RepuCoin: Your Reputation is Your Power

SEMINAR REPORT

Submitted By

HARIS WILSON
(PKD16CS027)

for the award of the degree

of

Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT ENGINEERING COLLEGE PALAKKAD

NOVEMBER 2019

CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT ENGINEERING COLLEGE PALAKKAD

SEMINAR REPORT

This is to certify that the report entitled **Soil Classification Using Machine Learning Methods and Crop Suggestion Based on Soil Series** *submitted by* **Rani Elizabeth Joy** *to the Department of Computer Science and Engineering, Government Engineering College Palakkad, in fulfillment of the requirement for the award of B-Tech Degree in Computer Science and Engineering, is a bonafied record of the seminar carried out by her.*

Place: Sreekrishnapuram

Date:

Prof. Binu R		
Seminar Guide	**Seminar Co-ordinator**	**Head of the Department**

ACKNOWLEDGEMENT

First of all I thank the Almighty, for giving me strength ,courage and knowledge to do this seminar successfully.

I express my heartfelt gratitude to the Principal Dr. P C Reghu Raj, Govt. Engineering College, Sreekrishnapuram, for his inspiration throughout the seminar.

I would also like to thank Dr. Reena Murali , Head Of the Department, Computer Science And Engineering, Govt. Engineering College, Palakkad for providing permission and availing all required facilities for undertaking the seminar in a systematic way.

I would like to appreciate the guidance given by Prof. Shibily Joseph ,Prof. Sony P ,Prof. Panchami U V and Prof. Joby N J, our Seminar Coordinators, Department of Computer Science And Engineering, Govt. Engineering College, Palakkad, for providing sincere guidance.

I am extremely grateful to Prof. Binu R, my Internal Guide, Govt. Engineering College, Palakkad, who guided me with his kind, ordinal, valuable suggestions.

I would also like to express my sincere gratitude to all teaching and non- teaching staffs of Department of Computer Science And Engineering, Govt. Engineering College, Palakkad, for the sincere directions imparted and the cooperation in connection with the seminar.

I am also thankful to my parents for the support given in connection with the seminar. Gratitude may be extended to all well-wishers and my friends who supported me for the seminar.

ABSTRACT

Existing proof-of-work cryptocurrencies cannot tolerate attackers controlling more than 50% of the network's computing power at any time, but assume that such a condition happening is "unlikely". However, recent attack sophistication, e.g., where attackers can rent mining capacity to obtain a majority of computing power temporarily, render this assumption unrealistic. This paper proposes RepuCoin, the first system to provide guarantees even when more than 50% of the system's computing power is temporarily dominated by an attacker. RepuCoin physically limits the rate of voting power growth of the entire system. In particular, RepuCoin defines a miner's power by its 'reputation', as a function of its work integrated over the time of the entire blockchain, rather than through instantaneous computing power, which can be obtained relatively quickly and/or temporarily. As an example, after a single year of operation, RepuCoin can tolerate attacks compromising 51% of the network's computing resources, even if such power stays maliciously seized for almost a whole year. Moreover, RepuCoin provides better resilience to known attacks, compared to existing proof-of-work systems, while achieving a high throughput of 10000 transactions per second (TPS).

Contents

List of Figures

Chapter 1

Introduction

BITCOIN is the most successful decentralized cryptocurrency to date. However, despite its enormous commercial success, many weaknesses have been associated with Bitcoin, including weak consistency, low transaction throughput and vulnerabilities to attacks, such as double spending attacks, eclipse attacks , selfish-mining attacks, and flash attacks.

Several promising existing solutions targeted the low throughput problem of Bitcoin. Nevertheless, these solutions either provide only probabilistic guarantees about transactions (weak consistency) or can provide strong consistency but suffer from liveness problems even when an attacker has a relatively small computing power. Moreover, the resilience of such solutions against attacks, such as selfish mining attacks where an attacker has more than 25% of the computing power, remains unsatisfactory. In addition, all existing contemporary proof-of-work (PoW) based variants of Bitcoin. rely on the assumption that an attacker cannot have more than 33% or 50% of computing power at any time. However, with the sophistication of attacks mounted on Bitcoin, e.g., flash attacks (a.k.a. bribery attacks), where an attacker can obtain a temporary majority (¿50%) of computing power by renting mining capacity, all these systems would fail. In brief, existing solutions that address the weaknesses associated with Bitcoin still suffer from significant shortcomings.

