Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Bitcoin Security under Temporary Dishonest Majority

Bachelor Thesis

Lukas Käppeli

lukask@student.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Georgia Avarikioti, Yuyi Wang
Prof. Dr. Roger Wattenhofer

September 24, 2018

Acknowledgements

I thank my supervisors Zeta Avarikioti and Yuyi Wang for their great support. This thesis was submitted to Financial Cryptography and Data Security 2019 as a paper under the same title. The paper is a joint work with my supervisors and the thesis has major overlap with it. At this point, I would like to thank Prof. Dr. Roger Wattenhofer for giving us the opportunity to submit the thesis for publication.

Abstract

Standard approaches for analysing Byzantine Agreement and Consensus Protocols typically just consider two types of players, honest and corrupted ones. These approaches are also used to provide security guarantees for Bitcoin, by assuming an upper bound on the fraction of corrupted players. This assumption doesn't reflect the reality, since it implies that every honest player is online during the whole protocol execution.

We provide a way to relax this assumption. By dividing the honest parties into alert (online) and sleepy (offline) parties, we can prove the security of Bitcoin under the relaxed assumption that we only require the expected number of alert parties to be larger than the corrupted players (by some factor). This setting allows even temporary dishonest majorities during the protocol execution. We will prove the security of the Bitcoin protocol by proving the three fundamental blockchain properties Chain-Growth, Common-Prefix and Chain-Quality. In the first part of this thesis, we are going to prove the three properties for a synchronous model. We then extend these results to the bounded-delay model and to a synchronous model, where we allow message losses. We provide remarkable results for the security of Bitcoin in all three models, especially in the one, where we allow message losses.

Keywords: Bitcoin, Security, Dishonest Majority, Offline Players, Sleepy Model

Contents

Chapter 1

Introduction

Bitcoin is the most popular cryptocurrency today. Since the introduction in 2008 [1], it did not just get the attention from the public, but also became a new research topic. By maintaining a public transaction ledger, called blockchain, Bitcoin provides a decentralized payment system. Bitcoin owners can execute transactions, which are broadcasted to all protocol participants via the Bitcoin peer-to-peer network. The protocol participants, known as miners, try to solve a Proof-of-Work in order to extend the blockchain with new blocks, containing a certain amount of transactions. For creating a block, a miner is rewarded with Bitcoins. These block rewards consist of transaction fees and new coins, which are created in this way.

Bitcoins permissionless setting allows anyone participating the consensus protocol (as a miner) without the need to authenticate himself. We refer to these anonymous protocol participants as players. Most of the existing approaches proving the security of protocols like Bitcoin by assuming fixed number of n players among at most t are corrupted. But this approach does not work well for analyzing protocols in a permissionless and thus highly dynamic setting. This follows from the fact that players can participate as they like and are not restricted to stay online for a certain time. A study on the Bitcoin network from [2] showed that from a set of players, online at some time, an expected fraction of 20% will be offline after six hours. This is consistent with the study from [3], observing only around 15% of the players still online after ten days. The model of [4] captures this impact of offline players, thus is able to reflect the effect of sporadic participation. We can show that Bitcoin (respectively the Backbone protocol from [5]) is provably secure under such assumptions and going even further by allowing temporary dishonest majorities, e.g. more corrupted than online players. Of course, this should only happen with small probability, but we will show that it is sufficient to have honest majority on expectation. Further, we introduce a parameter c, which upper bounds the number of blocks, contributed by the adversary. This is not needed to prove the security, but as [6] showed, if the adversary follows a selfish mining strategy, he can gain a much higher reward fraction compared to its fraction of the mining power. By choosing an appropriate value for c, it is possible to upper bound this advantage.

Another problem of Bitcoin is that network delays significantly affect the performance and security. By extending our synchronous model to the a semi-synchronous model, we are going to show that the upper bound on offline parties heavily depends on the maximum allowed message delay.

In the last section, we extend our analysis to a synchronous model, where we allow message losses. This follows the idea described in [7], where the adversary may perform an eclipse attack [8, 9] on some victim parties which enables him to control the parties view of the blockchain. We are going to show security under the assumption, that the adversary has the power to perform eclipse attacks to a certain number of parties, depending on the number of corrupted players.

Chapter 2

The model

We adapt the model of Garay et al. [5], since we intent to prove the security of the Backbone protocol, originally introduced in this work. We initially present all the components of a general model and then we parametrize the model to capture the three different models under which we later prove that the backbone protocol is secure.

2.1 The execution

We assume a fixed set of n parties, executing the Bitcoin backbone protocol. Each party can either be *corrupted*, *sleepy* or *alert*; *sleepy* is an offline honest node and *alert* an honest node that is actively participating in the protocol.

Involved programs

All programs are modeled as polynomially-bounded interactive Turing machines (ITM) that have communication, input and output tapes. An ITM instance (ITI) is an instance of an ITM running a certain program or protocol. Let the ITM $\mathcal{Z}$ denote the environment program that leads the execution of the Backbone protocol. Therefore $\mathcal{Z}$ can spawn multiple ITI's running the protocol. These instances are a fixed set of n parties, denoted by $P_1, \ldots, P_n$. The control program C, which is also an ITM, controls the spawning of these new ITI's and the communication between them. Further, C forces the environment $\mathcal{Z}$ to initially spawn an adversary $\mathcal{A}$. The environment will then activate each party in a round-robin way, starting with P_1. This is done by writing to their input tape. Each time, a corrupted party gets activated, $\mathcal{A}$ is activated instead. The adversary may then send messages $(\texttt{Corrupt}, P_i)$ to the control program and C will register the party P_i as corrupted, as long as there are less than $t < n$ parties corrupted. Further, the adversary can set each party asleep by sending a message $(\texttt{sleep}, P_i)$ to the control program. The control program C will set the party P_i asleep for the next round with probability s, without informing $\mathcal{A}$

if the instruction was successful or not.

Each party P_i has access to two ideal functionalities, the "random oracle" and the "diffusion channel", which are also modelled as ITM's. These functionalities, defined below, are used as subroutines in the Backbone protocol.

Rounds

A round of the protocol execution is a sequence of actions, performed by the different ITI's. In our setting, a round starts with the activation of the party P_1, which then performs the protocol-specific steps. By calling the below defined diffuse functionality, P_1 has finished it's actions for the current round and $\mathcal{Z}$ will activate P_2. If the party P_i is corrupted, $\mathcal{A}$ will be activated and if P_i is asleep, P_{i+1} gets activated instead. The round ends after P_n has finished. Rounds are ordered and therefore enumerated, starting from 1.

Views

Let us formally define the view of a party P. The only "external" input to the protocol is the security parameter κ. Therefore, we can consider κ to be constant over all rounds of the execution and we can exclude it from the random variable describing the view of a party. We denote by the random variable $VIEW_{\mathcal{A},\mathcal{Z}}^{P,t,n}$ the view of a party P after the execution of the Bitcoin backbone protocol in an environment $\mathcal{Z}$ and with adversary $\mathcal{A}$. The complete view over all n parties is the concatenation of their views, denoted by the random variable $VIEW_{\mathcal{A},\mathcal{Z}}^{t,n}$.

Communication and "hashing power"

The two ideal functionalities, which are accessible by the parties, model the communication between them and the way of calculating values of a hash function $H(\cdot) : \{0,1\}^* \to \{0,1\}^\kappa$ concurrently.

The random oracle functionality

The random oracle (RO) provides two functions, a calculation and a verification function. Each party is given a number of q calculation queries and unlimited verification queries per round. Thus, an adversary with t corrupted parties may query the random oracle for $t \cdot q$ calculation queries per round. Upon receiving a calculation query with some value x by a party P_i, the random oracle checks, whether x was already queried before. If not, the RO selects randomly $y \in \{0,1\}^\kappa$ and returns it. Further, the RO maintains a table and adds the pair (x, y) into this table. If x was already queried before, the RO searches in the table for the corresponding pair and returns the value y from it. It's easy to see that a

verification query now only returns true/valid, if such a pair exists in the table of the RO. Note that the RO can maintain tables for different hash functions and can be used for all hash functions we need.

The diffuse functionality

The diffuse functionality models the communication between the parties and thus maintains a *RECEIVE*() string for each party P_i. Note that this is not the same as the previously mentioned input tape. Each party can read the content of its *RECEIVE*() string at any time. The message delay is denoted by Δ, where $\Delta = 0$ corresponds to a synchronous setting.

The diffuse functionality has a *round* variable, which is initially set to 1. Each party P_i can send a message m, possibly empty, to the functionality, which then marks P_i as complete for the current round. We allow $\mathcal{A}$ to read all the messages that are sent by some P_i, without modifying, dropping or delaying it. When all parties and the adversary are marked as complete, the functionality writes all messages that are Δ rounds old to the *RECEIVE*() strings of either only the alert or all parties. We denote by B a Boolean function that indicates exactly that; if $B = 0$ the diffuse functionality writes all messages to the *RECEIVE*() strings of the alert parties, while if $B = 1$ the diffuse functionality writes all messages to the *RECEIVE*() strings of all parties. Each party can read the received messages in the next round being alert. At the end, *round* is incremented.

