Merkle Tree: A Fundamental Component of Big Data

A.Kubigenova, Al.Aktayeva, G. Yesmagambetova, V. Sukhomlin, A. Umbetov

Abstract **- Big data has unlocked the way for significant advances in various scientific fields. As a result, it has become a desirable topic in a both the academic world and IT businesses. ICT has contributed significantly to innovation, productivity, and competitiveness in big data management. However, many technological challenges related to data collection, storage, use, analysis, privacy, and trust need to be addressed. Big data poses challenges and approaches to overcoming these obstacles vary; data, processes, privacy, and governance are the most frequently mentioned challenges. In addition, inaccurate or misleading big data can lead to misinterpretation of results, negatively affecting user experience. In addition, inaccurate or misleading big data can also cause results to be misinterpreted, negatively impacting data security. This article discusses the challenges associated with significant data security and proposes some critical solutions to mitigate these challenges. A total of 45 articles were collected and examined to compile the article corpus, focusing on several essential issues of the big data security analysis field and their implications for multiple industries.**

Keywords **– big data, data management, internet of things, big data analytics, secure data processing, Merkle Trees, blockchain.**

I.INTRODUCTION

The term "big data" refers to the significant volume, velocity, and variety of information resources that require cost-efficient and innovative data processing methods rather than traditional approaches to improve understanding and decision-making. Emerging technologies and evolving methodologies are influencing future trends in big data analytics. Big data is a crucial aspect of data science. It explores various tools, procedures, and strategies for analysing vast and complex data sets, breaking them down, and systematically deriving insights and information. Understanding big data is a complex concept that requires thorough understanding by stakeholders, each with their preferences for architecture, suppliers, and technologies. Also, big data is a progressive concept that will continue to challenge human ability to manage vast amounts of constantly generated data that grows exponentially over time, with significant sources being modern data from digital communications and the Internet of Things (IoT) sector. Big data processing technologies are software tools designed primarily to analyse, process, and extract information from extensive data sets with highly intricate structures that traditional data processing technologies are unable to handle [1].

The current era of big data is characterised by the following key characteristics: volume, velocity, variety, veracity, variability, visualisation, and value. These characteristics provide a framework for understanding the unique challenges and opportunities that big data presents [1].

Understanding these characteristics of big data is key to using it correctly and is fundamental to those seeking to harness its potential. It provides a basis for selecting the right technologies, implementing effective data management strategies, and ultimately extracting meaningful insights from the vast and dynamic world of big data. However, these key characteristics are not enough to understand Big Data; it is about applying all of these characteristics to a complex problem, typically with multiple fuzzy logic variables. This underscores the need for proactive and effective data management strategies.

The use of big data extends the capabilities of classic statistically based analytical approaches by incorporating modern methods that leverage computer resources and methodologies to run analytical algorithms. This transition is critical as databases grow in size, variety, and complexity and become increasingly streaming-orientated.

Researchers have studied the background and concepts of big data, big data analytics, and cloud computing, from installation to implementation of real-world data processing applications (e.g., the process of setting up and familiarising with the working environments of big data analytics and big data processing systems); also the necessary coding technologies; and knowledge of the details of big data storage technologies, including their types, importance, durability, and availability, which reveal the differences in their properties [2].

Big data mainly includes big data sets in the form of structured, unstructured, and semi-structured data generated from, for example, healthcare, astronomy, social media, and earth sciences [3].

Today's interconnected digital world is experiencing a data explosion thanks to data logging, email, social media, and search engines. The advent of web technology has revolutionized user-generated content, leading to a significant increase in data generation. The recent surge in social media usage has further fuelled user-generated content, and now, real-time content generation has transformed this field into big data, making us more aware of the rapid changes in data generation.

With over a billion people actively using social media, the rapid generation of unstructured data presents a significant challenge. The sheer volume of this data necessitates a robust analysis and processing framework in the realm of big data [4].

Another primary source of big data is the recently emerging field of IoT, where machines generate data from various sensors, such as medical devices, temperature sensors, and myriad additional software modules and digital devices [5].

