DOI: 10.55648/1998-6920-2025-19-4-110-119 УДК 004.056

Метод обеспечения целостности агрегированных данных мониторинга с использованием технологии блокчейн

И. Р. Федоров

Санкт-Петербургский государственный университет аэрокосмического приборостроения
Национальный исследовательский университет ИТМО

Аннотация: В условиях роста сложности информационных систем и требований к обеспечению информационной безопасности возникает необходимость не только в сборе и анализе мониторинговых данных, но и в гарантированной неизменности этих данных во времени. Традиционные системы мониторинга не обеспечивают защиту от несанкционированных изменений истории метрик, что ограничивает их применение в контексте, требующего прозрачного аудита. В данной работе предлагается решить эту проблему за счет технологии блокчейн. Предложенный метод предполагает ежедневную выборку ключевых метрик мониторинга за прошедшие сутки с последующим хэшированием агрегированных данных. Полученный хэш передаётся в смарт-контракт, развёрнутый в публичном блокчейне Ethereum, где сохраняется вместе с меткой времени. Для верификации реализован программный модуль, повторно извлекающий данные из системы мониторинга и сравнивающий их хэш с зафиксированным в блокчейне значением. В ходе реализации прототипа была достигнута полная автоматизация процесса фиксации и последующей проверки агрегированных метрик. Верификация успешно определяет случаи подмены данных путём сравнения хэшей. Предложенный метод не исключает возможности фиксации подменённых данных в случае компрометации доверенной стороны, однако обеспечивает прозрачную и неизменяемую историю фиксаций, позволяя выявлять нарушения ретроспективно. Преимуществом является независимость от внутренней инфраструктуры организации и возможность верификации с использованием открытых инструментов.

Работа выполнена в рамках государственного задания (проект FSER-2025-0003).

Ключевые слова: блокчейн, информационная безопасность, верификация, система мониторинга, prometheus, ethereum.

Для цитирования: Федоров И. Р. Метод обеспечения целостности агрегированных данных мониторинга с использованием технологии блокчейн // Вестник СибГУТИ. 2025. Т. 19, № 4. С. 110–119. https://doi.org/10.55648/1998-6920-2025-19-4-110-119.

Статья поступила в редакцию 31.05.2025;
переработанный вариант – 30.07.2025;
принята к публикации 23.10.2025.

1. Введение

В условиях стремительного роста сложности информационных систем и увеличения требований к их надёжности и безопасности, системы мониторинга играют ключевую роль в обеспечении устойчивости цифровой инфраструктуры [1]. Такие популярные инструменты, как Prometheus [2], Zabbix [3] и InfluxDB [4], позволяют собирать и анализировать метрики работы систем в реальном времени, обнаруживать аномалии и сбои, а также подтверждать выполнение договорных обязательств по уровню обслуживания.

При этом одной из уязвимых сторон традиционных систем мониторинга остаётся возможность модификации или удаления исторических данных. В сценариях, связанных с расследованием инцидентов, регуляторным аудитом или судебными спорами, возникает необходимость гарантировать неизменность зафиксированных метрик задним числом. Это особенно актуально для организаций, работающих в сферах с повышенными требованиями к безопасности и прозрачности, например, в финансовом секторе, государственных учреждениях и в прочих критических инфраструктурах.

Одним из возможных решений данной проблемы является использование технологии блокчейн для анкоринга мониторинговых данных. В данной статье предлагается концепция мониторинга инфраструктуры, в рамках которой агрегированные метрики, собранные системой мониторинга, периодически хэшируются, а результат хэширования записывается в публичный блокчейн Ethereum [5]. Такой подход позволяет верифицировать подлинность и неизменность исторических данных, не раскрывая их содержимого, что делает возможным как соблюдение требований регуляторов, так и защиту конфиденциальной информации.