Note that in the case where $B = 1$, if a party is asleep at a round, it automatically gets marked as complete for this round. Further, upon waking up, it can read all the messages that were written to its *RECEIVE*() string while it was asleep.

Successful queries

A query to the RO oracle is successful, if the returned value $y < T$, where T is the difficulty parameter for the PoW function. The party, which have issued the query will then create a new valid block and may distribute it by the diffuse functionality. We denote the success probability of a single query by $p = Pr[y < T] = \frac{T}{2^\kappa}$. Note that in Bitcoin, the difficulty parameter is adjusted such that the block generation time is approximately ten minutes.

2.2 Sleepy, alert and corrupted

For each round i, we have at most t corrupted and $n_{honest,i} = n - t$ honest parties. Furthermore, the number of honest parties are divided to alert and sleepy parties, $n_{honest,i} = n_{alert,i} + n_{sleepy,i}$. We assume without loss of generality that no

corrupted party is asleep, since we only upper-bound the power of the adversary. Since $n_{alert,i}$ and $n_{sleepy,i}$ are random variables, we can also use their expected value. The expected value is constant over different rounds, thus we will refer to them as $E[n_{alert}]$ and $E[n_{sleepy}]$. Since each honest party is independently set to sleep with probability s and thus the random variable $n_{sleepy,i}$ is binomially distributed with parameters $(n-t)$ and s. Accordingly, $n_{alert,i}$ is also binomially distributed with parameters $(n-t)$ and $(1-s)$. Hence, $E[n_{sleepy}] = s \cdot (n-t)$ and $E[n_{alert}] = (1-s) \cdot (n-t)$.

2.3 Parametrized Model

Let $M(q, \Delta, B)$ be the model, defined in this section. In the following sections, we will look at three instantiations of this model. First of all, we are going to analyze the model $M(q, 0, 1)$, which corresponds to a synchronous setting, in which each party has the ability to make q queries to the random oracle and receives every message, even if the party is asleep. Then, we extend these results to the bounded delay model, which corresponds to $M(1, \Delta, 1)$. As before, every party will always receive messages, but we restrict q to be 1. In the last section, we analyze the model $M(q, 0, 0)$, which corresponds to the synchronous model, but we do not allow the diffuse functionality to write messages on the $RECEIVE()$ tapes of sleepy parties.

2.4 Properties

In order to prove the security of the Bitcoin backbone protocol, we are going to analyze three different properties, following the analysis of [5]. These properties are defined as predicates over $VIEW^{t,n}_{\mathcal{A},\mathcal{Z}}$, which will hold for all polynomially bounded environments $\mathcal{Z}$ and adversaries $\mathcal{A}$ with high probability.

Definition 2.1. Given a predicate Q and a bound $q, t, n \in \mathbb{N}$ with $t < n$, we say that the Bitcoin backbone protocol satisfies the property Q in the model $M(q, \Delta, B)$ for n parties, assuming the number of corruptions is bounded by t, provided that for all polynomial-time $\mathcal{Z}, \mathcal{A}$, the probability that $Q(VIEW^{t,n}_{\mathcal{A},\mathcal{Z}})$ is false is negligible in κ.

The following two Definitions concern the liveness and eventual consistency properties of the Backbone protocol. Using the notation of [5], we denote a chain C, where the last k blocks are removed, by $C^{\lceil k}$. Further, $C_1 \preceq C_2$ is denotes that C_1 is a prefix of C_2.

Definition 2.2. The chain growth property Q_{cg} with parameters $\tau \in \mathbb{R}$ and $s \in \mathbb{N}$ states that for any honest party P with chain C in $VIEW^{t,n}_{\mathcal{A},\mathcal{Z}}$, it holds

that for any $s+1$ rounds, there are at least $\tau \cdot s$ blocks added to the chain of P. [1]

Definition 2.3. The common-prefix property Q_{cp} with parameter $k \in \mathbb{N}$ states that for any pair of honest players P_1, P_2 adopting the chains C_1, C_2 at rounds $r_1 \leq r_2$ in $VIEW^{t,n}_{\mathcal{A},\mathcal{Z}}$ respectively, it holds that $C_1^{\lceil k} \preceq C_2$.

In order to argue about the number of adversarial blocks in a chain, we will use the chain quality property, as defined below:

Definition 2.4. The chain quality property Q_{cq} with parameters $\mu \in \mathbb{R}$ and $l \in \mathbb{N}$ states that for any honest party P with chain C in $VIEW^{t,n}_{\mathcal{A},\mathcal{Z}}$, it holds that for any ℓ consecutive blocks of C the ratio of adversarial blocks is at most μ.

[1] The Chain-Growth Property in [5] is defined slightly different: *.., it holds that for any s rounds, there are at least $\tau \cdot s$ blocks added to the chain of P..* By considering the proof for Theorem 3.8, one can see, why we use $s+1$ instead of s. It follows by the fact that the sum in Lemma 3.7 only goes from $i = r$ to $s-1$ and not to s.

Chapter 3

The q-bounded Synchronous Model without Message Loss

In this section, we analyze the Bitcoin backbone protocol in the previously defined model, instantiated as $M(q, 0, 1)$. This corresponds to the q-bounded synchronous setting from [5]. First, we define the success probabilities for the alert and corrupted parties, which will be used to prove the relations between them. At the end, we use these results to prove the chain growth, common prefix and chain quality properties.

Following the Definition of [5], let a successful round be a round in which at least one honest party solves a PoW. The random variable X_i is used to indicate successful rounds i by setting $X_i = 1$ and $X_i = 0$ otherwise. Further, we denote for a set of rounds S: $X(S) = \sum_{i \in S} X_i$. We note that if no party is asleep, we have $E[X_i] = Pr[X_i = 1] = 1 - (1 - p)^{q(n-t)}$, as in [5].

Lemma 3.1. *It holds that* $\frac{pqE[n_{alert}]}{1+pqE[n_{alert}]} \leq E[X_i] \leq pqE[n_{alert}]$.

Proof. By the definition of X_i, we know that $E[X_i] = E[1 - (1 - p)^{qn_{alert,i}}]$. Thus, the second inequality can easily be derived using Bernoulli. And for the first inequality holds:

$$
\begin{aligned}
E[X_i] &= \sum_{k=0}^{n-t} E[X_i | n_{alert,i} = k] \cdot Pr[n_{alert,i} = k] && \text{Law of total probability} \\
&= \sum_{k=0}^{n-t} \left(1 - (1-p)^{qk}\right) \cdot \binom{n-t}{k} (1-s)^k s^{n-t-k} && n_{alert} \sim \text{Bin}(n-t, 1-s) \\
&= \sum_{k=0}^{n-t} \binom{n-t}{k} (1-s)^k s^{n-t-k} \\
&\quad - \sum_{k=0}^{n-t} (1-p)^{qk} \binom{n-t}{k} (1-s)^k s^{n-t-k} \\
&= 1 - \left(s - (s-1)(1-p)^q\right)^{n-t} && \text{Binomial Theorem (twice)} \\
&\geq 1 - \left(s - (s-1)(1-pq)\right)^{n-t} && \text{By Bernoulli} \\
&\geq 1 - e^{-(1-s)(n-t)pq} && 1 - x \leq e^{-x} \\
&= 1 - \frac{1}{e^{pqE[n_{alert}]}} \\
&\geq 1 - \frac{1}{1 + pqE[n_{alert}]} && 1 + x \leq e^x \\
&= \frac{pqE[n_{alert}]}{1 + pqE[n_{alert}]}
\end{aligned}
$$

□

Further, we adapt the notation of a unique successful round from [5]. A round is called a unique successful round, if exactly one honest party obtains a PoW. Accordingly to the successful rounds, let the random variable Y_i indicate a unique successful round i with $Y_i = 1$ and $Y_i = 0$ otherwise. And for a set of rounds S, let $Y(S) = \sum_{i \in S} Y_i$.

Lemma 3.2. *It holds* $E[Y_i] = E[pqn_{alert,i}(1-p)^{q(n_{alert,i}-1)}] \geq E[X_i](1 - E[X_i])$.

Proof. To prove the required bounds, we need a few intermediary steps. Using Bernoulli, we can derive the following:

$$E[Y_i] = E[pqn_{alert,i}(1-p)^{q(n_{alert,i}-1)}] \geq E[pqn_{alert,i}(1 - pq(n_{alert,i} - 1))]$$

Then, we have to prove that $pqE[n_{alert}](1 - pqE[n_{alert}]) \geq E[X_i](1 - E[X_i])$. From the upper bound on $E[X_i]$, we can derive $E[X_i] = pqE[n_{alert}] - b$, for $b \geq 0$. Therefore:

$$\begin{aligned} E[X_i](1-E[X_i]) &= (pqE[n_{alert}]-b)(1-pqE[n_{alert}]+b) \\ &= pqE[n_{alert}](1-pqE[n_{alert}])-b^2-b+2pqE[n_{alert}]b \end{aligned}$$

In order to prove the required bound, it must hold that $0 \geq -b^2 - b + 2pqE[n_{alert}]b$, which is equivalent to $1 \geq E[X_i] + pqE[n_{alert}]$ and holds by the fact that $2E[X_i] \leq 1$. This is also required by the proof in [5], but not stated explicitly. Since in Bitcoin, $E[X_i]$ is between $2\% - 3\%$, the inequality can be justified.