This paper addresses these shortcomings —liveness of current high-throughput solutions, and vulnerability to attacks such as selfish mining and flash attacks. In particular, RepuCoin, the first system that can prevent attacks against an attacker who may possess more than 50% computing power of the entire network temporarily (e.g., a few weeks or even months). Our proof-of-concept implementation shows that while providing better security guarantees than predecessor protocols, RepuCoin also ensures a very high throughput (10000 transactions per second). In practice, Visa confirms a transaction within seconds, and processes 1.7k TPS on average. This shows that RepuCoin satisfies the required throughput of real world applications.

First, under proof-of-reputation a miner's decision power (i.e., the voting power for reaching consensus in the system) is given by its reputation. A miner's reputation is not measured by what which is called the miner's 'instantaneous' power, i.e., the miner's computing power in a short time range, as in classic PoW. Instead, the reputation is computed based on both the total amount of valid work a miner has contributed to the system and the regularity of that work, over the entire period of time during which the system has been active. It can be call this the miner's 'integrated power'. So, when an attacker joins the system at time t, even if it has a very strong mining ability that is, high computational (i.e., instantaneous) power, it would have no integrated power at time t, or even shortly after, as it did not contribute to the system before t.

Second, when a miner deviates from the system specifications, RepuCoin lowers the miner's reputation, and hence its integrated power, in consequence of this negative contribution. This prevents a powerful malicious miner from attacking the system repeatedly without significant consequences. In contrast, classic PoW systems either do not support any feature for punishing miners that do not abide by system specifications, or they punish these miners by merely revoking their rewards — this does not prevent them from attacking the system again immediately after.

RepuCoin provides deterministic guarantees on transactions by employing a reputation-based weighted voting consensus. Consensus is carried by a group formed of the top reputable miners. Every member of that group has a weight associated to its vote. The weight of a member's vote is the percentage of that member's reputation w.r.t. the entire group's reputation. Such weights ensure that one's voting power depends, not on the sheer instantaneous (computing) power — which is the enabler of flash attacks— but on the integrated power, which both takes time to build, and is built on the miner's honesty and historical performance. In fact, quantifying a miner's voting power based on its performance in the entire blockchain highlights the selfstabilizing characteristics of our approach: qualitatively, to acquire power in RepuCoin a miner is urged to exhibit normal, honest behavior; quantitatively, the speed with which a miner can gain power is dictated by the regularity and amount of that miner's contributions in the entire blockchain.

To illustrate the robustness of the design choices underlying RepuCoin, to present the analysis of the security provided by the mechanisms used in RepuCoin. In particular, It is shown that achieving the safety and liveness correctness conditions of the reputation-based weighted voting consensus protocol is physically guaranteed by the growth rate of the proof-of-reputation function, and the rate of decision power growth of the entire system is bounded. In addition, to present experiments exemplifying concrete values for the decision power growth in several situations, showing that the network achieves very high stochastic robustness against attacks on its liveness or safety. For instance, it can be demonstrated that after a single year of operation, RepuCoin is resilient to all attacks that compromise 26%, 33%, and 51% of the network's computing resources, even if such power stays maliciously confiscated for almost 100 years, 2 years, and 1 year respectively. Also, in the same setting, even if an attacker can afford to seize a huge computational power for a specific period, due to the cost of such attacks (e.g., 90for up to 3 months), it will not break RepuCoin. Moreover, to provide an analysis of the non-rationality of infiltration attacks, with a comparison of the cost of attacking different systems. .

Chapter 2

Literature survey

The blockchain technology is a relatively new approach in the field of information technologies. As one of its first implementations, bitcoin as a cryptocurrency has gained a lot of attention. Together with Ethereum, blockchain implementation with focus on smart contracts, they represent the very core of modern cryptocurrency development.In this paper brings a small introduction to the matter of blockchain and cryptocurrencies. It begin with a quick retrospective of some of the most famous solutions for decentralized digital money before Bitcoin, and then go into the very core of its function, together with Ethereum. These two cryptocurrencies hold majority of the cryptocurrency market capitalization. Of course, as it happens with new technologies, some limitations and problems emerged, and described them as well[1].

