# Формальная верификация и устойчивость к цензуре протоколов византийского консенсуса на примере IBFT

К. В. Зиборов* , Н. С. Бондарев‡ , Ю. А. Янович§

В работе выполнен формальный анализ семейства протоколов Istanbul BFT (IBFT), объединяющий уязвимую первую версию, исправленные промежуточные варианты и модификацию, обеспечивающую устойчивость к цензуре, в рамках единого подхода к верификации. На основе спецификаций TLA+ с помощью средства проверки моделей TLC (i) строятся контрпримеры, демонстрирующие нарушение согласованности в первой версии IBFT, (ii) подтверждаются свойства безопасности и живучести для исправленного варианта QBFT, и (iii) вводится строгая формализация устойчивости к цензуре для византийских протоколов с лидером как ограниченной гарантии качества цепочки, для которой стандартная ротация лидера может давать нарушение. В работе также предлагается и формально верифицируется правило выбора лидера *f-skip*, обеспечивающее устойчивость к цензуре без ослабления исходных гарантий безопасности и живучести.

**Ключевые слова:** IBFT, QBFT, BFT-консенсус, формальная верификация, TLA+, TLC, устойчивость к цензуре..

## 1. Введение

В системах распределённого реестра фундаментальной задачей является поддержание надежности системы в условиях наличия неисправных

* *Зиборов Кирилл Викторович* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: krl.ziborov@gmail.com, ORCID: 0000-0002-5676-9105.

*Ziborov Kirill Viktorovich* — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

‡ *Бондарев Никита Сергеевич* — студент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: bns150101@gmail.com, ORCID: 0009-0000-5240-0178.

*Bondarev Nikita Sergeevich* — student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

§ *Янович Юрий Александрович* — кандидат ф.-м. наук, Сколтех, e-mail: y.yanovich@skoltech.ru, ORCID: 0000-0003-4651-7585.

*Yanovich Yury Aleksandrovich* — Assistant Professor, Skolkovo Institute of Science and Technology.

или враждебных процессов. Протоколы консенсуса решают эту задачу, позволяя узлам соглашаться о единой упорядоченной последовательности транзакций. В то время как ранние публичные блокчейн-сети, самая известная из которых Bitcoin [15], использовали вычислительно затратный алгоритм Proof-of-Work, современные блокчейн-системы — как публичные, так и приватные — всё чаще используют устойчивые к византийским ошибкам (BFT) алгоритмы консенсуса. Корпоративные и приватные блокчейны используют BFT-алгоритмы с детерминированной окончательностью блоков при низких затратах ресурсов, тогда как многие открытые Proof-of-Stake системы применяют BFT-консенсус только после выбора комитета валидаторов [22, 23].

Протокол Practical Byzantine Fault Tolerance (PBFT) [24] заложил основу для этого класса алгоритмов, в то время как Istanbul BFT (IBFT) адаптировал PBFT к Ethereum-подобным разрешительным сетям [25]. IBFT организует выполнение в раунды, управляемые лидером, с двухфазным голосованием, достигая окончательности посредством фиксированного набора валидаторов, и используя Proof-of-Authority для устойчивости к атакам Сивиллы. Его принятие в корпоративных блокчейнах, таких как Quorum, сделало IBFT критическим компонентом реальных распределённых систем и репрезентативным примером BFT-консенсуса в производственных средах.

Однако требования к устойчивости консенсуса выходят за рамки классических свойств безопасности и живучести. В открытых блокчейнах устойчивость к цензуре, то есть гарантия того, что ни одна сущность не может произвольно исключать транзакции, является фундаментальным условием, тесно связанным с формальным свойством качества цепочки, которое обеспечивает справедливое распределение прав создания блоков [2]. Однако это свойство не является безусловным; оно может быть нарушено из-за дефектов протокола, регуляторного давления или экономических стимулов, таких как Maximal Extractable Value (MEV) [3, 4]. Хотя существующие исследования количественно оценили цензуру и её влияние на безопасность в системах Proof-of-Work (PoW) и Proof-of-Stake (PoS) [37], анализ BFT-консенсусов, управляемых лидером, лежащих в основе многих приватных сетей и комитетных стадий открытых сетей, остаётся недостаточно исследованным. В таких протоколах злонамеренный лидер получает привилегированное положение для цензурирования транзакций. Это создаёт необходимость в формализации и обеспечении устойчивости к цензуре в тех самых BFT-протоколах, которым доверяют финальность блоков в корпоративных и развивающихся PoS-экосистемах.

Несмотря на широкое принятие, IBFT прошёл путь через несколько версий со значительными доработками, направленными на корректность

протокола. Ранние реализации содержали критические уязвимости, нарушающие как свойство безопасности (safety), так и живучести (liveness) [27]. Последующие исправления устраняли эти проблемы, однако в литературе отсутствует единый формальный анализ, связывающий развитие протокола от уязвимых к верифицированным реализациям. Более того, хотя безопасность и живучесть получили внимание, свойство устойчивости к цензуре — потенциальная уязвимость, влияющая не только на IBFT, но и на BFT-протоколы с лидером в целом — остаётся неформализованным и непроверенным, создавая риски для систем, которые полагаются на эти механизмы консенсуса.

Эта работа закрывает эти пробелы посредством трёх основных вкладов, которые могут быть расширены за пределы IBFT:

1. **Всесторонний формальный анализ развития IBFT:** Нами был проведен формальный анализ промежуточных версий IBFT и представлены как верификация исправленных реализаций, так и контрпримеры, демонстрирующие конкретные угрозы в уязвимых версиях. Таким образом, был проведен полный формальный анализ истории развития этого популярного протокола.
2. **Новое формализованное определение устойчивости к цензуре:** Мы вводим первое строгое формальное определение устойчивости к цензуре для византийских протоколов консенсуса с лидером и доказываем, что стандартный IBFT нарушает это свойство. Эта уязвимость может быть обобщена на аналогичные протоколы с ротацией лидера, где злонамеренные лидеры могут бесконечно исключать транзакции.
3. **Модификация F-Skip с формальной верификацией:** Мы предлагаем и формально верифицируем лёгкую модификацию *f-skip*, которая обеспечивает устойчивость к цензуре при сохранении всех исходных гарантий безопасности и живучести. Этот механизм применим к любому основанному на лидере BFT-протоколу и предоставляет практическое, формально верифицированное решение проблемы цензуры в приватных блокчейнах и комитетном BFT-консенсусе в Proof-of-Stake системах.

Мы использовали TLA+ [14] для спецификации и средство проверки моделей TLC для исчерпывающей формальной проверки протокола. Результаты продемонстрированы на минимальных конфигурациях, которые охватывают существенное поведение протокола и при этом поддаются формальному анализу. Наш вклад даёт как специфичные для протокола

IBFT исправления, так и общие выводы, применимые к проектированию основанного на лидере консенсуса в современных блокчейн-системах.

Остальная часть работы организована следующим образом: в Разделе 2 обсуждаются связанные работы; Раздел 3 детализирует схему работы протокола IBFT; Раздел 4 представляет наши формальные спецификации и свойства; в Разделе 5 мы представляем формальный анализ версий IBFT на основе наших спецификаций; Раздел 6 детализирует подход к верификации и результаты; и Раздел 7 завершает работу направлениями будущих исследований.

## 2. Связанные работы

Формальная верификация протоколов консенсуса значительно эволюционировала от классических распределённых алгоритмов до блокчейн-специфичных дизайнов. Мы группируем связанные работы по четырём направлениям: методология формальной верификации; верификация протоколов консенсуса, включая свойства за пределами классических безопасности и живучести; анализ протокола IBFT и его модификаций; а также исследования устойчивости к цензуре в консенсусах.

### 2.1. Методы формальной верификации

Проверка моделей и интерактивное доказательство теорем составляют два основных подхода к верификации протоколов консенсуса. Инструменты проверки моделей, такие как TLA+/TLC [14], позволяют исчерпывающе исследовать модели с конечным числом состояний, генерируя контрпримеры в случае нарушения свойств протокола. Экосистема TLA+ была расширена инструментом Apalache [32], который даёт дополнительные возможности верификации и символьной проверки моделей. Недавние разработки также включают Quint [20] — современный язык спецификации, который компилируется в TLA+. Quint использовался для верификации протокола блокчейн-консенсуса ChonkyBFT.

Подход, связанный с интерактивным доказательством теорем, представленный работами в Coq [30] и Isabelle/HOL, обеспечивает машинопроверяемые доказательства корректности при явных предположениях, но проигрывает в трудоемкости. Velisarios [29] механизирует доказательства безопасности PBFT в Coq, тогда как Li и др. [12] формально верифицируют протокол Casper для Ethereum. Наша работа использует метод проверки моделей на TLA+, который балансирует между автоматизацией и строгой верификацией, являясь достаточным для анализа эволюции протокола и позволяя генерировать конкретные контрпримеры.