To conclude the proof, we just have to prove the following:

$$\begin{aligned} & E[pqn_{alert,i}(1-pq(n_{alert,i}-1))] && \geq pqE[n_{alert}] - (pq)^2 E[n_{alert}]^2 \\ \Leftarrow & E[{n_{alert}}^2] - E[n_{alert}] && \leq E[n_{alert}]^2 \end{aligned}$$

Which is equivalent to $Var[n_{alert}] \leq E[n_{alert}]$ and holds for the binomial distribution. □

Following the Definition of [5], let the random variable $Z_{ijk} = 1$ if the adversary obtains a PoW at round i by the j^{th} query of the k^{th} corrupted party. Otherwise, we set $Z_{ijk} = 0$. Summing up, gives us $Z_i = \sum_{k=1}^{t} \sum_{j=1}^{q} Z_{ijk}$ and $Z(S) = \sum_{i \in S} Z_i$. Then, the expected number of blocks that the adversary can mine in one round i is:

$$E[Z_i] = qpt = \frac{t}{E[n_{alert}]} pqE[n_{alert}] \leq \frac{t}{E[n_{alert}]} \cdot \frac{E[X_i]}{1-E[X_i]}$$

3.1 Temporary dishonest majority assumption

We assume the honest majority assumption holds on expectation. Specifically, for each round the following holds: $t \leq c \cdot (1-\delta) \cdot E[n_{alert}]$, where $\delta \geq 2E[X_i] + 2\epsilon$ and $c \in [0,1]$ is a constant. As in [5], δ refers to the advantage of the honest parties and ϵ will be defined in Definition 3.4.

From the expected honest majority assumption, we can derive a possible upper bound for s, depending on t, δ and c. Formally,

$$s \leq \frac{n - t - \frac{t}{c(1-\delta)}}{n-t} = 1 - \frac{1}{c(1-\delta)} \frac{t}{n-t}$$

3.2 Security analysis

The following two Definitions formalize typical executions of the Backbone protocol. Both of them are related to the hash functions, used for implementing the Backbone Protocol. Further, the parameters ϵ and η are introduced. According to [5], $\epsilon \in (0, 1)$ refers to the quality of concentration of random variables in typical executions and η corresponds to the parameter, determining block to round translation.

Definition 3.3 ([5], Definition 8)**.** An *insertion* occurs when, given a chain $\mathcal{C}$ with two consecutive blocks B and B', a block B^* is such that B, B^*, B' form three consecutive blocks of a valid chain. A *copy* occurs if the same block exists in two different positions. A *prediction* occurs when a block extends one which was computed at a later round.

Definition 3.4 ([5], Definition 9)**.** *(Typical execution).* An execution is $(\epsilon, \eta)-$ *typical* if, for any set S of consecutive rounds with $|S| \geq \eta\kappa$, the following holds:

a) $(1-\epsilon)E[X(S)] < X(S) < (1+\epsilon)E[X(S)]$

b) $(1-\epsilon)E[Y(S)] < Y(S)$

c) $Z(S) < (1+\epsilon)E[Z(S)]$

d) No insertions, no copies and no predictions occurred.

Theorem 3.5. *An execution is typical with probability* $1 - e^{-\Omega(\kappa)}$.

Proof. To prove a), b) and c), we can simply use a Chernoff bound by arguing that $E[X(S)]$, $E[Y(S)]$ and $E[Z(S)]$ are in $\Omega(|S|)$. The proof for the property d) is equivalent to [5], by reducing these events to a collision in one of the hash functions of the Bitcoin backbone protocol. Such collisions only happen with probability $e^{-\Omega(\kappa)}$.

□

The following Lemma shows the relations between the different expected values. The bounds are required in all proofs of the three properties and therefore essential. [1]

Lemma 3.6. *The following hold for any set S of at least $\eta\kappa$ consecutive rounds in a typical execution.*

a) $(1-\epsilon)E[X_i]|S| < X(S) < (1+\epsilon)E[X_i]|S|$

[1] The statement d) uses different factors as [5]. The problem is, that it's even not possible to prove the bounds from [5] with their Theorems, Lemmas and assumptions.

b) $(1-\epsilon)E[X_i](1-E[X_i])|S| < Y(S)$

c) $Z(S) < (1+\epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1-E[X_i]}|S| \leq c(1+\epsilon)(1-\delta)\frac{E[X_i]}{1-E[X_i]}|S|$

d) For $\sigma = (1-\epsilon)(1-E[X_i])$*:*

$$Z(S) < \Big(1+\frac{\delta}{\sigma}\Big)\frac{t}{E[n_{alert}]}X(S) \leq c\Big(1-\frac{\delta^2}{2\sigma}\Big)X(S)$$

e) $Z(S) < Y(S)$

Proof.

a) Follows directly from Definition 3.4.

b) By $E[Y_i] \geq E[X_i](1-E[X_i])$ and Definition 3.4.

c) The first inequality follows from Definition 3.4 and the second inequality follows from the honest majority assumption.

d) Let us first prove the first inequality:

$$\begin{aligned}
Z(S) &< (1+\epsilon)E[Z(S)] && \text{By Definition 3.4}\\
&= (1+\epsilon)E[Z_i]|S| && \text{Linearity of } E[Z(S)]\\
&\leq (1+\epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1-E[X_i]}|S| && \text{upper-bound for } E[Z_i]\\
&< (1+\epsilon)\frac{t}{E[n_{alert}]}\frac{1}{1-E[X_i]}\frac{1}{1-\epsilon}X(S) && \text{By a)}
\end{aligned}$$

So, the only thing left to prove is that $\frac{1+\epsilon}{1-\epsilon}\frac{1}{1-E[X_i]} \leq 1+\frac{\delta}{\sigma}$:

$$\frac{1+\epsilon}{1-\epsilon}\frac{1}{1-E[X_i]} - 1 = \frac{1+\epsilon-\sigma}{\sigma} = \frac{2\epsilon + E[X_i] - \epsilon E[X_i]}{\sigma} \leq \frac{\delta}{\sigma}$$

For the second inequality, we can use the honest majority assumption, which gives us:

$$\begin{aligned}
\Big(1+\frac{\delta}{\sigma}\Big)\frac{t}{E[n_{alert}]} &\leq c\cdot\Big(1+\frac{\delta}{\sigma}\Big)(1-\delta)\\
&= c\Big(1+\frac{\delta(1-\sigma-\delta)}{\sigma}\Big)\\
&\leq c\Big(1+\frac{\delta(\frac{\delta}{2}-\delta)}{\sigma}\Big) && \text{Using } 1-\sigma \leq \frac{\delta}{2}\\
&\leq c\Big(1-\frac{\delta^2}{2\sigma}\Big)
\end{aligned}$$

e) To prove $Z(S) < Y(S)$, we apply Lemma 3.6 b) and c) and then prove that

$$(1+\epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1-E[X_i]}|S| < (1-\epsilon)E[X_i](1-E[X_i])|S|$$

Dividing both sides with $E[X_i]|S|$ and multiplying with $(1-E[X_i])$ gives us:

$$(1+\epsilon)\frac{t}{E[n_{alert}]} < (1-\epsilon)(1-E[X_i])^2$$

By the honest majority assumption, we get $(1+\epsilon)\frac{t}{E[n_{alert}]} \leq c(1+\epsilon)(1-\delta)$. And since $c \in [0,1]$, we get $c(1+\epsilon)(1-\delta) \leq (1+\epsilon)(1-\delta)$. Thus, we only need to prove that:

$$\begin{aligned}
&(1+\epsilon)(1-\delta) && \leq (1-\epsilon)(1-E[X_i])^2 \\
\Leftrightarrow &1+\epsilon-\delta-\delta\epsilon && \leq 1-2E[X_i]+E[X_i]^2-\epsilon+2\epsilon E[X_i]-\epsilon E[X_i]^2 \\
\Leftrightarrow &2\epsilon+2E[X_i]+\epsilon E[X_i]^2 && \leq \delta+E[X_i]^2+2\epsilon E[X_i]+\delta\epsilon \\
\Leftrightarrow &\epsilon E[X_i]^2 && \leq E[X_i]^2+2\epsilon E[X_i]+\delta\epsilon
\end{aligned}$$

Where the last step was just applying $2E[X_i]+2\epsilon \leq \delta$. And since $\epsilon \in (0,1)$: $\epsilon E[X_i]^2 \leq E[X_i]^2$.

□

Next, we prove Bitcoin is secure under temporary dishonest majority in the q-bounded synchronous setting by proving the three properties defined in [5]: *chain growth*, *common prefix* and *chain quality*.

3.3 Chain-Growth

In the following, we provide and prove a lower bound for the chain growth of the backbone protocol. This property is generally known as liveness property. The following Lemma holds since it only depends on the random variable X_i.