Consensus is the key component and backbone of Blockchains, which use two main types of consensus mechanisms, namely proof-of-X based consensus, and Byzantinefault tolerant (BFT) consensus[8]. The former is generally permissionless, where anyone can join and leave the potentially large consensus group; and the latter is permissioned, where the set of participants running consensus is small and predefined. Proof-of-X based consensus has gained much interest since the use of Proof-of-work in Bitcoin, already described in the introduction. Proof-of-stake, which was first discussed in a Bitcoin community forum [4], has been proposed to use virtual voting to provide quicker transaction confirmation. Proof-of-space (a.k.a. proof-of-capacity) has been proposed to use physical storage resources to replace computing power in the proof-of-work mechanism. Proof-of-coin-age shares a similar concept as proofof-stake, as participants also perform virtual "mining" by demonstrating possession of a quantity of currency. Proofof-activity puts every coin owner into a mining lottery; periodically, winners are randomly determined by transactions. A winner is expected to respond with a signed message within a small time interval to claim its award. Proofof-elapsed time, proposed by Intel and implemented as a Hyperledger project, uses Intel SGX enabled CPUs to do virtual voting through a random sleeping time to replace the proof-of-work mechanism. Proof-of-membership system in ByzCoin creates a consensus group formed by recent PoW-based block creators, and this group runs a BFT protocol for reaching consensus[2]. However, it is shown that ByzCoin still suffers from selfish mining attack, and that it can permanently lose liveness during reconfiguration, if too many miners disappear in a short time or can repeatedly lose liveness temporarily. Unfortunately, none of the permissionless consensus protocols described is able to provide a meaningful security guarantee when an attacker is able to control a majority of mining power[7].

Chapter 3

System and threat model

System Context

RepuCoin makes use of two types of blocks, namely keyblock and microblock. Keyblocks are created through PoW, as a means to elect a leader, and they do not contain any transaction. Microblocks are proposed by a randomly selected leader to record transactions into the blockchain. All participants of the system, no matter whether they are miners or mobile clients, learn about new transactions and blocks in the same way as in BitCoin and other blockchains, through a peer-to-peer protocol.

In RepuCoin, we verify and commit microblocks, as well as decide on the keyblocks to be added to the blockchain, using Byzantine fault tolerance protocols (e.g., [28, 30]) with minor modifications. Such form of agreement prevents a malicious leader from double spending a coin, and resolves potential forks resulting from simultaneously mined keyblocks. According to Byzantine quorums theory, in order to reach an agreement, classic BFT protocols require votes from at least 2f + 1 nodes1 , to prevent an adversary controlling f nodes from breaking the protocol. In practice, however, an open BFT-based system cannot guarantee that an attacker will never be able to control more than f nodes. To enforce the assumption that no more than f Byzantine nodes are ever involved in a consensus, here to introduce novel mechanisms to make it infeasible, in practice, for an attacker to seize f nodes within the consensus group.

System Model

RepuCoin is a system composed of a non-predetermined number of nodes, called miners. Each miner has a reputation score, which determines that miner's ability of obtaining rewards. A miner's reputation score is based on the correctness of its behavior and its regularity in adding blocks to the existing chain, hence correlated with the miner's computing power. RepuCoin considers a network that is untrustworthy and unreliable. In addition, assume the network has partial synchrony. To address the above-mentioned uncertainty in the definition of the consensus quorums and outcomes for open BFT-based protocols, RepuCoin resorts to two main techniques. Firstly, rely on the notion of having a consensus group, i.e., a subset of the miners denoted by X capable of controlling the operations of RepuCoin, namely running the consensus protocol. Second, the voting rules in the underlying consensus schemes are not nominal, but based on a novel reputation-based weighted voting. To this end, refine the definition of quorums as follows: reaching agreement not only requires votes from 2f + 1 nodes, but also demands that these nodes collectively have more than 2/3 of the cumulative reputation of the consensus group.

The consensus group size is defined as the minimum number of miners with enough decision power (i.e., cumulative reputation) to ensure safe and live control of the system, given our quorum definition. Therefore, the consensus group members are obviously the miners with the highest reputation scores. This stratagem has two virtuous effects. First, RepuCoin's safety is guaranteed by consensus, which can be viewed as a deterministic control orchestrated by a set of miners with overwhelming cumulative reputation. By definition, such reputation itself gives an expectation of the correct behavior of these miners. Second, openess and fairness of RepuCoin relies on X being parametric and agnostic of identity of network members. At configuration time, the size of X is calculated by meeting a target percentage of the overall decision power, and so it can be large or small, depending on the mining pool composition, but not pre-determined. On the other hand, as miners gain or lose reputation, they can (by merit or demerit) enter and leave the consensus group.