The primary sources of big data constantly generate various data (structured, unstructured, and semi-structured) beyond the processing capabilities of modern database systems [6].

Managing and analysing large volumes of data and devices will be critical as look to the future. It's not just about databases; we also need to optimize our processing models to handle the typically low-value density of interest data [7].

II. Materials and methods

Merkle Trees play a crucial role in various applications within the realm of big data and secure data processing. Merkle trees are a structure that efficiently and securely verifies the contents of a large array and maintains data integrity. This structure helps verify the consistency and content of the data. Moreover, the model Merkle has been introduced as an innovative approach for data processing, combining data integrity verification mechanisms with deep learning models to achieve superior performance in tasks like financial behaviour detection and stock price prediction. This highlights the importance of data security in data processing [8].

Moreover, the concept of asynchronous computation using Merkle Trees has been suggested to reduce computational costs and enable parallel data processing in environments where low compute time is crucial [9]. However, challenges emerge with the performance of Merkle Trees in high-performance data systems due to expensive cryptographic operations and lack of parallelizability.

Research in [10] focuses on optimizing Merkle Tree structures to balance computational efficiency and data integrity. Additionally, Merkle Trees have been explored to enhance data integrity in decentralized systems, particularly in blockchain and IoT technologies, by addressing data falsification probabilities and optimizing structures for improved security [10].

Merkle Trees are extensively utilized in blockchain technologies like Bitcoin and Ethereum. They efficiently and securely authenticate vast amounts of data, making them a cornerstone of these systems [11].

For the first stage of the research process, known as the eligibility evaluation, 45 publications have been selected. A thorough analysis of the titles, abstracts, and primary contents of every article was done to ensure that the publications satisfied the inclusion criteria and that the study's goals were reached. Following this, as factually demonstrated, 23 papers were eliminated due to names that needed to fit the articles or incomplete texts that did not support the study's goals in their abstracts. As a result, 22 items are considered worthy of more examination.

Authors [12] proposed a new financial data processing model that provides a secure and efficient solution for financial data analysis by integrating the Merkle tree data integrity verification mechanism with the high performance of the Transformer model. [12] presented the Merkle Trees, which enhance data integrity in Big Data by combining data verification with Transformer models, improving security and performance in financial computations. This paper's main contribution is to propose a new financial data processing model that considers the efficiency of data processing and introduces a financial data security verification mechanism.

The authors in the paper [13] proposed the Merkle Trees enhances data integrity by calculating data falsification probabilities. They aid decentralized systems like blockchain and IoT and ensure secure and efficient data verification in Big Data environments.

Liu Zhenpeng et al. [14] provided the hierarchical Merkle hash tree-based data integrity auditing technique that minimises the authentication tree's scale using local signature aggregation technology and hierarchical principles, preventing improper retrieval. Additionally, this article states that the proposed method, Merkle Trees, enhances data integrity in big data by reducing computational and storage overhead, improving transparency and credibility, and enabling efficient auditing with hierarchical structures and monitoring mechanisms. According to this article, the suggested approach efficiently lowers the cloud service providers' resource consumption; theoretical analysis and comparison have demonstrated the high degrees of efficiency and safety of the agreement's solutions for auditing big data.

In [15] Merkle DAG structures, like M-DAG, utilize Merkle Trees to enhance data integrity in Big Data by enabling efficient multicopy verification through Boneh–Lynn–Shacham's (BSL). BLS signatures in decentralized environments. Based on the data attributes within the extensive data setting, a multiparty and effective audit mechanism for ensuring data integrity is developed in this study, leveraging blockchain technology. This enables efficient auditing of both small unstructured data and large-scale data through multiple copies. The verification of data integrity is carried out using smart contracts.

In this [16] paper Merkle Trees enhances data integrity in Big Data by efficiently mapping and identifying changes in large datasets. They also provide secure cloud storage with reduced encryption and decryption times.

Authors [17] proposed Merkle Trees enhances data integrity in Big Data by providing a decentralized and efficient verification technique, ensuring secure cross-chain data sharing with integrity and credibility. Merkle Trees can improve data integrity in Big Data by providing a safe and efficient way to verify data, ensuring credibility and integrity in cross-chain data-sharing environments.