Lemma 3.7 ([5], Lemma 7)**.** *Suppose that at round r, an honest party has a chain of length l. Then, by round $s \geq r$, every alert party has adopted a chain of length at least $l + \sum_{i=r}^{s-1} X_i$.*

Below, we prove Theorem 13 of [5] in this model.

Theorem 3.8. *In a typical execution, the chain-growth property holds with parameters* $\tau = (1-\epsilon)E[X_i]$ *and* $s \geq \eta\kappa$.

Proof. Note that it's sufficient to lower-bound the chain growth by only considering the random variable X_i, because if the adversary obtains a PoW, he either tries to "replace" some block from a honest node or he just adds the block to the chain. In both cases, the chain doesn't get smaller.

Then, for any set of rounds $S = \{r_1, \ldots, r_s\}$ with $|S| \geq \eta\kappa + 1$ holds by Lemma 3.7:

The chains of every honest player grows by at least $\sum_{i=1}^{|S|-1} X_{r_i} = X(S \setminus r_s)$. Using Lemma 3.6, we can derive

$$X(S \setminus r_s) > (1-\epsilon)E[X_i](|S|-1) \geq (1-\epsilon)E[X_i]\eta\kappa$$

□

Now, we prove an upper bound for the chain growth of the backbone protocol. The main difference to the lower bound is that we now also have to consider blocks, contributed by the adversary.

Lemma 3.9. *In* $\eta\kappa$ *consecutive rounds of a typical execution, less than* $(1+c)(1+\epsilon)\eta\kappa E[X_i]$ *blocks are computed.*

Proof. We know that for any set S of consecutive rounds, $X(S)+Z(S)$ blocks can be added in expectation. This is already an upper bound, since $X(S)+Z(S)$ blocks implies that the adversary does not replace any block from an honest party. We can upper bound $X(S)+Z(S)$ in the following way:

$$\begin{aligned}
X(S)+Z(S) &< X(S)\Big(1+c(1-\frac{\delta^2}{2\sigma})\Big) && \text{By Lemma 3.6 d)}\\
&\leq X(S)(1+c)\\
&< (1+c)(1+\epsilon)E[X_i]|S| && \text{By Lemma 3.6 a)}\\
&= (1+c)(1+\epsilon)\eta\kappa E[X_i]
\end{aligned}$$

□

3.4 Common Prefix

In the following, we prove that the common prefix property holds for a lower bounded parameter k. This proof provides the eventual consistency guarantees of the Bitcoin backbone protocol. The following Lemma holds, since the adversary

can try to replace blocks, computed by the honest parties. If he cannot or will not, the new block gets diffused and adapted by every other party in the next round.

Lemma 3.10 ([5], Lemma 6)**.** *Suppose the k^{th} block B of a chain C was computed by an honest party in a uniquely successful round. Then the k^{th} block in a chain C' is either B or has been computed by the adversary.*

Using this fact, we are able to prove the following Lemma and the common prefix Theorem, which are adapted from [5].

Lemma 3.11. *Assume a typical execution and consider two chains C_1 and C_2 such that $len(C_2) \geq len(C_1)$. If C_1 is adopted by an honest party at round r and C_2 is either adopted by an honest party or diffused at round r, then $C_1^{\lceil k} \preceq C_2$ and $C_2^{\lceil k} \preceq C_1$, for $k \geq (1+c)(1+\epsilon)\eta\kappa E[X_i]$.*

Proof. Equivalent to the proof in [5]. The Lemma can be shown by creating a set of rounds S, in which at least k blocks were created. Then, using Lemma 3.10, we can pair uniquely successful rounds in S with an adversarial block computed in S. In order to violate $C_1^{\lceil k} \preceq C_2$ and $C_2^{\lceil k} \preceq C_1$, it must hold that $Z(S) \geq Y(S)$. By Lemma 3.9, the properties of a typical execution have to apply for S, but then $Z(S) \geq Y(S)$ contradicts Lemma 3.6 e). □

Theorem 3.12. *In a typical execution, the common-prefix property holds with parameter $k \geq (1+c)(1+\epsilon)\eta\kappa E[X_i]$.*

Proof. Equivalent to the proof in [5]. By two considering chains C_1 and C_2, violating the common-prefix property, we can derive a contradiction. Therefore, let C_1 and C_2 be adopted by parties P_1 and P_2 at rounds r_1 and r_2. Let $r_1 \leq r \leq r_2$ be the round, in which some party P_i adopts a chain C_2' such that $C_1^{\lceil k} \not\preceq C_2'$. For the case $r = r_1$, the contradiction can be obtained by Lemma 3.11. And for $r_1 > r$, let C_1' be the chain, which P_i adopted at round $r-1$. Using Lemma 3.11, the Definition of r and that $len(C_2') \geq len(C_1)$ holds, because C_2' was preferred over C_1 by some party, we can derive the following:

$$\left(C_2'^{\lceil k} \preceq C_1'\right) \wedge \left(C_1^{\lceil k} \preceq C_1'\right) \wedge \left(len(C_2'^{\lceil k}) \geq len(C_1^{\lceil k})\right) \Longrightarrow C_1^{\lceil k} \preceq C_2'^{\lceil k}$$

This contradicts the Definition of r. □

3.5 Chain Quality

In this part, we are going to prove an upper bound on the adversarial blocks in a chain. Intuitively, this upper bound is approximately equal to $\frac{t}{E[n_{alert}]}$, but differs by some small factor, as we show in the following.

Theorem 3.13. *In a typical execution the chain-quality property holds with parameters* $\mu = (1 + \frac{\delta}{\sigma})\frac{t}{E[n_{alert}]} < c \cdot (1 - \frac{\delta^2}{2\sigma})$ *and* $\ell \geq (1 + c)(1 + \epsilon)\eta\kappa E[X_i]$.

Proof. Follows the same logic as the proof from [5]. We first define $L \geq \ell$ as the minimal number of consecutive blocks, where the first block was created by an honest party and some honest party tried to extend the chain ending at the last block. Then, we define the set of rounds S as the rounds, where the L blocks were created.

Then, let x denote the number of blocks, created by honest parties and included in the ℓ blocks. To get a contradiction, assume that:

$$x \leq \left(1 - \left(1 + \frac{\delta}{\sigma}\right)\frac{t}{E[n_{alert}]}\right) \cdot \ell \leq \left(1 - \left(1 + \frac{\delta}{\sigma}\right)\frac{t}{E[n_{alert}]}\right) \cdot L$$

.

Assuming a typical execution, we know that all L blocks are created during rounds in S. Further, $L \geq X(S)$ can be shown, using Lemma 3.7. Thus:

$$Z(S) \geq L - x \geq \left(1 + \frac{\delta}{\sigma}\right)\frac{t}{E[n_{alert}]} \cdot L \geq \left(1 + \frac{\delta}{\sigma}\right)\frac{t}{E[n_{alert}]}X(S)$$

Where $Z(S) \geq \left(1 + \frac{\delta}{\sigma}\right)\frac{t}{E[n_{alert}]}X(S)$ contradicts Lemma 3.6 d), since by Lemma 3.9, the rules of a typical execution apply for the set S. □

As a result of Theorem 3.13, we can finally make use of the constant c. As we just proved, the fraction of adversarial blocks is upper bounded by $c \cdot (1 - \frac{\delta^2}{2\sigma})$. This means that we can adjust the desired fraction, by changing c. Of course, since $1 - \frac{\delta^2}{2\sigma}$ is a bit lower than one and the ratio will be higher than c. But the difference will be very small and it's possible to adjust c until we have the desired bound. In order to restrict the advantage of selfish mining, it's possible to analyze the Bitcoin backbone protocol, by selecting the wished upper bound by setting c accordingly.

Chapter 4

The semi-Synchronous Model without Message Loss

In this section, we extend the previously seen results to the semi-synchronous (bounded delay) model. This means, that we allow Δ[1] delays for the messages, as described in the Definition of our model. In order to realize the proofs, we have to restrict q to be 1. And as in the last section, we do not assume message losses. Therefore, we refer in this section to the model $M(1, \Delta, 1)$.

Due the introduced network delays, we need to redefine unique successful rounds, because they do not provide the same guarantees in the this model. Especially, Lemma 3.10 will not hold in the new model. Therefore, we will introduce two new random variables, one for successful and one for unique successful rounds in the bounded delay model. Note, that the chances for the adversary do not change and we can use the bounds from the synchronous model.

