SCIENCE CHINA
Information Sciences

March 2020, Vol. 63 130105:1–130105:14
https://doi.org/10.1007/s11432-019-2698-1

• RESEARCH PAPER •
Special Focus on Security and Privacy in Blockchain-based Applications

Analysis of bitcoin backbone protocol in the non-flat model

Peifang NI[1,2,3], Hongda LI[1,2,3]* & Dongxue PAN[1,2,3]

[1]*State Key Laboratory of Information Security, Institute of Information Engineering,*
Chinese Academy of Sciences, Beijing 100093*, China;*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[3]*Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing* 100093*, China*

Received 15 June 2019/Revised 21 August 2019/Accepted 30 October 2019/Published online 11 February 2020

Abstract Owing to the novel proof-of-work based consensus algorithm, bitcoin has been the most successful decentralized cryptocurrency so far. In bitcoin system, parties (miners) compete to create blocks by doing publicly verifiable proofs of sequential work (proof-of-work) and the probability that a party wins the competition is proportional to the amount of computational power that he has invested. Note that its security holds under honest majority assumption in terms of the amount of computational power. In this paper, we provide the formal analysis of bitcoin backbone protocol in the non-flat model. Precisely, we rethink and redefine the model of computing puzzles to capture the real-world protocol execution, where each party owns different amount of computational power and does sequential computations towards a puzzle independently. Fortunately, our work obtains the better results in analyzing the security of bitcoin backbone protocol, which can reflect the real-world protocol execution better, without any additional assumptions but the honest majority assumption. Finally, we show that a robust public transaction ledger can be built on top of bitcoin backbone protocol in our model securely.

Keywords bitcoin system, proof-of-work, computational power, non-flat model, transaction ledger

Citation Ni P F, Li H D, Pan D X. Analysis of bitcoin backbone protocol in the non-flat model. Sci China Inf Sci, 2020, 63(3): 130105, https://doi.org/10.1007/s11432-019-2698-1

1 Introduction

Bitcoin system [1] is the first fully decentralized public transaction ledger (blockchain) maintained and extended by the parties via a consensus protocol under honest majority assumption. In Nakamoto's rigorous design, the selection of winner is implemented by solving a computational puzzle, which is a moderately hard hash inequality [2, 3]. And the opportunity of a party to be winner is related to the amount of computational power that he has invested. For achieving consistency among the honest parties, a final confirmed block is the one that has been at a deep enough position in the honest parties' local chains. For making the chain grow at a stable rate in the permissionless setting, it introduces a difficult target recalculation mechanism.

The core of bitcoin system, bitcoin backbone protocol, has been extracted and analyzed in the recent studies [4–6]. However, these analyzes are in the flat model, where all parties hold equal amount of computational power that they are allowed to make the same number of parallel queries to a hash function modeled as a random oracle per round (during which the computational power of each party is used up and then the round is increased by one). This is inconsistent with the real-world protocol execution, where

* Corresponding author (email: lihongda@iie.ac.cn)

each party owns different amount of computational power and does sequential computations towards a puzzle independently.

Although, it has been stressed that the non-flat model can be achieved by clustering several flat model parties into a virtual entity that with higher computational power [4–6]. However, this method is not precise to denote a real-world party's hashing power. Indeed, in a given period of time (a round), an honest party will not stop mining until he spends all of the computational power out and this means that the honest chains (the chains kept by the honest parties) may be increased by more than one blocks in one round. Notice that the hashing power of a party with computational power m is denoted as $1-(1-p(k))^m$ [4], which means that the party does m parallel computations towards a puzzle and his local chain can be extended with at most one block even if he succeeds more than one times in one round ($p(k)$ is the mining hardness function with security parameter k). That contradicts to the fact that he should do m sequential computations. Furthermore, as described in [4], this method brings an asymmetry that while the honest parties will not create more than one valid block per round, the adversary may use all its computational power and potentially compute more than one valid blocks. So this method in [4–6] reduces the hashing power of honest parties and has to depend on an additional strong assumption that $\alpha' \geqslant \gamma' > (1+\delta)\lambda\beta'$ (parameters $\delta > 0, \lambda \geqslant 1$), which means that, in one round, the probability that at least one honest party succeeds (α') is greater than the probability (γ') that exactly one honest party succeeds, which is greater than that of all the corrupted parties (β'). More formally, it not only requires that $n > 2t$ (the honest majority assumption), but also requires that $n - p(n-t)^2 > ((1+\delta)\ +1)t$, where n is the number of parties and t of them is controlled by the adversary.

1.1 Our contributions

In this paper, based on the model in [4], we formally discuss bitcoin backbone protocol in the non-flat model, where each party owns different amount of computational power. In our model, the parties share a synchronous communication network and the (fully) adaptive adversary controls a subset of parties. A party P_i holds computational power C_i implies that he can make C_i sequential computations towards a puzzle per round. The ratio of overall computational power controlled by the honest parties is $\frac{\lambda'}{1+\lambda'}$ ($\lambda' \in (1, +\infty)$) that captures the honest majority assumption.

It is true that an honest party with higher computational power can be treated as several clustered flat model parties as the adversary does. But the clustered parties should work together like the corrupted parties (controlled by the adversary) [4–6], rather than being independent with each other. For the above motivation, the main step of our work is to explore the real-world protocol execution deeply and give a more accurate estimate of the honest parties' hashing power.

In our model, to capture the real-world protocol execution, the amount of computational power is measured by the number of computations towards the puzzles per round. Now we can formally describe the bitcoin backbone protocol that consists of three algorithms called chain validation, chain comparison and proof-of-work. The definitions of these three algorithms are similar to that in [4], except that (1) in chain validation algorithm, the parameter ctr in each block records the number of computations towards this block and its validation is related to the computational power of its creator, (2) in chain comparison algorithm, the local best chain is the one that consists of the most amount of computational power among the set of valid chains, and (3) in proof-of-work algorithm, each party extends local chain independently by computing a hash function and continues this process until his computational power is used up.

Furthermore, considering the computational power that has been invested in blocks, we redefine the security properties. Chain growth means that the computational power of the honest parties' local chains should grow linearly to the number of rounds. Chain quality treats the parties' contributions as the computational power that they have invested in the chains held by the honest parties. Common prefix argues that the chains held by the honest parties enjoy the ever-growing computational power and share a common part.

As the first step of our analysis, we specify the environment of protocol execution. To avoid considering the variety of overall computational power in the permissionless setting, we divide protocol execution into some defined sets named standard execution with proper size, where the collective contributions of the honest parties and adversary do not deviate too much from their expectations, no lucky parties and no bad events occur with respect to the hash functions. Further, we prove that, with overwhelming probability, almost all polynomially bounded (with security parameter k) executions are standard and the chains held by the honest parties enjoy the three defined fundamental security properties.

The non-flat model allows us to describe the hashing power of the honest parties accurately. The corrupted parties controlled by the adversary make sequential computations towards the puzzles and extend local chains jointly, which is consistent with the models of [4–6]. And we realize that each honest party should be treated as the adversary. Concretely, after each computation, each party either updates local state, broadcasts the newly-mined block and continues to search the next answer or updates the local state and continues to extend it, and repeats this process until his computational power is used up. As a result, we get the expected contributions of the honest parties rather than the probability of success in one round.

Main results. Putting all the above together, we succeed in analyzing bitcoin backbone protocol in the non-flat model under honest majority assumption. Concretely, we give a more precise description of the honest parties' mining process, which is closer to the real-world protocol execution than the related studies [4–6]. As a result, the expectation of the honest parties' contributions (α) becomes bigger and the adversary's contributions (β) remain unchanged. So we obtain the higher chain growth rate $g = (1 - \varphi)\alpha$ (parameter $\varphi \in (0, 1)$) and chain quality $\mu = (1 - \delta)\frac{1}{\lambda'}$ (parameter $\delta \in (0, 1)$), and the faster consensus among the honest parties. We stress that the honest majority assumption ($\lambda' > 1$) is necessary or μ will be bigger than 1, which means that the best chain in the network is controlled by the adversary.