2. Обзор существующих решений

Системы мониторинга и наблюдаемости (observability) являются неотъемлемой частью при построении современной IT-инфраструктуры. В наиболее популярных на сегодняшний день решениях (например, Prometheus или InfluxDB), как правило, используются специализированные базы данных временных рядов (time series databases, TSDB), оптимизированные для записи большого количества метрик с высокой частотой. Для повышения надёжности и отказоустойчивости часто применяются репликация, резервное копирование и другие механизмы, обеспечивающие сохранность данных в случае сбоя. Однако подобные меры не решают проблему доверия к хранимым данным, особенно в контексте расследования инцидентов. В условиях, когда системный администратор или злоумышленник может иметь доступ к хранилищу метрик, отсутствуют технические гарантии, что данные не были подменены, удалены или модифицированы задним числом.

В целях гарантии неизменности данных некоторые организации используют:

1. WORM-хранилища (write once, read many), однако их стоимость и ограничения делают их не всегда применимыми [6].
2. SIEM-системы, которые фокусируются больше на событиях безопасности, чем на агрегированных метриках [7].
3. Архивирование и подпись бэкапов, что создаёт дополнительную нагрузку и требует доверия к подписывающей стороне.

Также следует отметить существующие решения в области применения технологии блокчейн с целью хранения данных для аудита, однако применение большинства из них ориентировано на определенную сферу деятельности (например, логистика или финтех), в то время как метрики мониторинга как объект аудита остаются слабо охваченными [8, 9, 10].

Таким образом, на сегодняшний день отсутствуют широкодоступные и масштабируемые решения, обеспечивающие подтверждаемую криптографически неизменность агрегированных метрик мониторинга. В данной работе для решения этой задачи предлагается использовать технологию блокчейн, так как она идеально подходит для сценариев, где требуется обеспечить прозрачность, неизменность и доверие к зафиксированным данным.

3. Анкоринг в блокчейне

Анкоринг (от англ. anchoring) – это процесс сохранения моментального снимка состояния системы в публичный источник, благодаря которому нет необходимости безоговорочно доверять администратору системы [11, 12]. Публичным источником в рамках данной работы будет выступать открытый блокчейн Ethereum. В отличие от полного хранения информации в блокчейне, анкоринг предполагает публикацию лишь короткого,

уникального криптографического отпечатка (например, SHA-256 хэша), который однозначно соответствует оригинальному содержимому. Таким образом достигается баланс между неизменяемостью, масштабируемостью и приватностью.

Публикация хэша в блокчейн выполняет две ключевые функции:

1. Timestamping (доказуемое время фиксации): благодаря такому свойству блокчейн, как неизменяемость, можно установить, что определённый набор данных существовал в конкретный момент времени.
2. Integrity verification (проверка целостности): если оригинальные данные были модифицированы хотя бы на один бит, их хэш изменится, в следствие чего сравнение с сохраненным в блокчейне значением это покажет.

Применение анкоринга можно встретить в различных сферах: логистика [13], образование [14], финтех и др. [15].

В контексте мониторинга анкоринг позволит создать «криптографический слепок» агрегированных метрик (например, суммарное количество ошибок за сутки) и зафиксировать его в блокчейне. При последующем аудите можно повторно вычислить хэш от тех же данных и убедиться, что они не были подменены. Таким образом, система получает доверенную внешнюю верификацию целостности при минимальных издержках.

Выбор блокчейна зависит от требований к стоимости, производительности, доступности и надёжности [16]. Существуют следующие виды блокчейна: публичный блокчейн (предоставляет высокий уровень доверия и децентрализации), частный блокчейн (есть возможность контроля доступа и более высокая скорость по сравнению с публичным).

Иногда также применяется гибридный подход: например, хранение хэша осуществляется в публичном блокчейне, а сами данные хранятся в IPFS (InterPlanetary File System) [17] или в S3-хранилище [18].

В рамках текущей работы был сделан выбор в пользу Ethereum ввиду простоты интеграции, так как в данном случае не требуется разворачивать платформу с нуля, можно воспользоваться уже готовой тестовой сетью. В следующем разделе будет представлена архитектура прототипа, реализующего подход анкоринга поверх системы мониторинга Prometheus.

4. Архитектура прототипа решения

На рис. 1 представлена архитектура прототипа, реализующего предложенный подход обеспечения неизменности мониторинговых данных путём их анкоринга в публичный блокчейн Ethereum. Архитектура состоит из четырёх основных компонентов, объединённых в единую цепочку формирования и верификации целостности метрик.

