Bitcoin Security under Temporary Dishonest Majority

Georgia Avarikioti, Lukas Käppeli, Yuyi Wang, and Roger Wattenhofer

ETH Zurich, Switzerland
{zetavar,yuwang,wattenhofer}@ethz.ch,
lukas.kaeppeli@hotmail.com

Abstract. We prove Bitcoin is secure under temporary dishonest majority. We assume the adversary can corrupt a specific fraction of parties and also introduce crash failures, i.e., some honest participants are offline during the execution of the protocol. We demand a majority of honest online participants on expectation. We explore three different models and present the requirements for proving Bitcoin's security in all of them: we first examine a synchronous model, then extend to a bounded delay model and last we consider a synchronous model that allows message losses.

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

1 Introduction

Bitcoin [10] is the predominant cryptocurrency today. Nevertheless, our understanding of Bitcoin' s correctness is limited. Only relatively recently, there have been attempts to formally capture Bitcoin' s security properties. In a seminal work, Garay et al. [7] proposed a formal framework (the "backbone protocol") to describe the Bitcoin system. They defined security properties for the backbone protocol and proved these both in the synchronous and bounded-delay model.

Our work extends the work of Garay et al. [7] in several dimensions. First, in contrast to our model, [7] assumed a constant honest majority of the participants. However, the Bitcoin protocol has been proven to be more fault-tolerant and able to allow for a majority of dishonest players, as long as this dishonest majority is only temporary. Specifically, in 2014 there was a majority takeover (approximately 54% of the network) by the mining pool GHash.io. The cost to perform such attacks have been studied in [3] and https://www.crypto51.app/. In this work, we extend the original work of Garay et al. [7] to capture these attacks, by allowing a temporary dishonest majority. We provide a formal analysis and investigate under which circumstances Bitcoin is secure when the honest majority holds only on expectation.

Second, motivated by a model of Pass and Shi [12], we not only have honest ("alert") or dishonest ("corrupted") nodes. Instead, there is a third group of nodes that are currently not able to follow the protocol. We call them "sleepy",

which really is a euphemism for nodes that are basically eclipsed from the action, for instance by a denial of service attack. Understanding this trade-off between corrupted and sleepy nodes gives us a hint whether a dishonest attacker should rather invest in more mining power (to get more corrupted players) or in a distributed denial of service architecture (to get more sleepy players).

Third, we introduce a parameter c, which upper bounds the number of blocks contributed by the adversary. This is not needed to prove security, but as [13] showed, if the adversary follows a selfish mining strategy, it can gain a 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.

Fourth, we study network delays since they 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 sleepy parties heavily depends on the maximum allowed message delay.

Finally, we extend our analysis to a synchronous model, where we allow message losses. This is inspired by an idea described in [8], where the adversary may perform an eclipse attack [14,15] on some victims which enables the adversary to control their view of the blockchain. We show security under the assumption that the adversary can eclipse a certain number of players, depending on the number of corrupted players.

The omitted proofs can be found in the appendix.

2   The Model

We adapt the model originally introduced by Garay et al. [7] to prove the security of the Backbone protocol. We initially present all the components of a general model and then 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.

Round. 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 $H(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^\kappa$ 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 [7]. These properties are defined as predicates over $VIEW_{\mathcal{A},\mathcal{Z}}^{t,n}$, which will hold for all polynomially bounded environments $\mathcal{Z}$ and adversaries $\mathcal{A}$ with high probability.

Definition 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 $Q(VIEW_{\mathcal{A},\mathcal{Z}}^{t,n})$ is false is negligible in κ.*

The following two Definitions concern the liveness and eventual consistency properties of the Backbone protocol. We are using the notation of [7]: We denote a chain C, where the last k blocks are removed, by $C^{\lceil k}$. Further, $C_1 \preceq C_2$ denotes that C_1 is a prefix of C_2.

Definition 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 $VIEW_{\mathcal{A},\mathcal{Z}}^{t,n}$, it holds that for any $s + 1$ rounds, there are at least $\tau \cdot s$ blocks added to the chain of P.* [1]

Definition 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 $VIEW_{\mathcal{A},\mathcal{Z}}^{t,n}$ 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:

---

[1] The Chain-Growth Property in [7] 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 1, one can see, why we use $s + 1$ instead of s. It follows by the fact that the sum in Lemma 13 only goes from $i = r$ to $s - 1$ and not to s.