## 2.2. Верификация протоколов консенсуса

Формальная верификация широко применялась к классическим распределённым алгоритмам, таким как Paxos [16] и Raft [28]. Многие созданные для блокчейнов протоколы, включая Tendermint [33], HotStuff [34] и TetraBFT [17], получили детальный формальный анализ. LibraBFT [35] представляет гибридную методологию верификации, сочетая проверку моделей с доказательством теорем для повышенной надежности.

Появление блокчейнов, использующих Proof-of-Stake (PoS), стимулировало усилия по верификации подобных протоколов. В частности, Ethereum 2.0 Beacon Chain был предметом нескольких исследований формальной верификации. Rashid и др. [8, 9] используют средство проверки моделей SPIN для верификации финализации чекпойнтов и процесса выхода валидаторов. Cassez и др. [10] применяют Dafny для верификации отсутствия ошибок времени выполнения в эталонной реализации Beacon Chain, тогда как Afzaal и др. [11] используют инструмент PAT для аналогичных целей. Эти работы демонстрируют растущую зрелость формальных методов для PoS-консенсуса.

Недавние работы расширили фокус за пределы базовых свойств безопасности и живучести, формализуя дополнительные гарантии консенсуса. Pass и Shi [13] вводят формальные определения справедливости в гибридных моделях консенсуса. Однако *устойчивость к цензуре*, в частности, для BFT-протоколов с лидером, используемых как в корпоративных блокчейнах, так и в комитетных PoS-системах, остаётся недостаточно формализованной. Существующие определения справедливости обычно связаны с упорядочиванием транзакций, а не гарантиями включения транзакций, оставляя разрыв в формальных моделях устойчивости к цензуре как отдельного свойства консенсуса.

## 2.3. Анализ и верификация IBFT

Существующие работы по анализу IBFT в основном состоят из аудитов безопасности, выявляющих уязвимости в ранних версиях [27]. Saltini и Hyland-Wood предоставляют наиболее полный анализ на сегодняшний день, выявляя проблемы безопасности и живучести в раннем IBFT и предлагая исправления. Однако их анализ ограничен первой версией IBFT и не предоставляет единого рассмотрения по промежуточным версиям, а также не изучает устойчивость к цензуре — критическое свойство для корпоративных блокчейн-приложений, где исключение транзакций создаёт бизнес-риски. В последующей работе [19] они также представили IBFT 2.0, пересмотренный вариант, который адресует проблемы безопасности и живучести. Модификация QBFT на основе варианта IBFT, описанного

Moniz[21], была специфицирована и частично верифицирована в Dafny. Доступные артефакты [18] сосредоточены на безопасности и не включают спецификацию или доказательство живучести, они также не связывают верифицированный дизайн QBFT с более ранними версиями IBFT.

Насколько нам известно, ни одна предшествующая работа не предоставляет: (1) полный путь формальной верификации через эволюцию IBFT от уязвимых к исправленным реализациям, (2) строгое формальное определение устойчивости к цензуре, применимое к основанным на лидере BFT-протоколам, или (3) формально верифицированную модификацию, такую как *f-skip*, которая обеспечивает устойчивость к цензуре при сохранении основных свойств протокола. Наша работа закрывает эти пробелы, предоставляя при этом выводы, применимые за пределами IBFT к более широкому классу основанных на ротации лидера BFT-протоколов консенсуса, используемых в современных блокчейн-системах.

## 2.4. Устойчивость к цензуре и безопасность блокчейнов

Изучение устойчивости к цензуре эволюционировало от концептуальной цели публичных систем к количественно определяемому свойству безопасности с прямым воздействием на целостность распределённого реестра. Формальная основа устойчивости к цензуре часто связывается с *качеством цепочки*, метрикой, которая гарантирует минимальную долю блоков, произведённых честными участниками [36]. Zhang и др. [37] устанавливают важную структуру для оценки качества цепочки в PoW-протоколах, показывая, что идеальное качество цепочки остаётся недостижимым и что его отсутствие позволяет front running и другие атаки. Эта формальная связь между качеством цепочки и безопасностью критична для нашей работы. Переходя от теории к практике, Wahrstätter и др. [2] предоставляют эмпирическое исследование, формализуя и количественно оценивая цензуру в Ethereum. Их вывод о том, что регуляторные санкции могут существенно компрометировать нейтральность блокчейна и увеличивать задержку транзакций, подчёркивает цензуру как ощутимую угрозу безопасности, а не только теоретическую проблему.

Дизайн протоколов предлагает два основных пути усиления устойчивости к цензуре. Во-первых, *безлидерные* дизайны консенсуса, такие как DBFT [5, 38], стремятся устранить единую точку отказа (и цензуры), присущую основанным на лидере протоколам, используя децентрализованную сборку блоков. Во-вторых, механизмы вроде Proposer-Builder Separation (PBS) в Ethereum [4] пытаются ограничить привилегии одного производителя блока. Однако, как показывают Heimbach и др., такие дизайны могут непреднамеренно централизовать привилегии производителя

блоков, потенциально усугубляя, а не уменьшая цензуру. Это подчёркивает сложность компромиссов в проектировании цензуроустойчивых систем.

Угроза цензуры внутренне связана с Maximal Extractable Value (MEV). Злонамеренные участники могут цензурировать транзакции, чтобы обеспечить прибыльный front running или sandwich-атаки [3]. Эффективность аукционов сборки блоков [7] и распространённость MEV-возможностей даже на сайдчейнах [1, 6] создают мощные финансовые мотивы для исключения транзакций. Эта экономическая составляющая требует, чтобы любая формальная модель устойчивости к цензуре для современных блокчейнов учитывала рациональных, ориентированных на прибыль противников, а не только произвольные византийские отказы.

Наша работа опирается на этот фундамент, но смещает фокус на *основанный на лидере BFT-консенсус*. В отличие от PoW/PoS-систем, изученных выше, BFT-протоколы предоставляют немедленную окончательность блоков и являются ключевым механизмом в работе корпоративных блокчейнов и комитетных стадий PoS-протоколов. Мы формализуем устойчивость к цензуре в рамках этой конкретной модели, доказывая нарушение этого свойства в стандартных протоколах с ротацией лидера, таких как IBFT, и предлагаем верифицируемую модификацию (*f-skip*), которая не требует радикального архитектурного перехода к безлидерному дизайну, тем самым предлагая практический путь к более сильным гарантиям для существующих, широко развёрнутых систем.

## 3. Протокол консенсуса IBFT

### 3.1. Модель системы и предположения

IBFT работает в частично синхронной системе с $N$ детерминированными процессами (валидаторами), из которых не более $F$ являются византийскими, требуя $N \geq 3F + 1$ для устойчивости. Процессы обмениваются сообщениями через надёжные, аутентифицированные каналы с гарантией доставки в перспективе, предполагая неизвестную, но конечную границу доставки сообщений $\Delta$. Криптографические примитивы обеспечивают целостность сообщений и предотвращают их подделку. Каждый процесс локально хранит текущую высоту блока $h$ и номер раунда $v$ внутри этой высоты.

## 3.2. Основная механика протокола

IBFT следует трёхфазному процессу принятия значения, адаптированному из PBFT, выполняемому на каждой высоте:

1. **Pre-Prepare:** Назначенный лидер для раунда $v$ предлагает блок для высоты $h$, рассылая всем валидаторам сообщение `PRE-PREPARE`, содержащее блок.
2. **Prepare:** Валидаторы проверяют предложенный блок и рассылают сообщения `PREPARE` при принятии. Блок становится *prepared* (подготовленным), когда валидатор получает $2F + 1$ совпадающих сообщений `PREPARE`, образующих кворум.
3. **Commit:** После подготовки блока валидаторы рассылают сообщения `COMMIT`. Блок становится *committed* (финализированным) после получения $2F + 1$ совпадающих сообщений `COMMIT`, после чего он необратимо добавляется в блокчейн.

Свойство пересечения кворумов (требуются $2F + 1$ сообщений) гарантирует, что любые два кворума пересекаются хотя бы в одном корректном процессе, гарантируя согласие между корректными валидаторами. Эта граница устойчивости $3F + 1$ фундаментальна для BFT-консенсуса.