Following the work of [5], let the random variable X'_i be defined such that for each round i, $X'_i = 1$, if $X_i = 1$ and $X_j = 0$, $\forall j \in \{i-\Delta+1, \ldots, i-1\}$. A round i is called Δ-isolated successful round, if $X'_i = 1$. Further, let $X'(S) = \sum_{i \in S} X'_i$. Using Bernoulli, we can derive the following bound on $E[X'_i]$:

$$E[X'_i] = E[X_i](1 - E[X_i])^{\Delta-1} \geq E[X_i](1 - (\Delta - 1)E[X_i])$$

In order to prove eventual consistency, we have to rely on a stronger events than just uniquely successful rounds. In [5], this is achieved by defining the random variable Y'_i such that for each round i, $Y'_i = 1$, if $Y_i = 1$ and $X_j = 0$, $\forall j \in \{i-\Delta+1, \ldots, i-1, i+1, \ldots, i+\Delta-1\}$. Then, a round i is called Δ-isolated unique successful round, if $Y'_i = 1$. Further, let $Y'(S) = \sum_{i \in S} Y'_i$. As before, we can lower bound $E[Y'_i]$ using Bernoulli:

$$E[Y'_i] = E[X_i](1 - E[X_i])^{2\Delta-1} \geq E[X_i](1 - (2\Delta - 1)E[X_i])$$

[1] According to Theorem 11 of [4], the parameter Δ has to be known by the honest parties to achieve state machine replication, e.g. achieving consensus.

4.1 Temporary dishonest majority assumption

We assume again honest majority on expectation, such that for each round $t \leq c \cdot (1-\delta) \cdot E[n_{alert}]$, where $\delta \geq 2\Delta E[X_i] + 4\epsilon + \frac{4\Delta}{\eta\kappa}$ and $c \in [0,1]$ is a constant. [2] The reason for the higher value of δ (compared to the synchronous model) is that $E[Y_i'] \leq E[Y_i]$ and we need a way to compensate this difference.

4.2 Security analysis

To prove chain growth, common prefix and chain quality, we first define typical executions for this model.

Definition 4.1 ([5], Definition 27)**.** An execution is (ϵ, η, Δ)-typical if, for any set S of consecutive rounds with $|S| \geq \eta\kappa$, the following hold.

a) $(1-\epsilon)E[X'(S)] < X'(S)$ and $X(S) < (1+\epsilon)E[X(S)]$

b) $(1-\epsilon)E[Y'(S)] < Y'(S)$

c) $Z(S) < (1+\epsilon)E[Z(S)]$

d) No insertions, no copies and no predictions occurred.

Theorem 4.2. *An execution is typical with probability* $1 - e^{-\Omega(\kappa)}$.

Proof. Equivalent to the proof of Theorem 3.5. □

The following Lemma corresponds to the semi-synchronous version of Lemma 3.6. Most of the relations follow the same structure and are similar to prove as in the synchronous model.

Lemma 4.3. *The following hold for any set S of at least $\eta\kappa$ consecutive rounds in a typical execution.*

a) $(1-\epsilon)E[X_i](1-E[X_i])^{\Delta-1}|S| < X'(S)$

b) $(1-\epsilon)E[X_i](1-E[X_i])^{2\Delta-1}|S| < Y'(S)$

c) $Z(S) < (1+\epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1-E[X_i]}|S| \leq c(1+\epsilon)(1-\delta)\frac{E[X_i]}{1-E[X_i]}|S|$

[2] One might notice that our lower bound of δ differs from the lower bound from [5]. First of all, they provided two different values for δ, where both of them are wrong in the sense that they are too small in order to prove the needed bounds.

d) Let $S' = \{r, \dots, r'\}$ with $|S'| \geq \eta\kappa$. For $S = \{r, \dots, r' + \Delta\}$ and $\sigma' = (1-\epsilon)(1-E[X_i])^{\Delta}$:

$$Z(S) < \left(1 + \frac{\delta}{2\sigma'}\right) \frac{t}{E[n_{alert}]} X'(S')$$

e) Let $S' = \{r, \dots, r'\}$ with $|S'| \geq \eta\kappa$. For $S = \{r - \Delta, \dots, r' + \Delta\}$:

$$Z(S) < Y'(S')$$

Proof.

a)

$$\begin{aligned}
(1-\epsilon)E[X_i](1-E[X_i])^{\Delta-1}|S| &\leq (1-\epsilon)E[X_i']|S| && \text{Definition of } E[X_i'] \\
&= (1-\epsilon)E[X'(S)] \\
&< X'(s) && \text{Definition 4.1 a)}
\end{aligned}$$

b)

$$\begin{aligned}
(1-\epsilon)E[X_i](1-E[X_i])^{2\Delta-1}|S| &\leq (1-\epsilon)E[Y_i']|S] && \text{Definition of } E[Y_i'] \\
&= (1-\epsilon)E[Y'(S)] \\
&< Y'(S) && \text{Definition 4.1 b)}
\end{aligned}$$

c) Equivalent to Lemma 3.6 c).

d) By applying a) and c), we only have to prove that the following holds:

$$\begin{aligned}
& (1+\epsilon)|S| && \leq (1-\epsilon)(1-E[X_i])^{\Delta}|S'|\left(1 + \frac{\delta}{2\sigma'}\right) \\
\Leftrightarrow & (1+\epsilon)\frac{|S|}{|S'|} && \leq \sigma'\left(1 + \frac{\delta}{2\sigma'}\right) \\
\Leftarrow & \frac{1+\epsilon+\frac{2\Delta}{\eta\kappa}}{\sigma'} - 1 && \leq \frac{\delta}{2\sigma'} \\
\Leftrightarrow & \frac{1+\epsilon+\frac{2\Delta}{\eta\kappa} - (1-\epsilon)(1-E[X_i])^{\Delta}}{\sigma'} && \leq \frac{\delta}{2\sigma'} \\
\Leftarrow & 2\epsilon + \frac{2\Delta}{\eta\kappa} + \Delta E[X_i] && \leq \frac{\delta}{2}
\end{aligned}$$

Where we applied Bernoulli to $(1 - E[X_i])^{\Delta}$. The proof is concluded by the Definition of δ.

e) By applying b) and c), we only have to prove that the following holds:

$$\begin{aligned}
&(1+\epsilon)\frac{t}{E[n_{alert}]}|S| && \leq (1-\epsilon)(1-E[X_i])^{2\Delta}|S'| \\
\Leftarrow &(1+\epsilon)c(1-\delta)\frac{|S|}{|S'|} && \leq (1-\epsilon)(1-E[X_i])^{2\Delta} \\
\Leftrightarrow &1-\delta && \leq \frac{(1-\epsilon)(1-E[X_i])^{2\Delta}}{c(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right)}
\end{aligned}$$

Therefore:

$$\begin{aligned}
1-\frac{(1-\epsilon)(1-E[X_i])^{2\Delta}}{c(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right)} &\leq \frac{(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right)-(1-\epsilon)(1-\Delta E[X_i])^2}{(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right)} \\
&= \frac{2\epsilon+2\Delta E[X_i]+(1+\epsilon)\frac{2\Delta}{\eta\kappa}-O(\Delta^2)}{(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right)} \\
&\leq 2\epsilon+2\Delta E[X_i]+\frac{4\Delta}{\eta\kappa} \\
&\leq \delta
\end{aligned}$$

Which again follows from the Definition of δ. Note that $O(\Delta^2)$ isn't as big as it seems to be. In fact, its equal to $\Delta^2 E[X_i]^2 - 2\Delta\epsilon E[X_i] - \epsilon\Delta^2 E[X_i]^2$, which will be relatively small.

□

4.3 Chain growth

Note, that the chain growth upper bound holds, as it is, since it only depends on $X(S)$ and $Z(S)$.

Lemma 4.4 ([5], Lemma 26)**.** *Suppose that at round r an honest party has a chain of length l. Then, by round $s \geq r+\Delta-1$, every honest party has adopted a chain of length at least $l+\sum_{i=r}^{s-\Delta} X'_i$.*

Proof. Equivalent to [5], by induction on $s-r-\Delta+1 \geq 0$. □

Theorem 4.5. *In a typical execution, the chain-growth property holds with parameters $\tau = (1-\epsilon)E[X_i](1-E[X_i])^{\Delta-1}$ and $s \geq \eta\kappa$.*

Proof. Equivalent to the proof from the synchronous case, but using Lemma 4.4 instead. □

4.4 Common prefix

The following Lemma is the adopted version of Lemma 3.11.

Lemma 4.6. *Assume a typical execution and consider two chains C_1 and C_2 at round r, such that $\text{len}(C_2) \geq \text{len}(C_1)$. For the same conditions of Lemma 3.11, it holds $C_1^{\lceil k} \preceq C_2$ and $C_2^{\lceil k} \preceq C_1$, for $k \geq (1+c)(1+\epsilon)\eta\kappa E[X_i] + 2\Delta$.*

Proof. The proof is the same as in Lemma 3.11, but considering a set $S' = \{i : r^* + \Delta < i < r - \Delta\}$ to contradict $Z(S) < Y'(S')$ from Lemma 4.3. □

Theorem 4.7. *In a typical execution, the common-prefix property holds with parameter $k \geq (1+c)(1+\epsilon)\eta\kappa E[X_i] + 2\Delta$.*

Proof. Equivalent to the proof from Theorem 3.12. □

4.5 Chain quality

Theorem 4.8. *In a typical execution, the chain-quality property holds with parameters $\mu = \left(1 + \frac{\delta}{2\sigma'}\right)\frac{t}{E[n_{alert}]}$ and $\ell \geq (1+c)(1+\epsilon)\eta\kappa E[X_i] + \Delta$.*

Proof. As in Theorem 3.13, we can argue that for $S' = \{r : r_1 \leq r \leq r_2 - \Delta\}$:

$$Z(S) \geq L - x \geq \left(1 + \frac{\delta}{2\sigma'}\right)\frac{t}{E[n_{alert}]}L \geq \left(1 + \frac{\delta}{2\sigma'}\right)\frac{t}{E[n_{alert}]}X'(S')$$

which is contradicting Lemma 4.3 d). □

Chapter 5

The q-bounded Synchronous Model with Message Loss

As in the synchronous case, we do not restrict the number of queries and assume no message delays. In the previous sections, we assumed that messages, sent from the diffusion functionality, will be written on the *RECEIVE*() string of each party. However, in this section, we assume that the messages only get written to the *RECEIVE*() strings of alert parties, i.e. sleepy parties do not receive messages. We therefore refer to the model instantiated as $M(q, 0, 0)$. This models the worst possible event of the reality, because in Bitcoin itself, parties that were offline will check on the currently longest chain, once they get back online. This model captures the effects if none of them receives one of the currently longest chains, thus are eventually a victim of an eclipse attack. This implies that it's not necessarily true that all parties' local chains have the same length.