Definition 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 $VIEW_{\mathcal{A},\mathcal{Z}}^{t,n}$, it holds that for any ℓ consecutive blocks of C the ratio of adversarial blocks is at most μ.*

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. Throughout the paper, $\epsilon \in (0, 1)$ refers to the quality of concentration of random variables in typical executions and η corresponds to the parameter, determining block to round translation.

Definition 5 ([7], 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 6 ([7], Definition 9). *(Typical execution). An execution is $(\epsilon, \eta)-$typical if, for any set S of consecutive rounds with $|S| \geq \eta\kappa$ and any random variable $X(S)$, the following holds:*

a) $(1 - \epsilon)E[X(S)] < X(S) < (1 + \epsilon)E[X(S)]$
b) No insertions, no copies and no predictions occurred.

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

Proof. To prove a), we can simply use a Chernoff bound by arguing that $E[X(S)]$ is in $\Omega(|S|)$. The proof for b) is equivalent to [7], by reducing these events to a collision in one of the hash functions of the Bitcoin backbone protocol. Such collisions only happen with probability $e^{-\Omega(\kappa)}$. □

3   The q-bounded Synchronous Model without Message Loss $M(q, 0, 1)$

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 in [7]. First, we define the success probabilities for the alert and corrupted parties, which are used to prove the relations between them. At the end, we use these results to show the properties of chain growth, common prefix and chain quality.

Following the definition in [7], let a successful round be a round in which at least one honest party solves a PoW. The random variable X_i indicates 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 $E[X_i] = Pr[X_i = 1] = 1 - (1 - p)^{q(n-t)}$.

Lemma 2. *It holds that $\frac{pqE[n_{alert}]}{1+pqE[n_{alert}]} \leq E[X_i] \leq pqE[n_{alert}]$.*

We also adapt the notation of a unique successful round from [7]. 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 indicates 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. *It holds* $E[Y_i] = E[pqn_{alert,i}(1-p)^{q(n_{alert,i}-1)}] \geq E[X_i](1 - E[X_i])$.

The proofs of both Lemma 2 and Lemma 3 can be found in Appendix A.

Let the random variable $Z_{ijk} = 1$ if the adversary obtains a PoW at round i by the jth query of the kth 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:

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

3.1 Temporary Dishonest Majority Assumption

We assume the honest majority assumption holds on expectation. In particular, 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 [7], δ refers to the advantage of the honest parties and ϵ is defined in Definition 6.

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

First of all, by Definition 6 the properties of the typical execution hold for the random variables $X(S), Y(S), Z(S)$, assuming $|S| \geq \eta\kappa$.

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. [2]

Lemma 4. *The following hold for any set S of at least $\eta\kappa$ consecutive rounds in a typical execution.*

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

---

[2] The statement d) uses different factors as [7]. The problem is, that it's even not possible to prove the bounds from [7] with their theorems, lemmas and assumptions.

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

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

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

Next, we prove Bitcoin is secure under temporary dishonest majority in the q-bounded synchronous setting by proving the three properties defined in [7]: *chain growth*, *common prefix* and *chain quality*. The proofs can be found in Appendix B.

4   The semi-Synchronous Model without Message Loss $M(1, \Delta, 1)$

In this section, we extend the previously seen results to the semi-synchronous (bounded delay) model. This means, that we allow Δ^3 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.

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 15 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.

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 stronger events than just uniquely successful rounds. In [7], 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:

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

---

[3] According to Theorem 11 of [12], 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. [4] 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

In this subsection, we prove Bitcoin is secure, i.e. the chain growth, common prefix and chain quality properties hold, for the semi-synchronous model without message loss. We note that the properties of the typical execution apply to the predefined random variables $(X'(S), Y'(S), Z(S))$, given that $|S| \geq \eta\kappa$.

The following lemma corresponds to the semi-synchronous version of Lemma 4. Most of the relations follow the same structure and are similar to prove as in the synchronous model.