For the application, we prove that a robust public transaction ledger, with security properties as persistence and liveness, can be built upon any blockchain protocol, with security properties as chain growth, chain quality and common prefix, in our non-flat model successfully.

1.2 Related work

The peer-to-peer system leads us to the permissionless setting, where messages are delivered via an unauthenticated network and parties can join or leave freely, such as the designed systems [7,8]. Obviously, it is crucial that these systems are designed to be secure against the adversarial behaviors. Byzantine agreement protocol tolerates a certain ratio of adversary and gets consensus among a fixed small group, such as PBFT [9], and some follow-on studies consider the performance and scalability [10,11]. However, in the permissionless setting, the above systems cannot resist against the sybil attack in that the adversary can spawn lots of parties and control majority of parties easily. Nakamoto proposed the first fully decentralized system [1] under honest majority assumption. Note that honest majority means that more than half of overall computational power in the network are controlled by the honest parties.

Considering the security of blockchain protocol, Nakamoto [1] provided initial arguments about preventing double-spending attack. Ref. [12] analyzes how bitcoin system uses broadcast mechanism to propagate transactions and blocks in the network. Ref. [13] extends Nakamoto's analysis to deal with (bounded) delays.

Recent studies [4–6, 14] have focused on the analysis of bitcoin backbone protocol, and three fundamental security properties of blockchain protocol, chain growth, chain quality and common prefix, are well defined. However, they only consider the flat model, where all parties are equal to hold the same amount of computational power. Bitcoin backbone protocol is extracted and analyzed formally in the static setting [4], and then proved in an asynchronous network [5], where the adversarial delay is a small prior bounded delay with $\Delta < \frac{1}{np}$. Ref. [14] proceeds to prove that blockchain is secure against the long delays with $\Delta > \frac{1}{np}$. Ref. [6] considers the recalculation of difficult target in the permissionless setting.

Additionally, on the other hand, the bitcoin mining protocol has been proved to be incentive-incompatible [15, 16]. Namely, the best strategy for rational parties is withholding their solutions for some period of time, rather than announcing them immediately, which leads to parties coalitions and reduces system decentralization. Refs. [17, 18] aim to design protocols, where the mining process is more collaborative and the parties are encouraged to solve puzzles together rather than compete. Owing to the fact that proof-of-work based protocols execute securely at the cost of consuming a huge amount of non-recyclable physical resources, another line of research intends to propose the resource-friendly protocols, such as the proof-of-stake based ones [19, 20].

Chaum et al. [21] proposed the first e-cash system, where a central bank is responsible for issuing and withdrawing coins, and a number of studies optimize it for better performance such as [22–24]. With the popularity of bitcoin, a multitude of cryptocurrency systems are proposed [25, 26] and actually, they are based on the same consensus protocol—the bitcoin backbone protocol.

1.3 Outline of the paper

The rest of this paper is organized as follows. In Section 2, we give preliminaries. In Section 3, a high overview of our analysis is presented. In Section 4, we show the formal security analysis of bitcoin backbone protocol and the comparison among the related studies. A concrete application of blockchain and the conclusion are presented in Sections 5 and 6, respectively.

2 Models and definitions

The formal cryptographic models of blockchain protocol that follow Canetti's formulation of real-world protocol execution [27, 28] have been studied in [4–6]. In this section, we extend their models to be non-flat model, where the amount of computational power held by each party corresponds to the real-world protocol execution.

2.1 Models

Employing elements from [27, 28], protocol executes in a multiparty setting and is driven by an environment program $\mathcal{Z}$. The adversary $\mathcal{A}$ can spawn and corrupt parties at any time adaptively, reorder messages and spoof the source of messages. We introduce two functionalities $\mathcal{F}_{\mathrm{NET}}$ and $\mathcal{F}_{\mathrm{MIN}}$ to describe the communication between parties and the process of mining new blocks in the course of protocol execution.

Communication model. The parties share a synchronous communication channel and messages delivery is achieved by a diffusion mechanism $\mathcal{F}_{\mathrm{NET}}$. $\mathcal{F}_{\mathrm{NET}}$ works in round and each party is allowed to get messages from $\mathcal{F}_{\mathrm{NET}}$ at any time. Formally, when $\mathcal{F}_{\mathrm{NET}}$ receives an instruction to diffuse a message m from party P_i, it inspects and sends m to the other parties, and then decides if P_i completes this round. At any time, $\mathcal{A}$ is allowed to get messages and specify the contents of the honest parties' receiving tapes. When all the parties complete the current round, $\mathcal{F}_{\mathrm{NET}}$ increases round by one.

Mining model. In order to describe the mining process of real-world protocol, we assume that each party has access to a functionality $\mathcal{F}_{\mathrm{MIN}}$ (Figure 1). Formally, $\mathcal{F}_{\mathrm{MIN}}$ maintains a local list $\mathcal{H}$ and a party P_i with computational power C_i is allowed to make C_i queries to $\mathcal{F}_{\mathrm{MIN}}$. For each query, P_i can get the answer with probability p. Note that at each query, $\mathcal{F}_{\mathrm{MIN}}$ returns the successful or failed message to P_i and waits for the next query.

2.2 Blockchain protocol in the non-flat model

Blockchain. We use the notations similar to those in [4] to describe the blockchain protocol and show how to decide the amount of computational power that a party has invested.

Blockchain is a sequence of blocks connected by hash values. The rightmost block denotes the head of chain $\mathrm{head}(\mathcal{C}) = (h, x, \mathrm{ctr}, \mathrm{TS})$ and each party extends chain $\mathcal{C}$ by producing a valid block as $B = (h', x', \mathrm{ctr}', \mathrm{TS}')$, where h' is the hash of $\mathrm{head}(\mathcal{C})$, x' is records (transactions), $\mathrm{ctr}' \in \mathbb{N}$ records the number of queries that a party has done towards this block and TS' is timestamp. Block B is valid if $(H(\mathrm{ctr}', G(h', x')) < T) \wedge (\mathrm{ctr}' \leqslant C)$, where $H(\cdot)$ and $G(\cdot)$ are hash functions with outputs in $\{0,1\}^k$, T is the current difficult target and C denotes the computational power of $B's$ creator.

FUNCTIONALITY $\mathcal{F}_{\text{MIN}}$

1. Setup.
   (a) The adversary $\mathcal{A}$ with computational power C_a and honest party P_i with computational power C_i, where $i \in \{1, \ldots, n_h^r\}$ and $\sum_{i=1}^{n_h^r} C_i > C_a$ (n_h^r is the number of honest parties at round r).
   (b) Security parameter k and PoW parameter p.
2. Upon receiving (Mining, h, C_i, P_i, r) from P_i, where $h \in \{0,1\}^k$, record $(P_i, C_i, b = 0, r)$ and proceed as follows. (P_i can make C_i sequential queries in this round).
   - **Stage 1**

   (a) If $b \leqslant C_i$,
   - with probability p, selects a pair (w, h'), where $w, h' \in \{0,1\}^k$, w is answer and $h' < T$ (T is the current difficult target). Set $b = b+1$ and $\mathcal{H} = \mathcal{H} \cup \{(w, h')\}$, update (P_i, C_i, b, r) and $\mathcal{H}$, send (Mining, $(w, h'), b, r$) to P_i and turns to Stage 2. (P_i succeeds to create a block);
   - with probability $1 - p$, set $b = b+1$, return (Mining, $\perp, b, r$) to P_i and turn to Stage 2. (P_i fails to create a block).

   (b) Else, abort and return (Mining, $\perp$) to P_i. (P_i fails to create a block).
   - **Stage 2**

   (a) Upon receiving (Mining, h'', b, P_i, r) from P_i at round r, turn to Stage 1. (P_i continues to mine blocks).
3. Upon receiving (Verify, w, h) from P_j.
   - If $(w, h) \in \mathcal{H}$, then return 1.
   - Otherwise, return 0.

Figure 1 Mining functionality $\mathcal{F}_{\text{MIN}}$.

