can generate blocks for many different versions of the block history. At the same time, validators can generate a long blockchain history in a short amount of time.

3.3.1 Nothing-At-Stake

The nothing-at-stake problem describes the fact that a validator can generate blocks for many different blockchain histories without being punished for it. There is nothing at stake for him to lose, so there is no incentive to resolve conflicts between different versions of the blockchain history.

Follow multiple histories. The fact that a validator will only receive a block reward for the generated blocks if they were generated on the blockchain history accepted by all network participants gives economic him incentive to generate blocks for all versions of the blockchain history. The strategy is superior as it guarantees the block rewards, but it stops conflict resolution. It impacts the stability of the consensus state.

In order to counter the incentive of generating blocks for all available blockchain histories, we need to introduce a way to punish validators for generating blocks on multiple blockchain histories. Some PoS blockchain networks do this by *blacklisting validators* working on more than one blockchain history, depriving them of block rewards [30].

Another possible solution is the introduction of so-called *slashing conditions* [31]. They define consensus rules under which validators will lose part of the funds they deposited in order to become validators. This inflicts economic loss on misbehaving validators which is more significant than the potential gain by following multiple blockchain histories.

3.3.2 Long-Range Attack

As costs for generating blocks in a PoS algorithm are negligible, any validator can generate as many as he wants. This allows the recreation of part or all of the blockchain history. A malicious group of validators can thus take over the blockchain network, starting at any point of the blockchain history

where they collectively held a sufficient amount of the staked tokens.

The exploit can be considered similar in effect to the majority hashrate attack for PoW blockchain networks. However, in the PoW context, the attacker still needs to invest the time to overcome the invested hashrate of the previous blockchain history. In the case of PoS, this is not the case, as block generation is cheap and the history can be recreated quickly. This makes the impact even more severe.

Finally, the exploit can be strengthened by combining it with the so-called stake-bleeding attack [34], where a validator will not generate blocks on the attacked blockchain history, thus slowing down its growth at the price of losing part of the staked funds. However, if his competing history prevails, the loss of funds will be reversed.

Attacks. The possible attacks are similar to the majority hashrate exploit: certain double spending, history reversal and transaction censorship. However, the history reversal is significantly more severe than for PoW blockchain networks. It impacts the same security criteria, respectively.

One solution to limit the impact of a long-range attack is the introduction of a *moving checkpoint*, which is a limit on the number of blocks that can be rewritten. This creates an upper bound on how far into the past history can be rewritten and introduces the concept of finality, which means that a block can never be changed after a certain point in time and can thus be considered final.

However, this approach does not address the need for new or reconnecting nodes to synchronize with the network over a long blockchain history. This leaves this relevant subset of nodes vulnerable. It can be addressed by introducing an *out-of-band history verification* of the blockchain, such as pulling checkpoints from a distributed network of trusted nodes.

Finally, *key-evolving cryptography* can be used to make sure that each private key can only be used to sign once, thus making it impossible to acquire past keys from validators to increase the share of validation power. Such approaches are still experimental and in the early stages of research.

4 Technical Vulnerabilities

4.1 Network Topology

Blockchain networks generally rely on rudimentary P2P gossip protocols to propagate messages, such as blocks and transactions, to all of the network participants. This network topology can be exploited to isolate certain nodes on the network and control the flow of messages between nodes. This allows can allow an attacker to manipulate the view of the consensus state for targeted nodes.

A popular approach for big service providers and miners in a blockchain network is to disable incoming connections and *establish connections to trusted nodes*. While this effectively solves the issue, it does not scale to all network participants, as it would not allow new participants to join the network anymore.

A more sophisticated countermeasure involven a number of advanced algorithms when deciding which connections to allow, effectively *randomizing the connections* we establish and maintain. This can be done by implementing random address selection, random address expiry, skewing preference to older known addresses, banning suspect messages and other tweaks. In combination, they eliminate the risk of a node having all its connections monopolized by a single adversary, thus countering the attack.

