Научная статья
УДК 004.056.2
https://doi.org/10.24143/2072-9502-2025-3-94-101
EDN ESXNZX

Метод проверки целостности журналов действий пользователей в облачных сервисах

В. В. Еремук[1]✉, В. А. Горошков[2], А. В. Касьянов[3], В. А. Ромашов[4], Д. П. Островский[5]

[1, 2, 4, 5]Национальный исследовательский университет ИТМО,
Санкт-Петербург, Россия, polar.vl@yandex.ru✉

[3]Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина),
Санкт-Петербург, Россия

Аннотация. Журналы действий пользователей являются одним из ключевых источников информации при расследовании инцидентов информационной безопасности в распределенных и облачных вычислительных системах. В ходе кибератак злоумышленники могут выполнять модификацию или удаление записей журналов действий пользователей с целью усложнения расследования инцидента. Аудит информационной безопасности часто осуществляется сторонними организациями, не имеющими легитимного доступа к пользовательским данным, из-за чего проверка целостности журналов действий пользователей без раскрытия данных журналов может быть невозможна, в то время как раскрытие данных журналов может нести юридические риски для поставщика облачных сервисов или организации-потребителя услуг. Объектом исследования являются журналы действий пользователей, размещенные в инфраструктуре облачных сервисов. Предмет исследования – методы и средства, обеспечивающие проверку целостности журналов действий пользователей без раскрытия содержимого журналов. Представлен метод на основе технологии блокчейн, позволяющий выполнять проверку целостности журналов действий пользователей без доступа к информации, хранящейся в журналах. Предложен метод организации хранения файлов журналов для внедрения разработанного метода. Описаны основные компоненты предлагаемого метода и процедуры взаимодействия между ними, включая процедуру инициализации нового блока, процедуру записи полезной нагрузки, процедуру записи конца блока. Описаны процедуры, выполняемые сторонним аудитором при выполнении проверок целостности журналов действий пользователей. Предусмотрена возможность выборочной проверки целостности на основе опубликованных хеш-сумм и структур данных типа «хеш-дерево», без необходимости доступа к данным, хранящимся в журналах. Разработанный метод может быть интегрирован в существующие распределенные и облачные вычислительные системы и может применяться как при расследовании инцидентов информационной безопасности, так и в ходе плановых проверок.

Ключевые слова: облачный сервис, распределенные вычисления, блокчейн, проверка целостности, расследование инцидентов информационной безопасности

Для цитирования: *Еремук В. В., Горошков В. А., Касьянов А. В., Ромашов В. А., Островский Д. П.* Метод проверки целостности журналов действий пользователей в облачных сервисах // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2025. № 3. С. 94–101. https://doi.org/10.24143/2072-9502-2025-3-94-101. EDN ESXNZX.

Original article

A method for verifying the integrity of user activity logs in cloud services

V. V. Eremuk[1]✉, V. A. Goroshkov[2], A. V. Kasyanov[3], V. A. Romashov[4], D. P. Ostrovsky[5]

[1, 2, 4, 5]ITMO University,
Saint Petersburg, Russia, polar.vl@yandex.ru✉

[3]Saint Petersburg Electrotechnical University,
Saint Petersburg, Russia

Abstract. User activity logs are one of the key sources of information when investigating information security incidents in distributed and cloud computing systems. During a cyberattack, attackers can modify or delete user activity log entries in order to complicate the investigation of the incident. Information security audits are often carried out by

third-party organizations that do not have legitimate access to user data, which may make it impossible to verify the integrity of user activity logs without disclosing log data, while disclosing log data may carry legal risks for a cloud service provider or a service consumer organization. The object of the study is user activity logs hosted in the cloud services infrastructure. The subject of the research is methods and tools that ensure the integrity of user activity logs without disclosing the contents of the logs. A method based on blockchain technology is presented that makes it possible to verify the integrity of user activity logs without access to information stored in the logs. A method of organizing the storage of log files for the implementation of the developed method is proposed. The main components of the proposed method and the procedures for interaction between them are described, including the procedure for initializing a new block, the procedure for recording the payload, and the procedure for recording the end of the block. The procedures performed by a third-party auditor when performing integrity checks of user activity logs are described. It is possible to selectively verify integrity based on published hash sums and hash tree data structures, without the need to access data stored in logs. The developed method can be integrated into existing distributed and cloud computing systems and can be used both in the investigation of information security incidents and during routine inspections.

