Method for Adaptive Allocation of Cryptographic Resources in Distributed Databases*

Serhii Zhebka[1,†], Viktoriia Zhebka[1,†], Hennadii Hulak[2,†], Roman Kyrychok[2,*,†] and Artem Platonenko[2,†]

[1] State University of Information and Communication Technologies, 7 Solomenskaya str., Kyiv, 03110, Ukraine
[2] Borys Grinchenko Kyiv Metropolitan University, 18/2 Bulvarno-Kudriavska str., Kyiv, 04053, Ukraine

Abstract

This paper presents the development of an adaptive method for optimizing the allocation of cryptographic resources in distributed databases. The method is based on dynamic system load analysis and adaptive redistribution of encryption keys depending on the threat level and current system state. The proposed approach utilizes clustering analysis and blockchain technologies to ensure a high level of security and efficient cryptographic resource management. Implementing this method minimizes data access time, enhances system resilience against man-in-the-middle (MITM) attacks, and optimizes node load distribution. Simulation results indicate a 15% reduction in access time and an improvement in system security.

Keywords

distributed databases, cryptographic keys, adaptive allocation, optimization method, information security, blockchain, MITM attack, clustering analysis

1. Introduction

In modern distributed databases, cryptographic methods are a key factor in ensuring information security. As data volumes grow and distributed systems expand, the challenge of optimal cryptographic resource management, particularly encryption key distribution, becomes increasingly important. Traditional key distribution methods often fail to account for dynamic system load changes, leading to excessive computational overhead or reduced security levels.

One of the critical threats to distributed databases is MITM attacks, which can compromise transmitted data if the system lacks adaptive cryptographic resource distribution mechanisms. Therefore, it is essential to develop methods that enable dynamic key management based on changing load conditions and security levels.

The problem of optimizing cryptographic resource allocation in distributed databases has gained particular relevance due to the increasing volume of information and the need for enhanced security. A major issue is the uneven distribution of load among system nodes, which can cause data access delays and increase the risk of MITM attacks. Moreover, existing methods often rely on static key distribution, making it difficult to respond effectively to dynamic system changes.

[1] examines cryptographic methods in distributed systems using identity-based encryption (IBE) algorithms. While effective for confidentiality, it lacks adaptability to varying threat levels.

[2] presents symmetric encryption with adaptive key management, focusing on automatic key updates when a threat threshold is exceeded. However, it does not address load balancing between nodes.

*CPITS 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2025, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
szhebka@hotmail.com (S. Zhebka); viktoria_zhebka@ukr.net (V. Zhebka); h.hulak@kubg.edu.ua (H. Hulak); r.kyrychok@kubg.edu.ua (R. Kyrychok); a.platonenko@kubg.edu.ua (A. Platonenko)
0009-0007-4620-9888 (S. Zhebka); 0000-0003-4051-1190 (V. Zhebka); 0000-0001-9131-9233 (H. Hulak); 0000-0002-9919-9691 (R. Kyrychok); 0000-0002-2962-5667 (A. Platonenko)

[3] proposes adaptive cryptographic protocols for key redistribution in distributed systems. Although resistant to MITM attacks, it does not incorporate dynamic load analysis for optimization.

[4] suggests optimizing key distribution in blockchain-based systems by analyzing node load, reducing access delays but not addressing adaptive redistribution under heightened threats.

[5] discusses cryptographic key management in cloud platforms, proposing adaptive redistribution approaches but without sufficient exploration of decentralized management and functional resilience.

The literature review highlights that most existing approaches focus on centralized systems or static key distribution. Adaptive cryptographic resource management methods, considering both threat levels and dynamic load changes, remain underexplored. This justifies the need for a method that combines adaptive key distribution with an optimization-based load-balancing approach.

Existing cryptographic resource management methods in distributed databases have several limitations, primarily due to their static approach to key distribution. These methods do not account for variable system load parameters and data access patterns, leading to increased encrypted data access times and security vulnerabilities against MITM attacks. The lack of adaptive key redistribution mechanisms can also cause excessive load on specific system nodes, reducing overall efficiency [6].