4.1.1 Eclipse Attack

During an eclipse attack, the attacker monopolizes all the connections of the targeted network participant, effectively controlling the messages which are relayed to the node [16]. This allows the attacker to present an inaccurate picture of the consensus state to the victim, or to delay reception of updates to the consensus state.

Block race engineering. Having control over the victim allows the attacker to delay messages on newly found blocks on the blockchain network. By targeting multiple miners, the attacker can withhold their blocks until several are found, which then compete for the block reward. It impacts the stability of the consensus state.

Miner removal. The attacker could also use the exploit to effectively re-

move the victim's hashrate from the blockchain network. This would make it easier to execute other attacks based on hashrate distribution. It impacts the stability of the consensus state.

Confirmed double spend. The attacker can eclipse a victim for a double spend, as well as some miners. He can then have his transaction mined by the eclipsed miners, forward the block to the merchant and execute the double spend. Subsequently, he will stop withholding the real block history of the blockchain network, which contains another transaction to overwrite the previous one. It impacts the immutability of the consensus state.

4.1.2 Balance Attack

The balance attack doesn't involve isolating a node or a group of nodes from the main network; rather, it attempts to become the only relay for messages between different partitions of the blockchain network by placing the attacker's nodes at important edges of the network graph [24]. This requires in-depth knowledge about the hashrate distribution and the communication topology of the underlying peer-to-peer network.

Rather than manipulating the view of the victims, the attacker can then play different network partitions against each other by deciding which progression of the blockhcain history to propagate and which one to delay or censor. While more difficult to exploit, it is also more difficult to detect and counter.

Attacks. The possible attacks mirror the ones for the eclipse attack. We can instigate block races, remove miners from the effective hashrate of the blockchain network and execute double spends.

In addition to previous mitigation strategies, we can use the fact that a balance attack requires advanced knowledge and monitoring of the network topology. We can therefore use traditional IT security controls to make it harder to execute BGP hijacking or DNS poisoning. Additionally, we can *encrypt communication* on the network.

5 Social Vulnerabilities

5.1 Node Owners

Blockchain technology is a complex topic and many average users do not have the background to fully understand the technical aspects of blockchain networks they use. As blockchain networks rely on the network effect to increase their usefulness, lowering participation in them has a significant impact. This can be done by discouraging potential network participants in a number of ways.

5.1.1 Content Insertion

Many blockchain networks allow nodes to include arbitrary data into transactions. A malicious actor can thus insert data of any form into the consensus state, which would exist in immutable form for the lifetime of the blockchain network. Depending on the data inserted, this can be abused in a number of ways.

Illegal content insertion. Illegal content can be inserted into the consensus state of blockchain networks. As has been demonstrated, this is not merely a theoretical possibility, but has been exploited in real blockchain networks before [33]. It is a point of contention whether this can have legal implications for the participants of the network. However, it can at least be an effective tool to drive fear, doubt and uncertainty (FUD) in a community. It impacts the accessibility of the consensus state.

Data growth acceleration. By including significant amounts of data into the blockchain, the consensus state can grow significantly and can expand to a size where some network participants decide it's too expensive to store it. Resource limitations would thus put a barrier around access to the blockchain network, reducing the number of participants. It impacts the accessibility of the consensus state.

In general, there is only one way to deal with undesired content insertion on blockchain network: not storing the content in question. This can be put into practice by implementing the possibility of *pruning blockchain data*, as already described in section 7 of the Bitcoin white paper [3]. They can then choose to discard certain content after validation of the related transactions and blocks has been completed.

An alternative approach is the implementation of a *light synchronization protocol* for the network [13]. It can take advantage of the properties of merkle trees in order to selectively validate blockchain data that the participant is interested in, without storing the full consensus state locally.

5.2 Project Organization

Blockchain networks are based on consensus rules enforced in a trustless manner between network participants. However, these consensus rules still need to be encoded in the software running the nodes of the network. If you manage to disrupt the organization of the project, you can change its direction and affect the usefulness of the blockchain network.

5.2.1 Governance Structure

Each blockchain project has a governance structure. In many cases, they are run in a mostly democratic way. This makes blockchain projects susceptible to be manipulated in the same way as modern elections in politics in order to affect certain changes [25].