Keywords: cloud services, distributed computing, blockchain, integrity verification, investigation of information security incidents

For citation: Eremuk V. V., Goroshkov V. A., Kasyanov A. V., Romashov V. A., Ostrovsky D. P. A method for verifying the integrity of user activity logs in cloud services. *Vestnik of Astrakhan State Technical University. Series: Management, computer science and informatics*. 2025;3:94-101. (In Russ.). https://doi.org/10.24143/2072-9502-2025-3-94-101. EDN ESXNZX.

Введение

Актуальность исследований обоснована растущим спросом на услуги поставщиков облачных сервисов, что сопровождается риском утечки и/или несанкционированного доступа к конфиденциальной информации. Согласно исследованию компании ИКС-ХОЛДИНГ, объем российского рынка облачных хранилищ в 2024 г. вырос до 165,6 млрд руб., что соответствует росту на 36,3 % относительно показателей 2023 г. [1]. Согласно данным компании Red Canary [2], в 2024 г. количество атак на поставщиков услуг облачных сервисов по всему миру возросло в 16 раз.

Одним из важнейших инструментов в процессе расследования инцидентов информационной безопасности (ИБ) в облачных сервисах являются журналы действий пользователей. В процессе атаки злоумышленники могут удалять журналы либо вносить в них недостоверные сведения, усложняя процесс расследования инцидентов ИБ. Таким образом, актуальной задачей является разработка метода проверки целостности журналов действий пользователей с целью обеспечения достоверности результатов, получаемых в ходе анализа журналов.

Проверка целостности журналов действий пользователей может выполняться не только в процессе расследования инцидентов ИБ, но и в ходе аудита ИБ. Поскольку аудит может выполняться сторонней организацией, важной задачей является обеспечение возможности проверки целостности журналов без раскрытия содержимого журналов и данных, над которыми выполнялись операции.

Объектом исследования являются журналы действий пользователей, хранящиеся в облачных сервисах, предметом исследования являются методы и средства проверки их целостности. В статье предложен метод, основанный на технологии блокчейн, позволяющий обеспечивать возможность проверки целостности журналов действий пользователей без раскрытия их содержимого. Описан способ организации хранения файлов журналов в облачных сервисах для поддержки возможности интеграции предложенного метода. Предложенный метод позволяет обеспечить гарантию достоверности результатов, получаемых при анализе журналов действий пользователей и, таким образом, востребован в области расследования инцидентов ИБ.

Структура файла журнала

Пусть G_1 и G_2 – циклическая мультипликативная группа большого простого порядка p. Пусть u, g – порождающие элементы группы G_1. Пусть $H: \{0,1\}^* \to G_1$ – криптографическая хеш-функция. Обозначим множество, состоящее из $U_{\max}$ пользователей, как $U = \{U_1, U_2, \ldots, U(u_{\max})\}$. Каждому пользователю U_k (где $1 \le k \le u_{\max}$) соответствует уникальный идентификатор пользователя u_k. Каждый пользователь U_k генерирует закрытый ключ $s_{u_k} \in Z_p$ и открытый ключ $p_{u_k} = g^{s_{u_k}}$. Поставщик облачных сервисов генерирует закрытый ключ $s_c \in Z_p$ и открытый ключ $p_c = g^{s_c}$. Каждый файл журнала F состоит из $b_{\max}$ последовательных блоков, т. е. $F = \left\{B_1, B_2, \ldots, B_{b_{\max}}\right\}$. Каждый блок состоит из $e_{\max}$ записей $e_{i,j}$, где i – номер блока, j – порядковый номер записи, т. е.

$$B_i = \left\{e_{i,1}, e_{i,2}, \ldots, e_{i,e_{\max}}\right\}.$$

Каждая запись журнала $e_{i,j} (1 \le j \le e_{\max})$ состоит из следующих элементов: идентификатор блока $\beta_{i,j}$ – порядковый номер блока в файле журнала, идентификатор записи $\xi_{i,j}$ – порядковый номер записи в блоке, идентификатор пользователя $\eta_{i,j}$, соответствующий некоторому u_k, метка времени $t_{i,j}$, полезная нагрузка $p_{i,j}$, содержащая сведения о выполненной операции.