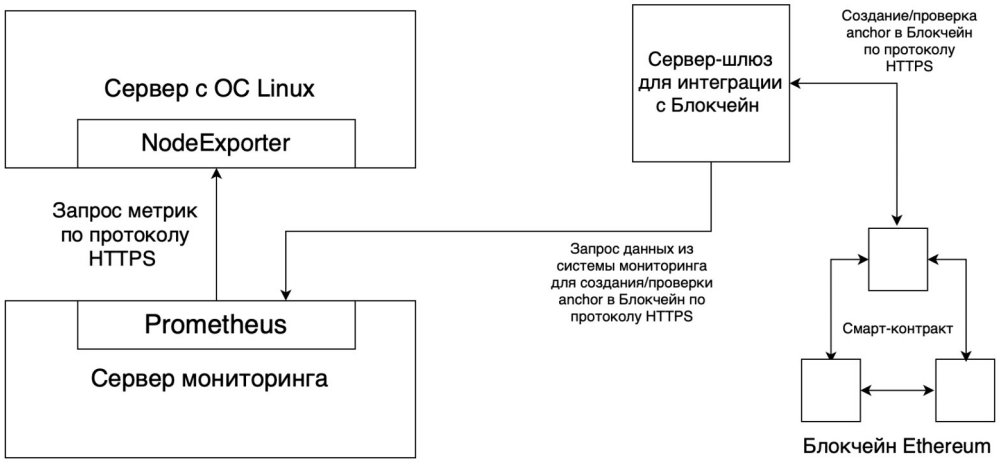

Рис. 1. Общая архитектура прототипа решения

4.1. Система мониторинга и сбора метрик

В качестве системы мониторинга используется Prometheus. За сбор системных метрик отвечает Node Exporter [19]. Node Exporter развёрнут на отдельном сервере Linux и экспортирует базовые метрики состояния операционной системы: загрузка CPU, утилизация оперативной памяти, место в файловой системе и др. На сервере мониторинга установлен Prometheus, который, в свою очередь, производит опрос Node Exporter по протоколу HTTPS с заданной периодичностью (по умолчанию каждые десять секунд) и сохраняет полученные метрики в своей time series database (TSDB). Также Prometheus предоставляет API доступ (/api/v1/query_range) для извлечения агрегированных метрик за нужный период времени.

4.2. Модуль анкоринга

Данный модуль реализован на языке программирования высокого уровня Python и может запускаться как вручную, так и по расписанию через планировщик задач (например, cron). Задача модуля – зафиксировать текущее состояние метрик в блокчейне. Процесс включает следующие этапы:

1. Формирование диапазона времени для предыдущих суток (например, с 00:00 до 23:59 в UTC).
2. Выполнение запроса к Prometheus с использованием вызова query_range.
3. Сериализация результата в строковое представление.
4. Вычисление SHA-256 хэша от сериализованных данных.
5. Вызов функции anchorHash(hash_string) смарт-контракта с передачей рассчитанного хэша.

Псевдокод модуля приведён в алгоритме 1. В результате выполнения транзакции в смарт-контракте создаётся запись с хэшем и меткой времени. Данная информация сохраняется в блокчейне и может быть проверена в будущем.

Алгоритм 1. ПРОЦЕДУРА АНКОРИНГА МЕТРИК В БЛОКЧЕЙНЕ

Входные данные: URL Prometheus-сервера P, адрес смарт-контракта C,
приватный ключ K
Выходные данные: Хэш агрегированных метрик за сутки
Вычислить временные параметры и сфомировать строку запроса к Prometheus
$T_{start} \leftarrow$ `UTC_date(now)` − 1 сутки
$T_{end} \leftarrow$ `UTC_date(now)`
$q \leftarrow$ строка запроса к Prometheus
Выполнить HTTP GET-запрос к API Prometheus
$D \leftarrow$ `GET`$(P$`/api/v1/query_range`$, \{q, T_{start}, T_{end}\})$
Сериализовать результат D в строку S
$H \leftarrow$ `SHA256`(S)
Построить транзакцию вызова смарт-контракта
$tx \leftarrow C$.`anchorHash`(H)
Отправить транзакцию: `sendTransaction`(tx, K)