Therefore, it is necessary to develop a method that ensures dynamic and optimized encryption key allocation according to the current system load, minimizes access delays, and enhances security without significantly increasing computational complexity.

The objective of this paper is to develop and justify an adaptive method for optimizing the allocation of cryptographic resources in distributed databases [7]. The proposed approach is based on dynamic system load analysis and seeks to balance security and performance. The main focus is on reducing data access times and improving system resilience to MITM attacks.

2. Key findings

The developed method for optimizing cryptographic resource allocation in distributed databases is based on adaptive encryption key management, considering system load variations and threat levels. The core idea is to use real-time dynamic request analysis and risk modeling to modify key distribution schemes accordingly. The approach accounts for user activity, data processing volume, and potential system vulnerabilities, allowing the selection of optimal encryption algorithms and key storage methods to balance security and performance [8].

The scientific novelty of the method lies in its adaptive key distribution approach, which leverages machine learning techniques and heuristic optimization algorithms. Specifically, neural network models are employed to predict system load changes and select cryptographic mechanisms accordingly [9]. This approach enables automatic adjustments in key length, encryption algorithms, and distribution methods, minimizing data compromise risks.

Key elements of the method include:

- Clustering analysis to identify user groups with similar request patterns.
- Distributed key management using blockchain mechanisms.

The implementation of the method not only enhances security without significantly increasing computational complexity but also reduces access time to encrypted data. This is especially important for large distributed systems, where high performance is critical. The optimization mechanism helps avoid overloading cryptographic processing servers by distributing the load according to the predicted user activity [10–14].

It is also worth noting the increased resilience of the system to man-in-the-middle attacks. The dynamic key distribution and their updating at random intervals make such attacks significantly more difficult to execute. Moreover, by integrating multi-factor authentication and biometric

access verification mechanisms, the risks of unauthorized key acquisition are greatly reduced [15–18]. As a result of implementing the proposed method, the overall security of distributed databases is enhanced, the risks of information compromise are minimized, and the use of cryptographic resources is optimized, ensuring stable system operation even during large-scale cyberattacks [19, 20].

The main scientific novelty of this method lies in the dynamic optimization of cryptographic resource distribution, based on adaptive real-time system load analysis. Traditional approaches either distribute keys statically or change them according to a predefined schedule, making them less flexible and potentially vulnerable to attacks [21, 22].

The proposed method uses a combination of machine learning and blockchain technologies for distributing and updating cryptographic keys, allowing for:

1. Predicting load changes in the distributed database and adapting the encryption strategy. The use of cluster analysis helps identify typical user request patterns and respond to anomalies.
2. Automatically changing keys according to risk levels. If the system detects suspicious activity, it can dynamically increase the frequency of key updates or switch to more secure encryption algorithms.
3. Reducing system load through a smart balance between cryptographic protection and query processing speed. The adaptive approach ensures resources are not overloaded during peak loads.
4. Distributing cryptographic keys via decentralized trust servers instead of a centralized key store significantly complicates their compromise and reduces the likelihood of man-in-the-middle attacks.
5. Recording key changes in the blockchain, ensuring their transparency, and authenticity, and enabling retrospective analysis for suspicious actions.

Thus, the main uniqueness of the method lies in the combination of adaptive load analysis, decentralized key management, and machine learning to create an optimal, attack-resistant, and high-performance cryptographic system for distributed databases.

Method for Adaptive Distribution of Cryptographic Resources in Distributed Databases

The input data for the proposed method of optimizing the distribution of cryptographic resources in distributed databases includes the number of nodes in the system N, the load level on each node L_i (ranging from 0 to 1, where 1 represents maximum load), the threat level for each node S_i (ranging from 0 to 1, where 1 indicates a critical threat), the total number of available cryptographic keys Kmax, and a threshold threat value θ, exceeding which triggers an emergency key update for the node to enhance the system's information security.

The algorithm consists of the following steps:

1. Normalization of Input Data

The values of load L_i and threat level S_i are transformed into the range [0,1] to ensure the correct operation of the algorithm. If the system reads absolute values, normalization is applied:

$$L_i^{norm} = \frac{L_i}{\max(L)} \tag{1}$$

$$S_i^{norm} = \frac{S_i}{\max(S)} \tag{2}$$

Normalization allows for the standardization of input data for further processing and ensures their comparability when calculating weight coefficients.