Threat Model Consider a malicious (a.k.a. Byzantine) adversary, who can arbitrarily delay, drop, re-order, insert, or modify messages. Also consider collusions of an arbitrary number of miners, to model a malicious real organization capable of deploying a significant number of virtual miners under its direct dependence. It is assumed that security of the used cryptographic primitives, including a secure hash function and a secure signature scheme. Such adversary can potentially control as many miners as it wishes, and coordinate them in real time with no delay. In consequence, the consensus group can be infiltrated by adversaries. However, assume that the adversary has the ability to control at most f (—X—1)/3 group members whose collective reputation is less than 1/3 of the cumulative reputation of the members of consensus group X. Under this assumption, the system is safe and live.

Chapter 4
Approach

Block Mining and Reward System

As mentioned earlier, in order to support higher throughput rates, RepuCoin decouples leader election (keyblocks) from transaction serialization (microblocks). Keyblock and leader election. Miners solve Bitcoinlike puzzles to create keyblocks, and receive rewards corresponding to keyblock creations. However, the keyblock creator is not necessarily the leader that commits transactions into microblocks. Rather, the leader is randomly elected from the reputable miners. The puzzle is defined as follows: H(prev_KB_hash $\|Nonce\|PK\|$) $< target$.

RepuCoin solves forked chains on the fly by dynamically forming the consensus group and agreeing on which chain to choose. The consensus group members are the top reputed miners in the mining network. The reputation score of miners can be calculated by using data from the blockchain, and is maintained locally by each miner. When different miners have the same reputation, a naive solution would be to order them according to their public key PK (a.k.a. address). However, this gives a miner with small PK an advantage. To avoid this, in RepuCoin, miners with the same reputation score are ordered by H(PK, R), where reputation R (and therefore the hash value) is updated each time a new keyblock becomes part of the blockchain.

Each time a new keyblock is created, the creator proposes it to the consensus group. The group verifies the received keyblocks, and runs the underlying Byzantine agreement protocol to decide which keyblock to choose (if multiple conflicting keyblocks are proposed). It can be called a keyblock that is agreed upon and signed by the group a pinned keyblock. A pinned keyblock is final and canonical, it defines the unique global blockchain from the genesis block up to the pinned keyblock. All keyblocks that conflict with a pinned keyblock are considered invalid. New keyblocks are mined based on the hash value of the previous pinned keyblock.

The (new) KB_hash is the hash of a pinned keyblock, i.e., all the material in the frame above, where K_sig is a signature on the hash value of (prev_KB_hash, Nonce, PK, R), and sig_KB_agmnt is the signed agreement from the consensus protocol on committing this keyblock. The first keyblock is called the genesis block (as in Bitcoin), which is defined as part of the system. Note that to verify a keyblock, consensus group members check the validity of K_sig, the solution to the mining puzzle, and the reputation R.

Each time a new keyblock is pinned, the next leader — which verifies transactions and commits them into microblocks — is selected as follows:

li: equals xj s.t. xj belongs to X j equals to H(K sigi) mod X

where li is the i-th leader determined by the hash value H(K_sigi) of the signature K_sigi contained in the i-th pinned keyblock, and X is the set of miners constituting the consensus group. Since a cryptographically secure hash function is considered a random oracle, the leader is selected randomly in the consensus group with probability 1/mod(X) . However, one concern with this leader selection process is that consensus group members can determine the following leader before pinning a block. Thus, a consensus member interested in getting more rewards would only accept (decide to pin) a block that makes itself the new leader. To address this issue, a simple approach is to determine a leader by using H(sigi) instead of H(K_sigi), where sigi is a signature on the current length of the blockchain issued by the keyblock creator. Note that it is important to issue this signature only after the keyblock has been pinned by the consensus group. This way, each consensus member would accept a block with equal probability.

Microblock

The current leader commits transactions into microblocks. To prevent double spending, each microblock is proposed to the consensus group before being accepted, and hence committing the transactions it encompasses. The group members verify the microblock, and initiate a consensus instance to agree on that microblock, which upon agreement is called a pinned microblock.