4.3. Смарт-контракт в Ethereum

Смарт-контракт реализован на специализированном языке программирования Solidity [20] и развёрнут в тестовой сети Ethereum Sepolia [21]. Он содержит:

1. функцию anchorHash(string), сохраняющую хэш и текущую метку времени в массив записей;
2. функцию getByDate(uint256), возвращающую хэш для указанной даты (округлённой до начала суток);
3. вспомогательная функция totalRecords(), возвращающая количество сохраненных хэшей.

Псевдокоды функций anchorHash и getByDate приведены в алгоритмах 2 и 3 соответственно. Смарт-контракт обеспечивает неизменность и верифицируемость сохранённой информации, а также возможность независимого аудита через обозреватель Etherscan или API [22].

Алгоритм 2. ФУНКЦИЯ ANCHORHASH(HASH) В СМАРТ-КОНТРАКТЕ

Входные данные: Строка *hash* (хэш SHA256 агрегированных метрик)
Выходные данные: Хэш и текущая метка времени добавлены в историю
timestamp ← `block.timestamp`
record ← структура {`hash:` *hash*, `timestamp:` *timestamp*}
Добавить *record* в массив *records*
Эмитировать событие `Anchored`*(hash, timestamp)*

Алгоритм 3. ФУНКЦИЯ GETBYDATE(UINT256) В СМАРТ-КОНТРАКТЕ

Входные данные: Дата *D*, за которую нужно получить хэш
Выходные данные: Хэш, метка времени
если *records.length = 0* **или** *records[D] = NULL*
  Возвратить ошибку: `No records`
иначе
  record ← *records[D]*
  вернуть (**record.hash, record.timestamp**)

4.4. Модуль верификации (verifier)

Модуль верификации также реализован на языке программирования высокого уровня Python и выполняет аналогичный запрос к Prometheus за те же сутки, как и модуль анкоринга. Процесс включает следующие этапы:

1. вычисление локального хэша текущих метрик за указанный диапазон времени;
2. получение хэша за эти же сутки из блокчейна через функцию смарт-контракта getByDate(uint256);
3. сравнение обоих значений.

Результат может быть представлен как булев флаг: true (данные не подменялись) или false (имеются расхождения). Это позволяет выявить ретроспективные изменения в данных мониторинга и использовать механизм для внешнего аудита. Псевдокод модуля приведён в алгоритме 4.

Алгоритм 4. ПРОЦЕДУРА ВЕРИФИКАЦИИ МЕТРИК НА ОСНОВЕ ХЭША ИЗ БЛОКЧЕЙНА

Входные данные: URL Prometheus-сервера *P*, адрес смарт-контракта *C*,
временной интервал *(Tstart , Tend)*
Выходные данные: Булево значение совпадения хэшей агрегированных метрик
D ← вычисление даты T_{start}, T_{end}
q ← строка запроса к Prometheus
Выполнить HTTP GET-запрос к API Prometheus
A ← `GET`*(**P**/`api/v1/query_range`, {**q**, T_{start}, T_{end}})*
Сериализовать результат *A* в строку *S*
H_{local} ← `SHA256`*(S)*
Получить нужный хэш из контракта:
H_{chain} ← *C.*`getByDate`*(D)*
если $H_{local} = H_{chain}$
  вернуть *true*
иначе
  вернуть *false*

Важным элементом архитектуры является защита от повторной подмены: если злоумышленник изменит метрики задним числом и повторно вызовет anchorHash(), он создаст новую запись, но не сможет перезаписать уже зафиксированный ранее хэш. Функция getByDate() позволяет однозначно сопоставить дату и ожидаемый хэш, даже если в контракте появилось несколько записей. Однако существует также ряд ограничений, которые необходимо учитывать при внедрении в реальных условиях:

1. Компрометация доверенной стороны. Так как одним из ключевых элементов архитектуры является модуль анкоринга, в случае его компрометации злоумышленник может сфальсифицировать метрики до формирования хэша и фиксации данных в блокчейне. Таким образом, в блокчейн будут попадать заведомо ложные данные. Одним из вариантов решения данной проблемы может быть добавление "избыточности" и запуск нескольких модулей анкоринга, работающих в кворуме. Также рекомендуется запускать модуль в защищенной среде выполнения.