## 3.3. Протокол смены раунда

Когда текущий лидер не прогрессирует (что определяется локальными таймаутами), валидаторы инициируют смену раунда. Критическое различие между версиями IBFT относится к их механизмам смены раунда:

### 3.3.1. IBFT 1.0 (уязвимая версия) [25]

Валидаторы отправляют сообщения `ROUND-CHANGE`, содержащие только номер нового раунда. Новый лидер предлагает произвольный блок, не учитывая ранее подготовленные блоки, что приводит к нарушениям свойства безопасности, когда конфликтующие блоки могут быть зафиксированы на одной и той же высоте.

### 3.3.2. IBFT 2.0 (исправленная промежуточная версия) [19]

Валидаторы прикрепляют (возможно пустой) *prepared certificate* к сообщениям `ROUND-CHANGE`, добавляя блок, для которого они наблюдали кворум `PREPARE`. Новый лидер собирает кворум `ROUND-CHANGE` (сертификат смены раунда) и должен предложить блок, на который ссылается

сертификат с наибольшим значением подготовленного раунда, если он существует; иначе он может предложить любой новый валидный блок, включая сертификат, обосновывающий предложение.

### 3.3.3. QBFT (исправленная промежуточная версия) [21]

Валидаторы включают пары $(pr, pv)$ в сообщения `ROUND-CHANGE`, где $pr$ — наибольший номер подготовленного раунда, а $pv$ — соответствующее подготовленное значение. Новый лидер должен предложить значение из наиболее подготовленного состояния, определяемого оператором `HighestPrepared`, если таковое существует. Это обеспечивает безопасность при смене раунда за счет правильной передачи состояния.

### 3.3.4. Современные верифицированные реализации

Последние версии IBFT включают комплексные исправления и были формально верифицированы[18]. Таким образом, было подтверждено выполнение свойств безопасности и живучести, однако все ещё отсутствует анализ дополнительных свойств, таких как устойчивость к цензуре.

## 3.4. Выбор лидера и ротация

IBFT использует детерминированный выбор лидера. Это гарантирует то, что каждый валидатор периодически становится лидером согласно предсказуемому расписанию. Хотя это обеспечивает простоту и справедливость в честных условиях, такая ротация позволяет византийским лидерам цензурировать транзакции во время своих периодов лидерства. Это может влиять не только на IBFT, но и на любой основанный на лидере BFT-протокол с предсказуемой ротацией. Детерминированная природа позволяет злонамеренным валидаторам предугадывать, когда они будут лидером, и стратегически исключать транзакции.

## 3.5. IBFT в более широком ландшафте консенсуса

Будучи производным от PBFT протоколом с немедленной окончательностью блоков, IBFT представляет класс алгоритмов консенсуса, используемых как в приватных блокчейнах, так и на уровне комитетного слоя консенсуса в Proof-of-Stake системах. Его эволюция от уязвимых к исправленным реализациям отражает вызовы в дизайне BFT-протоколов в целом, делая его ценным примером для формального анализа с более широкими последствиями для основанного на лидере византийского консенсуса.

# 4. Формальная спецификация и определения свойств

Этот раздел детализирует нашу методику формальной верификации, структуру наших спецификаций TLA+ и точные определения свойств, которые мы проверяем. Наш подход спроектирован так, чтобы быть одновременно специфичным для IBFT и обобщаемым на более широкий класс основанных на лидере BFT-протоколов консенсуса.

## 4.1. Методология формальной верификации

Мы используем язык спецификации TLA+ и средство проверки моделей TLC для нашего формального анализа. Этот подход предоставляет строгие математические рамки для моделирования конкурентных и распределённых систем, позволяя исчерпывающее исследование пространства состояний при ограниченных параметрах. Наша методика структурирована следующим образом:

**Разработка спецификации:** Мы создаём три отдельные TLA+ модуля, каждый моделирует критический этап эволюции IBFT:

1. `IBFT1_0.tla` моделирует уязвимый исходный протокол, опуская передачу состояния при смене раунда;
2. `QBFT.tla` моделирует исправленный промежуточный протокол, который включает полный механизм смены раунда;
3. `QBFT_FSkip.tla` моделирует исправленный протокол, усиленный нашим новым правилом выбора лидера *f-skip* для обеспечения устойчивости к цензуре.

**Формализация свойств:** Мы определяем базовые BFT-свойства (Agreement, Validity, Termination), а также новое формализованное определение устойчивости к цензуре как инварианты и темпоральные свойства TLA+.

**Проверка моделей:** Мы используем TLC, чтобы проверить эти свойства на минимальных, но репрезентативных конфигурациях системы (например, $N = 4$, $F = 1$). Эта ограниченная верификация достаточна, чтобы захватить существенные взаимодействия протокола и сгенерировать краткие контрпримеры для нарушений.

**Обобщение:** Хотя спецификации опираются на механику IBFT, определённые свойства, уязвимости и модификации (в частности *f-skip*) сформулированы для более широкой применимости к основанным на лидере BFT-протоколам.

## 4.2. Архитектура базовой спецификации

Все три спецификации TLA+ разделяют общую архитектурную основу, моделируя систему из $N$ детерминированных процессов (валидаторов) с не более чем $F$ византийскими отказами ($N \geq 3F + 1$). Общие элементы представлены в Таблице 1.

Таблица 1: Основные элементы спецификации TLA+

| **Тип элемента** | **Описание** |
|---|---|
| *Переменные состояния* | |
| `view_num[p]` | Текущий номер раунда для процесса $p$ |
| `height` | Текущая высота блока, в которой принимается значение |
| `pr[p]`, `pv[p]` | Наибольший подготовленный раунд и значение для процесса $p$ |
| `messages` | Мультимножество отправленных сообщений в протоколе (`PRE-PREPARE`, `PREPARE`, `COMMIT`) |
| `RCmessages` | Мультимножество сообщений `ROUND-CHANGE` для смены раунда |
| `decision[h][p]` | Значение, принятое процессом $p$ на высоте $h$ |
| `blockLeader[h]` | Лидер, чьё предложение было принято на высоте $h$ |
| *Константы и параметры* | |
| $N$, $F$ | Параметры устойчивости системы ($N \geq 3F + 1$) |
| `QUORUM` | Минимум сообщений для прогресса ($2F + 1$) |
| `Corr` | Множество корректных процессов, $\subseteq \{1, \ldots, N\}$, $\lvert\texttt{Corr}\rvert \geq N - F$ |
| `Values` | Множество валидных полезных нагрузок блоков |

**Спецификации действий:** Каждый модуль определяет набор атомарных переходов состояния, моделирующих действия протокола (например, `UponPrePrepare`, `UponPrepared`, `UponCommit`, `UponQRC` для обработки кворума сообщений смены раунда) и спецификацию допустимого византийского поведения.

Спецификации расходятся в двух ключевых аспектах: логике обоснования и выполнения смены раунда и правиле выбора лидера для заданной высоты и раунда.

## 4.3. Формальные определения свойств

Мы формально определяем и проверяем следующие свойства. Каждое свойство выражается как соответствующая формула TLA+, проверяемая TLC.

### 4.3.1. Согласованность (Safety)

Никакие два корректных процесса не принимают разные значения на одной и той же высоте. Это обеспечивает согласованность консенсуса. В TLA+ мы проверяем инвариант[1]:

```
Agreement ==
  \A h \in 1..MaxHeight, p, q \in Corr :
    (decision[h][p] /= NilValue /\ decision[h][q] /= NilValue)
    => decision[h][p] = decision[h][q]
```

### 4.3.2. Корректность (Safety)

Каждое принятое значение должно быть корректным (или пустым). Это предотвращает принятие произвольных значений:

```
Validity ==
  \A h \in 1..MaxHeight, p \in Corr :
    decision[h][p] \in Values \cup {NilValue}
```

### 4.3.3. Завершаемость (Liveness)

Протокол в конечном итоге примет значение для каждой высоты. Формально, для каждой высоты $h$ в конечном итоге верно, что корректный процесс принимает значение $v \in \text{Values}$ на $h$. В TLA+ мы проверяем темпоральное свойство:

```
Termination == <> \A h \in 1..MaxHeight :
    \E v \in Values : Committed(h, v)
```

[1] Далее мы используем стандартные обозначения TLA+. Например, \/ — это логическое И, ′ обозначает значение в следующем состоянии, а \A — это универсальный квантор $\forall$. <> $F$ — темпоральный оператор *eventually*, означающий, что всегда найдется состояние, в котором $F$ будет выполнено.

### 4.3.4. Устойчивость к цензуре: новое формализованное определение

Стандартные BFT-свойства не предусматривают исключение транзакций злонамеренным лидером. Мы вводим первое строгое определение устойчивости к цензуре для основанных на лидере BFT-протоколов консенсуса.