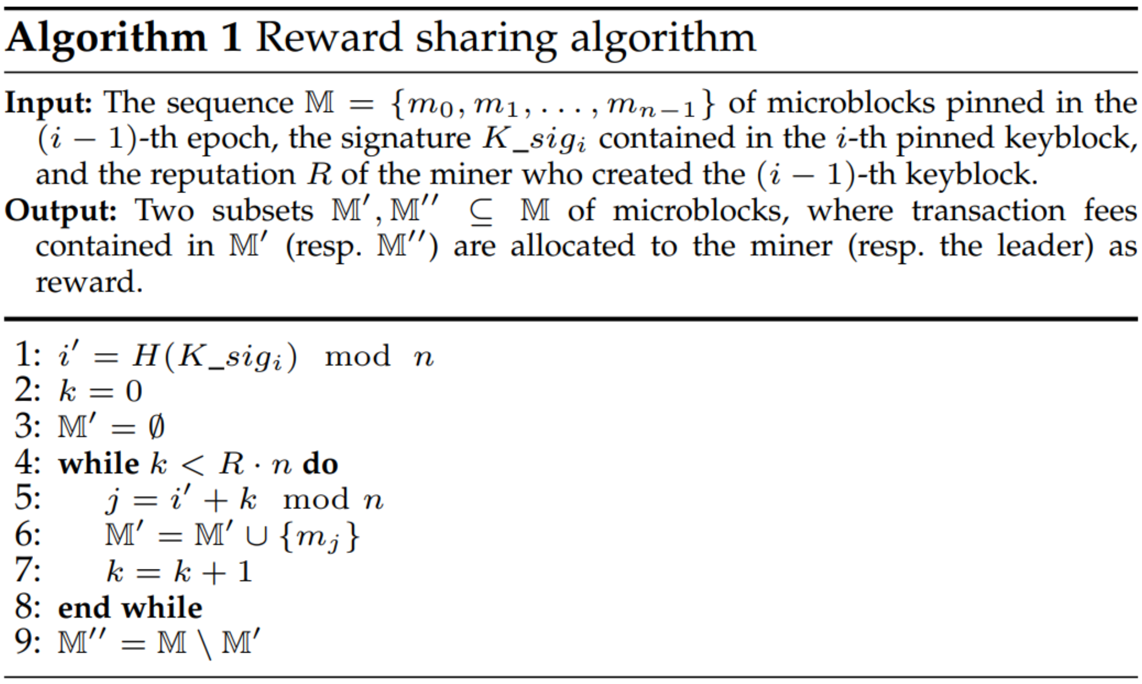
Reward system

In RepuCoin there are two types of rewards, namely mining rewards and transaction fees. Upon successfully mining a keyblock, a miner is entitled to get a reward, precisely if that miner's keyblock gets pinned. This mining reward is of a pre-set amount.

Every transaction within the microblock carries a transaction fee. The randomly elected leader shares the transaction fees with the miner of the pinned keyblock according to Algorithm 1. Roughly speaking, the miner's reputation determines the number of microblocks from which it can obtain transaction fees; the leader gets the rest. However, a leader that can determine the microblocks from which it gets transaction fees, may optimize its income by putting transactions with higher transaction fees into these microblocks.

To avoid this unfair game, RepuCoin uses the hash value of the next pinned keyblock, i.e., the keyblock pinned at the end of the new epoch, to decide which pinned microblocks are allocated to the miner and which go to the leader. Since the hash value of the next pinned keyblock cannot be predicted, RepuCoin eliminates the above situation.

More precisely, let M = {m0, m1, . . . , mn1} be the sequence of n microblocks that are pinned by the consensus group. Let R [0, 1] be the reputation score of the miner which creates the (i1)-th pinned keyblock. The transaction fees contained in the set M0 and M00 of microblocks are shared between the miner of the (i 1)-th pinned keyblock and the leader, respectively, as shown in Algorithm 1.



Algorithm 1 Reward sharing algorithm

Input: The sequence $\mathbb{M} = \{m_0, m_1, \ldots, m_{n-1}\}$ of microblocks pinned in the $(i-1)$-th epoch, the signature K_sig_i contained in the i-th pinned keyblock, and the reputation R of the miner who created the $(i-1)$-th keyblock.