Notations. The chain is denoted as $\mathcal{C}$. len$(\mathcal{C})$ is the length of $\mathcal{C}$ and $\mathcal{C}^{\lceil K}$ denotes the result by pruning the K rightmost blocks for that $K \in \mathbb{N}$ and $K < \text{len}(\mathcal{C})$; if $K \geqslant \text{len}(\mathcal{C})$, then $\mathcal{C}^{\lceil K} = \epsilon$ is an empty string. $\mathcal{C}_1 \preceq \mathcal{C}_2$ means that $\mathcal{C}_1$ is a prefix of $\mathcal{C}_2$. We use $\mathcal{C} := \mathcal{C}B$ to denote that B extends $\mathcal{C}$. $\text{Com}(B) = \text{ctr}$ and $\text{Com}(\mathcal{C}) = \sum_{i=1}^{\text{len}(\mathcal{C})} \text{Com}(B_i)$ denote the amount of computational power in B and $\mathcal{C}$, respectively. We use (different) computational power to denote (different) amount of computational power and honest chain to denote the chain kept by an honest party when the context is clear.

The bitcoin backbone protocol Π. The bitcoin backbone protocol is executed by arbitrary number of parties over an unauthenticated network. Each party tries to create a valid block by solving a computational puzzle independently. Here, we avoid specifying the type of values that the parties try to insert into blocks and the type of chain validation they preform (beyond checking for its structural properties with respect to the hash functions $G(\cdot), H(\cdot)$), and the way they interpret the chain. These can be handled by three external functions as $V(\cdot), I(\cdot)$ and $R(\cdot)$, which are specified by the application that runs on top of Π. Bitcoin backbone protocol Π is supported by three algorithms called chain validation, chain comparison and proof-of-work, and the detailed description is shown in Appendix A.

2.3 Desirable security properties

Three fundamental security properties have been well defined as chain growth, chain quality and common prefix [4,5,29]. Before giving our redefinitions, we give some useful terms.

Definition 1 (Round)**.** Round is the elementary unit of protocol execution. In our model, all parties complete the current round means that each party has used up the corresponding number of queries to $\mathcal{F}_{\text{MIN}}$ and then the current round is increased by one.

Definition 2. $\{\text{VIEW}_{\Pi,\mathcal{A},\mathcal{Z}}^{P,\text{Com}_h,\text{Com}_a}(z)\}_{z\in\{0,1\}^*}$ is a random variable ensemble that describes the view of party P in protocol Π under the environment $\mathcal{Z}$ and adversary $\mathcal{A}$, with input $z \in \{0,1\}^*$. $\text{Com}_h = \{\text{Com}_h^r\}_{r\in\mathbb{N}}$ and $\text{Com}_a = \{\text{Com}_a^r\}_{r\in\mathbb{N}}$ are computational power held by the honest and corrupted parties activated by $\mathcal{Z}$ at round r, respectively. We use $\text{VIEW}_{\Pi,\mathcal{A},\mathcal{Z}}^{P,\text{Com}_h,\text{Com}_a}$ for short.

Table 1 The parameters and symbols used in our analysis$^{a)}$

Parameter	Description
k	Security parameter
λ	The proportion of honest parties in number
λ'	The proportion of honest parties in the amount of computational power
δ	The advantage of honest parties, ($\lambda' > \frac{1}{1-\delta}$)
(Δ, s)	Determines how the amount of computational power fluctuates across rounds
ε	Determines the amount of computational power in a valid block
φ	The distance between variable and expectation in standard execution
K'	The number of consecutive blocks for recalculating difficult target
t	The number of consecutive rounds for chain-growth property
l	The number of consecutive blocks for chain-quality property
K	The number of consecutive blocks for common-prefix property

a) k, s, K', t, l, K are positive integers and $\lambda, \lambda', \delta, \Delta, \varepsilon, \varphi$ are positive reals, where $\lambda \in (0, +\infty)$, $\lambda' \in (1, +\infty)$, $\delta, \varepsilon, \varphi \in (0, 1)$, $\Delta \in [1, +\infty)$ and $0 < \varepsilon + \varphi < 1 - \delta < 1$.

Definition 3 (Chain growth property)**.** Consider protocol Π, chain growth property states that for any two honest parties P_1 and P_2 with $\mathrm{VIEW}_{\Pi,\mathcal{A},\mathcal{Z}}^{P_1,\mathrm{Com}_h,\mathrm{Com}_a} = \mathcal{C}_1$ and $\mathrm{VIEW}_{\Pi,\mathcal{A},\mathcal{Z}}^{P_2,\mathrm{Com}_h,\mathrm{Com}_a} = \mathcal{C}_2$ at round r_1 and r_2, respectively, it holds that $\mathrm{Com}(\mathcal{C}_2) - \mathrm{Com}(\mathcal{C}_1) \geqslant g \cdot t$, where $t = r_2 - r_1 \geqslant 0$ and g is the lower bound of chain growth rate.

Definition 4 (Chain quality property)**.** Consider protocol Π, chain quality property with parameters $\mu \in (0, 1)$ and $l \in \mathbb{N}$ states that for any honest party P with $\mathrm{VIEW}_{\Pi,\mathcal{A},\mathcal{Z}}^{P,\mathrm{Com}_h,\mathrm{Com}_a} = \mathcal{C}$ at round r and any large enough l consecutive blocks of $\mathcal{C}$ with computational power C, the total computational power contributed by the adversary $\mathcal{A}$ is at most $\mu \cdot C$.

Definition 5 (Common prefix property)**.** Consider protocol Π, common prefix property with parameter $K \in \mathbb{N}$ states that for any honest party P_1 with $\mathrm{VIEW}_{\Pi,\mathcal{A},\mathcal{Z}}^{P_1,\mathrm{Com}_h,\mathrm{Com}_a} = \mathcal{C}_1$ at round r_1 and P_2 with $\mathrm{VIEW}_{\Pi,\mathcal{A},Z}^{P_2,\mathrm{Com}_h,\mathrm{Com}_a} = \mathcal{C}_2$ at round r_2, where P_1 and P_2 can be the same party and $r_2 - r_1 \geqslant 0$, it holds that $\mathrm{Com}(\mathcal{C}_2) \geqslant \mathrm{Com}(\mathcal{C}_1)$ and $\mathcal{C}_1^{\lceil \mathrm{K}} \preceq \mathcal{C}_2$.

2.4 Application: robust public transaction ledger

A robust transaction ledger Π_{TL} is a protocol maintaining a ledger $\mathcal{L}$ of transactions ordered in the chain $\mathcal{C}$. Informally, during protocol execution, the winner collects a set of valid transactions as $x = \{\mathrm{tx}_1, \ldots, \mathrm{tx}_n\}$ from the network that is to be packed into the newly-mined block and then broadcasts the corresponding block. The other parties validate the received blocks and choose the valid one to extend local chains. As described in [4], $\mathcal{L}$ servers as an immutable bulletin board that each party has opportunity to post messages and everyone can read all the messages. The bulletin board should satisfy two security properties as persistence and liveness.

Definition 6 (Persistence)**.** With parameter $K \in \mathbb{N}$, if a block that contains transaction tx is at least K blocks away from the end of a ledger broadcasted by an honest party, then tx is stable and in the same position of the honest parties' local ledgers.

Definition 7 (Liveness)**.** With parameters $\omega, K \in \mathbb{N}$, if a valid transaction is given as an input to the honest parties continuously for ω consecutive rounds, then the honest parties will broadcast it in a block that more than K blocks away from the end of their ledgers.

3 Overview of the analysis

In this section, based on the parameters and symbols in Table 1, we give a high overview of our analysis. Precisely, we divide protocol execution into some disjoint sets of consecutive rounds that with defined characteristics and analyze the three security properties with respect to a set.

Notations. Protocol execution is denoted as E, which is a set of consecutive rounds. The number of parties is $n^r = n_h^r + n_a^r$ at round r, where n_h^r and n_a^r denote the number of honest and corrupted parties, and $\frac{n_h^r}{n_a^r} = \lambda > 0$. Party P_i with computational power C_i. $\mathrm{Com}^r = \mathrm{Com}_h^r + \mathrm{Com}_a^r$ denotes the total computational power in the network at round r, where Com_h^r and Com_a^r are held by the honest and corrupted parties, respectively. For honest majority assumption, we have $\frac{\mathrm{Com}_h^r}{\mathrm{Com}_a^r} = \lambda' > \frac{1}{1-\delta}$. $p = \frac{T}{2^k}$ is successful probability of each query to $\mathcal{F}_{\mathrm{MIN}}$. We use $\mathrm{negl}(\cdot)$ to denote the negligible function, whose output is smaller than the inverse of any polynomial $\mathrm{poly}(\cdot)$.