Change consensus rules. The community around a project is crucial for the exchange of ideas and the discussion of the roadmap. This makes blockchain networks arguable more susceptible to the manipulation of public opinion than any other project. The past has shown that even consensus rule changes are not impossible [17]. It impacts the stability of the consensus state.

Block project progress. By adopting changes which many community members disagree with, a conflict which drives many network participants away con be created at the heart of a blockchain project. This has happened previously in Bitcoin and has basically split the project in two [36]. It impacts the accessibility of the consensus state.

One initial defense against unexpected consensus rule changes and blocked progress is the clear *definition of the values* behind a blockchain project. This philosophy can serve as a guide for future decisions and can be the basis for the project roadmap.

A second important step is to create a *transparent project governance*, which serves as a way to achieve accountability by allowing all community members to audit the decisions and the related processes.

Finally, mechanisms to implement some sort of *voting on the blockchain* to gauge the opinion of community stakeholders is a valuable tool to avoid making decisions which will upset or aleniate a significant portion of the network participants.

5.2.2 Community Influence

Most blockchain projects are governed in a supposedly democratic way. This means that the community around a project is crucial for the exchange of ideas and the discussion of the roadmap. This makes blockchain networks arguable more susceptible to the manipulation of public opinion than any other project.

Personality cult. According to authority principle of persuasion defined by Dr. Robert Cialdini, people attribute higher value to the opinion of experts [2]. The principle of liking states that people prefer to follow those they like. Combining them, we could create leader personas shaping community opinion in a way favourable to the attacker. No criterion is directly impacted.

Social media manipulation. Two other principles exposed by Cialdini are those of commitment and social proof. Censorship and astroturfing can be used to create a wrong impression of the public opinion [6]. This can lead to pressure by the public to implement certain changes in the project. No criterion is directly impacted.

There is a lot of ongoing research into social media manipulation techniques using botnets [20]. Additionally, human beings are bad at being objective about the degree they are influenced [35]. As techniques become increasingly popular, techniques that use simple *scoring and thresholds* for punishment become less and less efficient [**?**].

On the other hand, it seems possible that some old techniques were good at identifying groups of users acting in synchronized manner [10]. Combined with a newer approach evaluating the *authenticity of messages* on social media [29], we believe a platform assisting network participants in forming an informed opinion could be built.

5.3 Out-Of-Band Incentives

Consensus is achieved on blockchain networks through the use of economic incentives. If a malicious actor tries to act against the consensus rules, he will usually incur a more or less severe economic punishment. However, the fact that such a loss can be countered by out-of-band payments makes the system vulnerable. As long as the counter-incentives are sufficient, the economic incentives of the system can be circumvented.

Attacks. In general, any form of acquiring extra hashrate allows an easier execution of attacks related to hashrate: double spends, history reversal and transaction censorship. However, they also directly impact the stability of the consensus state, even when not achieving sufficient hashrate accumulation to execute an attack.

One of the only suggested solutions for out-of-band incentives is currently to *counter-bribe* the miners in order to keep them away from the original briber and potentially get them back to mining normally [32]. This works because miners have incentive to preserve their block in the history in order to not lose their block reward.

Another suggested solution is to change to another PoW algorithm. This would allow the choice of an algorithm that is difficult to verify in smart contracts, making bribery more difficult. It would also punish miners for being bribed. This would however not be viable long term.

5.3.1 Hashrate Renting

The simplest form of acquiring extra mining power is through direct payments in return for acquiring a certain hashrate [18]. This can be done by using a cloud hashing provider, where you can rent hashrate as a service. Trust issues with this setup can be overcome by introducing a negative fee mining pool, where the operator overpays the miners to get them to join, so he gets to directly control the hashrate.

5.3.2 Contractual Bribery

A more advanced form of bribing miners can be executed by contractual agreement that is enforced on the blockchain. A simple version would simply create the desired forks with included bribery transactions; miners seeking profit would then mine this fork as it will make the competing history dominant, allowing them to receive the bribes contained in the transactions.