This change to the model leads to major differences compared to the results from the previous sections. In this case, unique successful rounds doesn't provide the same guarantees as before, especially Lemma 3.10 doesn't hold any more.

In the following, we denote by C_i the set of chains containing all longest chains that exist at round i. Further, we refer to the local chain of player P_j at round i by L_i^j.

The following Lemma shows the expected number of honest players, which have adopted one of the longest chains existing at the current round.

Lemma 5.1. *At every round i, there are expected $E[n_{alert}] = (1 - s)(n - t)$ parties j, such that $L_i^j \in C_i$.*

Proof. We will prove the Lemma by induction over all rounds of an execution. The base case is trivial, because at round 1, every party starts with the genesis block. Now for the step case, assume that the Lemma holds at round i. Then we show that it holds at round $i + 1$ too. In order to prove this, we perform a case distinction:

- Case $X_i = 0$: No new chains will be diffused, therefore no new chains can be adopted and we can apply the induction hypothesis.
- Case $Z_i = 0$: Analogue to the previous case.
- Case $X_i = 1$: (But $Y_i = 0$) Now we have to differentiate, if the new blocks extend some chain in C_i not:
  a) Some longest chain is extended:
  Every party, which is not asleep at round i will adopt one of the possibly multiple resulting new longest chains. Thus, there are expected $E[n_{alert}]$ alert parties which will have adopted one of the longest chains at round $i + 1$.
  b) No longest chain is extended:
  No honest party, whose local chain is already one of the currently longest chain will adopt a new chain, since it's length will not be larger than the length of its local chain. Thus, we can apply the induction hypothesis.
- Case $Y_i = 1$: As in the case before, every party, which was alert at round i, will adopt the resulting chain, if its length is larger than the length of its local chain. As before, there are $E[n_{alert}]$ alert parties which will have adopted one of the longest chains at round $i + 1$.
- Case $Z_i = 1$: Analogue to the previous case. But if the adversary withholds the found block, the case $Z_i = 0$ applies and at the round, where it diffuses this block, this case applies.

□

By the Lemma above, at every round i only expected $(1 - s)(n - t)$ parties j have a local chain $L_i^j \in C_i$. And a fraction of $(1 - s)$ of them will again be sleepy in the following rounds. Therefore, let $n^*_{alert,i}$ denote the number of alert parties j at round i, where $L_i^j \in C_i$.

It's easy to see that $n^*_{alert,i}$ is binomially distributed with parameters $(n - t)$ and $(1 - s)^2$. Let $E[n^*_{alert}] = (1 - s)^2(n - t)$ denote the expected value of $n^*_{alert,i}$, omitting the round index i, since the expected value is equal for all rounds. We define the random variable X_i^* which indicates, if at least one of the $n^*_{alert,i}$ parties solves a PoW at round i. Thus, we set $X_i^* = 1$, if some honest party j with $L_i^j \in C_i$ solves a PoW at round i and $X_i^* = 0$ otherwise. Further, we define for a set of rounds S: $X^*(S) = \sum_{i \in S} X_i^*$. The following Lemma can be proven, using the same argumentation as in the proof for the Lemma 3.1:

Lemma 5.2. *It holds that* $\frac{pqE[n^*_{alert}]}{1+pqE[n^*_{alert}]} \leq E[X_i^*] \leq pqE[n^*_{alert}]$.

Accordingly, let Y_i^* denote the random variable with $Y_i^* = 1$, if exactly one honest party j solves a PoW at round i and $L_i^j \in C_i$. Note that the resulting chain, will be the only longest chain. Further, for a set of rounds S let $Y^*(S) = \sum_{i \in S} Y_i^*$.

Lemma 5.3. *It holds* $E[Y_i^*] = E[pqn^*_{alert,i}(1-p)^{q(n^*_{alert,i}-1)}] \geq E[X_i^*](1-E[X_i^*])$.

Proof. The proof follows the exactly same steps as the proof for Lemma 3.2. □

5.1 Temporary dishonest majority assumption

In this setting, the honest majority assumption changes slightly. We cannot simply assume that t is smaller than some fraction of $E[n^*_{alert}]$, because we have also to consider parties j with $L_i^j \notin C_i$. We assume that for each round holds $t + (1-s)E[n_{sleepy}] \leq c \cdot (1-\delta) \cdot E[n^*_{alert}]$, where $\delta \geq 3\epsilon + 2E[X_i^*]$ and some constant $c \in [0,1]$. Note that $(1-s)E[n_{sleepy}]$ is the fraction of alert parties, working on shorter chains.

In order to compute the upper bound for s, we reformulate the honest majority assumption. Using the quadratic formula, this results in the following:

$$s \leq \frac{2c(1-\delta) - \sqrt{1 + 4(1 + c(1-\delta)\frac{t}{n-t})}}{2(1 + c(1-\delta))}$$

In the model description, we specified that the adversary is not informed if a party P_i is set to sleep, after sending an instruction $(\texttt{sleep}, P_i)$ to the control program C. This assumption is realistic since the adversary can not be certain about the success of his attempt to create a crash-failure. Further, allowing the adversary to know when he successfully set to sleep a node makes him quite powerful. Specifically, in our model we have a fraction of $1-s$ alert parties. Subtracting the parties, which are working on a longest chain, from the $(1-s)(n-t)$ parties, leaves us an expected fraction of $s(1-s)$ parties, which can be found on the left hand side of the honest majority assumption. If we would assume that the adversary knows, which parties are asleep at each round, we would have to change the temporary dishonest majority assumption to $t + E[n_{sleepy}] \leq c \cdot (1-\delta) \cdot E[n^*_{alert}]$. Then, the adversary could exploit this knowledge to his advantage and send sleep instructions to the parties working on the longest chains. To capture this adversarial behaviour a different model would be necessary (since s cannot be considered constant).

5.2 Security analysis

Lemma 5.4. *Suppose that at round r, the chains in C_i have size l. Then by round $s \geq r$, an expected number of $E[n_{alert}] = (1-s)(n-t)$ parties will have adapted a chain of length at least $l + \sum_{i=r}^{s-1} X_i^*$.*

Proof. By Lemma 5.1, for every round i, the expected number of parties j with $L_i^j \in C_i$ is $E[n_{alert}]$. Therefore, we only have to count the number of times, when one of these longest chains gets extended. □

In the following, we define a new variable ϕ and provide an upper bound for it. This is required for the proof of the common prefix property. Although the proven bound is not tight, it is sufficient for proving the desired properties.

Lemma 5.5. *The probability that the honest parties j with $L_i^j \notin C_i$ can create a new chain $C' \in C_r$ for some round $r \geq i$, before any chain from C_i gets extended is denoted by ϕ. It holds that:*

$$\phi \leq \frac{s}{1-s}$$

Proof. Without loss of generality, we may assume that all parties j with $L_i^j \notin C_i$ have the same local chain. Further, we can assume that this chain is just one block shorter than the currently longest chain. Thus, we search an upper bound for the probability that the parties $\{P_j\}_{L_i^j \notin C_i}$ are faster in solving two PoW's than the parties $\{P_j\}_{L_i^j \in C_i}$ solving one PoW.