Введем определение *устойчивости к цензуре для основанного на лидере BFT-консенсуса*:

Основанный на лидере византийский протокол консенсуса является *устойчивым к цензуре* с размером окна $W(N, F)$, если в любой последовательности из $W$ последовательно зафиксированных блоков по крайней мере один блок был предложен корректным лидером, где $W$ является функцией от $N$ и $F$.

Это свойство гарантирует, что злонамеренные лидеры не могут бесконечно доминировать в производстве блоков. Оно непосредственно соответствует гарантии *качества цепочки* как минимум $1/W$ в последовательности блоков. Протокол с предсказуемой ротацией лидера и без дополнительных механизмов (как стандартный IBFT) имеет неограниченный $W$, так как византийские лидеры могут подряд возглавлять произвольно большое число блоков, нарушая устойчивость к цензуре.

**Формализация как инварианта:** Для ограниченного `MaxHeight` мы проверяем инвариант, что в любом окне из $F + 1$ последовательно зафиксированных высот хотя бы один лидер корректен. Это выражено в TLA+ как:

```
HonestLeaderInFPlusOneBlocks ==
  \A h \in 1..(MaxHeight - F) :
    (\A i \in 0..F : blockLeader[h + i] /= -1)
    \* All heights in the window are committed
      => \E i \in 0..F : blockLeader[h + i] \in Corr
```

## 4.4. Модификация F-Skip: обобщаемый механизм

Чтобы обеспечить устойчивость к цензуре, мы предлагаем модификацию *f-skip*, лёгкое правило, применимое к любому основанному на лидере BFT-протоколу с предсказуемой ротацией.

**Базовое правило:** Валидатор не может быть лидером на высоте $h$, если он был лидером на любой из предшествующих $F$ зафиксированных высот (высоты $h - F$ по $h - 1$).

**Формальная реализация:** Это реализуется через два ключевых расширения спецификации:

**Отслеживание состояния:** Массив `blockLeader` записывает лидера для каждой зафиксированной высоты.

**Динамическая допустимость лидеров:**

```
AllowedLeaders(h) ==
    AllProcs \ {blockLeader[i] : i \in Max(1, h-F)..(h-1)}

Leader(h, v) ==
    LET candidates == AllowedLeaders(h) IN
    IF candidates = {} THEN DefaultLeader(h, v)
    ELSE KthMin(candidates, ((v-1) % |candidates|) + 1)
```

Лидер для $(h, v)$ затем выбирается детерминированной ротацией из множества `AllowedLeaders`$(h)$.

**Гарантия:** При не более чем $F$ византийских процессах принцип Дирихле гарантирует, что в любой последовательности из $F + 1$ последовательно зафиксированных блоков хотя бы один лидер должен быть корректным. Это ограничивает окно цензуры как $W = F + 1$. Модификация изменяет только выбор лидера, сохраняя все исходные свойства безопасности и живучести, зависящие от базовой логики принятия значения и смены раунда.

## 5. Анализ эволюции протокола IBFT

Мы применяем методику из Раздела 4 для анализа трёх ключевых версий IBFT. Для каждой мы описываем ключевые характеристики модели TLA+, проверяемые свойства и результаты проверки модели.

### 5.1. Уязвимый IBFT 1.0: спецификация и нарушение безопасности

**Обзор модели (`IBFT1_0.tla`):** Эта спецификация моделирует исходный, уязвимый протокол. Причина уязвимости кроется в отсутствии передачи состояния при смене раунда. Когда таймаут вызывает смену раунда, валидаторы увеличивают свой номер раунда, но не отправляют сообщений `ROUND-CHANGE`, передающих их подготовленное состояние $(pr, pv)$.

**Ключевой дефект в спецификации:** Действие смены раунда неполное:

```
Timeout(p) ==
    /\ processState' = [processState EXCEPT ![p].view_num =
       processState[p].view_num + 1]
    /\ UNCHANGED <<messages>>  \* No ROUND-CHANGE messages sent
```

Следовательно, новый лидер не имеет информации о ранее подготовленных блоках и может предложить произвольный новый блок, даже если другой блок был подготовлен в предыдущем раунде.

**Результаты верификации:** Наша проверка модели выявила, что:

- **Agreement нарушается**. TLC генерирует контрпример трассы, где византийский лидер отправляет конфликтующие сообщения `PRE-PREPARE` в раунде 1. Корректные процессы разделяются в ходе `PREPARE`-фазы. Последующие корректные лидеры, не зная об этом конфликте из-за отсутствия передачи состояния, предлагают разные значения, что приводит к двум различным принятым значениям на одной и той же высоте.
- **Validity** выполнялось тривиально, так как все предложенные значения принадлежат множеству `Values`.

Это подтверждаёт известную уязвимость безопасности IBFT 1.0. Контрпример конкретно демонстрирует, как протокол может нарушать свойство согласованности консенсуса.

## 5.2. Исправленный QBFT: безопасность и живучесть без устойчивости к цензуре

**Обзор модели (`QBFT.tla`):** Эта спецификация моделирует исправленную версию протокола (QBFT). Существенное исправление заключается во введении полноценного протокола смены раунда, где сообщения `ROUND-CHANGE` передают наиболее подготовленное состояние отправителя $(pr, pv)$.

**Ключевой механизм** — `HighestPrepared`, безопасность протокола зависит от этого оператора, который обрабатывает кворум $Q$ сообщений `ROUND-CHANGE`:

```
HighestPrepared(Q) ==
    IF Q = {} THEN <<NilView, NilValue>>
    ELSE
        LET prSet == {m.pr : m \in Q}
            maxPr == Maximum(prSet) IN
        IF maxPr = NilView THEN <<NilView, NilValue>>
        ELSE
            LET msgsAtMax == {m \in Q : m.pr = maxPr} IN
            LET rep == CHOOSE m \in msgsAtMax : TRUE IN
            <<maxPr, rep.pv>>
```

Новый лидер раунда $v$ должен предложить компоненту значения (второй элемент) кортежа, возвращаемого `HighestPrepared`($Q$), где $Q$ — кворум сообщений `ROUND-CHANGE` для раунда $v$. Это гарантирует, что любой ранее подготовленный блок будет предложен повторно, сохраняя согласованность при смене раунда.

**Результаты верификации:** Наша проверка модели подтвердила, что:

- **Agreement** и **Validity выполняются**. Проверка TLC подтверждает за счёт явного перебора пространства состояний, что механизм `HighestPrepared` предотвращает принятие конфликтующих блоков на одной и той же высоте.
- **Termination выполняется**. Протокол в конечном итоге фиксирует блок на каждой высоте при условиях частичной синхронности и справедливости лидера.
- **Устойчивость к цензуре** (`HonestLeaderInFPlusOneBlocks`) **нарушается**. TLC находит контрпример, где один и тот же византийский процесс записывается как `blockLeader` для двух последовательных высот (например, высоты 1 и 2 при $F = 1$). Это возможно, потому что стандартная round-robin ротация лидера (`Leader`$(h, v) = ((h + v - 2) \bmod N) + 1$) не накладывает ограничений на то, как часто византийский валидатор может быть лидером.

Таким образом, для QBFT выполнены классические BFT-свойства безопасности и живучести, но не обеспечена устойчивость к цензуре. Один злонамеренный валидатор, являясь лидером, может бесконечно цензурировать транзакции, предлагая пустые или выборочные блоки.

## 5.3. QBFT с F-Skip: безопасный, живой и устойчивый к цензуре вариант

**Обзор модели** (`QBFT_FSkip.tla`)**:** Эта спецификация расширяет `QBFT.tla` исключительно изменением логики выбора лидера для реализации правила *f-skip* (Раздел 4.4). Базовая логика принятия значения и смены раунда остаётся неизменной.

**Ключевая модификация — динамическая допустимость лидера:** Лидер больше не определяется постоянной формулой. Вместо этого множество допустимых лидеров для высоты $h$ исключает тех, кто был лидером на предыдущих $F$ блоках:

```
AllowedLeaders(h) ==
    AllProcs \ {blockLeader[i] : i \in Max(1, h-F)..(h-1)}
```

```
Leader(h, v) ==
    LET candidates == AllowedLeaders(h) IN
    IF candidates = {} THEN DefaultLeader(h, v)
    ELSE KthMin(candidates, ((v-1) % |candidates|) + 1)
```

**Результаты верификации:** Наша проверка модели верифицировала, что:

- **Agreement**, **Validity** и **Termination выполняются**. Проверка TLC подтверждает, что модификация *f-skip* не затрагивает безопасность и живучесть базового протокола. Механизм `HighestPrepared` остаётся эффективным.
- **Устойчивость к цензуре (`HonestLeaderInFPlusOneBlocks`) выполняется**. Для ограниченной модели (`MaxHeight` $= 3$, $F = 1$) TLC подтверждает сохранение инварианта того, что в каждом окне из $F + 1$ (то есть 2 в проверяемой конфигурации) последовательных принятых блоков хотя бы один лидер честный. Это формально доказывает, что *f-skip* обеспечивает желаемое свойство устойчивости к цензуре.