2. Calculation of the Weight Coefficient for Each Node

The importance of each node for key distribution is determined, taking into account the load and threat level. The weight coefficient is defined as:

$$W_i = \alpha L_i^{norm} + \beta S_i^{norm} \tag{3}$$

where $\alpha = 0.7$, $\beta = 0.3$ are weight coefficients that determine the influence of load and threat.

This approach allows for consideration of both the current load on the nodes and potential risks arising from a high threat level.

3. Threat Level Check for Each Node

For each node, we check whether the threat level Si exceeds the threshold value θ. If $S_i > \theta$ (critical threat), an emergency key update is performed. Keys are distributed evenly among all nodes to avoid risk concentration:

$$K_i = K_{total} / N, \tag{4}$$

where K_{total} is the total number of cryptographic keys that need to be distributed among the nodes of the distributed database. This value is fixed and determined based on the system's overall data protection requirements and the number of active nodes in the system.

At the same time, a notification is sent to the administrator about the increased threat level.

If $S_i \leq \theta$ (normal operating mode), the keys are distributed proportionally to the weight coefficients W_i. Each node receives keys according to the formula:

$$K_i = \frac{K_{\max} \cdot W_i}{\sum_{j=1}^{N} W_j} \tag{5}$$

This approach minimizes risks in critical threat situations and ensures an optimal distribution of keys under normal operating conditions.

4. Determining the Final Key Distribution

The values of K_i are rounded to the nearest whole number to obtain the number of keys for each node. Additionally, a check is performed to ensure that the total number of distributed keys equals K_{total}.

5. Recording Updated Keys in the System

At this stage, the cryptographic key table in the distributed database is updated, after which the system sends the updated keys to the corresponding nodes. If some nodes still face a high threat level, a re-evaluation is triggered to provide an additional layer of security.

Since an uneven distribution of cryptographic resources may lead to server overload, it is proposed to minimize the variation in load across the nodes. The optimization model for minimizing load variation is defined as:

$$V(t) = \sum_{i=1}^{N} (L_i(t) - L_{avg}(t))^2, \quad L_{avg}(t) = \frac{1}{N} \sum_{i=1}^{N} (L_i(t) \tag{6}$$

where $L_{avg}(t)$ is the average system load, and N is the number of servers in the system.

To balance the load, an adaptive redistribution of cryptographic keys is performed. Servers with a load coefficient exceeding the permissible value $Ki > \alpha$ are identified as nodes that require a reduction in cryptographic load, while nodes with Kj can take on additional computations.

The process of key redistribution is described by the key migration equation between nodes:

$$M_{ij} = \gamma(L_i - L_j), \quad \forall i, j \in N, \quad L_i > L_j \tag{7}$$

where γ is the adaptation coefficient, which defines the speed of key redistribution.

The key redistribution between nodes is determined by the migration equation:

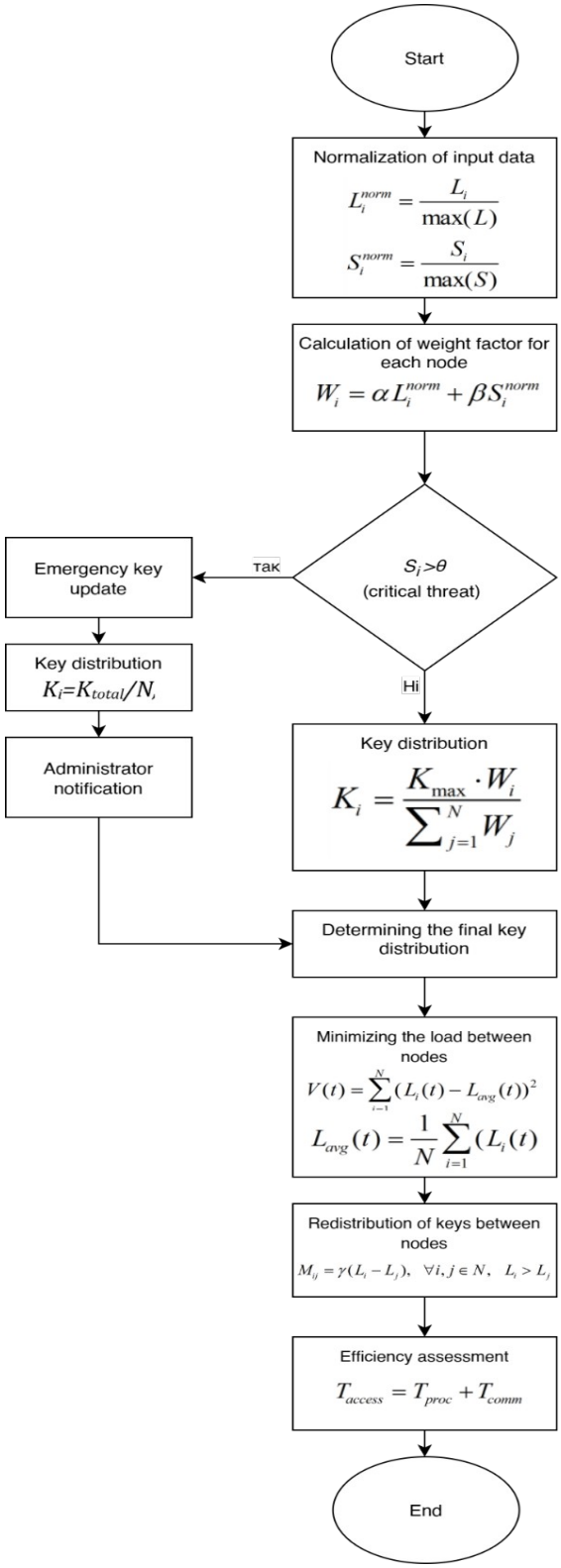

Figure 1: Flowchart of the Adaptive Cryptographic Resource Allocation Method in Distributed Databases

Efficiency Evaluation

The efficiency of the proposed method is assessed based on the average access time to encrypted data.

$$T_{access} = T_{proc} + T_{comm} \tag{8}$$

where T_{proc} is the processing time of the request on the server, and T_{comm} is the delay associated with the transmission of cryptographic keys. Minimizing T_{comm} through adaptive key distribution helps improve the overall system performance. The method involves dynamic updating of keys K_i depending on the load parameters. If:

$$L_i > \theta \cdot \frac{\sum_{j=1}^{N} L_j}{N} \tag{9}$$

where θ is the coefficient of load imbalance, then a portion of the decryption requests is transferred to other nodes with lower load:

$$L_j = L_j + \alpha \cdot (L_i - L_{avg}), \quad L_i = L_i - \alpha \cdot (L_i - L_{avg}) \tag{10}$$

where $L_{avg} = \frac{\sum_{j=1}^{N} L_i}{N}$, α—coefficient of distribution.

Encryption keys are updated if the node is overloaded or its performance is significantly lower than the average:

$$K_i \to K_j, \quad if \ C_i < \beta \cdot C_{avg} \tag{11}$$

where C_{avg} is the average node performance, β is the threshold for key update.

Performance Evaluation

Unlike traditional methods, the proposed approach involves the dynamic distribution of cryptographic resources in distributed databases using an adaptive key distribution method based on the current system load. The methods selected for comparison (AES-GCM, RSA-2048, ECDSA) are the most commonly used in cryptographic data protection and employ different encryption approaches:

- AES-GCM—a representative of symmetric encryption that ensures fast data processing.
- RSA-2048—one of the classical asymmetric algorithms used for key exchange and digital signatures.
- ECDSA—a digital signature algorithm based on elliptic curve cryptography that provides high security with a shorter key length.
- The proposed method uses an adaptive approach to key distribution, which reduces latency and increases security without placing unnecessary load on the system.