This paper [18] focuses, which the Merkle Trees enhances data integrity in Big Data by securing large datasets with blockchained structures, ensuring tamper-proof verification through vertical blockchains from root to leaf nodes.

In [19] Merkle Trees improves data integrity in Big Data by providing secure outsourced data systems with high performance through innovative techniques despite initial performance challenges. [20] proposed the Merkle Hash Trees ensures data integrity by verifying the completeness, correctness, and freshness of outsourced data in cloud services, enhancing integrity and security in Big Data environments.

In [21] Merkle Trees provides a succinct way for verifiers to ensure data integrity by receiving short proofs from provers, enhancing security in cryptographic protocols.

Merkle Trees play a crucial role in ensuring data security by providing a way to prove data integrity and validity through efficient computation and storage of hash values [22].

One-time signatures can provide satisfactory security, but their one-time nature can become a severe problem. When using one-time signatures, it will be necessary to transfer the entire volume of information each time, and a new address will be needed for each transaction. A new public key will need to be published with each transaction. In addition, storing new information transactions will gradually require more and more time to find them. To solve the problem, the signature scheme is expanded by conducting multiple signatures based on several critical pairs for each address. Multiple signatures are used based on a binary hash tree, the Merkle tree.

The ingenious design of a Merkle tree efficiently breaks down large amounts of data into smaller, manageable blocks. It then consolidates all the transaction data in a block to create a single digital fingerprint, significantly expediting the verification process.

The construction of a Merkle tree is a fascinating interplay of concatenations and hashings. Different pairs of nodes are concatenated and then hashed, resulting in the Merkle tree's root. The structure of a Merkle tree progresses from the bottom up, from the root to the leaves, showcasing the power of cryptographic hashing.

The construction of a Merkle tree is a systematic and reliable process. Transactions from leaf nodes are paired up to form non-leaf nodes, and this process continues in a predictable manner until we reach the root node (see Fig.1).

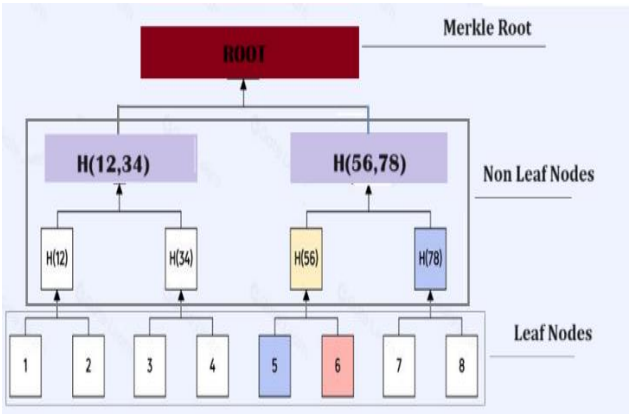

Figure 1. Hash tree - Merkle tree diagram

For example, consider a block with eight different transactions: T1, T2, T3, T4, T5, T6, T7, and T8. Each transaction is hashed to produce H1, H2, H3, H4, H5, H6, H7, and H8. The hashes are then concatenated and hashed again to produce H(12), H(34), H(56), and H(78). The result is concatenated and hashed again to produce H (1234) and H (5678). The next step will produce H (12345678) as the Merkle root. The diagram below shows the Merkle tree constructed from 8 transactions in a block.

Merkle Trees and Merkle Roots in Big Data with Blockchain Technology. Blockchain consists of chains of blocks. One block can contain up to thousands of different transactions. The root hash obtained at the end of the Merkle tree summarises all the transactions contained in that block. This makes the verification process efficient, and any change will be easily detected.

The resulting Merkle root, a key component stored in the block header, is integral to the transmission process. In a cryptocurrency network, for instance, the block header is hashed, not the individual transactions. The presence of the Merkle root in the block header ensures that any change in the original data is immediately detected, making the entire system impervious to tampering.