Output: Two subsets $\mathbb{M}', \mathbb{M}'' \subseteq \mathbb{M}$ of microblocks, where transaction fees contained in $\mathbb{M}'$ (resp. $\mathbb{M}''$) are allocated to the miner (resp. the leader) as reward.

```
1: $i' = H(K\_sig_i) \mod n$
2: $k = 0$
3: $\mathbb{M}' = \emptyset$
4: while $k < R \cdot n$ do
5:     $j = i' + k \mod n$
6:     $\mathbb{M}' = \mathbb{M}' \cup \{m_j\}$
7:     $k = k + 1$
8: end while
9: $\mathbb{M}'' = \mathbb{M} \setminus \mathbb{M}'$
```

Figure 4.1: Algorithm 1

Block Pinning

In RepuCoin, use consensus to pin both keyblocks and microblocks. Transactions belonging to pinned microblocks cannot be unrolled at a later point in time.

Byzantine fault-tolerant consensus algorithms typically rely on processes voting. A process needs to collect a quorum of votes on a given value/action for that value/action to be considered legal by the system. The size of the quorum is selected in a way that guarantees (i) safety of decisions, e.g., avoiding conflicting decisions, and (ii) liveness of the system, i.e., miners should be able to hear eventually a number of votes on some value/action from a quorum of miners. It has been shown that in systems, as soon as 1 3 or more of the miners are compromised, an attacker can make the system inconsistent — an attacker can make different parts of the system decide differently . In order to make our system robust to such attacks, it is proposed to modify the traditional nominal voting mechanism, i.e., hearing from a sufficient number (quorum) of miners, by requiring as well to hear from a sufficient number of miners such that their added reputation is above a defined threshold. Such a modification prevents an attacker from breaking the correctness of the system directly upon compromising any 1/3 of the miners: it should compromise as well enough miners that their added reputation is at least 1/3 of the total reputation of the consensus group.

Committing microblocks

. The leader of the current epoch, issues transactions in the form of microblocks. After generating a microblock, the leader initiates a consensus instance for this microblock proposing an accept to commit that microblock. Other consensus members will propose to either accept or decline (by not accepting) this microblock, depending on the transactions contained within and of course their validity. In other words, if transactions within the micro-block are invalid then members should decline committing that microblock. The leader continues to issue microblocks that are proposed to the consensus group for validation and commitment, until a new leader is elected.

Committing keyblocks

Upon successfully mining a keyblock, the miner of that block sends that keyblock to all members of the consensus group. Upon receiving a keyblock, this group member initiates a consensus instance proposing the received keyblock (first received keyblock in the case when many such keyblocks are received). As a result, the members of the consensus group decide on a single keyblock to be part of the blockchain. The miner of that block is termed as the "winner". The hash of the new keyblock output by the consensus decides which member of the consensus group becomes the leader of the current epoch as previously mentioned.

Upon receiving a keyblock, this group member initiates a consensus instance proposing the received keyblock (first received keyblock in the case when many such keyblocks are received). As a result, the members of the consensus group decide on a single keyblock to be part of the blockchain. The miner of that block is termed as the "winner". The hash of the new keyblock output by the consensus decides which member of the consensus group becomes the leader of the current epoch as previously mentioned.

A member of the consensus group that successfully decides on the identity of the new leader stops validating microblocks relative to the previous leader. Afterwards, that member initiates a consensus instance to agree on the total set of committed microblocks. Consensus group members need to agree on the total set of microblocks, since a leader is selected from this group. A leader that does not know the total set of committed microblocks might propose microblocks that are in conflict with committed ones and accordingly lose its reputation, not out of maliciousness but simply out of lack of knowledge. To avoid this situation, every member after reaching a decision on the identity of the new leader submits to consensus the largest sequence number of microblocks that have been committed along with a verifiable proof of this claim. As such, having a sufficient number N of signatures on a decision constitute a proof of its validity. Let the number N of signatures is sufficient if N greater than or equal to 2f + 1 and if the total reputation of the miners issuing these N signatures is more than 2/3 of the total reputation of the group.