A more advanced version uses one blockchain network with a smart contract to reward miners for solutions, while another blockchain network is the one the attack is run against [19]. This will over-incentivize the creation of solutions that favour the attacker's intent, thus disrupting the economic equilibrium of the network and the underlying security guarantees.

Next to guaranteeing the attacker to outpace the public blockchain history with his private blockchain history if all miners behave rationally, this constellation can be profitable in an of itself, as it allows the attacker to outpace all other miners on the network, even with less than majority hashrate, thus allowing him to censor competing blocks and receive all of the rewards.

6 Case Study

6.1 Context Definition

The risks for blockchain networks and their consensus state security have been established through research in the previous section. A summary of the identified risks is available in Appendix A for easy reference. The controls related to the mitigation of various vulnerabilities, attack methods and attacks is available as a list for each vulnerability in Appendix B.

The created references will now be applied to a real blockchain project to get practical risk management results. During our analysis, the simple method of equating risk score to probability times impact will be used. We will use the scoring as specified below.

Probability	
5	Always
4	Likely
3	Possible
2	Unlikely
1	Never

Impact	
5	Critical
4	Significant
3	Moderate
2	Marginal
1	Negligible

The risk management will be done for a real blockchain network, Alvalor, which is currently under active development. Little public information is available, but the author is familiar with the technical design. This will allow the results of this paper to be put into practice and have a real use beyond the academic value.

Alvalor is a minimalist blockchain network based on a simple proof-of-work mechanism. For our purposes, the transactions can be considered as simple, conveying no additional data other than the transfer details, thus rendering them fixed in size. Once the project launches, the hashrate of the blockchain network should be comparatively low.

On the foundational layer, which is the only one relevant for our analysis, it can thus be compared to a multitude of other small blockhain projects and a lot of considerations should be applicable to a decent range of blockchain networks. The recommendations at the end of the section could therefore be translated and applied widely.

6.2 Risk Analysis

6.2.1 Identification

In its current design, Alvalor uses a simple PoW consensus algorithm, which means that all risks from A.1.1 and A.1.2 are applicable, while risks of section A.1.3 are not.

The network topology is based on a simple gossip protocol in a peer-to-peer network and is therefore subject to all of the risks listed under A.2.

No content insertion is possible, so section A.3.1 can be discarded. Section A.3.2 and A.3.3 merit consideration in our case.

This leaves us with the following list of exploits to consider:

1. Majority & minority hashrate
2. Finney attack, block discarding & selfish mining
3. Eclipse & balance attack
4. Governance hijacking & community manipulation
5. Hashrate renting & contractual bribery

6.2.2 Assessment

Rather than assessing the impact of each attack, we will assess the possible impact of each exploit that enables attacks. This is more meaningful in the context of blockchain as the mere possibility of a risk materializing already impacts the security guarantees of the blockchain network.

Majority hashrate	
Probability	2
Impact	5
Total	10

Minority hashrate	
Probability	3
Impact	3
Total	9

Finney attack	
Probability	4
Impact	2
Total	8

Block discarding	
Probability	2
Impact	1
Total	2

Selfish mining	
Probability	3
Impact	3
Total	9

Eclipse attack	
Probability	2
Impact	2
Total	4

Balance attack	
Probability	1
Impact	3
Total	3

Governance hijacking	
Probability	3
Impact	3
Total	9

Community manipulation	
Probability	2
Impact	4
Total	8

Hashrate renting	
Probability	3
Impact	2
Total	6

Contractual bribery	
Probability	2
Impact	3
Total	6

6.2.3 Triage

In our risk treatment, we want to focus on high and medium exploits for mitigation, while choosing risk acceptance for the risks at the low end of the scale. We will use the following categorization for the exploits.

- Low: 1-4
- Medium: 5-7
- High: 8+

The following high risk exploits should be addressed with controls: majority hashrate, minority hashrate, selfish mining, governance hijacking, finney attack and community manipulation.

The following medium risk exploits should be address with controls: hashrate renting and contractual bribery.

The following low risk exploits will be accepted: eclipse, balance attack and block discarding.