Таким образом, модификация *f-skip* успешно преобразует протокол в устойчивый к цензуре, сохраняя при этом свойства безопасности и живучести. Она предоставляет практическое, формально верифицированное решение уязвимости цензуры, присущей стандартным основанным на лидере BFT-протоколам, таким как IBFT.

## 6. Формальная верификация и результаты

### 6.1. Настройки и методология верификации

Мы верифицировали все версии протокола с помощью средства проверки моделей TLC версии 1.7.4 в ограниченной конфигурации, обеспечивающей исчерпывающий явный перебор пространства состояний при фиксированных значениях параметров. Такая конфигурация позволяет воспроизводимо обнаруживать уязвимости, получать минимальные контрпримеры и одновременно сохранять вычислительную реализуемость экспериментов.

**Аппаратно-программная среда.** Эксперименты выполнялись на машине с процессором Intel Core i7-12700H (2,70 ГГц), 8 ГБ ОЗУ и SSD объёмом 1 ТБ. Для запуска использовалась среда TLA+ Toolbox 1.7.4.

**Параметры проверки.** Мы использовали минимальные конфигурации, которые удовлетворяют условию устойчивости $N \geq 3F + 1$. В

таблице 2 приведены значения параметров, использованные для каждого варианта протокола, включая оптимизированные настройки для различных классов свойств.

Таблица 2: Параметры верификации для каждого варианта протокола

| **Параметр** | **IBFT 1.0** | **QBFT (без f-skip)** | **QBFT (с f-skip)** |
|---|---|---|---|
| $N$ (валидаторы) | 4 | 4 | 4 |
| $F$ (византийские) | 1 | 1 | 1 |
| Корректные процессы (`Corr`) | $\{2, 3, 4\}$ | $\{2, 3, 4\}$ | $\{2, 3, 4\}$ |
| Византийский процесс | $\{1\}$ | $\{1\}$ | $\{1\}$ |
| `Values` | `{"A "B"}` | `{"A "B"}` / `{"A"}`* | `{"A "B"}` / `{"A"}`* |
| `MaxHeight` | 1 | 2 | 2 / 3* |
| `MaxView` | 4 | 3 / 4* | 3 / 5* |
| `MaxMessages` | 50 | 30 / 40* | 30 / 50* |

*Разные конфигурации использовались для верификации разных свойств (см. ниже).

**Стратегия верификации.** Для каждой версии протокола мы проверяли четыре класса свойств, сформулированных в разделе 4.3: (1) корректность типов (`TypeOK`) как базовую проверку согласованности модели; (2) *Agreement* и *Validity* как инварианты безопасности; (3) *Termination* как темпоральное свойство живучести при допущениях справедливости (частичная синхронность и в конечном итоге прогрессирующий лидер); (4) инвариант устойчивости к цензуре `HonestLeaderInFPlusOneBlocks`. Для отдельных классов свойств использовались различные наборы параметров, позволяющие существенно уменьшать пространство состояний без изменения семантики проверяемого свойства (например, сокращение множества `Values` при проверке устойчивости к цензуре).

## 6.2. IBFT 1.0: подтверждена уязвимость безопасности

Для IBFT 1.0 применялись два набора параметров, отдельно оптимизированные под проверку `TypeOK` и поиск контрпримера для *Agreement*:

- TypeOK: `Values` $= \{"A"\, "B"\}$, `MaxView` $= 4$, `MaxHeight` $= 1$, `MaxMessages` $= 50$
- Agreement: `Values` $= \{"A"\, "B"\}$, `MaxView` $= 3$, `MaxHeight` $= 1$, `MaxMessages` $= 50$

**Проверяемые свойства.** Проверялись *Agreement* и *Validity* (см. раздел 4.3).

**Результаты.** TLC обнаружил контрпример, нарушающий *Agreement*, после перебора порядка $5{,}9 \cdot 10^5$ состояний за 7 с. Трасса контрпримера

иллюстрирует типичное нарушение безопасности: византийский процесс 1 рассылает конфликтующие сообщения `PRE-PREPARE` со значениями “A” и “B” в раунде 1; корректные процессы разделяются по подготовленным значениям; далее, из-за отсутствия передачи подготовленного состояния при смене раунда, последующие лидеры не обязаны переиспользовать уже подготовленное значение и могут предложить альтернативу. В результате на одной и той же высоте фиксируются несовместимые решения (например, процесс 2 фиксирует “A”, а процесс 3 фиксирует “B”). Таким образом, эксперименты воспроизводимо подтверждают известную уязвимость IBFT 1.0: отсутствие передачи состояния при смене раунда допускает конфликтующие принятые значения.

### 6.3. QBFT без f-skip: безопасность выполняется, но устойчивость к цензуре нарушается

Для QBFT без f-skip использовались два набора параметров:

- Agreement/Validity: $\texttt{Values} = \{\text{"}A\text{"}\,\text{"}B\text{"}\}$, $\texttt{MaxView} = 3$, $\texttt{MaxHeight} = 2$, $\texttt{MaxMessages} = 30$
- Устойчивость к цензуре: $\texttt{Values} = \{\text{"}A\text{"}\}$, $\texttt{MaxView} = 4$, $\texttt{MaxHeight} = 2$, $\texttt{MaxMessages} = 40$

**Проверяемые свойства.** Проверялись инварианты безопасности (*Agreement*, *Validity*), темпоральное свойство живучести (*Termination*) при допущениях справедливости, а также инвариант устойчивости к цензуре `HonestLeaderInFPlusOneBlocks` (см. раздел 4.3).

**Результаты.** Проверка TLC подтверждаёт, что *Agreement* и *Validity* выполняются (порядка $1{,}18 \cdot 10^9$ состояний за 4 ч), а *Termination* выполняется при заданных допущениях справедливости. Однако инвариант устойчивости к цензуре нарушается: TLC находит контрпример, в котором византийский процесс 1 фиксируется как `blockLeader` на высотах 1 и 2, уже после перебора порядка $5{,}5 \cdot 10^5$ состояний за 12 с. Причина нарушения заключается в том, что при стандартной ротации лидеров и отсутствии дополнительных ограничений лидера византийский валидатор может за счёт асимметричных задержек сообщений и управления моментом смены раунда добиваться последовательного лидерства на соседних высотах, тем самым демонстрируя, что одних лишь безопасности и живучести недостаточно для обеспечения устойчивости к цензуре.

### 6.4. QBFT с f-skip: все свойства выполняются

Для QBFT с f-skip применялись два набора параметров:

- Agreement/Validity: `Values` $= \{"A", "B"\}$, `MaxView` $= 3$, `MaxHeight` $= 2$, `MaxMessages` $= 30$
- Устойчивость к цензуре: `Values` $= \{"A"\}$, `MaxView` $= 5$, `MaxHeight` $= 2$ (и 3)*, `MaxMessages` $= 50$

*Примечание.* Для проверки инварианта `HonestLeaderInFPlusOneBlocks` достаточно, чтобы `MaxHeight` покрывал хотя бы одно окно длины $F + 1$; при $F = 1$ минимально достаточно `MaxHeight`$=2$.

**Проверяемые свойства.** Проверялись те же свойства, что и для QBFT без f-skip, с учётом модификации выбора лидера, обеспечивающей `HonestLeaderInFPlusOneBlocks`.

**Результаты.** Проверка TLC подтверждает выполнение всех свойств: *Agreement* и *Validity* выполняются (порядка $5{,}77 \cdot 10^8$ состояний за 3 ч); *Termination* сохраняется при тех же допущениях справедливости; инвариант устойчивости к цензуре выполняется (порядка $5{,}13 \cdot 10^8$ состояний за 6 ч). Механизм f-skip запрещает любому валидатору быть лидером более одного раза в любом окне из $F+1$ последовательно зафиксированных блоков. При $F = 1$ это означает, что в любых двух последовательных блоках по крайней мере один лидер обязан быть корректным, и, следовательно, "окно цензуры" ограничено сверху значением $F$ последовательных блоков под византийским лидерством.

## 6.5. Сводка результатов верификации

В таблице 3 сведены результаты верификации для всех трёх вариантов протокола на основе фактических запусков TLC. В совокупности результаты иллюстрируют эволюцию IBFT: от небезопасной версии (IBFT 1.0) к версии, удовлетворяющей свойствам безопасности и живучести (QBFT), и далее к версии, обеспечивающей устойчивость к цензуре (QBFT с f-skip).