Lemma 5. *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) $Z(S) < (1 + \epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1-E[X_i]}|S| \leq c(1 + \epsilon)(1 - \delta)\frac{E[X_i]}{1-E[X_i]}|S|$
d) *Let $S' = \{r, \ldots, r'\}$ with $|S'| \geq \eta\kappa$. For $S = \{r, \ldots, r' + \Delta\}$ 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 $S' = \{r, \ldots, r'\}$ with $|S'| \geq \eta\kappa$. For $S = \{r - \Delta, \ldots, r' + \Delta\}$:*

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

The proof of Lemma 5 as well as the proofs of the security properties can be found in Appendix C.

5   The q-bounded Synchronous Model with Message Loss $M(q, 0, 0)$

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

---

[4] One might notice that our lower bound of δ differs from the lower bound from [7]. 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.

written to the $RECEIVE()$ strings of alert parties, i.e. sleepy parties do not receive messages. 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 15 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 6. *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: $X^*(S) = \sum_{i \in S} X_i^*$.

Lemma 7. *It holds that* $\frac{pqE[n^*_{alert}]}{1+pqE[n^*_{alert}]} \leq E[X_i^*] \leq pqE[n^*_{alert}]$.

Proof. The lemma can be proven using the same argumentation as in the proof for the Lemma 2. □

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 8. *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. □

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 (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 behavior a different model would be necessary (since s cannot be considered constant).

5.2   Security Analysis

For this section, we note that the properties of an typical execution apply for the random variables $X^*(S), Y^*(S)$ and $Z(S)$, given that $|S| \geq \eta\kappa$.

Lemma 9. *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 6, 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. $\square$

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 10. *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 2, 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:

$$\sum_{k=2}^{\infty}(k-1)E[\tilde{X}_i]^2(1-E[\tilde{X}_i])^{k-2}(1-E[X_i^*])^k$$

$$= \frac{E[\tilde{X}_i]^2}{(1 - E[\tilde{X}_i])^2} \cdot \sum_{k=2}^{\infty} (k-1)\left((1 - E[\tilde{X}_i])(1 - E[X_i^*])\right)^k$$

$$= \frac{E[\tilde{X}_i]^2(1 - E[X_i^*])^2}{(E[\tilde{X}_i] + E[X_i^*] - E[\tilde{X}_i]E[X_i^*])^2}$$

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:

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

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

$$\frac{b^2}{(1+a)^2(1-ab)^2(a+ab+b)^2} \le \frac{s}{1-s}$$
$$\Leftrightarrow ab \le (1+a)^2(1-ab)^2(a+ab+b)^2$$

The inequality holds, since $(1+a)^2(1-ab)^2 \ge 1$ and $ab \le (a+ab+b)^2$. $\quad\square$

The lemma below replaces Lemma 15. The possibility to have chains of different length at the same round offers various ways to replace a block from round i, where $Y_i^* = 1$. Thus, we cannot use the same arguments as in Lemma 15.

Lemma 11. *Suppose the kth block B of a chain C was computed at round i, where $Y_i^* = 1$. Then with probability at least $1 - \phi$, the kth block in a chain C' will be B or requires at least one adversarial block to replace B.*

As in the previous sections, the properties of the typical execution hold and executions are typical with high probability, by Lemma 1.

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 12. *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) $Z(S) < (1 + \epsilon)\frac{t}{E[n_{alert}^*]}\frac{E[X_i^*]}{1 - E[X_i^*]}|S| < (1 + \epsilon)\left(c(1 - \delta) - \frac{s}{1-s}\right)\frac{E[X_i^*]}{1 - E[X_i^*]}|S|$
d) *For $\sigma^* = (1 - \epsilon)(1 - E[X_i^*])$:*

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

e)

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

Next, we prove Bitcoin is secure in the synchronous model with message loss. The proof can be found in Appendix D.

6   Security Analysis 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 $\delta = 0.07^5$. 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 $\delta = 0.075$.
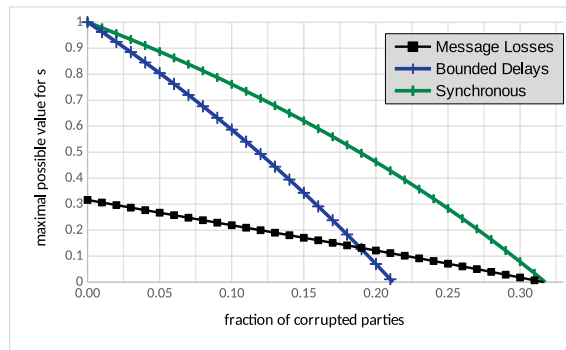
Fig. 1. This figure shows the upper bound on the fraction of sleepy parties, depending on the fraction of corrupted parties.