6.3 Security Conclusions

6.3.1 Observations

Our first observation is that the fundamental stability of the proof-of-work consensus algorithm is one of the main concerns in the light of more recent discoveries in terms of vulnerabilities of the theoretical model.

The second observation is that social vulnerabilities, which are not part of most blockchain's security considerations, rank on the same level as these fundamental issues.

The third observation is that out-of-band incentives are a non-negligible risk to the stability of blockchain security guarantees and should be a part of the threat model surrounding blockchain networks.

6.3.2 Recommendations

In order to mitigate vulnerabilities around hashrate distribution and block propagation, we recommend the implementation of the following controls: B1.3 Confirmation wait period, B.1.5 Memory pool monitoring, B.1.7 Blockchain checkpoints and B.2.3 Extended block relaying.

While not directly mitigating all risks, it puts the power into the hand of the network participants, which will be able to design the best strategies for their use case while making the necessary tradeoffs for their situation. A good default implementation of such strategies should be provided.

In order to mitigate vulnerabilities around the project organization, we recommend the implementation of the following controls: B.6.1 Definition of values, B.6.2 Transparent governance and B.6.3 Onchain voting.

While novel approaches such as the evaluation of authenticity of social media messages seem promising, more work in the area is needed before an implementation is possible without spending tremendous resources. Using a human-centered approach seems like the most pragmatic way at this point in time.

We do not recommend currently recommend the implementation of any controls regarding the out-of-bound incentive vulnerabilities. However, we strongly recommend defining them as a relevant concern in the threat model

of the project.

Indeed, none of the current approaches seems to be satisfactory for the time being and additional work in the area is required. However, the progress should be continuously monitored and it is recommend to keep an eye on the surfacing of such practices in the wild.

7 Conclusion

This paper took a look at risk management for blockchain networks from the perspective of protecting the security of the consensus state and thus ensuring the security guarantees that make blockchain networks uniquely useful in the modern information technology world.

One interesting observation was that the author found vaste amounts of research literature on the theoretical soundness of the security model, a growing body of academic literature on the technical aspects of the P2P network, but almost no consideration for the social aspects of blockchain security.

Indeed, no blockchain project has considered attacks on the consensus model through out-of-band incentives as part of their threat model. Nor has any community made an effort to create a platform for the exchange of ideas that is resistent censorship and manipulation.

In conclusion, it can be said that, as blockchain networks will grow in significance and the advantages to be gained from exploiting them will increase, significant challenges remain on the horizon and will have to be tackled in order to guarantee the continued usefulness of this disruptive new technology.

References

[1] International Standards Organization, "Information technology – Security techniques – Code of practice for information security controls", ISO/IEC 27002, June 2005.

[2] Robert B. Cialdini, "The 6 Principles of Persuasion", *Influence: The Psychology of Persuasion*, HarperBusiness, 2006.

[3] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", `https://bitcoin.org/bitcoin.pdf`, October 2008.

[4] Carlos Laorden, Borja Sanz, Gonzalo Alvarez and Pablo García Bringas, "A Threat Model Approach to Threats and Vulnerabilities in On-line Social Networks", `http://paginaspersonales.deusto.es/claorden/publications/2010/sanz_RECSI10_A%20Threat%20Model%20Approach%20to%20Attacks%20and%20Countermeasures%20in%20OSN.pdf`, January 2010.

[5] Éric Dubois, Patrick Heymans, Nicolas Mayer and Raimundas Matulevičius, "A Systematic Approach to Define the Domain of Information System Security Risk Management", `https://pdfs.semanticscholar.org/ddb5/ff3f13160733b1ec11b34683e0264a09067e.pdf`, May 2010.

[6] Jacob Ratkiewicz, Michael D. Conover, Mark Meiss, Bruno Gonçalves, Alessandro Flammini and Filippo Menczer, "Detecting and Tracking Political Abuse in Social Media", `https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/download/2850/3274`, July 2011.

[7] Michael D. Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Goncalves, Filippo Menczer and Alessandro Flammini, "Political Polarization on Twitter", `https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2847/3275`, July 2011.