In order to prove that, we have to introduce a new random variable $\tilde{X}_i$, with $\tilde{X}_i = 1$ if some honest party j with $L_i^j \notin C_i$ solves a PoW. By the same argumentation as in Lemma 3.1, we can argue that $\frac{pq(1-s)E[n_{sleepy}]}{1+pq(1-s)E[n_{sleepy}]} \leq E[\tilde{X}_i] \leq pq(1-s)E[n_{sleepy}]$. Therefore, the upper bound on the required probability is:

$$\begin{aligned}
&\sum_{k=2}^{\infty}(k-1)E[\tilde{X}_i]^2(1-E[\tilde{X}_i])^{k-2}(1-E[X_i^*])^k \\
=&\frac{E[\tilde{X}_i]^2}{(1-E[\tilde{X}_i])^2} \cdot \sum_{k=2}^{\infty}(k-1)\big((1-E[\tilde{X}_i])(1-E[X_i^*])\big)^k \\
=&\frac{E[\tilde{X}_i]^2(1-E[X_i^*])^2}{(E[\tilde{X}_i]+E[X_i^*]-E[\tilde{X}_i]E[X_i^*])^2}
\end{aligned}$$

Now, let $a := pq(1-s)^2(n-t) = pqE[n_{alert}^*]$ and $b := pqs(1-s)(n-t) = pq(1-s)E[n_{sleepy}]$. Then by the Definition of $E[\tilde{X}_i]$ and $E[X_i^*]$ holds:

$$\begin{aligned}\frac{E[\tilde{X}_i]^2(1-E[X_i^*])^2}{(E[\tilde{X}_i]+E[X_i^*]-E[\tilde{X}_i]E[X_i^*])^2} &= \frac{b^2\left(\frac{1}{1+a}\right)^2}{\left(\frac{a}{1+a}+\frac{b}{1+b}-ab\right)^2} \\ &= \frac{b^2}{(1+a)^2(1-ab)^2(a+ab+b)^2}\end{aligned}$$

Thus, $\phi \leq \frac{s}{1-s}$ is equivalent to:

$$\begin{aligned}&\frac{b^2}{(1+a)^2(1-ab)^2(a+ab+b)^2} && \leq \frac{s}{1-s} \\ \Leftrightarrow &b^2(1-s) && \leq s(1+a)^2(1-ab)^2(a+ab+b)^2 \\ \Leftrightarrow &ab && \leq (1+a)^2(1-ab)^2(a+ab+b)^2\end{aligned}$$

The inequality holds, since $(1+a)^2(1-ab)^2 \geq 1$ and $ab \leq (a+ab+b)^2$. □

The following Lemma replaces Lemma 3.10. The possibility to have chains of different length at the same round offers various ways to replace a block from a round i, where $Y_i^* = 1$. Therefore, we cannot use the same arguments as in Lemma 3.10.

Lemma 5.6. *Suppose the k^{th} block B of a chain C was computed at round i, where $Y_i^* = 1$. Then with probability at least $1-\phi$, the k^{th} block in a chain C' will be B or requires at least one adversarial block to replace B.*

Proof. There are several ways to replace a block from a round i, where $Y_i^* = 1$:

1) The adversary has a precomputed block B', replacing B directly. Thus, in the same round, $\mathcal{A}$ diffuses the chain C', where the last added block is B'.

2) The parties j, with $L_i^j \notin C_i$ and thus with $L_i^j \neq C$ solve a PoW at some round $r \geq i$, leading to a new block on some chain C', which has the same length as C. Then, $\mathcal{A}$ extends and diffuses C' before C gets further extended.

3) The adversary solves a PoW and creates some chain C'. As in the second case, we may assume that the length of C and C' is equal. Then at some round $r \geq i$, either $\mathcal{A}$ or at least one party j with $L_r^j = C'$ solve a PoW, resulting in the creation of the set C_r.

4) The parties j with $L_i^j \notin C_i$ are faster in solving two PoW's, before C gets extended.

The cases 1) - 3) involve at least one adversarial block, as required. And by Lemma 5.5, we know that the case 4) only happens with probability ϕ. □

As in the previous sections, we are going to define typical executions and then prove that executions are typical with high probability.

Definition 5.7. An execution is (ϵ, η)-typical if, for any set S of consecutive rounds with $|S| \geq \eta\kappa$, the following hold.

a) $(1-\epsilon)E[X^*(S)] < X^*(S)$ and $X(S) < (1+\epsilon)E[X(S)]$

b) $(1-\epsilon)E[Y^*(S)] < Y^*(S)$

c) $Z(S) < (1+\epsilon)E[Z(S)]$

d) No insertions, no copies and no predictions occurred.

Theorem 5.8 ([5], Theorem 28)**.** *An execution is typical with probability* $1 - e^{-\Omega(\kappa)}$.

Proof. Equivalent to the proof of Theorem 3.5.

□

Since we allow message losses in this model, we require more unique successful rounds than in other models. This leads to a different bound in part e) of the following Lemma.

Lemma 5.9. *The following hold for any set S of at least $\eta\kappa$ consecutive rounds in a typical execution.*

a) $(1-\epsilon)E[X_i^*]|S| < X^*(S)$

b) $(1-\epsilon)E[X_i^*](1-E[X_i^*])|S| < Y^*(S)$

c) $Z(S) < (1+\epsilon)\frac{t}{E[n^*_{alert}]}\frac{E[X_i^*]}{1-E[X_i^*]}|S| < (1+\epsilon)\big(c(1-\delta) - \frac{s}{1-s}\big)\frac{E[X_i^*]}{1-E[X_i^*]}|S|$

d) For $\sigma^* = (1-\epsilon)(1-E[X_i^*])$*:*

$$Z(S) < \Big(1 + \frac{\delta}{\sigma^*}\Big)\frac{t}{E[n^*_{alert}]}X^*(S) \leq c\Big(1 - \frac{\delta^2}{2\sigma^*}\Big)X^*(S)$$

e)

$$Z(S) < Y^*(S)(1-\epsilon)(1-\phi)$$

Proof.

a) Follows directly from Definition 5.7.

b) By $E[Y_i^*] \geq E[X_i^*](1 - E[X_i^*])$ and Definition 5.7.

c) The first inequality follows from Definition 5.7 and the second inequality follows from the honest majority assumption.

d) Let us first prove the first inequality:

$$\begin{aligned}
Z(S) &< (1+\epsilon)E[Z(S)] && \text{By Definition 5.7 c)}\\
&= (1+\epsilon)E[Z_i]|S| && \text{Linearity of } E[Z(S)]\\
&\leq (1+\epsilon)\frac{t}{E[n^*_{alert}]}\frac{E[X_i^*]}{1-E[X_i^*]}|S| && \text{upper-bound for } E[Z_i]\\
&< (1+\epsilon)\frac{t}{E[n^*_{alert}]}\frac{1}{1-E[X_i^*]}\frac{1}{1-\epsilon}X^*(S) && \text{By a)}
\end{aligned}$$

So, the only thing left to prove is that $\frac{1+\epsilon}{1-\epsilon}\frac{1}{1-E[X_i^*]} \leq 1 + \frac{\delta}{\sigma^*}$:

$$\frac{1+\epsilon}{1-\epsilon}\frac{1}{1-E[X_i^*]} - 1 = \frac{1+\epsilon}{\sigma^*} - 1 = \frac{1+\epsilon-\sigma^*}{\sigma^*} \leq \frac{2\epsilon + E[X_i^*]}{\sigma^*} \leq \frac{\delta}{\sigma^*}$$

For the second inequality, we can apply the honest majority assumption and just need to prove:

$$\begin{aligned}
& c\Big(1+\frac{\delta}{\sigma^*}\Big)\Big((1-\delta) - \frac{s}{1-s}\Big) && \leq c\Big(1 - \frac{\delta^2}{2\sigma^*}\Big)\\
\Leftarrow & \Big(1+\frac{\delta}{\sigma^*}\Big)(1-\delta) && \leq 1 - \frac{\delta^2}{2\sigma^*}\\
\Leftrightarrow & 1 + \frac{\delta}{\sigma*} - \delta - \frac{\delta^2}{\sigma^*} && \leq 1 - \frac{\delta^2}{2\sigma^*}\\
\Leftrightarrow & \frac{\delta}{\sigma^*} - \delta && \leq \frac{\delta^2}{2\sigma^*}\\
\Leftarrow & 1 - \sigma^* && \leq \frac{\delta}{2}
\end{aligned}$$

which follows from the Definition of σ^* and δ.

e) The proof goes straight forward, using parts b) and c):

$$
\begin{aligned}
& Z(S) && < Y^*(S)(1-\epsilon)(1-\phi) \\
\Leftarrow & (1+\epsilon)E[Z_i]|S| && \leq (1-\epsilon)^2 E[Y_i^*]|S|(1-\phi) \\
\Leftarrow & (1+\epsilon)\frac{t}{E[n^*_{alert}]} && \leq (1-\epsilon)^2(1-E[X_i^*])^2(1-\phi) \\
\Leftarrow & (1+\epsilon)\big(c(1-\delta)-\frac{s}{1-s}\big) && \leq (1-\epsilon)^2(1-\phi)(1-2E[X_i^*]+E[X_i^*]^2) \\
\Leftarrow & 1-\delta-\frac{s}{1-s}+\epsilon && \leq 1-2\epsilon-\phi+\epsilon\phi-2E[X_i^*](1-\epsilon)^2(1-\phi) \\
\Leftarrow & 3\epsilon+2E[X_i^*]+\phi-\frac{s}{1-s} && \leq \delta \\
\Leftarrow & 3\epsilon+2E[X_i^*] && \leq \delta
\end{aligned}
$$

where the last step follows from Lemma 5.5 and the inequality holds according to the Definition of δ.