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 [4], where the upper 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%.

7   Related Work

To model temporary dishonest majority in Bitcoin we used an idea, originally introduced by Pass and Shi [12]. In this work, they introduced the notion of

---

[5] 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.

sleepy nodes, i.e. nodes that go offline during the execution of the protocol, and presented a provably secure consensus protocol. In this paper, we model the dynamic nature of the system by additionally allowing the adversary to set parties to sleep, thus enabling temporary dishonest majority.

Bitcoin has been studied from various aspects and multiple attacks have been proposed, concerning the network layer [8,11,1] as well as the consensus algorithm (mining attacks) [6,13,5,9]. The most famous mining attack is selfish mining [6], where a selfish miner can withhold blocks and gain disproportionate revenue compared to his mining power. The chain quality property, originally introduced in [7], encapsulates this ratio between the mining power and the final percentage of blocks, and thus rewards, the adversary owns. On the other hand, Heilman et al. [8] examined eclipse attacks on the Bitcoin's peer-to-peer network. In turn, Nayak et al. [11] presented a novel attack combining selfish mining and eclipse attacks. They showed that in some adversarial strategies the victims of an eclipse attack can actually benefit from being eclipsed. Our last model, where offline parties do not get the update messages, captures this attack.

8   Conclusion & Future Work

In this paper, we prove Bitcoin is secure under temporary dishonest majority. Specifically, we extended the framework of Garay et al. [7] to incorporate offline nodes and allow the adversary to introduce crush failures. This way we can relax the honest majority assumption and allow temporary dishonest majority. We prove Bitcoin' s security by showing that under an expected honest majority assumption the following security properties hold: chain growth, common prefix and chain quality.

We examine three models: the synchronous model, the bounded delay model and the synchronous model with message loss. The first two models result in similar bounds regarding the fractions of corrupted and sleepy parties. In contrast, the last model that allows message losses when a party goes offline is less resilient to sleepy behavior. This is expected since this model captures the nature of eclipse attacks where the adversary can hide part of the network form an honest party and either waste or use to his advantage the honest party's mining power. We illustrate in Figure 6 the upper bounds on the fraction of sleepy parties depending on the fraction of corrupted parties for all three models.

For future work, we did not consider the bounded delay with message loss model. We expect the difference on the results from synchronous to bounded delay model to be similar to the model without message loss. Another interesting future direction is to consider a more powerful adversary, who knows whether his attempt to set a party to sleep is successful or not.

9   Acknowledgments

References

1. Apostolaki, M., Zohar, A., Vanbever, L.: Hijacking bitcoin: Routing attacks on cryptocurrencies. In: 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017. pp. 375–392 (2017)
2. Bahack, L.: Theoretical bitcoin attacks with less than half of the computational power (draft). In: IACR Cryptology ePrint Archive, Report 2013/868 (Nov 2013)
3. Bonneau, J.: Hostile blockchain takeovers (short paper). In: Bitcoin18: Proceedings of the 5th Workshop on Bitcoin and Blockchain Research (2018)
4. Decker, C., Wattenhofer, R.: Information propagation in the bitcoin network. In: IEEE P2P 2013 Proceedings (Sep 2013)
5. Eyal, I.: The miner's dilemma. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015. pp. 89–103 (2015)
6. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: Commun. ACM (Nov 2013)
7. Garay, J.A., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II. pp. 281–310 (2015)
8. Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on bitcoin's peer-to-peer network. In: Proceedings of the 24th USENIX Conference on Security Symposium. pp. 129–144. SEC'15, USENIX Association, Berkeley, CA, USA (2015)
9. Kwon, Y., Kim, D., Son, Y., Vasserman, E.Y., Kim, Y.: Be selfish and avoid dilemmas: Fork after withholding (FAW) attacks on bitcoin. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. pp. 195–209 (2017)
10. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. In: https://bitcoin.org/bitcoin.pdf (Oct 2008)
11. Nayak, K., Kumar, S., Miller, A., Shi, E.: Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. 2016 IEEE European Symposium on Security and Privacy (EuroSP) pp. 305–320 (2015)
12. Pass, R., Shi, E.: The sleepy model of consensus. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. pp. 380–409. Springer International Publishing, Cham (2017)
13. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in bitcoin. In: Financial Cryptography and Data Security - 20th International Conference, FC 2016, Christ Church, Barbados, February 22-26, 2016, Revised Selected Papers. pp. 515–532 (2016)
14. Singh, A., Ngan, T.W.J., Druschel, P., Wallach, D.S.: Eclipse attacks on overlay networks: Threats and defenses. In: IEEE Infocom 2006 (Apr 2006)
15. Sit, E., Morris, R.: Security considerations for peer-to-peer distributed hash tables. In: Springer, pp. 261269 (Oct 2002)