These tools, the Merkle Trees and Merkle Roots play a crucial role in maintaining the validity of transactions without the need to load the entire network. This reliability instils confidence in the robustness of blockchain technology.

The tree has been calculated from the leaves to the root. Each node leaf is calculated as a hash from the generated public key. The remaining nodes are calculated by obtaining a hash from the concatenation (glueing) of the child nodes. In this way, the entire tree is calculated up to the root.

A unique feature of the Merkle tree is that the existence of any node or leaf can be cryptographically proven by computing the root (See Table1). The message signature is created using the private key from the selected key pair.

Table 1. Advantages of Merkle Tree of Big Data

n/n	Advantages	Note

1.	Efficient data verification process	The Merkle tree provides an efficient means of verifying transactions without consuming much computing power.
2.	Less memory footprint	Verifying transactions using the Merkle tree does not require loading all the data. It requires less space for computation compared to other data structures.
3.	Fast transactions	Since transactions are paired together and a single hash is created, the transmission of information over the network becomes faster. This is one of the main reasons why cryptocurrency transfers are very fast.
4.	Tamper detection	The Merkle tree allows you to detect when a transaction has been tampered with. When a transaction is hashed and stored, a change in the original information also causes a change in the hash. This can be determined by comparing the current hash with the hash stored in the block header.

Signature verification involves computing the root based on the transmitted parameters and comparing it with a reusable public key. These parameters are:

1) Signature.

2) Root.

3) A one-time key, the private part of which was used to sign the message.

4) Hashes from the tree lying on the path from the selected leaf to the root.

One-time Merkle signatures simplify the process by eliminating the need to transmit a separately selected one-time public key. Instead, this key can be obtained from the message signature. All that needs to be transmitted is its number, which reflects its position in the tree. This simplicity enhances the understanding and confidence in the process.

Merkle trees, composed and calculated from public keys, allow the publication of only the root of the tree instead of the whole set of keys. This increases the size of the signature by including part of the tree in the signature, but it also allows the verification of many signatures using only one hash. Thus, with a tree depth of N, it is possible to sign 2^N messages.

Data security management helps to understand the type of data used in the organisation, and it carefully checks and manages the data, detecting errors and eliminating them by implementing an ideal solution, which otherwise leads to the risk of loss of reputation and vulnerability of data. Furthermore, it ensures that the data provided is secure and accurate, reduces the cost of data management, and complies with standards.

III. Result and Discussion

Utilising a Merkle Tree significantly reduces the volume of data that an authorised entity needs to uphold for authentication purposes. This is achieved through the segregation of data verification from the data itself. The Merkle Tree, whether stored locally or within a decentralised system, offers various benefits, including enhanced security:

- The Merkle Tree provides a straightforward method to ensure data integrity and dependability.
- Minimal memory or disk space requisites enable rapid verifications.
- Moreover, the verification and administration process is simplified, as only a limited quantity of data needs to be transmitted across networks, making the system user-friendly and efficient.

The Merkle tree model has expanded the one-time signature (OTS) concept. This approach's fundamental idea is to amalgamate numerous one-time public keys into a framework to collectively derive a value that can symbolise them. This resulting value has been designated as the public key of the Merkle tree. The structure of the hash tree allows each message to be individually signed using a distinct pair of OTS private keys; a maximum of 2^h messages can be signed, with h representing the height of the Merkle tree. The complete generation of the Merkle tree is essential before it can be utilised, with the leaves depicting the one-time signature public keys in a specific sequence.

The tree's bottom node has been the leaf nodes' hash value, for instance $\alpha_{34} = g(pk_5)$. Every internal node has been represented by the hash value of its two concatenated child nodes, which can be stated as,

$$\alpha_{xy} = g\left(\alpha_{(x+1)(2y)} \parallel \alpha_{(x+1+(2y+1)}\right) \qquad (1)$$

where g() - the hash function, ‖ - the connector.