Обозначим результат конкатенации полей β_i, $\xi_{i,j}$, u_k, T, $P_{i,j}$ записи $e_{i,j}$ как $\mu_{i,j}$:

$$\mu_{i,j} = \left(\beta_{i,j} \left\| \xi_{i,j} \right\| \eta_{i,j} \left\| t_{ij} \right\| p_{i,j}\right).$$

Служебные записи журнала (первая и последняя в блоке) имеют структуру, аналогичную другим записям, однако не содержат сведений о действиях пользователя и в качестве полезной нагрузки содержат строки BEGIN и END соответственно. В качестве транзакций в рамках настоящего исследования используются метки записей журнала. Метки записей, хранимые в одном блоке файла журнала B, должны быть записаны в виде транзакций в один блок C в блокчейне. При этом блок C не должен содержать другие транзакции. В рамках данного исследования не рассматриваются детали реализации хранения данных блокчейна на отдельных узлах сети, а также особенности конкретных технологий, используемых для реализации блокчейн-системы.

Поставщик услуг предоставляет потребителю (организации) [3] доступ к вычислительным ресурсам. Потребитель предоставляет доступ к вычислительным ресурсам пользователям. Каждому пользователю соответствует пара «открытый/закрытый ключ». Пользователи взаимодействуют с облачными сервисами, выполняя операции в рамках предоставленных им прав. Пользователи выполняют действия над распределенно расположенными данными и программами, что приводит к созданию записей журнала действий. Записи журнала сохраняются в хранилище (далее – «диск»), управляемом поставщиком облачных сервисов. Информация для проверки целостности данных журналов публикуется в блокчейн и на сетевой ресурс, принадлежащий потребителю и управляемый потребителем (далее – «внутренний сетевой ресурс»). У поставщика облачных сервисов отсутствует возможность удалять или модифицировать существующие данные на внутреннем сетевом ресурсе, но присутствует возможность записывать новые данные. При выполнении проверки целостности данных журналов аудитор получает доступ к внутреннему сетевому ресурсу, блокчейну, списку пользователей и их открытым ключам, но не получает доступ к содержимому записей журналов.

Инициализация нового блока в файле журнала

Процесс инициализации нового блока в файле журнала представлен на рис. 1.

Рис. 1. Процесс инициализации нового блока в файле журнала

Fig. 1. The process of initializing of a new block in the log file

Создание новой записи журнала инициируется некоторым пользователем U_k, выполнившим действие над данными в облачном сервисе. Для начала записи нового блока журнала B_i поставщик облачных сервисов в момент времени t_1 создает служебную запись $e_{i,1} = \{i, 1, u_k, t_1, \text{BEGIN}\}$, не содержащую сведений о выполняемых действиях. Далее поставщик облачных сервисов находит подпись $\lambda\left(\mu_{i,1}\right)$ посредством ключа s_c и отправляет пользователю U_k сообщение M_1:

$$m_1 = \mu_{i,1} \| \lambda\left(\mu_{i,1}\right).$$

После получения сообщения m_1 пользователь проверяет подпись $\lambda\left(\mu_{i,1}\right)$ посредством открытого ключа поставщика облачных сервисов p_c. Если подпись недействительна, пользователь отправляет поставщику облачных сервисов сообщение об ошибке. Иначе пользователь генерирует метку записи:

$$\sigma_{i,1} = H(\mu_{i,1} \| 0)^{p_{u_k}}.$$

Далее метка $\sigma_{i,1}$ записывается в блокчейн и отправляется поставщику облачных сервисов. После получения значения метки служебная запись $e_{i,1}$ записывается на диск и публикуется на внутренний сетевой ресурс. Служебная запись $e_{i,1}$ будет доступна аудитору при выполнении проверки целостности журналов действий пользователей.

Запись полезной нагрузки

Процесс записи полезной нагрузки представлен на рис. 2.