Upon reaching a decision on the new leader and on the global set of committed microblocks, each consensus group member also sends a message to notify the winner, the current leader, and the newly elected leader of this result. A consensus group member waits to either hear from consensus or from a sufficient number N0 of other group members about the identity of the new leader and the global set of committed microblocks, before it adopts that member as leader. Saying N0 is sufficient if the total reputation of the issuers of the N0 signatures is more than 1/3 of the total reputation of the group and if N0 greater than or equal to f + 1. In that case, the current leader simply stops issuing microblocks and the new leader takes over proposing microblocks.

Optimizing agreement on committed microblocks

order to have agreement on the set of committed microblocks without resorting to a consensus instance. Due to the PoW, it is known that mining a keyblock successfully takes a certain time depending on the mining difficulty (e.g. on average 10 minutes in Bitcoin). If assumed that a leader issues microblocks to be committed at a pre-specified rate, then assume that on average a leader commits m microblocks per epoch. Accordingly, all consensus group members do not validate or commit more than m microblocks for any given leader. Upon reaching a decision on the identity of a new leader (as a result of having a new keyblock mined) a consensus group member only initiates consensus on the set of committed microblocks if it has not seen m committed microblocks from the previous leader.

The benefits of fixing the number of microblocks (that a leader can commit) to m microblocks per epoch extends beyond having a fast and efficient agreement on the set of committed microblocks. It can also be used to incentivise leaders not to hinder throughput, e.g., a malicious leader in the worst case might decide not to submit any microblocks to intentionally stall the throughput. However now, since a leader is expected to commit m microblocks per epoch, leaders which cannot meet that constraint can be punished for example by decreasing their reputation and hence decreasing their chances of staying part of the consensus group and becoming leaders again.

Reputation System

Firstly highlight the shortcomings of previous systems. For example, a proof-of-work based system requires a miner to show that it has done some work in order to include its set of proposed transactions, and hence extend the chain. Thus, a miner that has a high computing power can join the system at any time and can play attacks. Similarly, in the proof-of-membership system, a miner has to show that it has created enough blocks recently to demonstrate its computing power, then it can issue microblocks and can gain power in the consensus protocol. Again, an attacker with higher computing power can join the system at any time and can break the system. However, with proof-of-reputation, in addition to creating enough recent keyblocks, a miner has to show that it has behaved honestly and created keyblocks regularly for a period of time before being able to launch any attacks on the system.

In particular, Ext belongs to [0, 1] is the (optional) external source reputation of the miner. For example, when Citybank joins RepuCoin, it may have a starting reputation that is higher than a random individual joiner. In RepuCoin, this is encoded by using Ext3 . H belongs to 0, 1 is the honesty of the miner, which is set to “1” for each new joiner, and is set to “0” if a miner has misbehaved.

The reputation function

. It is intended to define the social objectives of reputation in RepuCoin in a precise and parameterizable way. Those objectives are: (i) careful start, through an initial slow increase; (ii) potential for quick reward of mature participants, through fast increase in midlife; (iii) prevention of over-control, by slow increase near the top.

The formula defining the progression (resp. regression) of reputation, f(x) above, is a sigmoid function. It ensures that miners, at the start, can only increase their reputation slowly, even if having a strong computing power. A miner needs to stay in the system and behave honestly for a long enough period, to progressively increase its reputation up to the turning point, where it is trusted enough to be incentivized to make it grow more quickly, to more interesting levels. And finally, the curve inflects again, so that the reputation does not grow forever, but asymptotically reaches a plateau that promotes a balance of power amongst miners. The reputation function is also parameterized, to allow to mark these points precisely.It is denoted by a block chunk (or just ‘chunk’ for simplicity) a sequence of successive keyblocks in the blockchain. Blocks chunks satisfy the following: (i) all block chunks are of the same size, and (ii) any keyblock is included in exactly one block chunk.

Chapter 5

Performance evaluation

Implementation

Extend the BFT-SMaRt library for our RepuCoin implementation and deploy each member of the consensus group on a different machine, each having the following specifications: Dell FC430, Intel Xeon E5-2680 v3 @2.5GHz, 48GB RAM. To simulate wide-area network conditions, impose a round-trip network latency of 200 ms between any two machines, and a maximum communication bandwidth between any pair of machines to 35 Mbps. To better simulate the system in the real world scenario, and make use of the mining power distribution from the Bitcoin mining network5 . More details and justifications of our setting, including the choice of PoW mining rate and mining power distribution, can be found in our technological report.