Definition 8 ((Δ, s)-respecting environment)**.** For any set of consecutive rounds S and $|\mathcal{S}| \leqslant s$, a sequence $\{\mathrm{Com}^r\}_{r\in\mathcal{S}}$ is (Δ, s)-respecting if it satisfies that $\max_{r\in\mathcal{S}}\{\mathrm{Com}^r\} \leqslant \Delta \min_{r\in\mathcal{S}}\{\mathrm{Com}^r\}$.

Definition 9 (Valid block)**.** The block B is valid if $(1-\varepsilon)\frac{1}{p} \leqslant \mathrm{Com}(B) \leqslant (1+\varepsilon)\frac{1}{p}$, where $\varepsilon \in (0,1)$.

Definition 10 (Good round)**.** The round r is good if there are no invalid blocks created during r.

The following lemma tells us that there is no lucky party that can create much more blocks than expectation.

Lemma 1. Let E be an execution, then during round $r \in E$, with probability $\mathrm{negl}(k)$, there is a lucky party can create a block B with $\mathrm{Com}(B) < (1-\varepsilon)\frac{1}{p}$.

Proof. Suppose that party P extends chain $\mathcal{C}$ to $\mathcal{C}'$ with block B at round r successfully and $\mathrm{Com}(\mathcal{C}') = \mathrm{Com}(\mathcal{C}) + L$. Let $\mathcal{J} = \{1, 2, \ldots, L\}$ be the index set that P has queried. Then we have $|\mathcal{J}| < (1-\varepsilon)\frac{1}{p}$ that implies B is a guessed block, which occurs with probability exponentially small in k.

Corollary 1. Let E be an execution, then during round $r \in E$, with probability $\mathrm{negl}(k)$, the number of blocks created by the adversary is more than $p \times \mathrm{Com}_a^r$.

Proof. Suppose-towards a contradiction-that, for some round $r \in E$, the number of adversary blocks is $M > \frac{p\times\mathrm{Com}_a^r}{1-\varepsilon}$, which implies that at least one block B with computational power smaller than $(1-\varepsilon)\frac{1}{p}$. Following from Lemma 1 B is a guessed block.

Proposition 1. With overwhelming probability, almost all rounds are good.

Proof. By Lemma 1, for any round r, with overwhelming probability, the computational power invested in each block is at least $(1-\varepsilon)\frac{1}{p}$. Note that the adversary may create a block with computational power much more than $(1+\varepsilon)\frac{1}{p}$ by gathering all the computational power and this block will be discarded by the honest parties.

The following lemma shows that, in our model, choosing the valid chain with the most amount of computational power is equal to choosing the longest valid chain in the network.

Lemma 2. Let E be an execution, an honest party P with local best chain $\mathcal{C}$ at round $r \in E$. Then, with overwhelming probability, $\mathcal{C}$ is the longest valid chain in the network.

Proof. Suppose-towards a contradiction-that, there is a valid chain $\mathcal{C}'$ that $\mathrm{len}(\mathcal{C}^{'}) > \mathrm{len}(\mathcal{C})$ and $\mathrm{Com}(\mathcal{C}') < \mathrm{Com}(\mathcal{C})$. Based on the definition of valid block, there is at least one block $B \in \mathcal{C}'$ with $\mathrm{Com}(B) < (1-\varepsilon)\frac{1}{p}$, which occurs with probability $\mathrm{negl}(k)$, so that $\mathcal{C}'$ is invalid.

Two main parameters. In our model, it is direct to quantify the parties' contributions as computational power that they have invested. For the honest parties, they try to extend local best chains independently, so that, at each query, the honest chains can be increased by at most one block. During one round, each party queries $\mathcal{F}_{\mathrm{MIN}}$ sequentially to extend local chain, so it is potential that the honest chains are increased by more than one blocks in the synchronous network. For the adversary, he makes sequential queries to $\mathcal{F}_{\mathrm{MIN}}$ in one round. Given above, we show two main parameters α, β that denote the contributions of the honest and corrupted parties per round, respectively.

Without loss of generality, let $\{(P_i, C_i) : i \in \{1, 2, \ldots, n_h^r\}\}$ be the set of honest parties with corresponding computational power at round r and $c = \max\{C_i : i \in \{1, 2, \ldots, n_h^r\}\}$. For $j \in \{1, 2, \ldots, c\}$, we define two variables X_j^r and $C_{h,j}^r$ as follows. At round r, if at leat one honest parties succeed at the j-th query, then $X_j^r = 1$, otherwise $X_j^r = 0$. If $X_j^r = 1$, then set $C_{h,j}^r = c_{h,j}^r = \max\{\mathrm{Com}(B_{j,s}^r) : 1 \leqslant s \leqslant n_{h,j}^r\}$, where s is the index of blocks created at the j-th query and $n_{h,j}^r$ is the total number of honest parties of the j-th query; otherwise, $C_{h,j}^r = 0$. For an execution E, let $X(E) = \sum_{r\in E}\sum_{j=1}^{c} C_{h,j}^r$. With respect to the adversary, we define the similar variables Y_j^r and $C_{a,j}^r$, and let $Y(E) = \sum_{r\in E}\sum_{j=1}^{\mathrm{Com}_a^r} C_{a,j}^r$.

(1) $\alpha = \sum_{j=1}^{c} C_{h,j}^{r} = \sum_{j=1}^{c} \Pr[X_j^r = 1] \cdot c_{h,j}^{r}$. It is the expected computational power contributed by the honest parties at round r.

(2) $\beta = \sum_{j=1}^{\mathrm{Com}_a^r} C_{a,j}^{r} = \sum_{j=1}^{\mathrm{Com}_a^r} \Pr[Y_j^r = 1] \cdot c_{a,j}^{r}$. It is the expected computational power contributed by the corrupted parties at round r.

Note that α and β capture the essence that each honest party does sequential computations towards a puzzle and extends local chain, independently. However, the corrupted parties can collude and make full use of their computational power. For convenience, we consider the following bounds:

$$\mathbb{E}[C_h^r] = \alpha = \sum_{j=1}^{c} (1-(1-p)^{n_{h,j}^r}) \cdot c_{h,j}^r \geqslant \sum_{j=1}^{c} \frac{pn_{h,j}^r}{1+pn_{h,j}^r} \cdot c_{h,j}^r \geqslant \frac{1-\varepsilon}{1+pc'} \cdot \mathrm{Com}_h^r, \tag{1}$$

where the last inequality follows from the definition of valid block, $c' = \max\{n_{h,j}^r, 1 \leqslant j \leqslant c\}$ and $\sum_{j=1}^{c} n_{h,j}^r = \mathrm{Com}_h^r$. For the adversary, we have

$$\mathbb{E}[C_a^r] = \beta = \sum_{j=1}^{\mathrm{Com}_a^r} p \cdot c_{a,j}^r \leqslant (1+\varepsilon) \cdot \mathrm{Com}_a^r = \frac{1+\varepsilon}{\lambda'} \cdot \mathrm{Com}_h^r \leqslant \frac{1+\varepsilon}{1-\varepsilon} \cdot \frac{1+pc'}{\lambda'} \alpha, \tag{2}$$

where the third equality follows from the relation between Com_h^r and Com_a^r, and the last inequality follows from the lower bound of α.

Standard execution. This notion divides protocol execution into some sets of consecutive rounds with specific characteristics. During a given execution E, the parties perform Bernoulli trials and succeed with fixed probability, so that the valid blocks with fixed computational power in expectation. In this way, we consider an execution E is standard as the following definition.