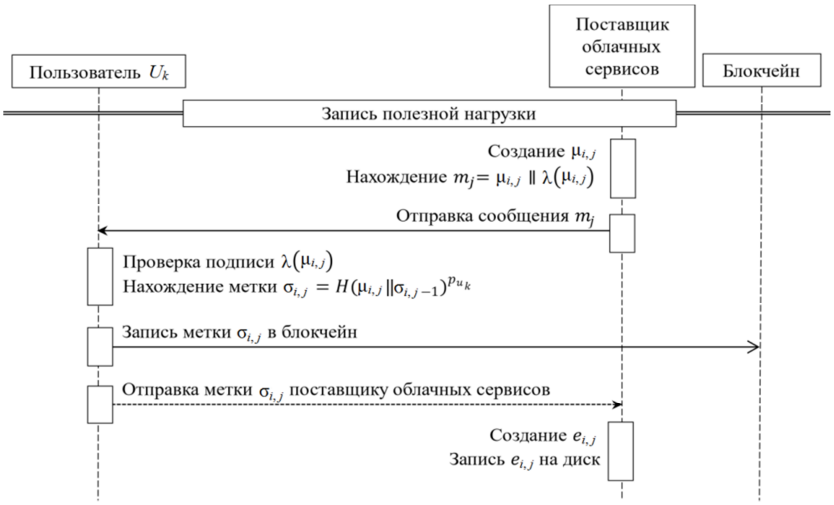

Рис. 2. Процесс записи полезной нагрузки

Fig. 2. The process of payload recording

Выполнение операции пользователем U_k над данными в облачном сервисе приводит к созданию информации о выполненной операции, которая будет записана в файл журнала действий пользователей в качестве полезной нагрузки $p_{i,j}$, где j – порядковый номер записи в текущем блоке. Таким образом, $\mu_{i,j} = (i \| j \| t_j \| u_k \| p_{i,j})$, где t_j – время создания $\mu_{i,j}$.

Поставщик облачных сервисов находит подпись $\lambda\left(\mu_{i,j}\right)$ и отправляет пользователю U_k сообщение m_j:

$$m_j = \mu_{i,j} \| \lambda\left(\mu_{i,j}\right). \quad (1)$$

Пользователь U_k проверяет подпись $\lambda\left(\mu_{i,j}\right)$ и в случае успешной проверки находит значение метки:

$$\sigma_{i,j} = H(\mu_{i,j} \| \sigma_{i,j-1})^{p_{u_k}}. \quad (2)$$

Метка $\sigma_{i,j}$ записывается в блокчейн и отправляется поставщику облачных сервисов. После получения метки поставщик облачных сервисов создает и записывает на диск $e_{i,j}$.

Запись конца блока

Процесс записи конца блока представлен на рис. 3.

При достижении количества записей $e_{\max} - 1$ поставщик облачных сервисов должен записать служебную запись $e_{i,e_{\max}}$ и перейти к созданию следующего блока. Метку служебной записи формирует пользователь, чьи действия привели к созданию записи $e_{i,e_{\max}-1}$. Поставщик облачных сервисов формирует $\mu_{i,e_{\max}} = (i \| e_{\max} \| u_k \| t_1 \| \text{END})$. Аналогично записи $e_{i,1}$ служебная запись $e_{i,e_{\max}}$ не содержит сведений о выполненной операции. Аналогично выражениям (1), (2) выполняется нахождение значений $m_{e_{\max}}$ и $\sigma_{i,e_{\max}}$. Метка $\sigma_{i,e_{\max}}$ записывается в блокчейн. После получения значения метки поставщик облачных сервисов записывает $e_{i,e_{\max}}$ на диск. Служебная запись $e_{i,e_{\max}}$ публикуется на внутренний сетевой ресурс.

Рис. 3. Процесс записи конца блока

Fig. 3. The process of end of the block recording

Проверка целостности журналов

Процесс проверки целостности журналов представлен на рис. 4–6.