Consensus goup

Consider consensus groups that initially control from about 50% to 98.1% of computing power. With the current Bitcoin mining power distribution, the corresponding consensus group sizes would range from 4 to 19. However, that security is hampered for consensus groups that control more than 90% computing power, given Bitcoin's computing power distribution, can present the performance results with consensus group controlling computing power from 44.7% to 90%.

Consensus Latency

To measure the latency of our consensus implementation , and compare it with the latency of the original BFT-SMaRt. Such a comparison illustrates the timing overhead that is incurred relative to using our reputationbased weighted voting mechanism. BFT-SMaRt requires at least 2f + 1 members to agree on a value, while our reputation-based weighted voting variant requires in addition that these members (which are at least 2f + 1) have collectively more than 2/3 of the reputation of the entire consensus group.

Throughput

For example, using 2MB microblocks, the throughput increases from slightly more than 10000 TPS with a consensus group of size 13 (controlling 90% computing power), to 22500 TPS with a consensus group of size 4 (controlling 44.7% computing power). Second, for all group sizes, one can see, as expected, that the throughput tends to increase as blocks become larger, and this is what can be observed up to 2MB. For example, when the consensus controls 90% computing power of the entire network (group size of 13), the throughput for blocks of 512KB, 1MB, and 2MB, is respectively equal to 6200, 9400, and 10000 TPS. It is observed that using larger block sizes (e.g., 4MB), decreases the throughput. However, this outlier is an artefact of the underlying protocol use, i.e., the BFTSMaRt library, whose sheer performance, as a regular BFT protocol, is seemingly affected for very large block sizes.

Chapter 6

Conclusion

RepuCoin provides proof-of-reputation as an alternative way to provide a strong deterministic consensus, and be robust against attacks, in a permission-less distributed blockchain system. All BFT-based blockchain systems are bound to the coverage of the assumption on the maximum number of faulty players, f, or their decision power quota thereof. RepuCoin, although belonging to that generation of systems, is the first to deploy effective mitigation measures that reduce brittleness in the face of overwhelming adversary power, where other systems give in. Namely, it provides security guarantees against an attacker who can control a majority of the overall computing power for a duration that increases with the joining time of the attacker. Based on the strong deterministic guarantee derived from reputation-based weighted voting, the robustness of RepuCoin grows with legitimate operation time: the later the attacker joins, the more secure the system is. For example, an attacker that joins the system after it has been operating for a year, would need at least 51computing power and would need to behave correctly in the system for 10 months before being able to successfully make RepuCoin lose liveness. Breaking RepuCoin's safety is at least as difficult as breaking its liveness. Further discussions on secure bootstrapping, formal security analysis, and applying our PoR to other virtual mining systems (e.g. proof of stake) can be found in our technological report.

BIBLIOGRAPHY

[1] IEEE ”Bitcoin: A Peer-to-Peer Electronic Cash System” by Satoshi Nakamoto

[2] ” “Double-spending Fast Payments in Bitcoin” by Shuai Wang , Liwei Ouyang, Yong Yuan , Senior Member, IEEE, Xiaochun Ni, Xuan Han, and Fei-Yue Wang , Fellow, IEEE

[3] IEEE ”Blockchains and Smart Contracts for the Internet of Things” by Ghassan O. Karame, Elli Androulaki, and Srdjan Capkun.

[4] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever.”Hijacking Bitcoin: Routing Attacks on Cryptocurrencies”

[5] Arthur Gervais.”Tampering with the Delivery of Blocks and Transactions in Bitcoin”.

[6] Ethan Heilman, ”Eclipse Attacks on Bitcoin’s Peer-to-Peer Network”.

[7] Ittay Eyal and Emin Gun Sirer ” “Majority Is Not Enough: Bitcoin Mining Is Vulnerable”.

[8] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar, ”Optimal selfish mining strategies in Bitcoin”

[9] Joseph Bonneau. ”Why Buy When You Can Rent? - Bribery Attacks on Bitcoin-Style Consensus”.

[10] Lightning Network. 2017. URL: https : / / lightning . network/.

[11] Joseph Poon and Thaddeus Dryja. ”The bitcoin lightning network: Scalable off-chain instant payments”

[12] Kyle Croman. ”On scaling decentralized blockchains”

[13] Yonatan Sompolinsky and Aviv Zohar. ” “Secure highrate transaction processing in bitcoin”