[8] Gavin Andresen, "Neutralizing a 51% attack", `http://gavintech.blogspot.com/2012/05/neutralizing-51-attack.html`, May 2012.

[9] Meni Rosenfeld, "Analysis of hashrate-based double-spending", `https://arxiv.org/pdf/1402.2009.pdf`, December 2012.

[10] Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow and Christos Faloutsos, "CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks", `http://alexbeutel.com/papers/www2013_copycatch.pdf`, May 2013.

[11] Ittay Eyal and Gün Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable", https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf, November 2013.

[12] Lear Bahack, "Theoretical Bitcoin Attacks with less than Half of the Computational Power", http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.473.2485&rep=rep1&type=pdf, December 2013.

[13] Vitalik Buterin, "Light client protocol", https://github.com/ethereum/wiki/wiki/Light-client-protocol, July 2014.

[14] Martijn Bastiaan, "Preventing the 51%-Attack: a Stochastic Analysis of Two Phase Proof of Work in Bitcoin", http://referaat.cs.utwente.nl/conference/22/paper/7473/preventingthe-51-attack%20-a-stochastic-analysis-of-two-phase-proof-of-work-in-bitcoin.pdf, January 2015.

[15] Ayelet Saphirshtein, Yonatan Sompolinsky and Aviv Zohar, "Optimal Selfish Mining Strategies in Bitcoin", https://fc16.ifca.ai/preproceedings/30_Sapirshtein.pdf, July 2015.

[16] Ethan Heilman, Alison Kendler, Aviv Zohar and Sharon Golberg, "Eclipse Attacks on Bitcoin's Peer-to-Peer Network", https://eprint.iacr.org/2015/263.pdf, August 2015.

[17] Mike Hearn, "On consensus and forks", https://medium.com/@octskyward/on-consensus-and-forks-c6a050c792e7, August 2015.

[18] Joseph Bonneau, "Why Buy When You Can Rent? Bribery Attacks on Bitcoin-Style Consensus", https://www.springer.com/cda/content/document/cda_downloaddocument/9783662533567-c2.pdf?SGWID=0-0-45-1587311-p180215767, February 2016.

[19] Jason Teutsch, Sanjay Jain and Prateek Saxena, "When cryptocurrencies mine their own business", https://people.cs.uchicago.edu/~teutsch/papers/repurposing_miners.pdf, February 2016.

[20] Norah Abokhodair, Daisy Yoo and David W. McDonald, "Dissecting a Social Botnet: Growth, Content and Influence in Twitter", http://www.pensivepuffin.com/dwmcphd/papers/Abokhodair.SocialBotnet.CSCW2015.pdf, April 2016.

[21] Siamak Solat and Maria Potop-Butucaru, "ZeroBlock: Timestamp-Free Prevention of Block-Withholding Attack in Bitcoin", https://arxiv.org/pdf/1605.02435.pdf, May 2016.

[22] Jinxue Zhang, Rui Zhang, Yanchao Zhang and Guanhua Yan, "The Rise of Social Botnets: Attacks and Countermeasures", https://arxiv.org/pdf/1603.02714.pdf, March 2016.

[23] George Bissias, Brian Levine, A. Pinar Ozisik and Gavin Andresen, "An Analysis of Attacks on Blockchain Consensus", https://arxiv.org/pdf/1610.07985.pdf, October 2016.

[24] Christopher Natoli and Vincent Gramoli, "The Balance Attack Against Proof-Of-Work Blockchains: The R3 Testbed as an Example", https://arxiv.org/pdf/1612.09426.pdf, December 2016.

[25] Hunt Allcott and Matthew Gentzkow, "Social Media and Fake News in the 2016 Election", https://web.stanford.edu/~gentzkow/research/fakenews.pdf, January 2017.

[26] Reu Zhang and Bart Preneel, "Publish or Perish: A Backward-Compatible Defense against Selfish Mining in Bitcoin", https://www.esat.kuleuven.be/cosic/publications/article-2746.pdf, April 2017.