A   Bounds for (unique) successful rounds

Lemma 2. *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:

$$
\begin{aligned}
E[X_i] &= \sum_{k=0}^{n-t} E[X_i | n_{alert,i} = k] \cdot Pr[n_{alert,i} = k] \\
&= \sum_{k=0}^{n-t} \left(1 - (1-p)^{qk}\right) \cdot \binom{n-t}{k}(1-s)^k s^{n-t-k} \\
&= 1 - \left(s - (s-1)(1-p)^q\right)^{n-t} \geq 1 - \left(s - (s-1)(1-pq)\right)^{n-t} \\
&\geq 1 - e^{-(1-s)(n-t)pq} = \frac{pqE[n_{alert}]}{1 + pqE[n_{alert}]}
\end{aligned}
$$

$\square$

Lemma 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. 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 [7], 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. $\square$

B Bitcoin Security in $M(q, 0, 1)$

In this Section we prove the three properties, as stated in [7], for Bitcoin security under the synchronous model with temporary dishonest majority as defined in 3.1. First, we prove the following lemma for a typical execution, which we later use to prove the desired properties.

Lemma 4. *The following hold for any set S of at least $\eta\kappa$ consecutive rounds in a typical execution.*

a) $(1 - \epsilon)E[X_i]|S| < X(S) < (1 + \epsilon)E[X_i]|S|$
b) $(1 - \epsilon)E[X_i](1 - E[X_i])|S| < Y(S)$
c) $Z(S) < (1 + \epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1 - E[X_i]}|S| \le c(1 + \epsilon)(1 - \delta)\frac{E[X_i]}{1 - E[X_i]}|S|$
d) *For $\sigma = (1 - \epsilon)(1 - E[X_i])$:*

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

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

Proof.
a) Follows directly from Definition 6.
b) By $E[Y_i] \ge E[X_i](1 - E[X_i])$ and Definition 6.
c) The first inequality follows from Definition 6 and the second inequality follows from the honest majority assumption.
d) Let us first prove the first inequality:

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

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

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

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

$$\left(1 + \frac{\delta}{\sigma}\right)\frac{t}{E[n_{alert}]} \le c \cdot \left(1 + \frac{\delta}{\sigma}\right)(1 - \delta) \le c\left(1 + \frac{\delta(\frac{\delta}{2} - \delta)}{\sigma}\right) \le c\left(1 - \frac{\delta^2}{2\sigma}\right)$$

e) To prove $Z(S) < Y(S)$, we apply Lemma 4 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:

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

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

$\square$

B.1 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 13 ([7], 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 [7] in this model.

Theorem 1. *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 13:

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 4, we can derive

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

$\square$

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 14. *In $\eta\kappa$ consecutive rounds of a typical execution, it holds that 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 $X(S) + Z(S)$ using Lemma 4 a) and d) in the following way:

$$X(S)+Z(S) < X(S)\Big(1+c(1-\frac{\delta^2}{2\sigma})\Big) < (1+c)(1+\epsilon)E[X_i]|S| = (1+c)(1+\epsilon)\eta\kappa E[X_i]$$

$\square$

B.2   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. [6]

Lemma 15 ([7], Lemma 6). *Suppose the kth block B of a chain C was computed by an honest party in a uniquely successful round. Then the kth 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 property, which are adapted from [7].