Definition 11 (Standard execution)**.** An execution E in (Δ, s)-respecting environment is standard, if, for any set of consecutive rounds $S \subseteq E$, the followings are satisfied:

(a) $(1-\epsilon)\frac{K}{p}/\alpha \leqslant |S| \leqslant |E| \leqslant (1+\epsilon)\frac{K'}{p}/(\alpha+\beta)$ ($K = 6$ and $K' = 2016$ blocks in bitcoin system);

(b) each round $r \in S$ is good;

(c) $(1-\varphi)\alpha\,|S| \leqslant X(S) \leqslant (1+\varphi)\alpha\,|S|$ and $(1-\varphi)\beta\,|S| \leqslant Y(S) \leqslant (1+\varphi)\beta\,|S|$;

(d) no insertion, no copies and no predictions occur with respect to the hash functions.

Theorem 1. With overwhelming probability, almost all polynomially bounded executions E in (Δ, s)-respecting environment are standard.

Proof. (a) We limit the size of an execution. In the permissionless protocol, it is necessary to recalculate difficult target for a new epoch. For convenience, we consider executions with proper size (e.g., an epoch), where there is no difficult target recalculation point. Note that, an epoch is selected to be long enough, and we set the size of E with upper bound $\frac{K'}{p}/(\alpha+\beta)$ and lower bound $\frac{K}{p}/\alpha$ such that there is no difficult target recalculation point and at least one valid block is stable during E. With overwhelming probability, the protocol execution can be divided into such sets of consecutive rounds.

(b) The blocks created during an execution should with reasonable computational power. By Proposition 1, we have that a round is good with overwhelming probability.

(c) We show that for each execution, the distance between the variable and its expectation is reasonable. By the definitions of α and β, we have that $\mathbb{E}[X(S)] = \alpha\,|S|$ and $\mathbb{E}[Y(S)] = \beta\,|S|$. By Chernoff bound, with probability at least $1 - \mathrm{e}^{-\Omega(|S|)}$, we have $(1-\varphi)\alpha\,|S| \leqslant X(S) \leqslant (1+\varphi)\alpha\,|S|$ and $(1-\varphi)\beta\,|S| \leqslant Y(S) \leqslant (1+\varphi)\beta\,|S|$.

(d) We show that insertion and copy imply collision towards the hash function. Suppose that blocks B_1 and B_2 are two consecutive valid blocks, then a valid block B_3 is inserted into B_1 and B_2 or B_2 extends another chain successfully. That implies that B_2 connects to two distinct blocks, so that a collision occurs. With the security of hash function, it happens with probability at most $\mathrm{e}^{-\Omega(k)}$. Prediction means one can predict the output of hash function, with the security parameter k, we have that it happens with probability $\frac{1}{2^k}$.

4 Analysis of bitcoin backbone protocol Π

In this section, we show our detailed security analysis of the bitcoin backbone protocol Π presented in Section 2.

4.1 Achieving chain growth property

Chain growth property states that the computational power of chains held by the honest parties grows with a lower bound.

Lemma 3. E is a standard execution in (Δ, s)-respecting environment, suppose that, at the beginning of round $r \in E$, an honest party holds chain $\mathcal{C}^r$ with computational power C^r. Then, at the beginning of round $r+1 \in E$, the computational power of honest parties' local chain satisfies $C^{r+1} \geqslant C^r + \sum_{j=1}^{c} C^r_{h,j}$.

Proof. First, suppose $c=1$ that means each honest party can make at most one query to $\mathcal{F}_{\rm MIN}$ during round r. Observe that the corrupted parties can choose to hide or broadcast the newly-mined blocks, so that the honest parties' chains will increase with at least $C^r_{h,1} = c^r_{h,1}$ if at least one honest blocks created and $C^r_{h,1} = 0$ for the worst case.

For the induction step, we assume that the inequality is true for $c-1$, then we have that $C^{r+1} \geqslant C^r + \sum_{j=1}^{c-1} C^r_{h,j}$. If no honest parties succeed at the c-th query, it holds that $C^{r+1} \geqslant C^r + \sum_{j=1}^{c} C^r_{h,j}$. Note that if at least one honest party succeeds at the c-th query, every party will receive these newly-mined blocks (an adversary block maybe included) and extend local chain by one of them, so that $C^{r+1} \geqslant C^r + \sum_{j=1}^{c-1} C^r_{h,j} + C^r_{h,c} = C^r + \sum_{j=1}^{c} C^r_{h,j}$.

Lemma 4. E is a standard execution in (Δ, s)-respecting environment, suppose that, at the beginning of round $r_1 \in E$, an honest party holds chain $\mathcal{C}_1$ with computational power C_1. Then, at the beginning of round $r_1 + t \in E$ $(t > 0)$, each honest party's local chain with computational power $C_2 \geqslant C_1 + \sum_{r=r_1}^{r_1+t-1} \sum_{j=1}^{c} C^r_{h,j}$.

Proof. From Lemma 3, the honest parties' chains increase by at least $\sum_{j=1}^{c} C^r_{h,j}$ during one round. So it holds that $C_2 \geqslant C_1 + \sum_{r=r_1}^{r_1+t-1} \sum_{j=1}^{c} C^r_{h,j}$ for $t > 0$.

Theorem 2. E is a standard execution in (Δ, s)-respecting environment, for any two rounds $r_1, r_2 \in E$, two honest parties with $\mathrm{VIEW}^{P_1,\mathrm{Com}_h,\mathrm{Com}_a}_{\Pi,\mathcal{A},\mathcal{Z}} = \mathcal{C}_1$ and $\mathrm{VIEW}^{P_2,\mathrm{Com}_h,\mathrm{Com}_a}_{\Pi,\mathcal{A},\mathcal{Z}} = \mathcal{C}_2$ at r_1 and r_2, respectively. Then with probability at least $1 - \mathrm{e}^{-\Omega(t)}$, it holds that $\mathrm{Com}(\mathcal{C}_2) - \mathrm{Com}(\mathcal{C}_1) \geqslant g \cdot t$, where $r_2 - r_1 = t \geqslant 0$ and $g = (1-\varphi)\alpha$.

Proof. Suppose that, at the beginning of round r_1, an honest party holds chain $\mathcal{C}_1$ with computational power C_1 and at the beginning of round $r_2 = r_1 + t$ $(t \geqslant 0)$, an honest party holds chain $\mathcal{C}_2$ with computational power C_2. We have that $\Pr[C_2 - C_1 \geqslant g \cdot t] \geqslant 1 - \mathrm{e}^{-\Omega(t)}$.

For any round $r \in E$, we have $\mathbb{E}[C^r_h] = \alpha$ (in Section 3). Let $\mathcal{W}$ be the total amount of computational power contributed by the honest parties during t consecutive rounds in expectation, we have $\mathcal{W} = \alpha \cdot t$. By Chernoff bound, we have $\Pr[\sum_{r=r_1}^{r_1+t-1} \sum_{j=1}^{c} C^r_{h,j} < (1-\varphi)\mathcal{W}] < \mathrm{e}^{-\Omega(t)}$. From Lemma 4, we have $C_2 \geqslant C_1 + \sum_{r=r_1}^{r+t-1} \sum_{j=1}^{c} C^r_{h,j}$. Thus,

$$\Pr[C_2 - C_1 \leqslant \textstyle\sum_{r=r_1}^{r+t-1} \sum_{j=1}^{c} C^r_{h,j} < (1-\varphi)\mathcal{W}] < \mathrm{e}^{-\Omega(t)}$$

implies that

$$\Pr[C_2 - C_1 \geqslant (1-\varphi)\mathcal{W}] \geqslant 1 - \mathrm{e}^{-\Omega(t)}.$$

Let $g = (1-\varphi)\alpha$.

4.2 Achieving chain quality property

Chain quality property states that the ratio of computational power contributed by the adversary in a continuous part of the honest parties' local chains has an upper bound.

Lemma 5. E is a standard execution in (Δ, s)-respecting environment, and then any $l \in \mathbb{N}$ consecutive blocks with computational power C of a chain are created in at least $\frac{C}{\alpha+\beta}$ consecutive rounds.