Table 1

Comparison of Cryptographic Methods

Parameter	AES-GCM	RSA-2048	ECDSA	The proposed method
Encryption type	Symmetric	Asymmetric	Asymmetric (signature)	Hybrid (adaptive)
Productivity	High	Low	High	High

Key length	128/192/256 bit	2048 bit	256 bit	Dynamic (adaptive)
Computational costs	Low	High	Medium	Optimized
Security	High (with correct nonce)	High with large keys	High	High (adaptive key generation)
Usage	Data encryption	Key transmission protection	Digital signature	Dynamic key distribution in a distributed database

Table 2
Comparison of data access latency (in milliseconds, approximately)

System load	AES-GCM	RSA-2048	ECDSA	The proposed method
Low	1-2 ms	10-20 ms	5-8 ms	2-3 ms
Medium	3-5 ms	30-50 ms	8-12 ms	3-5 ms
High	7-10 ms	80-150 ms	15-25 ms	5-7 ms

The proposed method provides **lower latency for accessing encrypted data** compared to asymmetric algorithms (RSA, ECDSA) and slightly exceeds AES-GCM in speed, compensating for this with **adaptive security**.

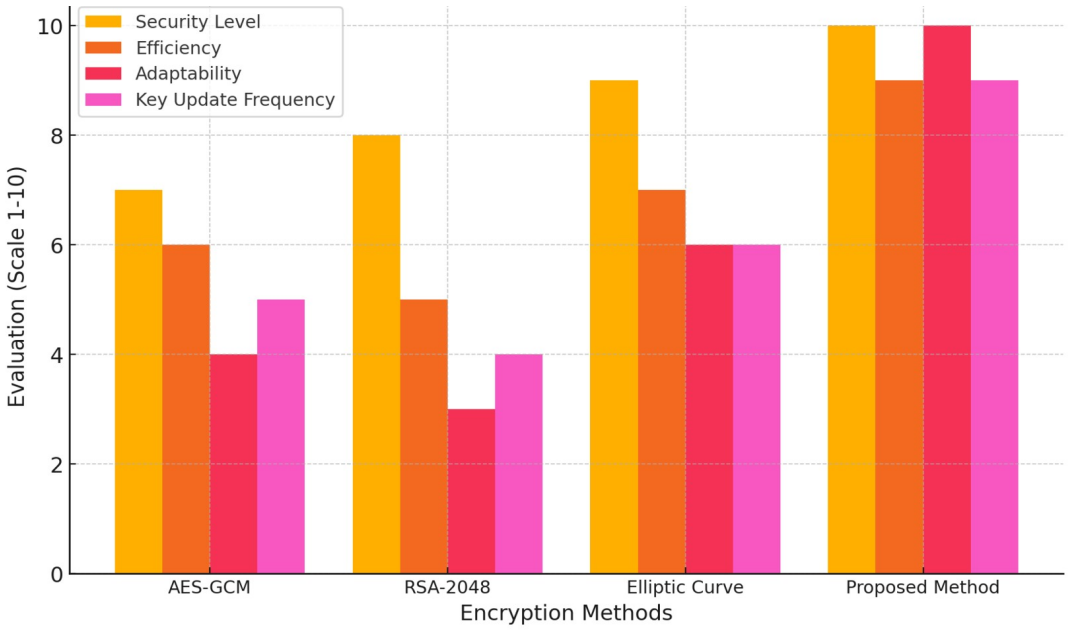

Figure 2: Comparison of encryption methods

The graph demonstrates that the proposed method offers a higher level of security, efficiency, adaptability, and key update frequency.

Conclusions

The proposed method for optimizing the distribution of cryptographic resources in distributed databases improves the efficiency of encryption key management by reducing the load on nodes and ensuring an even distribution of data processing requests. The use of an adaptive approach, based on dynamic system load analysis, helps avoid bottlenecks in encryption and decryption processes, which is crucial for scalable decentralized platforms such as blockchain networks and distributed data storage systems. The proposed solution strikes a balance between security and data access speed, significantly reducing the risks of man-in-the-middle attacks and enhancing the system's resilience to potential threats.

The results show that the use of an adaptive key redistribution mechanism can significantly reduce cryptographic operation latency while lowering computational costs without compromising security. The practical application of the method is possible in blockchain systems, distributed databases, secure cloud storage, and financial and corporate systems, where data processing speed and protection against unauthorized access are critical.