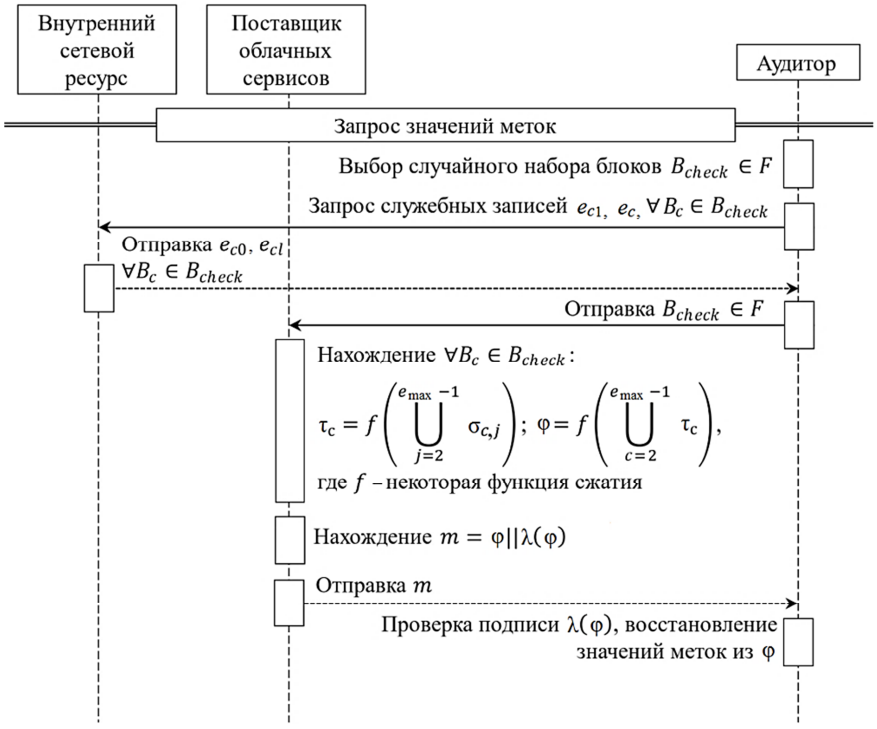

Рис. 4. Процесс проверки целостности журналов – этап запроса значений меток записей

Fig. 4. Logs integrity checking process – the stage of requesting the values of entries tags

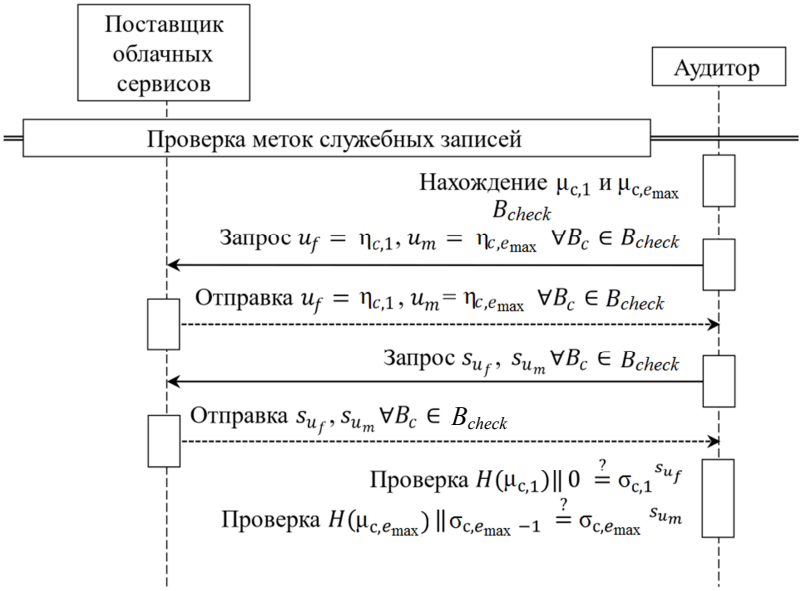

Рис. 5. Процесс проверки целостности журналов – этап проверки меток служебных записей

Fig. 5. Logs integrity checking process – the stage of service entries verification

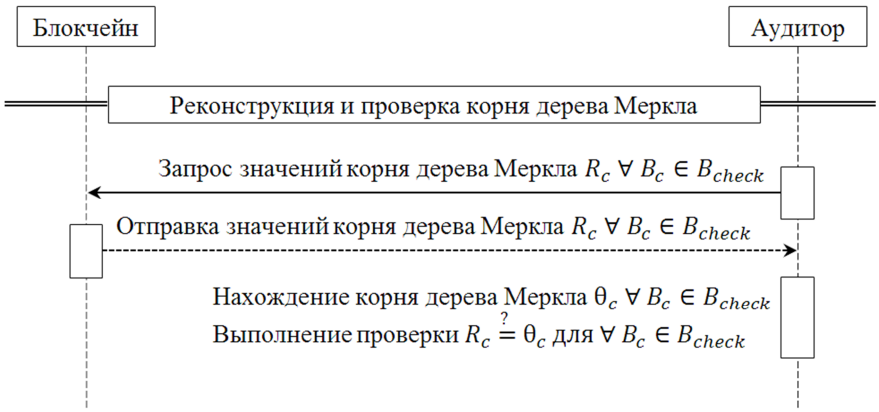

Рис. 6. Процесс проверки целостности журналов – этап реконструкции и проверки корня дерева Меркла