Proof. Gathering all the resources in the network, the expected computational power that a chain increased is $\alpha+\beta$ per round. Let $S = \{r', \ldots, r^*\} \subseteq E$ be a set of consecutive rounds, during which the l consecutive blocks are created.

Towards a contradiction that $|S| < \frac{C}{\alpha+\beta}$. Let $C = \sum_{r\in S} C^r$, where C^r denotes the increased computational power during round r. It implies that at least one C^r $(r \in S)$ satisfies the inequality $C^r > (1+\varphi)(\alpha+\beta)$. However, by Chernoff bound, it holds that $\Pr[C^r > (1+\varphi)(\alpha+\beta)] < \mathrm{e}^{-\Omega(C)}$ $(r \in S)$.

Theorem 3. E is a standard execution in (Δ, s)-respecting environment, for any honest party P with $\mathrm{VIEW}_{\Pi,\mathcal{A},Z}^{P,\mathrm{Com}_h,\mathrm{Com}_a} = \mathcal{C}$ at round $r \in E$ and any $l \in \mathbb{N}$ consecutive blocks of $\mathcal{C}$ with computational power C. Then with probability at least $1-\mathrm{e}^{-\Omega(C)}$, it holds that the computational power contributed by the adversary is at most $\mu \cdot C$, where $\mu = (1-\delta)\frac{1}{\lambda'} < (1-\delta)^2$.

Proof. Let $\mathcal{C} = B_1 \ldots B_{\mathrm{len}(\mathcal{C})}$ be a chain held by an honest party P at round r. For any $l \in \mathbb{N}$ (large enough) consecutive blocks $B_v, \ldots, B_u$ with computational power C. We assume that B_v and B_u are created by the honest parties at round r_v and r_u, respectively ($r_v = 1$ if B_v is the genesis block). $S = \{r_v, \ldots, r_u\} \subseteq E$ is the set of consecutive rounds that these l blocks create, then $|S| \geqslant \frac{C}{\alpha+\beta}$ (Lemma 5).

Let $X(S)$ and $Y(S)$ denote the computational power contributed by the honest and corrupted parties during S. By Chernoff bound, with overwhelming probability, we have that

$$\begin{aligned} Y(S) &\leqslant (1+\varphi)\beta\,|S| \leqslant (1+\varphi)\cdot\frac{1+\varepsilon}{1-\varepsilon}\cdot\frac{1+pc'}{\lambda'}\alpha\,|S| \\ &= \frac{1+\varphi}{1-\varphi}\cdot\frac{1+\varepsilon}{1-\varepsilon}\cdot\frac{1}{\lambda'}(1-\varphi)\alpha\,|S| < (1-\delta)\frac{1}{\lambda'}\cdot X(S). \end{aligned} \tag{3}$$

The first inequality follows from the definition of standard execution (part (c)), the second one follows from (2) and the last one follows from $0 < \varepsilon+\varphi < 1-\delta < 1$. Note that the equality comes from the fact that $pc' \ll 1$, so $1+pc' \approx 1$. Then, we have

$$\mu = \frac{Y(S)}{C} < \frac{Y(S)}{X(S)} < (1-\delta)\frac{1}{\lambda'} < (1-\delta)^2. \tag{4}$$

4.3 Achieving common prefix property

Common prefix property ensures that the honest parties' local chains share a common part by pruning the last $K \in \mathbb{N}$ blocks and enjoy the growing amount of computational power. We first analyze two possible cases that may cause the honest parties' local chains diverge.

Case 1. For some consecutive queries, more than one honest parties succeed and the newly-mined blocks with the same computational power at the same query.

Case 2. At some time, the adversary broadcasts a forked hidden chain that is better than the honest parties' local chains.

First, based on the synchronous network and chain comparison algorithm, the honest parties extend local chains with the same computational power, which means that the chains held by honest parties with the same amount of computational power and (different) blocks.

Second, by the honest majority assumption, the adversary extends a hidden chain independently, the hidden chain cannot be better than the honest parties' local ones. It implies that the computational power of the hidden chain is smaller than the honest parties' when considering a period of time.

Recall the definition of best chain that the honest parties will not choose a chain that is worse than their local chains at any time as stated in Lemma 6. Then we prove that with overwhelming probability, the honest chains will not diverge by more than K blocks because of Case 1 and the hidden chain kept by the adversary for a long time cannot be better than the public best chain (Case 2).

Lemma 6. E is a standard execution in (Δ, s)-respecting environment, and an honest party holds the best local chain $\mathcal{C}_1$ with $\mathrm{Com}(\mathcal{C}_1) = C_1$ at round r_1. At round r_2 $(r_2 \geqslant r_1, r_1, r_2 \in E)$, the computational power of honest parties' local chain is C_2. Then with overwhelming probability, it holds that $C_2 \geqslant C_1$.

Proof. At round r_1, $\mathcal{C}_1$ with computational power C_1 is a local chain of an honest party, which means that all the chains are held by the honest parties with computational power C_1. If no valid blocks are broadcasted during r_1 to r_2, which means that no honest parties succeed during these rounds (the adversary may hide the newly-mined blocks), then we have $C_2 = C_1$. On the contrary, the honest parties will update local chains to hold a better local state. So we have that $C_2 > C_1$.

Lemma 7. *E is a standard execution in (Δ, s)-respecting environment. Assume that two honest parties adopt two distinct chains $\mathcal{C}_1$ and $\mathcal{C}_2$ at round $r \in E$. Then with overwhelming probability, $\mathcal{C}_1$ and $\mathcal{C}_2$ cannot diverge by more than $K \in \mathbb{N}$ blocks in Case 1.*

Proof. Consider the last common block of $\mathcal{C}_1$ and $\mathcal{C}_2$ that are created by an honest party at round r^* ($r^* = 1$ if no such block). Let $S = \{j : r^* < j \leqslant r\} \subseteq E$ be a set of consecutive rounds. Note that if an honest party succeeds at a query during S, there must be more than one honest parties succeed at the same query (Case 1). Let $Q = \{Q_i : i \in Z = \{1, \ldots, q\}\}$ be the set of queries during S. For convenience, we consider the condition that if one honest party succeeds at the i-th ($i \in Z$) query, exactly another honest party succeeds at the same query.

Let variable $Q_i = 1$ if there are new blocks created at the i-th query, otherwise, $Q_i = 0$. We claim that $\sum_{i=1}^{q} Q_i \geqslant K$. Let $p_i^* =\Pr[Q_i = 1] = \binom{n_{h,i}^S}{2} \cdot p^2 \cdot (1-p)^{n_{h,i}^S - 2}$, where $n_{h,i}^S$ is the number of honest parties for the i-th query during S. Then we set $R = \{v_1, v_2, \ldots, v_K, \ldots\} \subseteq Z$ as the set of indexes of successful queries and we have

$$\Pr[\textstyle\sum_{i=1}^{q} Q_i \geqslant K] = \prod_{i \in R} p_i^* \cdot \prod_{i \in Z/R} (1 - p_i^*).$$

Lemma 8. *E is a standard execution in (Δ, s)-respecting environment. Assume that $\mathcal{C}_1$ is held by an honest party and $\mathcal{C}_2$ is broadcasted by the adversary at round $r \in E$. With overwhelming probability, we have $\mathrm{Com}(\mathcal{C}_1) > \mathrm{Com}(\mathcal{C}_2)$ (Case 2).*

Proof. Consider the last common block of $\mathcal{C}_1$ and $\mathcal{C}_2$ that is created by an honest party at round r^* ($r^* = 1$ if no such block). Let $S = \{i : r^* < i \leqslant r\} \subseteq E$ be a set of consecutive rounds. Towards contradiction, we claim that $X(S) \leqslant Y(S)$. There is no "bad events" occurred in the standard execution, so that the diverge parts of $\mathcal{C}_1$ and $\mathcal{C}_2$ are created during S. By inequality (3), we have that $X(S) > Y(S)$.