The public key of the Merkle tree, or root node pub, has been at the top and represents a commitment. Identifying the leaf nodes that are a part of this commitment can be possible. The signer first computes the values of the leaf nodes for the digest $d = g(\mathrm{M})$. The signer then chooses an unused "i-th" one-time private signature key sk_i, to generate a signature δ_d for this message d-digest $\iota \in (0, \dots, 2^h - 1)$, where h - the height of the tree, and each pair of OTS private keys can only be used once.

The signature result is $sigh = (i, \delta_d, pk_i, Auth_i)$,

including the corresponding OTS public key pk_i and the authentication path *i.* The authentication path *i* consists of the extra nodes of all nodes on the path from the "i-th" leaf node to the root. For example, $Auth_5 = \{\alpha_{35}, \alpha_{23}, \alpha_{10}\}$. Then, assume that the Merkle tree public key pub has been previously passed to the verifier. When the verifier receives the message M and the signature, it verifies the signature in two steps: First, it uses the OTS public vital pk_i to verify the signature δ_d of the message digest d. If δ_d is a valid signature of d, compute the root value by concatenating $g(pk_i)$и $Auth_i$.

When the resulting root value matches a known public key, the signature token should be accepted. Without knowing the OTS private key sk, an attacker cannot forge a signature δ_d' for a forged message digest d'.

The Merkle tree uses a single public key pub to provide proof of the legitimacy of the 2^h one-time signature public keys pk. pk should be used to verify the correspondence between d and δ_d'. This means that the first step cannot be successfully verified unless the attacker has the correct OTS private key. When an attacker decides to forge a single OTS key pair, since pk' in $sigh'$ - provided by the signer, σ_d' 'will be misled into successful verification. However, since pk' is not involved in the Merkle tree generation, the OTS public key legitimacy verification using pub and $Auth_i$ will fail. This means that the fake OTS key pair will not pass the second authentication step.

Digital signature technology has been prevalent in authentication protocols to demonstrate the sender's possession of a private key without disclosing any details about the private key. If the sender transmits the public key, it then employs the private key to authenticate its identity, thereby verifying its identity, i.e.

$$sign'(Hash(ID||counter), sk), pk \quad (2),$$

where $sign'$ () is an asymmetric signature scheme and Hash () is a secure hash function.

However, the utilisation of this authentication mechanism, which relies on stateless signature technology, is limited to a single use. This limitation arises from the fact that any individual capable of intercepting the communication exchanges can exploit the identical data for authentication purposes. Within a robust authentication framework, the signature system must exhibit resistance against tampering, while the overall protocol architecture should demonstrate resilience against replay assaults and man-in-the-middle intrusions. Furthermore, certain research endeavours have identified a susceptibility associated with the adoption of hash-based signature methodologies for authentication procedures, exemplified by references [24].

Although they use random numbers, tokens, or other mechanisms to improve the protocol's security, no study uses a formal method to analyse their security. The client request authentication message is as follows:

$$sign'(Hash(ID||counter), sk), pk \quad (3),$$

where counter is a variable starting at 0 and will be incremented after each successful authentication.

A Merkle tree and an essential one-time signature (OTS) technique have been used as the signature scheme in the authentication protocol. There is no requirement for an extra counter because the Merkle tree tracks the state. It is referred to as a prototype since the only way to adjust the authentication system's performance is to swap out the sign() function for a version of another hash-based signature scheme. We use a variety of hash-based signature variations along with their time-to-memory properties to authenticate a cluster of Internet of Things devices that can precisely match their processing power and data storage capacity. Extensive data security management incorporates specific privacy safeguards and establishes crucial regulations.

IV. Summary

We have entered a unique era in the development of data analytics, made possible by the emergence of big data, technology, and innovative information management and analytics tools. The combination of these trends means that there are tools to promptly analyse genuinely massive amounts of data. With big data, vast amounts of generated data are managed and analysed, then used for effective modelling.

In cybersecurity, big data analytics has found widespread use in fraud detection. In addition, data encryption using quantum computing is an emerging technology that makes it difficult for hackers to interpret encrypted data. The analysis demonstrates that a model utilizing a Merkle tree accomplishes the anticipated security and efficiency objectives for big data.