Fig. 6. Logs integrity checking process – the stage of reconstruction and verification of the Merkle tree root

Для выполнения проверки целостности журналов аудитор случайным образом выбирает набор блоков $B_{check} \in F$. Для каждого блока $B_c \in B_{check}$ аудитор запрашивает ранее опубликованные служебные записи $e_{c,1}$ и $e_{c,e_{max}}$ и отправляет поставщику облачных сервисов список блоков B_{check}. Поставщик облачных сервисов находит для каждого блока $B_c \in B_{check}$:

$$\tau_c = f\left(\bigcup_{j=2}^{e_{max}-1} \sigma_{c,j}\right); \quad (3)$$

$$\varphi = f\left(\bigcup_{c=2}^{e_{max}-1} \tau_c\right). \quad (4)$$

В выражениях (3), (4) f – некоторая функция сжатия. Поставщик облачных сервисов находит подпись $\lambda(J)$ для значения φ и отправляет аудитору сообщение

$$m = \varphi \| \lambda(\varphi).$$

Получив сообщение m, аудитор проверяет подпись и в случае успешной проверки восстанавливает из φ значения меток записей. Для каждого блока $B_c \in B_{check}$ аудитор выполняет следующие действия:

1. Аудитор получает служебные записи $e_{c,1}$ и $e_{c,e_{max}}$. Аудитор находит значения $\mu_{c,1}$ и $\mu_{c,e_{max}}$

посредством конкатенации полей $e_{c,1}$ и $e_{c,e_{\max}}$ соответственно.

2. Для записи $e_{c,1}$ аудитор получает идентификатор пользователя $u_f = \eta_{c,1}$ и запрашивает значение закрытого ключа s_{u_f} пользователя u_f.

3. Аудитор проверяет выполнение условия

$$H(\mu_{c,1}) \| 0 \stackrel{?}{=} \sigma_{c,1}{}^{s_{u_f}}. \quad (5)$$

4. Для записи $e_{c,e_{\max}}$ аудитор получает идентификатор пользователя $u_m = \eta_{c,e_{\max}}$ и запрашивает значение закрытого ключа s_{u_m} пользователя u_m.

5. Аудитор проверяет выполнение условия

$$H(\mu_{c,e_{\max}}) \| \sigma_{c,e_{\max}-1} \stackrel{?}{=} \sigma_{c,e_{\max}}{}^{s_{u_m}}. \quad (6)$$

Если существует $B_c \in B_{check}$ такой, что не выполняется хотя бы одно из условий (5), (6), проверка целостности завершается с ошибкой.

В случае успешной проверки согласно условиям (5), (6) для каждого $B_c \in B_{check}$ аудитор запрашивает из блокчейна значение корня дерева Меркла R_c [4–6]. Далее для каждого B_c аудитор выполняет следующие действия:

1. Используя значения $\sigma_{c,1}, \ldots, \sigma_{c,e_{\max}-1}$, аудитор находит значение корня дерева Меркла:

$$\theta_c = H\Big(H\Big(\ldots H\Big(H\big(\sigma_{c,1}\big) | H\big(\sigma_{c,2}\big)\Big) H\big(\sigma_{c,3}\big)\ H\big(\sigma_{c,4}\big)\Big)\ldots\Big).$$

2. Аудитор проверяет выполнение условия для каждого $B_c \in B_{check}$:

$$R_c \stackrel{?}{=} \theta_c. \quad (7)$$

Если существует $B_c \in B_{check}$ такой, что не выполняется условие (7), проверка завершается с ошибкой.

Заключение

Предложен метод проверки целостности журналов действий пользователей облачных сервисов на основе технологии блокчейн, позволяющий выполнять проверку целостности без раскрытия данных журналов. Описан способ организации хранения файлов журналов в облачных сервисах для поддержки возможности интеграции предложенного метода. Предложенный метод позволяет обеспечить гарантию достоверности результатов, получаемых при анализе журналов действий пользователей, и, таким образом, востребован в области расследования инцидентов информационной безопасности.

Список источников

1. 2024 Threat Detection Report // Red Canary. URL: https://resource.redcanary.com/rs/003-YRU-314/images/2024 ThreatDetectionReport_RedCanary.pdf (дата обращения: 06.02.2025).