Theorem 4. *E is a standard execution in (Δ, s)-respecting environment, for any two honest parties with $\mathrm{VIEW}_{\Pi,\mathcal{A},Z}^{P_1,\mathrm{Com}_h,\mathrm{Com}_a} = \mathcal{C}_1$ and $\mathrm{VIEW}_{\Pi,\mathcal{A},Z}^{P_2,\mathrm{Com}_h,\mathrm{Com}_a} = \mathcal{C}_2$ at round $r_1, r_2 \in E$ ($r_2 \geqslant r_1$), respectively. Then, with probability at least $1 - \mathrm{e}^{-\Omega(K)}$, it holds that $\mathrm{Com}(\mathcal{C}_2) \geqslant \mathrm{Com}(\mathcal{C}_1)$ and $\mathcal{C}_1^{\lceil K} \preceq \mathcal{C}_2$, where $K \in \mathbb{N}$.*

Proof. Consider two chains $\mathcal{C}_1$ and $\mathcal{C}_2$ with computational power C_1 and C_2 held by honest parties P_1 and P_2 at the corresponding rounds r_1 and r_2 ($r_1 \leqslant r_2$). Following from Lemma 6, we have $C_2 \geqslant C_1$. Following from Lemmas 7 and 8, we have that $\mathcal{C}_1$ and $\mathcal{C}_2$ cannot diverge with more than K blocks. Consequently, with the probability at least $1 - \mathrm{e}^{-\Omega(K)}$, we have $(C_2 \geqslant C_1) \wedge (\mathcal{C}_1^{\lceil K} \preceq \mathcal{C}_2)$.

4.4 More discussion

In the real-world, protocol (e.g., bitcoin system) executes in the permissionless setting, where the overall amount of computational power varies with the parties join or leaves the network, and the parties communicate with each other through an asynchronous network, where the messages are delivered with delays. What's more, the protocol is executed by the non-flat parties who hold the different amount of computational power and the security holds under honest majority assumption. Refs. [4–6] analyzed bitcoin backbone protocol in the flat model. As analyzed in Section 1, in addition to the honest majority assumption (denoted by HMA), a stronger assumption (denoted by SA) is required in their models [4–6]. To show a clear line of problem-solving process, we show the comparison of the execution environments of the analytical models between [4–6] and ours (Table 2).

As shown in Table 2, our study improves the analytical model to be non-flat and proceeds the problem-solving into the next step, and extends the study [4] directly. We succeed in analyzing bitcoin backbone protocol in the non-flat model under honest majority assumption.

Table 2 The execution environments of analytical models

Environment	Ref. [4]	Ref. [5]	Ref. [6]	Ours
Permissionless setting	No	No	Yes	No
Asynchronous network	No	Yes	No	No
Non-flat parties	No	No	No	Yes
Assumptions	HMA and SA	HMA and SA	HMA and SA	HMA

Table 3 The comparison of results between [4] and ours

Parameter	Ref. [4]	Ours
HPoHP	$\alpha'' = \frac{q(n-t)}{1+pq(n-t)}$	$\alpha \geqslant \frac{q(n-t)}{1+pc'}$
HPoA	$\beta'' = qt$	$\beta = qt$
CGR	$g' = (1-\varphi)\alpha''$	$g = (1-\varphi)\alpha$
CQ	$\mu' = (1+\frac{\delta}{2})\frac{t}{n-t}$	$\mu = (1-\delta)\frac{t}{n-t}$

To compare the results with [4] more clearly, we use α' ($\gamma' = \alpha' - \alpha'^2$), β', g' and μ' to denote the corresponding parameters in [4]. For convenience, we measure contributions of parties by the computational power that they have invested uniformly. Formally, we use $\alpha'' = \alpha' \cdot \frac{1}{p} = \frac{q(n-t)}{1+pq(n-t)}$ and $\beta'' = \beta' \cdot \frac{1}{p} = qt$ to denote the expected amount of computational power contributed by the honest parties and the adversary in one round [4], respectively, where n is the number of parties and t of them can be controlled by the adversary, and q denotes the number of hashing queries that a party can perform per round. Further, based on the relations that $\text{Com}_h^r = q(n-t)$ and $\text{Com}_a^r = qt$, we can get that $\alpha = \sum_{j=1}^{c} C_{h,j}^r \geqslant \frac{1}{1+pc'} \cdot \text{Com}_h^r = \frac{q(n-t)}{1+pc'}$ and $\beta = qt$, where $c' = \max\{n_{h,j}^r, 1 \leqslant j \leqslant c\}$ is the maximum number of honest parties at the j-th query of round r and satisfies $\text{Com}_h^r = \sum_{j=1}^{c} n_{h,j}^r = q(n-t)$. As a result, we have that $\alpha > \alpha''$, and $\beta = \beta''$.

In security analysis, we can see that the hashing power of the honest parties (denoted by HPoHP) is improved ($\alpha > \alpha''$) obviously and the hashing power of the adversary (denoted by HPoA) remains unchanged as expected ($\beta = \beta''$). As a result, we obtain the higher lower bound of chain growth rate (denoted by CGR) with $g = (1-\varphi)\alpha > g' = (1-\varphi)\alpha''$, the higher chain quality (denoted by CQ) with $\mu = (1-\delta)\frac{\text{Com}_a^r}{\text{Com}_h^r} < \mu' = (1+\frac{\delta}{2})\frac{t}{n-t}$, where $\frac{\text{Com}_a^r}{\text{Com}_h^r} = \frac{t}{n-t} < 1$ is honest majority assumption, and the quicker consensus among honest parties (a fewer time to create K valid blocks). The detailed comparison of results between [4] and ours is presented in Table 3.

From the results, our proposed model is workable and meaningful. However, our model is not perfect in that we divide the execution into several sets, analyze the parties' actions in a relative static setting and consider the synchronous network. In the following work, we will extend our work by combing with the models in [5, 6] and ultimately achieve security analysis of the backbone protocol in a model that is indistinguishable from the real-world protocol execution.

5 Application

In this section, we introduce a concrete application of blockchain. Formally, we show that a robust public transaction ledger Π_{PL} can be established upon the blockchain protocol Π securely.

5.1 Transaction ledgers

A transaction ledger can be extracted as a vector of transaction sequence as $\mathcal{L} = \{x_1, \ldots, x_m\}$, where $x_i = \{\text{tx}_1, \ldots, \text{tx}_n\}$ is a set of transactions packed in a block B_i. Note that the position of transaction $\text{tx}_j \in x_i$ in ledger $\mathcal{L}$ is (i, j). A transaction tx_j^i is the deliver of coins from the payer's accounts named input transactions to the payee's accounts named output transactions and it is valid such that $\text{Valid}(\text{tx}_j^i) = 1$ if (1) the issuer (payer) is the owner of the input transactions, i.e., he owns the secret keys sk that match with the accounts of input transactions, and (2) there is no transaction tx' that conflicts with

tx^i_j, i.e, tx^i_j and tx' have the same input transactions. Ledger $\mathcal{L}$ is valid if each transaction is valid that $\{\mathrm{Valid}(\mathrm{tx}^i_j) = 1 | i = 1, \ldots, m; j = 1, \ldots, n\}$.

A robust transaction ledger protocol Π_{TL} can be realized by instantiating the backbone protocol Π [4]. Formally, each party can fetch messages from the network that contains transactions x and chains $\mathcal{C}$, then validates and obtains the current valid ledger $\mathcal{L}$, and the party P with computational power C executes protocol Π to compete and extend ledger $\mathcal{L}$ via packing the valid transactions $x' \subset x$ into the newly-created block. The functions $V(\cdot)$, $R(\cdot)$ and $I(\cdot)$ that specifies the parties' actions are instantiated as follows.

- **Content validation function** $V(\cdot)$. $V(\mathcal{L}) = 1$ if and only if each vector $x_i \in \mathcal{L}$ is valid, such that $\{\mathrm{Valid}(\mathrm{tx}^i_j) = 1 | i = 1, \ldots, m; j = 1, \ldots, n\}$.
- **Chain reading function** $R(\cdot)$. If $V(\mathcal{L}) = 1$, then $R(\mathcal{C}) = \mathcal{L}$.
- **Input contribution function** $I(\cdot)$. $I(\mathcal{L}, x)$ takes the current ledger $\mathcal{L}$ and a set of transactions x receives from the network as inputs and outputs $\mathcal{L} := \mathcal{L}||x'$, where x' is the largest valid subsequence of x (such that $x' \subset x$) with respect to the transactions in $\mathcal{L}$. Note that $I(\cdot)$ is executed by the parties who win the current competition.