2. Отсутствие неизменности до момента анкоринга. Значения метрик можно изменить в системе мониторинга до момента формирования хэша и записи его в блокчейн. В случае, если злоумышленник получит доступ к хранилищу системы мониторинга до запуска модуля анкоринга, у него будет возможность изменить данные задним числом. В качестве решения предлагается сократить период анкоринга (например, до 1 часа).

3. Стоимость и масштабируемость. Запись транзакции в публичный блокчейн подразумевает финансовые затраты (газ). Стоимость транзакции может варьироваться от 0.2 до 2 долларов в зависимости от загруженности сети и текущей цены газа. В периоды повышенной активности комиссия может временно достигать 5 долларов и выше за транзакцию. При высокой частоте анкоринга решение может стать экономически нецелесообразным, поэтому в дальнейших исследованиях планируется рассмотреть альтернативные блокчейн платформы, например, Polygon [23], Tezos [24] или Hyperledger Fabric [25]. Ключевые отличия перечисленных блокчейн-платформ представлены в таблице 1.

Таблица 1. Сравнение блокчейн-платформ

Платформа	**Класс**	**EVM-совместимость**	**Размер комиссии**
Ethereum	Публичный	Есть	от 0.2
Polygon	Публичный	Есть	от 0.01
Tezos	Публичный	Нет	от 0.005
Hyperledger Fabric	Частный	Нет	0

Согласно таблице 1 можно отметить, что Ethereum и Polygon поддерживают EVM, что делает возможным прямой перенос смарт-контрактов между этими платформами. Это существенно упрощает реализацию и поддержку решений, изначально разработанных для Ethereum. Polygon при этом предлагает значительно более низкую стоимость транзакций, что делает его более предпочтительным вариантом при высокой частоте запуска модуля анкоринга.

Tezos и Hyperledger Fabric не поддерживают EVM, следовательно, миграция на эти платформы требует переписывания логики смарт-контрактов. Однако они также демонстрируют низкую стоимость транзакций, что делает их потенциально привлекательными для использования в случаях с высокими требованиями к масштабируемости. Hyperledger Fabric отличается также тем, что разворачивается как частный блокчейн. Такой класс блокчейнов подходит для закрытых инфраструктур, где публичная верификация данных не требуется. Это позволяет полностью избежать издержек, связанных с использованием публичных платформ, а также обеспечивает более строгий контроль над доступом и управлением.

5. Заключение

В данной статье был предложен и реализован метод обеспечения целостности агрегированных данных мониторинга на основе механизма анкоринга хэшей в публичный блокчейн. Подход позволяет обеспечить доказуемую неизменность метрик, собранных системой мониторинга Prometheus, с возможностью последующей верификации путём повторного хэширования и сопоставления с ранее зафиксированным значением в блокчейне. Разработанный прототип включает Python-модули для анкоринга и верификации, а также смарт-контракт на языке Solidity, развёрнутый в тестовой сети Ethereum Sepolia.

В качестве дальнейших направлений исследования планируется:

1. Реализация поддержки множественных источников метрик (Zabbix, InfluxDB и др.).
2. Реализация поддержки безопасной верификации через web-интерфейс или с помощью инструмента Grafana [26].
3. Реализация поддержки альтернативных способов хранения, включая использование IPFS для архивных данных и zk-SNARK доказательств вместо открытых хэшей.
4. Проведение масштабных нагрузочных тестов, включая анализ влияния объёма метрик на производительность и стоимость транзакций в основной сети Ethereum.
5. Реализация поддержки альтернативных блокчейн-платформ (Polygon, Hyperledger Fabric).

Предложенный в работе подход может быть полезен в системах, где целостность мониторинга критична – например, в финансовом секторе и государственных органах.