□

5.3 Chain-growth

Note that the chain growth upper bound holds with the parameters from the synchronous model without message losses and we just have to prove the new lower bound. Intuitively, if we set $s = 0$, the upper bound is equal to the upper bound from the synchronous model without message losses.

Theorem 5.10. *In a typical execution, the chain-growth property holds with parameters $\tau = (1-\epsilon)E[X_i^*]$ and $s \geq \eta\kappa$.*

Proof. Equivalent to the proof from the synchronous case, but using Lemma 4.4 instead. □

5.4 Common-prefix

Only the proof for the common prefix Lemma changes, since we have several possibilities to replace a block in this new model. This is formalized in the proof of the Lemma below.

Lemma 5.11. *The Lemma 3.11 holds in this model with the same parameters as in the synchronous model without message losses.*

Proof. The proof follows the same logic as in the synchronous model without message losses. We create a set of round S, in which at least k blocks were

created. In order to violate $C_1^{\lceil k} \preceq C_2$ and $C_2^{\lceil k} \preceq C_1$, each block created in a round $i \in S$, where $Y_i^* = 1$ has to be replaced. Using Lemma 5.6, it holds that an expected fraction of $1-\phi$ block replacements involve at least one adversarial block. Using Chernoff, we can argue that for $|S|$ rounds, the fraction of such block replacements is lower bounded by $(1-\epsilon)(1-\phi)$ with probability $1-e^{\Omega(\kappa)}$. Thus, $C_1^{\lceil k} \preceq C_2$ and $C_2^{\lceil k} \preceq C_1$ is violated if and only if $Z(S) > Y^*(S)(1-\epsilon)(1-\phi)$. And by Lemma 3.9, the properties of a typical execution have to apply for S, but then $Z(S) > Y^*(S)(1-\epsilon)(1-\phi)$ contradicts Lemma 5.9 e). □

Theorem 5.12. *In a typical execution, the common-prefix property holds with parameter* $k > (1+c)(1+\epsilon)\eta\kappa E[X_i]$.

Proof. Equivalent to the proof from the synchronous model without message losses, but using Lemma 5.11 instead. □

5.5 Chain-Quality

Theorem 5.13. *In a typical execution, the chain-quality property holds with parameters* $\mu = (1+\frac{\delta}{\sigma^*})\frac{t}{E[n^*_{alert}]} \leq c \cdot (1-\frac{\delta^2}{2\sigma^*})$ *and* $\ell \geq (1+c)(1+\epsilon)\eta\kappa E[X_i]$.

Proof. We can use the same arguments as in the proof from the synchronous model without message losses. Assume that $x \leq (1-(1+\frac{\delta}{\sigma^*})\frac{t}{E[n^*_{alert}]}) \cdot \ell \leq (1-(1+\frac{\delta}{\sigma^*})\frac{t}{E[n^*_{alert}]}) \cdot L$. Then, by Lemma 5.4, it holds that $L \geq X^*(S)$, which implies that $Z(S) \geq L - x \geq (1+\frac{\delta}{\sigma^*})\frac{t}{E[n^*_{alert}]} \cdot L \geq (1+\frac{\delta}{\sigma^*})\frac{t}{E[n^*_{alert}]} X^*(S)$. The contradiction is obtained by Lemma 5.9 d). □

Chapter 6

Main results

As a result of the temporary dishonest majority assumptions, we have derived upper bounds for the probability s as shown in Figure 6. Therefore, we fixed $c = 0.5$ to limit the advantage of an adversary, following a Selfish Mining strategy. Further, we have chosen for all three models $\epsilon = 0.005$. For the synchronous model without message losses, we set $E[X_i] = 0.03$, which results in $\delta = 0.07$.[1] For the Semi-Synchronous model, we set also $E[X_i] = 0.03$, resulting in $E[X_i'] = 0.022$. For $\Delta = 10$, we then get $\delta = 0.46$. And for the synchronous model with message losses, we have chosen $E[X_i^*] = 0.03$, which results in $\delta = 0.075$.

One might be wondering how we could allow such high values for s. We have fixed $E[X_i]$, respectively $E[X_i^*]$, for our calculations. We can do this without loss of generality, since these expected values are dependent on p, which depends on the difficulty parameter T. The adjustment of T, used to regulate the block generation rate, depends on the fraction of sleepy parties, because they do not provide computational power (e.g. new blocks) to the blockchain.

These results are also consistent with the results from [10], where the upper

[1] Note that δ is dependent on $E[X_i]$, which is again dependent on s. If we would remove this dependency, the results would be at most 2% better than the actual results shown in Figure 6.

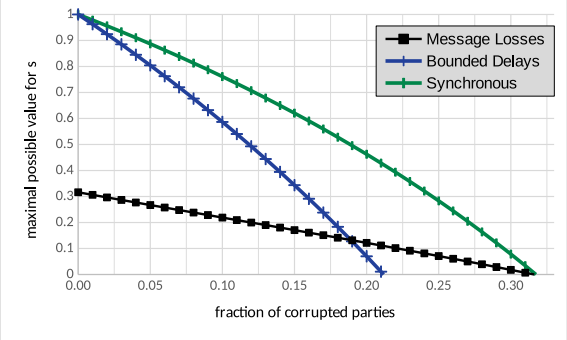

Figure 6.1: This figure shows the upper bound on the fraction of sleepy parties, depending on the fraction of corrupted parties.

bound on the adversarial fraction is stated at 49.1%. If we set $c = 1$ and $s = 0$, due the value of δ, we get an maximal possible adversarial fraction of 48.5%.

Chapter 7

Related Work

This work is built upon the ideas and results of two papers. The Backbone protocol and the framework for the security proofs can be found in [5]. Further, a large part of the model is adapted from [5], which is founded on the work of Ran Canetti in [11, 12, 13]. The sleepy model was introduced in [4], in which Rafael Pass and Elaine Shi present a provably secure protocol when parties can go offline for a certain amount of time. But the proofs are based on the assumption that the fraction of corrupted over the alert parties is always less than some constant. As we showed, we can even relax this assumption for Bitcoin by just assuming honest majority on expectation.

Ouroboros, described in [14], is also proven in a sleepy model. The parties, called stakeholders, can be offline for some time, but are required to be online at important events such as during the slot, where they are elected slot leaders. Since in Ouroboros, the parties receive their block reward when they are elected slot leader, they have a much higher interest in being online at this certain time slot, where in Bitcoin the overall online time (combined with the hashing power) determines the rewards. This is then captured in our model, where each player is round independent set to sleep with some probability.

Chapter 8

Conclusion

In this thesis we combined and extended the work of [5] and [4]. We were able to prove the security of Bitcoin, respectively the security of the Backbone protocol in the sleepy model, by just assuming honest majority on expectation. This shows the resistance of the Bitcoin protocol, which is able to recover from peaks of adversarial power above 50%. The q-bounded synchronous model, where we allow message losses provide one of the most realistic abstractions of the reality and shows nevertheless the security of Bitcoin. The results imply that Bitcoin can withstand very powerful adversaries who can perform eclipse attacks on a number of honest parties.

However, we did not consider a more powerful adversary, who is informed about the success of introduced crash-failures. This would be very interesting to study, since this involves a non-constant probability s and new strategy for the adversary.

Bibliography

[1] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. In: https://bitcoin.org/bitcoin.pdf. (October 2008)

[2] Biryukov, A., Khovratovich, D., Pustogarov, I.: Deanonymisation of clients in bitcoin p2p network. In: http://arxiv.org/abs/1405.7418. (July 2014)

[3] Donet, J.A.D., Perez-Sola, C., Herrera-Joancomart, J.: The bitcoin p2p network. In: https://fc14.ifca.ai/bitcoin/papers/bitcoin14_submission_3.pdf. (March 2014)

[4] Pass, R., Shi, E.: The sleepy model of consensus. In: https://eprint.iacr.org/2016/918.pdf. (May 2017)

[5] Garay, J.A., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: IACR Cryptology ePrint Archive, 2014:765. (June 2017)

[6] Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in bitcoin. In: http://arxiv.org/abs/1507.06183. (July 2015)

[7] Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on bitcoin's peer-to-peer network. In: Cryptology ePrint Archive, Report 2015/263. (March 2015)

[8] Singh, A., Ngan, T.W.J., Druschel, P., Wallach, D.S.: Eclipse attacks on overlay networks: Threats and defenses. In: IEEE Infocom 2006. (April 2006)

[9] Sit, E., Morris, R.: Security considerations for peer-to-peer distributed hash tables. In: Springer, pp. 261-269. (October 2002)

[10] Decker, C., Wattenhofer, R.: Information propagation in the bitcoin network. In: IEEE P2P 2013 Proceedings. (September 2013)

[11] Canetti, R.: Security and composition of multiparty cryptographic protocols. In: J. Cryptology (2000) 13: 143-202. (January 2000)

[12] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Cryptology ePrint Archive, Report 2000/067. (December 2000)

[13] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Proceedings 2001 IEEE International Conference on Cluster Computing. (October 2001)

[14] Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: A provably secure proof-of-stake blockchain protocol. In: https://eprint.iacr.org/2016/889.pdf. (August 2017)