Lemma 16. *Assume a typical execution and consider two chains C_1 and C_2 such that $\text{len}(C_2) \geq \text{len}(C_1)$. 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 [7]. The lemma can be shown by creating a set of rounds S, in which at least k blocks were created. Then, using Lemma 15, 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 14, the properties of a typical execution have to apply for S, but then $Z(S) \geq Y(S)$ contradicts Lemma 4 e). $\square$

---

[6] In order to replace a block, the adversary may use a Block Discarding Attack, described in [2]. Thus, the adversarial block is likely to be a precomputed block, where the adversary performed a block withholding attack (e.g. Selfish Mining), analyzed in [6].

Theorem 2. *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 [7]. 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 $C_1^{\lceil k} \npreceq C_2'$. For the case $r = r_1$, the contradiction can be obtained by Lemma 16. And for $r_1 > r$, let C_1' be the chain, which P_i adopted at round $r - 1$. Using Lemma 16, 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

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

which contradicts the Definition of r.   □

B.3   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. *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 [7]. 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 13. 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 4 d), since by Lemma 14, the rules of a typical execution apply for the set S.   □

As a result of Theorem 3, 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, 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.

C   Bitcoin Security in $M(1, \Delta, 1)$

In this Section, we prove Bitcoin is secure under temporary dishonest majority, as defined in 4.1, in the bounded delay model. We first prove Lemma 5 and later the three security properties.

Lemma 5. *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) $Z(S) < (1 + \epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1-E[X_i]}|S| \le c(1+\epsilon)(1-\delta)\frac{E[X_i]}{1-E[X_i]}|S|$
d) *Let $S' = \{r, \ldots, r'\}$ with $|S'| \ge \eta\kappa$. For $S = \{r, \ldots, r' + \Delta\}$ 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 $S' = \{r, \ldots, r'\}$ with $|S'| \ge \eta\kappa$. For $S = \{r - \Delta, \ldots, r' + \Delta\}$:*

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

Proof.
a) By $(1 - \epsilon)E[X_i](1 - E[X_i])^{\Delta-1}|S| \le (1 - \epsilon)E[X_i']|S| < X'(S)$
b) By $(1 - \epsilon)E[X_i](1 - E[X_i])^{2\Delta-1}|S| \le (1 - \epsilon)E[Y_i']|S| < Y'(S)$
c) Equivalent to Lemma 4 c).
d) By applying a) and c), we only have to prove that the following holds:

$$2\epsilon + \frac{2\Delta}{\eta\kappa} + \Delta E[X_i] \qquad \le \frac{\delta}{2}$$

$$\Leftrightarrow \frac{1 + \epsilon + \frac{2\Delta}{\eta\kappa}}{\sigma'} - 1 \qquad \le \frac{\delta}{2\sigma'}$$

$$\Rightarrow (1 + \epsilon)|S| \qquad \le (1 - \epsilon)(1 - E[X_i])^{\Delta}|S'|\left(1 + \frac{\delta}{2\sigma'}\right)$$

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

$$1 - \delta \qquad\qquad \leq \frac{(1-\epsilon)(1-E[X_i])^{2\Delta}}{c(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right)}$$

$$\Leftrightarrow (1+\epsilon)c(1-\delta)\frac{|S|}{|S'|} \qquad \leq (1-\epsilon)(1-E[X_i])^{2\Delta}$$

$$\Rightarrow (1+\epsilon)\frac{t}{E[n_{alert}]}|S| \qquad \leq (1-\epsilon)(1-E[X_i])^{2\Delta}|S'|$$

Therefore:

$$1 - \frac{(1-\epsilon)(1-E[X_i])^{2\Delta}}{c(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right)} \leq \frac{(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right) - (1-\epsilon)(1-\Delta E[X_i])^2}{(1+\epsilon)\left(1+\frac{2\Delta}{\eta\kappa}\right)}$$

$$\leq 2\epsilon + 2\Delta E[X_i] + \frac{4\Delta}{\eta\kappa}$$

$$\leq \delta$$

$\square$

C.1 Chain growth

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

Lemma 17 ([7], 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 [7], by induction on $s - r - \Delta + 1 \geq 0$. $\square$

Theorem 4. *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 17 instead. $\square$

C.2 Common prefix

The following lemma is the adopted version of Lemma 16.