Utilising big data, blockchain technology can store various data in a hashed chain format, making it impossible for unauthorised access. As a result, this technology can store sensitive data that should not be changed. It also provides public access to data when needed.

Big data technology and cybersecurity are directly proportional, with the development of technology paving the way for the development of cybersecurity. In future work, there will be a continued investigation into the enhanced integration of large-scale data and integrity validation methodologies, including strategies to mitigate substantial storage costs through the placement of data blocks or labels within the big data.

References

[1]. S. Uthayasankar, K. Muhammad, I. Zahir and W. Vishanth *"Critical analysis of Big Data challenges and analytical methods,"* Journal of Business Research, vol.70, 2016, *https://doi.org/10.1016/j.jbusres.2016.08.001*

[2]. A.A. Kharlamov, M. Pilgun "*Data Analytics for Predicting Situational Developments in Smart Cities: Assessing User Perceptions*," Sensors, vol.24(15):4810, 2024, *https://doi.org/10.3390/s24154810*

[3]. L. Theodorakopoulos, A. Theodoropoulou and C. Halkiopoulos "*Enhancing Decentralized Decision-Making with Big Data and Blockchain Technology: A Comprehensive Review*," Applied Sciences, vol.14(16):7007, 2024, *https://doi.org/10.3390/app14167007*

[4]. S. Azzabi, Z. Alfughi and A. Ouda "*Data Lakes: A Survey of Concepts and Architectures*," Computers, vol.13(7):183, 2024, *https://doi.org/10.3390/computers13070183*

[5]. M. Halaweh, A.E. Massry "*Conceptual Model for Successful Implementation of Big Data in Organizations,"* Journal of International Technology and Information Management, vol. 24(2), pp.21–34, 2015, https://doi.org/10.58729/1941-6679.1039

[6]. Z. Chang, J. Wu, H. Liang, Yong Wang, Y. Wang, X. Xiong *" A Review of Power System False Data Attack Detection Technology Based on Big Data*," Information, vol.15(8):439, 2024, *https://doi.org/10.3390/info15080439*

[7]. H. S. Munawar, F. Ullah, S. Qayyum, D. Shahzad "*Big Data in Construction: Current Applications and Future Opportunities*," Big Data and Cognitive Computing, 2022; vol.6(1):18, 2022 *https://doi.org/10.3390/bdcc6010018*

[8]. U. Sivarajah, S. Kumar, V. Kumar, S. Chatterjee, J. Li, "*A study on big data analytics and innovation: From technological and business cycle perspectives*," Technol. Forecast. Soc. Chang., vol.202(C): 123328, 2024, *https://doi.org/10.1016/ j.techfore.2024.123328*

[9]. A. Kharangate "*Asynchronous Merkle Trees*," arXiv.org,arXiv:2311.17441v1, 2023, *https://doi.org/*10.48550/arxiv.2311.17441

[10]. D. Williams and Emin Gun Sirer, "*Optimal parameter selection for efficient memory integrity verification using Merkle hash trees*," In Proc. *3rd IEEE International Symposium on Network Computing and Applications (NCA 2004),* pp. 383-388, 2004, *https://doi.org/10.1109/NCA.2004.1347805*

[11]. J. A. Alzubi "*Blockchain-based Lamport Merkle digital signature: Authentication tool in IoT healthcare*," Journal Computer Communications, vol.170, pp. 200-208, 2021, *https://doi.org/*10.1016/j.comcom.2021.02.002

[12]. X. Wang, W. Lin, W. Zhang, Y. Huang, Z. Li, Q. Liu, X. Yang, Y.Yao, & C. Lv, C. "*Integrating Merkle Trees with Transformer Networks for Secure Financial Computation",* Applied Sciences, vol.14(4):1386, 2024 *https://doi.org/10.3390/app14041386*

[13]. O. Kuznetsov, A. Rusnak, A. Yezhov, K. Kuznetsova, D. Kanonik & O. Domin, O. "*Evaluating the Security of Merkle Trees in the Internet of Things: An Analysis of Data Falsification Probabilities,"abs/2404.12093*, 2024, *https://doi.org/10.48550/ arxiv.2404.12093*