[27] Vitalik Buterin, "Introduction to Cryptoeconomics", https://vitalik.ca/files/intro_cryptoeconomics.pdf, April 2017.

[28] Mauro Conti, Sandeep Kumar E, Chhagan Lal and Sushmita Ruj, "A Survey on Security and Privacy Issues of Bitcoin", https://arxiv.org/pdf/1706.00916.pdf, June 2017.

[29] Aviad Elyashar, Jorge Bendahan and Rami Puzis, "Has the Online Discussion Been Manipulated? Quantifying Online Discussion Authenticity within Online Social Media", https://arxiv.org/pdf/1708.02763.pdf, August 2017.

[30] Wenting Li, Sébastien Andreina, Jens-Matthias Bohli and Ghassan Karame, "Securing Proof-of-Stake Blockchain Protocols", https://pdfs.semanticscholar.org/ebfb/57843cdf23ce6fe7007c0f1ea233eca4b71e.pdf, September 2017.

[31] Vitalik Buterin and Virgil Griffith, "Casper the Friendly Finality Gadget", https://arxiv.org/pdf/1710.09437.pdf, October 2017.

[32] Patrick McCorry, Alexander Hicks and Sarah Meiklejohn, "Smart Contracts for Bribing Miners", http://homepages.cs.ncl.ac.uk/patrick.mccorry/minerbribery.pdf, January 2018.

[33] Roman Matzutt, Jens Hiller, Martin Henze, Jan Henrik Ziegeldorf, Dirk Müllmann, Oliver Hohlfeld and Klaus Wehrle, "A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin", https://fc18.ifca.ai/preproceedings/6.pdf, March 2018.

[34] Peter Gaži, Aggelos Kiayias and Alexander Russell, "Stake-Bleeding Attacks on Proof-of-Stake Blockchains", https://eprint.iacr.org/2018/248.pdf, March 2018.

[35] Nabeel Gillani, Ann Yuan, Martin Saveski, Soroush Vosoughi and Deb Roy, "Me, My Echo Chamber, and I: Introspection on Social Media Polarization", https://arxiv.org/pdf/1803.01731.pdf, April 2018.

[36] Alyssa Hertig, "Bitcoin Cash Fork Leaves Users Behind, But Does It Matter?", https://www.coindesk.com/bitcoin-cash-fork-leaves-users-behind-matter/, May 2018.

Acronyms

DAG directed acyclical graph. 7

FUD fear, doubt and uncertainty. 19

P2P peer-to-peer. 8, 11, 17, 24, 28

PoS proof-of-stake. 7, 15, 16

PoW proof-of-work. 7, 10–12, 16, 22, 24

Glossary

block A block represents a bundle of transactions which represent a non-conflicting valid state transaction for the current consensus state on a blockchain network. It is secured with a hash that contains a certain number of zero bits at the beginning in order to make it difficult to find a valid solution.. 14–17, 19

block reward The block reward is an economic incentive given to the generator of a valid block in exchange for the hashrate he invested to discover a block with the necessary amount of zero bits at the beginning of the validation hash.. 10, 12–15, 22

blockchain network A blockchain network is a peer-to-peer network where the participants run a consensus algorithm with a number of consensus rules in order to find agreement on the state of the data in the system.. 2, 6–11, 15–25, 27, 28

consensus state The consensus state of a blockchain network describes the part of the state which all participants have agreed on in their common blockchain history and which is therefore shared amongst all participants on the network.. 6–15, 17–20, 22, 23, 28

double spend A double spend is the action of spending the same funds on a blockchain network more than once. Fundamentally, this is the problem that consensus algorithms try to solve by deciding between conflicting transactions. A successful double spend allows a malicious actor to trick a recipient of a transaction into delivering value for funds that will later become void. 9, 13, 18

hashrate The hashrate is the amount of hashing operations a miner can execute on a blockchain network in one second. The higher the hashrate, the higher the chance for the miner to discover a valid hash for the next block generation, thus increasing the chance of gaining the block reward.. 10–13, 16, 18, 22–24