## 6.6. Ограничения и вопросы масштабируемости

Наш подход наследует типичные ограничения подхода, связанного с проверкой моделей.

**Взрыв пространства состояний.** Экспоненциальный рост числа состояний ограничивает анализ малыми конфигурациями ($N = 4$, $F = 1$). Тем не менее, именно минимальная конфигурация $N = 3F + 1$ соответствует "границе устойчивости" и воспроизводит наиболее сложные сценарии корректности, поэтому она является информативной для выявления уязвимостей и проверки исправлений.

**Ограниченность по высотам и раундам.** Верификация выполнялась для ограниченных значений высоты (1–3) и номера раунда (3–5). Это

Таблица 3: Сводка результатов верификации для вариантов IBFT

| **Свойство** | **IBFT 1.0** | **QBFT (без f-skip)** | **QBFT (с f-skip)** |
|---|---|---|---|
| **Agreement (безопасность)** | Нарушено | Выполняется | Выполняется |
| **Validity (безопасность)** | Выполняется | Выполняется | Выполняется |
| **Termination (живучесть)** | Н/П | Выполняется* | Выполняется* |
| **Устойчивость к цензуре** | Н/П | Нарушено | Выполняется |
| **Перебор состояний (TypeOK)** | $\sim 4{,}6 \cdot 10^6$ | $\sim 1{,}18 \cdot 10^9$ | $\sim 5{,}77 \cdot 10^8$ |
| **Перебор состояний (Agreement)** | $\sim 5{,}9 \cdot 10^5$ | $\sim 1{,}18 \cdot 10^9$ | $\sim 5{,}77 \cdot 10^8$ |
| **Перебор состояний (Censorship)** | Н/П | $\sim 5{,}5 \cdot 10^5$ | $\sim 5{,}13 \cdot 10^8$ |
| **Суммарное время верификации** | 61 с | 4 ч | 9 ч |

*Живучесть проверялась при допущениях справедливости

не является доказательством для неограниченных исполнений, однако на практике такие границы превышают типичные шаблоны исполнения, необходимые для воспроизведения нарушений безопасности или атак цензуры.

**Чувствительность к параметрам.** Мы наблюдали значимую зависимость размеров пространства состояний от выбора параметров. Например, сокращение `Values` с `{"A "B"}` до `{"A"}` резко уменьшает пространство состояний при проверке устойчивости к цензуре, не влияя на корректность проверяемого инварианта.

**Параметризованная верификация.** Для получения гарантий при произвольных $N$ и $F$ в перспективе целесообразно использовать параметризованную проверку моделей или доказательство теорем. Тем не менее, для целей данной работы (демонстрация дефектов, исправлений и их эффектов) ограниченная проверка моделей особенно ценна тем, что выдаёт конкретные, интерпретируемые контрпримеры.

Несмотря на указанные ограничения, наша верификация:

1. воспроизвела нарушение безопасности в IBFT 1.0;
2. подтвердила выполнение свойств безопасности и живучести в QBFT;

3. выявила уязвимость устойчивости к цензуре при стандартной ротации лидера;
4. формально подтвердила, что модификация f-skip обеспечивает устойчивость к цензуре, не нарушая остальные свойства.

# 7. Заключение и направления дальнейших исследований

В данной работе представлен последовательный путь формальной верификации протокола консенсуса IBFT, который устраняет разрыв между анализом исторически уязвимых версий и современными исправленными реализациями, а также расширяет класс проверяемых гарантий за пределы традиционных безопасности и живучести. Мы внесли три ключевых вклада. Во-первых, мы дали целостный формальный анализ эволюции IBFT, подтвердив корректность промежуточных исправлений и построив воспроизводимые контрпримеры для ошибочных версий. В частности, спецификации TLA+ и эксперименты TLC демонстрируют, что отсутствие передачи состояния при смене раунда в IBFT 1.0 приводит к нарушению согласованности принятых значений, тогда как исправленные варианты (QBFT) сохраняют безопасность за счёт механизма `HighestPrepared`. Во-вторых, мы сформулировали строгое определение устойчивости к цензуре для BFT-протоколов с лидером и показали, что стандартный IBFT (и более общий класс протоколов с предсказуемой ротацией лидера) не удовлетворяет этому свойству: византийский лидер может исключать транзакции, сохраняя формальные свойства безопасности и живучести. В-третьих, мы предложили и формально верифицировали модификацию *f-skip* — лёгкий механизм, применимый к BFT-протоколам с лидером, который обеспечивает устойчивость к цензуре и при этом не нарушает исходные гарантии безопасности и живучести. Верификация подтверждает, что f-skip ограничивает “окно цензуры” сверху величиной $F$ последовательных блоков под византийским лидерством.

Практическая значимость результатов заключается в следующем. Механизм f-skip даёт простое и обратно совместимое усиление для существующих реализаций IBFT: требуется лишь модификация выбора лидера, при этом достигается формально подтверждённая устойчивость к цензуре — критически важная гарантия для корпоративных блокчейнов, где избирательное исключение транзакций напрямую трансформируется в операционные и бизнес-риски. Более общий вывод состоит в том, что одних только безопасности и живучести недостаточно для практического консенсуса: свойства “справедливости” и противодействия цензуре долж-

ны быть явно заложены в протокол и формально проверены, особенно в архитектуре с выбранным лидером. Наконец, мы демонстрируем прикладную ценность ограниченной проверки моделей: даже минимальные конфигурации ($N = 4$, $F = 1$) способны воспроизводить существенные сценарии и порождать контрпримеры, полезные для проектирования и отладки протоколов.

В качестве направлений дальнейших исследований представляются наиболее перспективными следующие задачи. (i) Переход от ограниченной проверки к параметризованной верификации, обеспечивающей гарантии при произвольных $N$ и $F$, за счёт параметризованного model checking и/или доказательства теорем. (ii) Адаптация f-skip к динамическим наборам валидаторов (вход/выход валидаторов), что критично для практических систем с изменяющимся составом участников. (iii) Перенос формализации устойчивости к цензуре и механизма f-skip на другие протоколы консенсуса с лидером (например, HotStuff, Tendermint, LibraBFT) для усиления гарантий в более широком ландшафте BFT-консенсуса. (iv) Интеграция f-skip в промышленные клиенты (например, GoQuorum, Hyperledger Besu) с последующей оценкой влияния на производительность и задержки. (v) Расширение модели противника в сторону рациональных, экономически мотивированных атак (в частности, связанных с MEV), чтобы формальная модель лучше отражала реальные стимулы цензуры и манипуляций порядком включения транзакций.

Эволюция IBFT от уязвимой версии к формально проверенному и устойчивому к цензуре варианту подчёркивает практическую роль формальных методов в проектировании блокчейн-консенсуса. Наши результаты показывают, что формальная верификация является не только академическим упражнением, но и необходимым инженерным инструментом для обеспечения надёжности распределённых систем, на которых строятся современные блокчейн-приложения. В совокупности мы связываем анализ эволюции протокола, формализацию нового свойства и практический механизм его обеспечения, тем самым внося вклад как в понимание IBFT, так и в более широкую область формальной верификации византийского консенсуса.

## Доступность данных

Все спецификации TLA+, конфигурации TLC и скрипты верификации доступны по адресу:
`https://github.com/BondarevNS/ibft-tla-plus-verification`

## Список литературы


[1] Vostrikov D., Madhwal Y., Seoev A., Smirnova A., Yanovich Yu., Smirnov A., Gorgadze V., "Unpacking Maximum Extractable Value on Polygon: A Study on Atomic Arbitrage", *arXiv preprint*, 2025.

[2] Wahrstätter A., Ernstberger J., Yaish A., Zhou L., Qin K., Tsuchiya T., Steinhorst S., Svetinovic D., Christin N., Barczentewicz M., Gervais A., "Blockchain Censorship", *Proceedings of the ACM Web Conference 2024*, 2024, 1632–1643. DOI: 10.1145/3589334.3645431.

[3] Daian P., Goldfeder S., Kell T., Li Y., Zhao X., Bentov I., Breidenbach L., Juels A., "Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability", *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, 910–927. DOI: 10.1109/SP40000.2020.00040.

[4] Heimbach L., Kiffer L., Ferreira Torres C., Wattenhofer R., "Ethereum's Proposer-Builder Separation: Promises and Realities", *Proceedings of the 2023 ACM on Internet Measurement Conference*, 2023, 406–420. DOI: 10.1145/3618257.3624824.