Литература

1. *Hamou-Lhadj W.* Observability of Software Computing Systems: Challenges and Opportunities // 2022 3rd International Conference on Embedded & Distributed Systems (EDiS). Oran, Algeria, 2022, p. 5-5.
2. Prometheus [Электронный ресурс] // URL: `https://prometheus.io` (дата обращения: 18.09.2025).
3. Zabbix [Электронный ресурс] // URL: `https://www.zabbix.com/ru` (дата обращения: 18.09.2025).
4. InfluxDB [Электронный ресурс] // URL: `https://www.influxdata.com` (дата обращения: 18.09.2025).
5. Arigela S. S. D., Voola P. Blockchain Open Source Tools: Ethereum and Hyperledger Fabric. *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF),* Chennai, India, 2023, P. 1-8.
6. Hsu W. W., Ong S. WORM storage is not enough [Technical Forum]. *IBM Systems Journal,* 2007, v. 46, № 2. pp. 363-369.
7. SIEM-системы в управлении информационной безопасностью: учебное пособие / С. В. Беззатеев, С. Г. Фомичева. Санкт-Петербург: ГУАП, 2021. 131 с.
8. Fahrezy M. D., Tjahyadi R., Kurniawati H. Blockchain Adoption in Financial Audit: A Review. *2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS),* Bandung, Indonesia, 2025, p. 1-6.
9. Liu Z., Zhang X., Li G., Cui H.,Wang J., Xiao B. A Secure and Reliable Blockchain-based Audit Log System. *2024 IEEE International Conference on Communications*, Denver, CO, USA, 2024, p. 2010-2015.
10. Younas A., Kassim A. A. M. A Review of Auditor's Role in Blockchain-Based Supply Chain Management. *2024 1st International Conference on Logistics (ICL),* Jeddah, Saudi Arabia, 2024, p. 1-7.

11. Korepanova D., Nosyk M., Ostrovsky A., Yanovich Y. Building a Private Currency Service Using Exonum. *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom),* Sochi, Russia, 2019, p. 1-3.
12. Prada-Delgado M. A., Dittmann G., Circiumaru I., Jelitto J. A Blockchain-Based Crypto-Anchor Platform for Interoperable Product Authentication. *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, Daegu, Korea, 2021, pp. 1-5.
13. *Федоров И. Р.* Применение технологии блокчейн в управлении цепочками поставок // Сборник трудов IX конгресса молодых ученых, 15 апреля – 18 апреля 2020. С. 144-146.
14. Блокчейн-платформа для образования [Электронный ресурс] // URL: `https://www.comnews.ru/digital-economy/content/112755/2018-04-19/blokcheyn-platforma-dlya-obrazovaniya-zarabotaet-v-novom-uchebnom-godu` (дата обращения: 18.09.2025).
15. *Беззатеев С. В., Федоров И. Р., Федосенко М. Ю.* Перспективы внедрения технологии блокчейн в производственные процессы отечественных компаний // Проблемы информационной безопасности. Компьютерные системы. 2022. Т. 51, № 3. С. 96-120.
16. *Fedorov I. R., Pimenov A. V., Panin G. A., Bezzateev S. V.* Blockchain in 5G Networks: Perfomance Evaluation of Private Blockchain // Wave Electronics and its Application in Information and Telecommunication Systems (WECONF 2021). Saint-Peresburg, 2021, P. 9470519.
17. Zheng Q., Li Y., Chen P., Dong X. An Innovative IPFS-Based Storage Model for Blockchain. *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Santiago, Chile, 2018, p. 704-708.
18. Khande R., Rajapurkar S., Barde P., Balsara H., Datkhile A. Data Security in AWS S3 Cloud Storage. *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT),* Delhi, India, 2023, p.1-6.
19. Node Exporter, GitHub [Электронный ресурс] // URL: `https://github.com/prometheus/node_exporter` (дата обращения: 18.09.2025).
20. Solidity Programming Language [Электронный ресурс] // URL: `https://soliditylang.org` (дата обращения: 18.09.2025).
21. Sepolia Testnet [Электронный ресурс] // URL: `https://sepolia.dev` (дата обращения: 18.09.2025).
22. Etherscan [Электронный ресурс] // URL: `https://etherscan.io` (дата обращения: 18.09.2025).
23. Aung M. T., Thein N. N. M. A Comparative Study of Ethereum and Polygon for Implementing NFT-Based Certification Systems. *2024 5th International Conference on Advanced Information Technologies (ICAIT),* Yangon, Myanmar, 2024, p. 1-6.
24. Allombert V., Bourgoin M., Tesson J. Introduction to the Tezos Blockchain. *2019 International Conference on High Performance Computing & Simulation (HPCS),* Dublin, Ireland, 2019, p. 1-10.
25. Kaushal R. K., Kumar N. Blockchain Implementation with Hyperledger Fabric and Approach for Performance Evaluation. *2023 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS),* New Raipur, India, 2023, p. 1-5.
26. Grafana [Электронный ресурс] // URL: `https://grafana.com` (дата обращения: 18.09.2025).