[14]. Liu Zhenpeng, Wang Shuo, Duan Sichen, Ren Lele & Wei Jianhang "*Dynamic Data Integrity Auditing Based on Hierarchical Merkle Hash Tree in Cloud Storage*," Electronics, vol.12:717, 2023, *https://doi.org 10.3390/electronics12030717*

[15]. J. Wu, S. A. Haider, M. Bhardwaj, A. Sharma & P. Singhal "*Blockchain-Based Data Audit Mechanism for Integrity over Big Data Environments,"* Security and Communication Networks*,* pp.1-9, 2022, *https://doi.org/10.1155/2022/8165653*

[16]. J. S. Jayaprakash, K. Balasubramanian, R. Sulaiman, M. K. Hasan, B. D. Parameshachari & C. Iwendi "*Cloud Data Encryption and Authentication Based on Enhanced Merkle Hash Tree Method,"* Computers, Materials & Continua, vol.*72*(1), 2022, *https://doi.org/10.32604/cmc.2022.021269*

[17]. R. Wang & S. Zhong & Qin Zhou & J. Tu "*A Trustworthy Data Verification Technique for Cross-Chain Data Sharing Based on Merkle Trees,"* In Proc. Conference: International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), pp.1- 6, 2023, *DOI:10.1109/icdcece57866.2023. 10150492*

[18]. Tyson Winarski "*Big Data Blockchains with Merkle Trees*," Patent US10896171B2, 2021

[19]. M. El-Hindi, T. Ziegler & C. Binnig "*Towards Merkle Trees for High-Performance Data Systems*," In Proc. Conference: SIGMOD 2023, https://doi.org/10.1145/3595647.3595651

[20]. M.S. Niaz & G. Saake, "*Merkle hash tree based techniques for data integrity of outsourced data*," CEUR Workshop Proceedings, vol.1366, pp. 66-71, 2015,

[21]. L. Chen, & R. Movassagh "*Quantum Merkle Trees,"* Quantum vol.8:1380, 2024, *https://doi.org/10.22331/q-2024-06-18-1380*

[22]. S. Jing, X. Zheng, & Z. Chen "*Review and Investigation of Merkle Tree's Technical Principles and Related Application Fields,"* In Proc. Conference: International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), pp.86-90, 2021, *DOI:10.1109/CAIBDA53561.2021.00026*

[23]. S. Han, K. Xu, Zh. Zhu, S. Guo, H. Liu & Z. Li "*Hash-Based Signature for Flexibility Authentication of IoT Devices*," Wuhan Univ. J. Nat. Sci., vol.27(1), pp.1-10, 2022, *https://doi.org/10.1051/wujns/2022271001*

[24]. S. Han S, Y. Wang, D. Shen & C.Wang "*A Multi-Party Privacy-Preserving Record Linkage Method Based on Secondary Encoding*," Mathematics, vol.12(12):1800, 2024, *https://doi.org/10.3390/math12121800*

Manuscript received October 12, 2024.The study is part of a doctoral thesis titled "Models and methods of post-quantum cryptographic security of big data."

1. A.Kubigenova, doctoral student of S.Seifullin Kazakh Agro Technical Research University, Astana, Kazakhstan, phone: 87024584790; e-mail: akku_kubigenova@mail.ru

2. A.Aktayeva , Associate Professor of the Department of Information Systems and Informatics, Kokshetau University named after A. Myrzakhmetov, Kokshetau, Kazakhstan. e-mail: aktaewa@list.ru.

3. G. Yesmagambetova, Lecturer of the Department of Information Systems, Sh.Ualikhanov Kokshetau University, Kokshetau, Kazakhstan. e-mail: gal.esm@mail.ru

4. V. Sukhomlin, Department of Information Security Lomonosov Moscow State University, Moscow, Russian Federation, e-mail: sukhomlin@mail.ru

5. A. Umbetov, Physics and Mathematics School Nazarbayev Intellectual Lyceum, Almaty, Kazakhstan, e-mail: altynbek.umbetov@gmail.com