5.2 Constructing a robust public ledger from blockchain

We now prove that, based on the backbone protocol Π, we construct a secure transaction ledger protocol Π_{TL}. Precisely, we prove that the three defined security properties of bitcoin backbone protocol (Subsection 2.3) guarantee the two properties of public transaction ledger (Subsection 2.4).

Theorem 5. E is a standard execution in (Δ, s)-respecting environment, with parameter $K \in \mathbb{N}$, persistence property holds with probability at least $1 - \mathrm{e}^{-\Omega(K)}$.

Proof. Let $\mathcal{C}_1$ be a chain held by an honest party P_1 at round $r_1 \in E$ and tx be a valid transaction packed in block B. Assume that B is at least K deep in $\mathcal{C}_1$, thus $B \in \mathcal{C}_1^{\lceil K}$. Towards a contradiction, let $\mathcal{C}_2$ be a chain held by an honest party P_2 at round r_1 and $B \notin \mathcal{C}_2^{\lceil K}$.

By Theorem 4, we have that $\mathcal{C}_2^{\lceil K} \preceq \mathcal{C}'_2$, where $\mathcal{C}'_2$ is the local chain held by P_2 at round r_2 ($r_2 \geqslant r_1$), and $\mathcal{C}_1^{\lceil K} \preceq \mathcal{C}'_2$. So we have that $B \in \mathcal{C}_2^{\lceil K}$ and $\mathrm{tx} \in \mathcal{C}_2^{\lceil K}$, which happened with probability at least $1 - \mathrm{e}^{-\Omega(K)}$.

Theorem 6. E is a standard execution in (Δ, s)-respecting environment, with parameters $K \in \mathbb{N}$ and $\omega = \frac{2K(1+\varepsilon)}{(1-\varphi)\alpha p}$, liveness property holds with probability at least $1 - \mathrm{e}^{-\Omega(K)}$.

Proof. With the chain growth property (Theorem 2), after ω consecutive rounds the length of honest parties' chains will be increased with at least $2K$ blocks.

Assume that a valid transaction tx is an input of honest parties for at least ω rounds. The chain quality property (Theorem 3) guarantees that at least one of the blocks in $\mathcal{C}^{\lceil K}$ is created by an honest party, where $\mathcal{C}$ is an honest party's local chain. Then tx would be in this block and at a position that is at least K blocks away from the end of honest parties' ledgers. What's more, honest parties will drop a block if a conflicting transaction tx' is included successfully by the adversary.

6 Conclusion

In this paper, we extend the studies of [4–6] to analyze bitcoin backbone protocol in the non-flat model. Our main work is modeling each honest party's mining process as the way of the adversary in previous studies and then getting the expectation of the whole honest parties' hashing power. Compared with previous studies, our model is closer to the real-world protocol and we get the better results in security analysis of bitcoin backbone protocol without any additional assumptions but the honest majority assumption.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2017YFB0802500) and Beijing Municipal Science and Technology Project (Grant No. Z191100007119007).

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2008. http://bitcoin.org/bitcoin.pdf

2 Dwork C, Naor M. Pricing via processing or combatting junk mail. In: Advances in Cryptology—CRYPTO'92. Berlin: Springer, 1993. 139–147

3 Rivest R L, Shamir A, Wagner D A. Time-Lock Puzzles and Timed-Release Crypto. Technical Report, Cambridge, 1996

4 Garay J, Kiayias A, Leonardos N. The bitcoin backbone protocol: analysis and applications. In: Advances in Cryptology—EUROCRYPT 2015. Berlin: Springer, 2015. 281–310

5 Pass R, Seeman L, Shelat A. Analysis of the blockchain protocol in asynchronous networks. In: Advances in Cryptology—EUROCRYPT. Berlin: Springer, 2017. 643–673

6 Garay J, Kiayias A, Leonardos N. The bitcoin backbone protocol with chains of variable difficulty. In: Advances in Cryptology—CRYPTO 2017. Berlin: Springer, 2017. 291–323

7 Ratnasamy S, Francis P, Handley M, et al. A scalable content-addressable network. SIGCOMM Comput Commun Rev, 2001, 31: 161–172

8 Druschel P, Rowstron A. Past: persistent and anonymous storage in a peer-to-peer networking environment. In: Proceedings of IEEE Workshop on Hot Topics in Operating Systems, 2001. 65–70

9 Castro M, Liskov B. Practical byzantine fault tolerance and proactive recovery. ACM Trans Comput Syst, 2002, 20: 398–461

10 Abd-El-Malek M, Ganger G R, Goodson G R, et al. Fault-scalable byzantine fault-tolerant services. SIGOPS Oper Syst Rev, 2005, 39: 59–74

11 Clement A, Wong E L, Alvisi L, et al. Making byzantine fault tolerant systems tolerate byzantine faults. In: Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, Boston, 2009. 153–168

12 Decker C, Wattenhofer R. Information propagation in the bitcoin network. In: Proceedings of International Conference on Peer-To-Peer Computing, 2013. 1–10

13 Sompolinsky Y, Zohar A. Secure high-rate transaction processing in bitcoin. In: Financial Cryptography and Data Security. Berlin: Springer, 2015. 507–527

14 Wei P, Yuan Q, Zheng Y, et al. Security of the blockchain against long delay attack. In: Advances in Cryptology—ASIACRYPT 2018. Berlin: Springer, 2018. 250–275

15 Tsabary I, Eyal I. The gap game. In: Proceedings of ACM International Conference on Systems and Storage, 2018. 132

16 Eyal I, Sirer E G. Majority is not enough: bitcoin mining is vulnerable. Commun ACM, 2018, 61: 95–102

17 Sarkar P. Multi-stage proof-of-work blockchain. IACR Cryptology ePrint Archive, 2019, 2019: 162

18 Szalachowski P, Reijsbergen D, Homoliak I, et al. StrongChain: transparent and collaborative proof-of-work consensus. 2019. ArXiv: 1905.09655

19 David B, Gaži P, Kiayias A, et al. Ouroboros praos: an adaptively-secure, semi-synchronous proof-of-stake blockchain. In: Proceedings of International Conference on the Theory & Applications of Cryptographic Techniques. Berlin: Springer, 2018. 66–98

20 Badertscher C, Gazi P, Kiayias A, et al. Ouroboros genesis: composable proof-of-stake blockchains with dynamic availability. In: Proceedings of Computer and Communications Security, 2018. 913–930

21 Chaum D, Rivest R L, Sherman A T. Blind signatures for untraceable payments. In: Advances in Cryptology. Berlin: Springer, 1983. 199–203

22 Baldimtsi F, Chase M, Fuchsbauer G, et al. Anonymous transferable e-cash. In: Public-Key Cryptography—PKC 2015. Berlin: Springer, 2015. 101–124

23 Tewari H, Hughes A. Fully anonymous transferable ecash. IACR Cryptol ePrint Archive, 2016, 2016: 107

24 Canard S, Pointcheval D, Sanders O, et al. Divisible e-cash made practical. IET Inf Secur, 2015, 10: 332–347

25 Miers I, Garman C, Green M, et al. Zerocoin: anonymous distributed e-cash from bitcoin. In: Proceedings of 2013 IEEE Symposium on Security and Privacy, 2013. 397–411

26 Sasson E B, Chiesa A, Garman C, et al. Zerocash: decentralized anonymous payments from bitcoin. In: Proceedings of 2014 IEEE Symposium on Security and Privacy (SP), 2014. 459–474

27 Canetti R. Security and composition of multiparty cryptographic protocols. J Cryptol, 2000, 13: 143–202

28 Canetti R. Universal composable security: a new paradigm for cryptographic protocols. In: Proceedings of IEEE Symposium on Foundations of Computer Science, 2001

29 Kiayias A, Panagiotakos G. Speed-security tradeoffs in blockchain protocols. IACR Cryptol ePrint Archive, 2015, 2015: 1019