miner A miner is a block generator in a proof-of-work blockchain network. A miner will execute a hash algorithm repeatedly to find a valid solution for the block header puzzle, thus the comparison to "mining".. 14, 17, 18

mining pool When mining alone, miners are subject to significant variance between block discoveries. Pooling hashrate allows them to diminish variance and receive income at more predictable intervals. Some protocols were specifically designed to make communal mining possible with auditable reward distributions.. 10, 14, 22

network effect The network effect explains the phenomenon of a network exponentially gaining in usefulness with a linearly growing number of participants.. 19

node A node is a participant in a blockchain network, relaying messages such as transactions and blocks in a peer-to-peer fashion.. 17, 19

transaction A transaction in the context of a blockchain network is a transfer of value from one account to another account in the consensus state. It can optionally include data and manipulate data stored in the consensus state. It is always applied atomically as part of a state transition.. 9–13, 17, 19, 23, 24

validator A validator is a block generator in a proof-of-stake blockchain network. A validator will stake some of his funds and in return has a proportional probability to be selected to generate the next block for the blockchain network.. 15, 16

A Risks

A.1 Theoretical Risks		
A.1.1 Hashrate Distribution		
Majority hashrate	*Certain double spend*	Immutability
Majority hashrate	*History reversal*	Immutability, stability
Majority hashrate	*Transaction censorship*	Accessibility
Minority hashrate	*Probabilistic double spend*	Immutability
Minority hashrate	*Transaction throttling*	Accessibility
A.1.2 Block Propagation		
Finney attack	*Unconfirmed double spend*	Immutability
Block discarding	*Unfair economic loss*	Stability
Selfish mining	*Unfair economic gain*	Stability
A.1.3 Cheap Validation		
Nothing-at-stake	*Follow multiple histories*	Stability
Long-range attack	*Certain double spend*	Immutability
Long-range attack	*History reversal*	Immutability, stability
Long-range attack	*Transaction censorship*	Accessibility
A.2 Technical Risks		
A.2.1 Network Topology		
Eclipse attack	*Block race engineering*	Stability
Eclipse attack	*Miner removal*	Stability
Eclipse attack	*Confirmed double spend*	Immutability
Balance attack	*Block race engineering*	Stability
Balance attack	*Miner removal*	Stability
Balance attack	*Confirmed double spend*	Immutability
A.3 Social Risks		
A.3.1 Node Owners		
Content insertion	*Illegal content insertion*	Accessibility
Content insertion	*Data growth acceleration*	Accessibility
A.3.2 Project Organization		
Governance hijacking	*Change consensus rules*	Stability
Governance hijacking	*Block project progress*	Accessibility
Community manipulation	*Personality cult*	–
Community manipulation	*Social media manipulation*	–
A.3.3 Out-Of-Band Incentives		
Hashrate renting	–	Stability
Contractual Bribery	–	Stability

B Controls

B.1 Hashrate Distribution
B.1.1 Automatic hashrate balancing
B.1.2 Two-phase proof-of-work
B.1.3 Confirmation wait period
B.1.4 History transaction weighting
B.1.5 Memory pool monitoring
B.1.6 Anonymous transactions
B.1.7 Blockchain checkpoints
B.2 Block Propagation
B.2.1 Unpublished block punishment
B.2.2 Pool block rewards
B.2.3 Extended block relaying
B.2.4 Mining parent randomization
B.3 Cheap validation
B.3.1 Validator blacklist
B.3.2 Slashing conditions
B.3.3 Moving checkpoints
B.3.4 Out-of-band verification
B.3.5 Key-evolving signatures
B.4 Network Topology
B.4.1 Trusted connections
B.4.2 Randomized connections
B.4.3 Encrypted communication
B.5 Node Owners
B.5.1 Blockchain pruning
B.5.2 Light synchronization
B.6 Project Organization
B.6.1 Definition of values
B.6.2 Transparent governance
B.6.3 Onchain voting
B.6.4 Behaviour scoring
B.6.5 Authenticity evaluation
B.7 Out-Of-Band Incentives
B.7.1 Counter-bribery
B.7.2 Proof-of-work change