[5] Afanasyeva A., Kameskiy D., Telnov S., Yanovich Yu., "Leaderless Byzantine Fault-Tolerant Consensus Protocol for Blockchains", *Proceedings of the 2023 6th International Conference on Blockchain Technology and Applications*, 2023, 78–84. DOI: 10.1145/3651655.3651665.

[6] Seoev A., Gremyachikh L., Smirnova A., Madhwal Y., Kalacheva A., Belousov D., Zubov I., Smirnov A., Fedyanin D., Gorgadze V., Yanovich Yu., "The Bidding Games: Reinforcement Learning for MEV Extraction on Polygon Blockchain", *arXiv preprint*, 2025.

[7] Öz B., Sui D., Thiery T., Matthes F., "Who Wins Ethereum Block Building Auctions and Why?", *6th Conference on Advances in Financial Technologies*, 2024, 22:1–22:25. DOI: 10.4230/LIPIcs.AFT.2024.22.

[8] Rashid M., Rasool I., Zafar N.A., Afzaal H., "Formal Modeling and Verification of Validator Voluntarily Exit in Ethereum 2.0 Beacon Chain", *2023 2nd International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE)*, 2023, 1–6. DOI: 10.1109/ETECTE59617.2023.10396687.

[9] Rashid M., Rasool I., Zafar N.A., Afzaal H., "Formal Modeling and Verification of Justification and Finalization of Checkpoints in Ethereum 2.0 Beacon Chain", *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, 2024, 1–6. DOI: 10.1109/KHI-HTC60760.2024.10481930.

[10] Cassez F., Fuller J., Asgaonkar A., "Formal Verification of the Ethereum 2.0 Beacon Chain", *Lecture Notes in Computer Science*, **13243** (2022), 167–182. DOI: 10.1007/978-3-030-99524-9_9.

[11] Afzaal H., Zafar N.A., Tehseen A., Kousar S., Imran M., "Formal Verification of Justification and Finalization in Beacon Chain", *IEEE Access*, **12** (2024), 55077–55102. DOI: 10.1109/ACCESS.2024.3389551.

[12] Li E., Serbanuta T., Diaconescu D., Zamfir V., Rosu G., "Formalizing Correct-by-Construction Casper in Coq", *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020, 1–3. DOI: 10.1109/ICBC48266.2020.9169468.

[13] Pass R., Shi E., “Hybrid consensus: Efficient consensus in the permissionless model”, *Leibniz International Proceedings in Informatics (LIPIcs), DISC 2017*, **91. DOI:** 10.4230/LIPIcs.DISC.2017.39 (2017).

[14] Lamport L., *Specifying Systems*, Addison–Wesley, 2002. DOI: 10.5555/579617.

[15] Nakamoto S., “Bitcoin: A peer-to-peer electronic cash system”, *White paper*, 2008.

[16] Lamport L., “TLA+ specification of Paxos”, *Online artifact*, 2008.

[17] Losa G., “TLA+ specification of TetraBFT”, *Online artifact*, 2024.

[18] Saltini R., “QBFT Formal Specification and Verification in Dafny”, *Online artifact*, 2024.

[19] Saltini R., Hyland-Wood D., “IBFT 2.0: A Safe and Live Variation of the IBFT Blockchain Consensus Protocol for Eventually Synchronous Networks”, *arXiv preprint arXiv:1909.10194*, 2019. DOI: 10.48550/arXiv:1909.10194.

[20] França B., Kolegov D., Konnov I., Prusak G., “ChonkyBFT: Consensus Protocol of ZKsync”, *arXiv preprint arXiv:2503.15380*, 2025.

[21] Moniz H., “The Istanbul BFT Consensus Algorithm”, *arXiv preprint arXiv:2002.03613*, 2020. DOI: 10.48550/arXiv.2002.03613.

[22] Buterin V., Griffith V., “Casper the Friendly Finality Gadget”, *CoRR, abs/1710.09437*, 2017.

[23] Buterin V., Hernandez D., Kamphefner T., Pham K., Qiao Z., Ryan D., Sin J., Wang Y., Zhang Y.X., “Combining GHOST and Casper”, *CoRR, abs/2003.03052*, 2020.

[24] Castro M., Liskov B., “Practical Byzantine fault tolerance”, *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI '99)*, 1999, 173–186. DOI: 10.1145/296806.296824.

[25] Ethereum Community, “Istanbul Byzantine Fault Tolerance (EIP-650)”, *Online proposal*.

[26] ConsenSys, “Quorum Blockchain”, *Online project*.

[27] Saltini R., “Correctness Analysis of IBFT”, *CoRR, abs/1901.07160*, 2019.

[28] Ongaro D., “Consensus: Bridging theory and practice”, *Ph.D. thesis, Stanford University*, 2014.

[29] Rahli V., Vukotic I., Völp M., Esteves-Verissimo P., “Velisarios: Byzantine Fault-Tolerant Protocols Powered by Coq”, *Programming Languages and Systems (ESOP 2018), Lecture Notes in Computer Science*, **10801** (2018), 619–650. DOI: 10.1007/978-3-319-89884-1_22.

[30] Alturki M.A., Chen J., Luchangco V., Moore B., Palmskog K., Peña L., Rosu G., “Towards a verified model of the Algorand consensus protocol in Coq”, *Formal Methods. FM 2019 International Workshops, Lecture Notes in Computer Science*, **12232** (2020), 362–367. DOI: 10.1007/978-3-030-54994-7_27.

[31] Konnov I., Kukovec J., Tran T.-H., “TLA+ model checking made symbolic”, *Proc. ACM Program. Lang.*, **3**:OOPSLA (2019), 123. DOI: 10.1145/3360549.

[32] Konnov I.V., Kuppe M.A., Merz S., “Specification and Verification with the TLA+ Trifecta: TLC, Apalache, and TLAPS”, *Leveraging Applications of Formal Methods*, 2022. DOI: 10.1007/978-3-031-19849-6_6.

[33] Braithwaite S., Buchman E., Konnov I., Milosevic Z., Stoilkovska I., Widder J., Zamfir A., “Formal Specification and Model Checking of the Tendermint Blockchain Synchronization Protocol”, *2nd Workshop on Formal Methods for*

*Blockchains (FMBC 2020), OASIcs*, **84** (2020), 10:1–10:8. DOI: 10.4230/OASIcs.FMBC.2020.10.

[34] Kukharenko V., Ziborov K., Sadykov R., Rezin R., "Verification of HotStuff BFT Consensus Protocol With TLA+/TLC in an Industrial Setting", *SHS Web of Conferences*, **93** (2021), 01006. DOI: 10.1051/shsconf/20219301006.

[35] Carr H., Jenkins C., Moir M., Miraldo V.C., Silva L., "Towards Formal Verification of HotStuff-Based Byzantine Fault Tolerant Consensus in Agda", *NASA Formal Methods (NFM 2022), Proceedings*, 2022, 616–635. DOI: 10.1007/978-3-031-06773-0_33.

[36] Garay J., Kiayias A., Leonardos N., "The Bitcoin Backbone Protocol: Analysis and Applications", *Advances in Cryptology – EUROCRYPT 2015, Lecture Notes in Computer Science*, **9057** (2015), 281–310.

[37] Zhang S., Lee J., "Analysis of the main consensus protocols of blockchain", *ICT Express*, 2019, 1–7. DOI: 10.1016/j.icte.2019.08.001.

[38] Crain T., Gramoli V., Larrea M., Raynal M., "DBFT: Efficient Leaderless Byzantine Consensus and its Application to Blockchains", *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, 2018, 1–8.

Статья поступила 21 января 2026 г.

## Formal Verification and Censorship Resistance of Byzantine Consensus Protocols: An IBFT Case Study

K. V. Ziborov, N. S. Bondarev, Y. A. Yanovich

We present a formal study of the Istanbul BFT (IBFT) protocol family that connects its vulnerable early design, corrected intermediate variants, and a censorship-resistant modification within a unified verification framework. Using TLA+ specifications and TLC model checking, we (i) obtain counterexample traces demonstrating safety violations in the first IBFT version, (ii) verify safety and liveness for a corrected QBFT variant, and (iii) formalize censorship resistance for leader-based Byzantine consensus as a bounded chain-quality requirement and show that standard leader rotation can violate it. We then introduce and verify the *f-skip* leader-selection rule, which enforces censorship resistance while preserving the original safety and liveness guarantees.

**Keywords:** IBFT, QBFT, Byzantine consensus, formal verification, TLA+, TLC, censorship resistance..

# References

[1] Vostrikov D., Madhwal Y., Seoev A., Smirnova A., Yanovich Yu., Smirnov A., Gorgadze V., "Unpacking Maximum Extractable Value on Polygon: A Study on Atomic Arbitrage", *arXiv preprint*, 2025.