Lemma 18. *Assume a typical execution and consider two chains C_1 and C_2 at round r, such that $len(C_2) \geq len(C_1)$. For the same conditions of Lemma 16, 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 16, but considering a set $S' = \{i : r^* + \Delta < i < r - \Delta\}$ to contradict $Z(S) < Y'(S')$ from Lemma 5. $\square$

Theorem 5. *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 2. $\square$

C.3   Chain quality

Theorem 6. *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, 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 5 d).   □

D   Bitcoin Security in $M(q, 0, 0)$

In this Section, we prove Bitcoin is secure under temporary dishonest majority, as defined in 5.1, in the synchronous setting with message losses. We first prove Lemma 11 regarding block replacement, then we prove the properties for a typical execution in this setting, as stated in Lemma 12, and lastly the three security properties.

Lemma 11. *Suppose the kth block B of a chain C was computed at round i, where $Y_i^* = 1$. Then with probability at least $1 - \phi$, the kth 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 10, we know that the case 4) only happens with probability ϕ.   □

Lemma 12. *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) $Z(S) < (1 + \epsilon)\frac{t}{E[n_{alert}^*]}\frac{E[X_i^*]}{1 - E[X_i^*]}|S| < (1 + \epsilon)\left(c(1 - \delta) - \frac{s}{1-s}\right)\frac{E[X_i^*]}{1 - E[X_i^*]}|S|$

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

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

e)

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

Proof.
a) Follows directly from Definition 6.
b) By $E[Y_i^*] \ge E[X_i^*](1 - E[X_i^*])$ and Definition 6.
c) The first inequality follows from Definition 6 and the second inequality follows from the honest majority assumption.
d) Let us first prove the first inequality:

$$
\begin{aligned}
Z(S) &< (1 + \epsilon)E[Z(S)] &&\text{By Definition 6 c)}\\
&\le (1 + \epsilon)\frac{t}{E[n_{alert}^*]}\frac{E[X_i^*]}{1 - E[X_i^*]}|S| &&\text{upper-bound for } E[Z_i]\\
&< (1 + \epsilon)\frac{t}{E[n_{alert}^*]}\frac{1}{1 - E[X_i^*]}\frac{1}{1 - \epsilon}X^*(S) &&\text{By a)}
\end{aligned}
$$

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

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

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

$$
\begin{aligned}
1 - \sigma^* &\le \frac{\delta}{2}\\
\Rightarrow \left(1 + \frac{\delta}{\sigma^*}\right)(1 - \delta) &\le 1 - \frac{\delta^2}{2\sigma^*}\\
\Rightarrow c\left(1 + \frac{\delta}{\sigma^*}\right)\left((1 - \delta) - \frac{s}{1 - s}\right) &\le c\left(1 - \frac{\delta^2}{2\sigma^*}\right)
\end{aligned}
$$

which follows from the Definition of σ^* and δ.
e) The proof goes straight forward, using parts b) and c):

$$
\begin{aligned}
3\epsilon + 2E[X_i^*] &\le \delta\\
\Rightarrow 3\epsilon + 2E[X_i^*] + \phi - \frac{s}{1 - s} &\le \delta\\
\Rightarrow (1 + \epsilon)\left(c(1 - \delta) - \frac{s}{1 - s}\right) &\le (1 - \epsilon)^2(1 - \phi)(1 - 2E[X_i^*] + E[X_i^*]^2)
\end{aligned}
$$

$$\Rightarrow (1+\epsilon)\frac{t}{E[n^*_{alert}]} \qquad\qquad \leq (1-\epsilon)^2(1-E[X_i^*])^2(1-\phi)$$

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

where the last step follows from Lemma 10 and the inequality holds according to the Definition of δ.

$\square$

D.1   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 7. *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 17 instead. $\square$

D.2   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 19. *The Lemma 16 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 11, it holds that an expected fraction of $1 - \phi$ block replacements involve at least one adversarial block. Using Chernoff, we can argue that the number 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 14, the properties of a typical execution have to apply for S, but then $Z(S) > Y^*(S)(1-\epsilon)(1-\phi)$ contradicts Lemma 12 e). $\square$

Theorem 8. *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 19 instead. $\square$

D.3   Chain-Quality

Theorem 9. *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 9, 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 12 d). $\qquad\square$