2. Российский рынок облачных инфраструктурных сервисов 2024 // АО «ИКС-ХОЛДИНГ». URL: https:// survey.iksconsulting.ru/page59801703.html (дата обращения: 06.02.2025).

3. ГОСТ Р ИСО 9127-94. Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов. М.: Госстандарт России, 1995. 8 с.

4. Szydlo M. Merkle Tree Traversal in Log Space and Time // International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. P. 541–554. DOI: 10.1007/978-3-540-24676-3_32.

5. Becker G. Merkle signature schemes, merkle trees and their cryptanalysis // Ruhr-University Bochum, Tech. Rep. 2008. V. 12. P. 19.

6. Li H., Lu R., Zhou L., Yang B., Shen X. An efficient merkle-tree-based authentication scheme for smart grid // IEEE Systems Journal. 2013. V. 8. N. 2. P. 655–663. DOI: 10.1109/JSYST.2013.2271537.

References

1. 2024 Threat Detection Report. *Red Canary.* Available at: https://resource.redcanary.com/rs/003-YRU-314/images/2024 ThreatDetectionReport_RedCanary.pdf (accessed: 06.02.2025).

2. Rossiiskii rynok oblachnykh infrastrukturnykh servisov 2024 [The Russian market of cloud infrastructure services 2024]. *AO «IKS-KhOLDING».* Available at: https://survey. iksconsulting.ru/page59801703.html (accessed: 06.02.2025).

3. *GOST R ISO 9127-94. Sistemy obrabotki informatsii. Dokumentatsiia pol'zovatelia i informatsiia na upakovke dlia potrebitel'skikh programmnykh paketov* [ISS R ISO 9127-94. Information processing systems. User documentation and packaging information for consumer software packages]. Moscow, Gosstandart Rossii Publ., 1995. 8 p.

4. Szydlo M. Merkle Tree Traversal in Log Space and Time. *International Conference on the Theory and Applications of Cryptographic Techniques*. Berlin, Heidelberg, Springer Berlin Heidelberg, 2004. Pp. 541-554. DOI: 10.1007/978-3-540-24676-3_32.

5. Becker G. Merkle signature schemes, merkle trees and their cryptanalysis. *Ruhr-University Bochum, Tech. Rep*., 2008, vol. 12, p. 19.

6. Li H., Lu R., Zhou L., Yang B., Shen X. An efficient merkle-tree-based authentication scheme for smart grid. *IEEE Systems Journal*, 2013, vol. 8, no. 2, pp. 655-663. DOI: 10.1109/JSYST.2013.2271537.

Статья поступила в редакцию 22.04.2025; одобрена после рецензирования 16.06.2025; принята к публикации 24.07.2025
The article was submitted 22.04.2025; approved after reviewing 16.06.2025; accepted for publication 24.07.2025

Информация об авторах / Information about the authors

Владимир Вадимович Еремук – аспирант факультета безопасности информационных технологий; Национальный исследовательский университет ИТМО; polar.vl@yandex.ru

Вячеслав Александрович Горошков – заведующий лабораторией факультета безопасности информационных технологий; Национальный исследовательский университет ИТМО; gorosvia@ya.ru

Александр Владимирович Касьянов – аспирант кафедры информационной безопасности; Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина); kasjanov@inbox.ru

Виктор Андреевич Ромашов – аспирант факультета безопасности информационных технологий; Национальный исследовательский университет ИТМО; whiviktor@gmail.com

Данил Павлович Островский – аспирант факультета безопасности информационных технологий; Национальный исследовательский университет ИТМО; dan97_@mail.ru

Vladimir V. Eremuk – Postgraduate Student of the Faculty of Secure Information Technologies; ITMO University; polar.vl@yandex.ru

Vyacheslav A. Goroshkov – Head of the Laboratory of the Faculty of Secure Information Technologies; ITMO University; gorosvia@ya.ru

Aleksandr V. Kasyanov – Postgraduate Student of the Department of Information Security; Saint Petersburg Electrotechnical University; kasjanov@inbox.ru

Viktor A. Romashov – Postgraduate Student of the Faculty of Secure Information Technologies; ITMO University; whiviktor@gmail.com

Danil P. Ostrovsky – Postgraduate Student of the Faculty of Secure Information Technologies; ITMO University; dan97_@mail.ru