Федоров Иван Романович

к.т.н., доцент кафедры информационной безопасности, Санкт-Петербургский государственный университет аэрокосмического приборостроения (ГУАП, 190000, Санкт-Петербург, ул. Большая Морская, д. 67), тел. +7 995 598 40 16, e-mail: `ivanfedorov@guap.ru`, ORCID ID: 0000-0003-2422-4714.

Авторы прочитали и одобрили окончательный вариант рукописи.

Method for Ensuring Integrity of Aggregated Monitoring Data Using Blockchain Technology

Ivan R. Fedorov

Saint-Petersburg State University of Aerospace Instrumentation (SUAI)
ITMO University

Abstract: As the complexity of information systems increases and the requirements for information security become more stringent, there arises a need not only for the collection and analysis of monitoring data but also for ensuring their immutability over time. Traditional monitoring systems do not provide protection against unauthorized modifications of metric history, which limits their applicability in contexts that require transparent auditing. This work proposes to address this issue using blockchain technology. The proposed approach involves daily extraction of key monitoring metrics for the previous day, followed by hashing of the aggregated data. The resulting hash is transmitted to a smart contract deployed on the public Ethereum blockchain, where it is stored along with a timestamp. For verification, a software module was developed that retrieves data from the monitoring system and compares their hash with the one stored on the blockchain. A prototype implementation has achieved full automation of the process of recording and subsequently verifying aggregated metrics. The verification procedure reliably detects any data tampering by comparing hashes. While the proposed method does not eliminate the risk of recording tampered data in the event of a trusted party being compromised, it does ensure a transparent and immutable history of recordings, allowing for retrospective detection of violations. A key advantage is its independence from the organization's internal infrastructure and the ability to perform verification using open tools.

Keywords: blockchain, information security, verification, monitoring system, prometheus, ethereum.

For citation: Fedorov I. R. Method for Ensuring Integrity of Aggregated Monitoring Data Using Blockchain Technology. *Vestnik SibGUTI*, 2025, vol. 19, no. 4, pp. 110-119. https://doi.org/10.55648/1998-6920-2025-19-4-110-119.

The article was submitted: 31.05.2025;
revised version: 30.07.2025;
accepted for publication 23.10.2025.

References

1. Hamou-Lhadj W. Observability of Software Computing Systems: Challenges and Opportunities. *2022 3rd International Conference on Embedded & Distributed Systems (EDiS)*, Oran, Algeria, 2022, p. 5.
2. Prometheus, available at: https://prometheus.io (accessed: 18.09.2025).
3. Zabbix, available at: https://www.zabbix.com/ru (accessed: 18.09.2025).
4. InfluxDB, available at: https://www.influxdata.com (accessed: 18.09.2025).
5. Arigela S. S. D., Voola P. Blockchain Open Source Tools: Ethereum and Hyperledger Fabric. *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF),* Chennai, India, 2023, P. 1-8.
6. Hsu W. W., Ong S. WORM storage is not enough [Technical Forum]. *IBM Systems Journal,* 2007, v. 46, № 2. pp. 363-369.
7. SIEM-systems in information security management. S. V. Bezzateev, S. G. Fomicheva. Saint-Petersburg: SUAI, 2021, p. 131.