Future research prospects are related to improving dynamic system state monitoring mechanisms, including the development of machine learning algorithms to predict node load and automatically adjust the key distribution strategy. Additionally, further experimental studies are necessary to determine the optimal parameters for adaptive redistribution based on the architecture of the distributed system, particularly analyzing the impact of various consensus models on the performance of cryptographic operations. An important direction is also the study of integrating the proposed method with modern cryptographic algorithms, such as post-quantum cryptography, which will enhance resilience to potential future threats.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, SIAM J. Comput. 32(3) (2003). doi:10.1137/s0097539701398521

[2] K. Paterson, G. Price, A comparison of adaptive encryption techniques in distributed systems, IEEE Transactions on Information Forensics and Security, 13(2) (2018).

[3] M. Bellare, P. Rogaway, Adaptive protocols for public-key encryption, in: Advances in Cryptology, EUROCRYPT, 1998.

[4] S. Kumar, T. Utsab, Adaptive cryptographic access control for blockchain systems, Int. J. Adv. Comput. Sci. Appl. (2021).

[5] P. Venkatram, A new lossless data compression algorithm exploiting positional redundancy, arXiv, 2021. doi:10.48550/arXiv.2107.13801

[6] V. Astapenya, et al., Analysis of ways and methods of increasing the availability of information in distributed information systems, in: 2021 IEEE 8th Int. Conf. on Problems of Infocommunications, Science and Technology (2021). doi:10.1109/picst54195.2021.9772161

[7] V. Grechaninov, et al., Models and methods for determining application performance estimates in distributed structures, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288, 2022, 134–141.

[8] A. Bessalov, V. Sokolov, S. Abramov, Efficient commutative PQC algorithms on isogenies of Edwards curves, Cryptography 8(3), iss. 38 (2024) 1–17. doi:10.3390/cryptography8030038

[9] Y. Kostiuk, et al., Integrated protection strategies and adaptive resource distribution for secure video streaming over a Bluetooth network, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3826 (2024) 129–138.
[10] A. V. Halchenko, Instrumental means of cryptographic systems based on deniable encryption, Ph.D. Dissertation, Zaporizhzhia National University, 2021.
[11] A. Gharout, M. Merabti, D. Llewellyn-Jones, Adaptive group key management protocol for wireless dynamic groups, J. UCS 18(6) (2012) 874–898.
[12] F. Pierazzi, M. Colajanni, Performance and cost evaluation of an adaptive encryption architecture for cloud databases, IEEE Trans. Cloud Comput. 2(2) (2014) 153–166.
[13] NIST, Recommendation for key management: Part 1 – General, NIST special publication 800-57 Part 1 Revision 5, 2020.
[14] S. Kumar, T. Utsab, A. Raychoudhury, Image compression using approximate matching and run length, Int. J. Adv. Comput. Sci. Appl. 2(6) (2011).
[15] S. Romanenko, Cryptanalysis and cryptographic protocols, Uzhhorod National University, 2020.
[16] Y. Galchenko, Information security: Cryptographic methods and protocols, National Aviation University, 2010.
[17] A. Astrakhantsev, Models and methods for improving the security and quality of data transmission in mobile communication systems, Ph.D. Dissertation, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," 2021.
[18] F. Pierazzi, M. Colajanni, Performance and cost evaluation of an adaptive encryption architecture for cloud databases, IEEE Transactions on Cloud Computing, 2(2) (2014) 153–166.
[19] S. Belattaf, M. Merabti, D. Llewellyn-Jones, Reliable and adaptive distributed public-key management infrastructure for the Internet of Things, J. Netw. Comput. Appl., 123 (2018) 94–108.
[20] V. Malinov, et al., Cryptocurrency as a tool for attracting investment and ensuring the strategic development of the bioenergy potential of processing enterprises in Ukraine, in: Lecture Notes on Data Engineering and Communications Technologies, vol. 195, 2024, 387–405. doi:10.1007/978-3-031-54012-7_17
[21] V. Zhebka, et al., Methodology for choosing a consensus algorithm for blockchain technology, in: Digital Economy Concepts and Technologies, vol. 3665, 2024, 106–113
[22] V. Malinov, et al., Biomining as an effective mechanism for utilizing the bioenergy potential of processing enterprises in the agricultural sector, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3421, 2023, 223–230.