[2] Wahrstätter A., Ernstberger J., Yaish A., Zhou L., Qin K., Tsuchiya T., Steinhorst S., Svetinovic D., Christin N., Barczentewicz M., Gervais A., "Blockchain Censorship", *Proceedings of the ACM Web Conference 2024*, 2024, 1632–1643. DOI: 10.1145/3589334.3645431.

[3] Daian P., Goldfeder S., Kell T., Li Y., Zhao X., Bentov I., Breidenbach L., Juels A., "Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability", *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, 910–927. DOI: 10.1109/SP40000.2020.00040.

[4] Heimbach L., Kiffer L., Ferreira Torres C., Wattenhofer R., "Ethereum's Proposer-Builder Separation: Promises and Realities", *Proceedings of the 2023 ACM on Internet Measurement Conference*, 2023, 406–420. DOI: 10.1145/3618257.3624824.

[5] Afanasyeva A., Kameskiy D., Telnov S., Yanovich Yu., "Leaderless Byzantine Fault-Tolerant Consensus Protocol for Blockchains", *Proceedings of the 2023 6th International Conference on Blockchain Technology and Applications*, 2023, 78–84. DOI: 10.1145/3651655.3651665.

[6] Seoev A., Gremyachikh L., Smirnova A., Madhwal Y., Kalacheva A., Belousov D., Zubov I., Smirnov A., Fedyanin D., Gorgadze V., Yanovich Yu., "The Bidding Games: Reinforcement Learning for MEV Extraction on Polygon Blockchain", *arXiv preprint*, 2025.

[7] Öz B., Sui D., Thiery T., Matthes F., "Who Wins Ethereum Block Building Auctions and Why?", *6th Conference on Advances in Financial Technologies*, 2024, 22:1–22:25. DOI: 10.4230/LIPIcs.AFT.2024.22.

[8] Rashid M., Rasool I., Zafar N.A., Afzaal H., "Formal Modeling and Verification of Validator Voluntarily Exit in Ethereum 2.0 Beacon Chain", *2023 2nd International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE)*, 2023, 1–6. DOI: 10.1109/ETECTE59617.2023.10396687.

[9] Rashid M., Rasool I., Zafar N.A., Afzaal H., "Formal Modeling and Verification of Justification and Finalization of Checkpoints in Ethereum 2.0 Beacon Chain", *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, 2024, 1–6. DOI: 10.1109/KHI-HTC60760.2024.10481930.

[10] Cassez F., Fuller J., Asgaonkar A., "Formal Verification of the Ethereum 2.0 Beacon Chain", *Lecture Notes in Computer Science*, **13243** (2022), 167–182. DOI: 10.1007/978-3-030-99524-9_9.

[11] Afzaal H., Zafar N.A., Tehseen A., Kousar S., Imran M., "Formal Verification of Justification and Finalization in Beacon Chain", *IEEE Access*, **12** (2024), 55077–55102. DOI: 10.1109/ACCESS.2024.3389551.

[12] Li E., Serbanuta T., Diaconescu D., Zamfir V., Rosu G., "Formalizing Correct-by-Construction Casper in Coq", *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020, 1–3. DOI: 10.1109/ICBC48266.2020.9169468.

[13] Pass R., Shi E., “Hybrid consensus: Efficient consensus in the permissionless model”, *Leibniz International Proceedings in Informatics (LIPIcs), DISC 2017*, **91. DOI:** 10.4230/LIPIcs.DISC.2017.39 (2017).

[14] Lamport L., *Specifying Systems*, Addison–Wesley, 2002. DOI: 10.5555/579617.

[15] Nakamoto S., “Bitcoin: A peer-to-peer electronic cash system”, *White paper*, 2008.

[16] Lamport L., “TLA+ specification of Paxos”, *Online artifact*, 2008.

[17] Losa G., “TLA+ specification of TetraBFT”, *Online artifact*, 2024.

[18] Saltini R., “QBFT Formal Specification and Verification in Dafny”, *Online artifact*, 2024.

[19] Saltini R., Hyland-Wood D., “IBFT 2.0: A Safe and Live Variation of the IBFT Blockchain Consensus Protocol for Eventually Synchronous Networks”, *arXiv preprint arXiv:1909.10194*, 2019. DOI: 10.48550/arXiv:1909.10194.

[20] França B., Kolegov D., Konnov I., Prusak G., “ChonkyBFT: Consensus Protocol of ZKsync”, *arXiv preprint arXiv:2503.15380*, 2025.

[21] Moniz H., “The Istanbul BFT Consensus Algorithm”, *arXiv preprint arXiv:2002.03613*, 2020. DOI: 10.48550/arXiv.2002.03613.

[22] Buterin V., Griffith V., “Casper the Friendly Finality Gadget”, *CoRR, abs/1710.09437*, 2017.

[23] Buterin V., Hernandez D., Kamphefner T., Pham K., Qiao Z., Ryan D., Sin J., Wang Y., Zhang Y.X., “Combining GHOST and Casper”, *CoRR, abs/2003.03052*, 2020.

[24] Castro M., Liskov B., “Practical Byzantine fault tolerance”, *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI ’99)*, 1999, 173–186. DOI: 10.1145/296806.296824.

[25] Ethereum Community, “Istanbul Byzantine Fault Tolerance (EIP-650)”, *Online proposal*.

[26] ConsenSys, “Quorum Blockchain”, *Online project*.

[27] Saltini R., “Correctness Analysis of IBFT”, *CoRR, abs/1901.07160*, 2019.

[28] Ongaro D., “Consensus: Bridging theory and practice”, *Ph.D. thesis, Stanford University*, 2014.

[29] Rahli V., Vukotic I., Völp M., Esteves-Verissimo P., “Velisarios: Byzantine Fault-Tolerant Protocols Powered by Coq”, *Programming Languages and Systems (ESOP 2018), Lecture Notes in Computer Science*, **10801** (2018), 619–650. DOI: 10.1007/978-3-319-89884-1_22.

[30] Alturki M.A., Chen J., Luchangco V., Moore B., Palmskog K., Peña L., Rosu G., “Towards a verified model of the Algorand consensus protocol in Coq”, *Formal Methods. FM 2019 International Workshops, Lecture Notes in Computer Science*, **12232** (2020), 362–367. DOI: 10.1007/978-3-030-54994-7_27.

[31] Konnov I., Kukovec J., Tran T.-H., “TLA+ model checking made symbolic”, *Proc. ACM Program. Lang.*, **3**:OOPSLA (2019), 123. DOI: 10.1145/3360549.

[32] Konnov I.V., Kuppe M.A., Merz S., “Specification and Verification with the TLA+ Trifecta: TLC, Apalache, and TLAPS”, *Leveraging Applications of Formal Methods*, 2022. DOI: 10.1007/978-3-031-19849-6_6.

[33] Braithwaite S., Buchman E., Konnov I., Milosevic Z., Stoilkovska I., Widder J., Zamfir A., “Formal Specification and Model Checking of the Tendermint Blockchain Synchronization Protocol”, *2nd Workshop on Formal Methods for Blockchains (FMBC 2020), OASIcs*, **84** (2020), 10:1–10:8. DOI: 10.4230/OASIcs.FMBC.2020.10.

[34] Kukharenko V., Ziborov K., Sadykov R., Rezin R., “Verification of HotStuff BFT Consensus Protocol With TLA+/TLC in an Industrial Setting”, *SHS Web of Conferences*, **93** (2021), 01006. DOI: 10.1051/shsconf/20219301006.

[35] Carr H., Jenkins C., Moir M., Miraldo V.C., Silva L., “Towards Formal Verification of HotStuff-Based Byzantine Fault Tolerant Consensus in Agda”, *NASA Formal Methods (NFM 2022), Proceedings*, 2022, 616–635. DOI: 10.1007/978-3-031-06773-0_33.

[36] Garay J., Kiayias A., Leonardos N., “The Bitcoin Backbone Protocol: Analysis and Applications”, *Advances in Cryptology – EUROCRYPT 2015, Lecture Notes in Computer Science*, **9057** (2015), 281–310.

[37] Zhang S., Lee J., “Analysis of the main consensus protocols of blockchain”, *ICT Express*, 2019, 1–7. DOI: 10.1016/j.icte.2019.08.001.

[38] Crain T., Gramoli V., Larrea M., Raynal M., “DBFT: Efficient Leaderless Byzantine Consensus and its Application to Blockchains”, *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, 2018, 1–8.

Received on January 21, 2026