8. Fahrezy M. D., Tjahyadi R., Kurniawati H. Blockchain Adoption in Financial Audit: A Review. *2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS),* Bandung, Indonesia, 2025, p. 1-6.
9. Liu Z., Zhang X., Li G., Cui H.,Wang J., Xiao B. A Secure and Reliable Blockchain-based Audit Log System. *2024 IEEE International Conference on Communications*, Denver, CO, USA, 2024, p. 2010-2015.
10. Younas A., Kassim A. A. M. A Review of Auditor's Role in Blockchain-Based Supply Chain Management. *2024 1st International Conference on Logistics (ICL),* Jeddah, Saudi Arabia, 2024, p. 1-7.
11. Korepanova D., Nosyk M., Ostrovsky A., Yanovich Y. Building a Private Currency Service Using Exonum. *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom),* Sochi, Russia, 2019, p. 1-3.
12. Prada-Delgado M. A., Dittmann G., Circiumaru I., Jelitto J. A Blockchain-Based Crypto-Anchor Platform for Interoperable Product Authentication. *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, Daegu, Korea, 2021, pp. 1-5.
13. Fedorov I. R. Application Of Blockchain Technology In Supply Chain Managemenet. *Collection of works of the IX Congress of young scientists*, 15 april – 18 april, 2020. pp. 144-146.
14. Blockchain platform for education, available at: `https://www.comnews.ru/digital-economy/content/112755/2018-04-19/blokcheyn-platforma-dlya-obrazovaniya-zarabotaet-v-novom-uchebnom-godu` (accessed: 18.09.2025).
15. Bezzateev S. V., Fedorov I. R., Fedosenko M. Y. The Perspective For Introduction Of Blockchain Technology Into The Production Processes Of Russian Companies. Information Security Problems. Computer Systems, 2022, v. 51, № 3. pp. 96-120.
16. Fedorov I. R., Pimenov A. V., Panin G. A., Bezzateev S. V. Blockchain in 5G Networks: Perfomance Evaluation of Private Blockchain. *Wave Electronics and its Application in Information and Telecommunication Systems (WECONF 2021),* Saint-Peresburg, 2021, P. 9470519
17. Zheng Q., Li Y., Chen P., Dong X. An Innovative IPFS-Based Storage Model for Blockchain. *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Santiago, Chile, 2018, p. 704-708.
18. Khande R., Rajapurkar S., Barde P., Balsara H., Datkhile A. Data Security in AWS S3 Cloud Storage. *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT),* Delhi, India, 2023, p.1-6.
19. Node Exporter, GitHub, available at: `https://github.com/prometheus/node_exporter` (accessed: 18.09.2025).
20. Solidity Programming Language, available at: `https://soliditylang.org` (accessed: 18.09.2025).
21. Sepolia Testnet, available at: `https://sepolia.dev` (accessed: 18.09.2025).
22. Etherscan, available at: `https://etherscan.io` (accessed: 18.09.2025).
23. Aung M. T., Thein N. N. M. A Comparative Study of Ethereum and Polygon for Implementing NFT-Based Certification Systems. *2024 5th International Conference on Advanced Information Technologies (ICAIT),* Yangon, Myanmar, 2024, p. 1-6.
24. Allombert V., Bourgoin M., Tesson J. Introduction to the Tezos Blockchain. *2019 International Conference on High Performance Computing & Simulation (HPCS),* Dublin, Ireland, 2019, p. 1-10.
25. Kaushal R. K., Kumar N. Blockchain Implementation with Hyperledger Fabric and Approach for Performance Evaluation. *2023 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS),* New Raipur, India, 2023, p. 1-5.
26. Grafana, available at: `https://grafana.com` (accessed: 18.09.2025).

Ivan R. Fedorov

PhD (Engineering), Associate Professor of the Department of Information Security, Saint-Petersburg State University of Aerospace Instrumentation (SUAI, Russia, 190000, Saint Petersburg, Bolshaya Morskaya street, 67), phone: +7 995 598 40 16, e-mail: `ivanfedorov@guap.ru`, ORCID ID: 0000-0003-2422-4714.