Mathematical model and software architecture of verifiable audit of compliance with information security policy based on blockchain anchoring*

Valeriia Balatska[1,*,†], Orest Polotai[1,†], and Nazarii Dmytriv[2,†]

[1] Lviv State University of Life Safety, 35 Kleparivska str., 79007 Lviv, Ukraine
[2] Private Higher Education Establishment "European University," 16-V Vernadskyi bl., 03115 Kyiv, Ukraine

Abstract

The paper addresses the problem of ensuring verifiable audit of information system events for compliance with an information security policy under conditions of distributed data processing and storage. The relevance of the study is determined by the fact that traditional centralized logging tools do not provide a sufficient level of immutability, evidential reliability, and independent verification of audit records, especially in environments with increased requirements for integrity, access controllability, and accountability of operations. The purpose of the study is to develop a mathematical model and software architecture of a verifiable audit system for compliance with information security policy, which combines mechanisms for formalized verification of security events, assessment of the level of compliance with established rules, secure off-chain storage of full audit records, and blockchain anchoring of their cryptographic fingerprints. An audit event model, an information security policy model, and a function for verifying event compliance with established access and data processing rules are formalized. An integral indicator of compliance with information security policy and an audit trust indicator are proposed, taking into account record integrity, authenticity of event sources, auditability, security of interaction channels, and cryptographic key management. To ensure audit immutability and evidential reliability, a hybrid on-chain/off-chain model is proposed, in which full event logs are stored outside the blockchain, while only cryptographic hash anchors, timestamps, digital signatures, and links to previous records are written to the distributed ledger. The software architecture of the system is developed, including modules for audit event collection, policy compliance verification, security indicator assessment, off-chain storage, blockchain anchoring, and log integrity verification. The practical value of the proposed approach lies in its applicability to the development of secure audit subsystems in corporate and governmental information systems, where accountability, access controllability, and protection of audit evidence are critical.

Keywords

information security audit; information security policy; verifiable audit; blockchain anchoring; data integrity; event logging; access control; distributed information systems.

1. Introduction

Under the current conditions of organizational digital transformation, information systems increasingly function as distributed hardware and software environments. Within these environments, data processing, transmission, and storage are facilitated through an array of interconnected services, databases, API interfaces, and cloud components [1, 2]. In such a landscape, the information security policy (ISP) serves as the fundamental mechanism for regulating resource access, defining permissible data operations, controlling inter-system interaction, and recording security events.

Standardized approaches to information security management—most notably ISO/IEC 27001:2022—explicitly mandate systematic monitoring, evaluation of security control effectiveness, and internal auditing. Furthermore, the modern NIST CSF 2.0 framework emphasizes the necessity of governance, measurability, and accountability within cybersecurity processes [3, 4].

* CSDP'2026: Cyber Security and Data Protection, May 7, 2026, Lviv, Ukraine
* Corresponding author.
† These authors contributed equally.
v.balatska@ldubgd.edu.ua (V. Balatska); o.polotaj@ldubgd.edu.ua (O. Polotai); nazarii.dmytriv@gmail.com (N. Dmytriv)
0000-0002-6262-6792 (V. Balatska); 0000-0003-4593-8601 (O. Polotai); 0009-0007-0326-3622 (N. Dmytriv)

However, the mere existence of a formalized information security policy does not guarantee the ability to prove its actual enforcement within an information system. In practice, policy compliance monitoring is predominantly implemented through centralized event logs, which remain vulnerable to unauthorized modification, selective deletion, record tampering, or the compromise of privileged accounts. Consequently, foundational NIST log management recommendations stress that logging must be not only comprehensive but also secure, manageable, and suitable for subsequent analysis and verification [5, 6]. Yet, in large-scale distributed environments, this is insufficient, as the additional challenge arises of proving log integrity post-event and enabling the independent verification of audit results [7]. The problem is exacerbated by the fact that modern information systems require more than just event logging; they necessitate the verification of actual actions against established security policy rules. Recent research in information security and compliance advocates for a transition from the formal existence of policies to mechanisms for automated non-compliance detection, risk assessment, and continuous requirement monitoring in dynamic environments [8]. Specifically, approaches have been proposed where automation tools and smart contracts are utilized to identify policy violations, mitigate the human factor, and enhance response efficiency. Simultaneously, research into policy verification for distributed systems demonstrates that without a formal apparatus for rule representation and execution verification, it is impossible to ensure the evidence-based consistency of security requirements in multi-component environments.

In this context, blockchain technologies are viewed as a promising instrument for increasing trust in audit records due to their properties of immutability, distributed storage, cryptographic linking, and traceability of changes. A number of contemporary studies have demonstrated that a blockchain-based approach can significantly strengthen the auditability of sensitive data access, ensure transparency in event recording, and enhance the evidentiary value of logs [9, 10]. For instance, research into blockchain-enabled EHR access auditing shows the feasibility of using an immutable audit trail to control access to medical data. Meanwhile, review and applied works from 2024–2025 confirm the effectiveness of blockchain approaches for tamper-evident logging, traceability, and non-repudiation in distributed systems [11]. Nevertheless, authors emphasize that migrating full logs to the blockchain is inefficient due to scalability constraints, storage costs, processing latency, and the risk of exposing sensitive metadata. Consequently, modern literature increasingly justifies the use of a hybrid on-chain/off-chain model. In this model, comprehensive event logs are stored in traditional or specialized off-chain repositories, while the distributed ledger records only cryptographic fingerprints (hashes), timestamps, digital signatures, or other service-level verification attributes. This approach combines the efficiency of processing large volumes of event data with cryptographically verifiable immutability and independent integrity checks. Furthermore, research into blockchain-based access control frameworks suggests that blockchain should be considered not merely as a storage medium but as a trust infrastructure for verifiable access control, decision tracing, and confirmation of policy compliance [12].

Despite the significant volume of research, existing approaches are primarily focused either on general blockchain auditing or on narrow application domains such as medical records, IIoT, smart contracts, or internal audits. A notable research gap remains: the development of a unified mathematical model and software architecture for the verifiable audit of information security policy compliance that simultaneously:

1. Formalizes the audit event and the policy rule;
2. Provides a quantitative assessment of compliance;
3. Supports secure off-chain storage of full logs;
4. Implements blockchain anchoring for evidentiary weight and independent verification.

This scientific and practical gap determines the relevance of this study. Thus, there is a critical need to develop a mathematical model and software architecture for a verifiable audit system of information security policy compliance based on blockchain anchoring. Such a system would ensure formalized control over security rule execution, quantitative evaluation of the audit circuit

state, cryptographically confirmed log immutability, and the capacity for independent verification of audit results within distributed information systems.

2. Problem statement

This problem gains particular urgency in distributed information systems (DIS), where audit events are generated at various points across the information circuit, transmitted via network channels, processed by diverse software modules, and stored in heterogeneous repositories. In such systems, a dual challenge arises: on one hand, the need to verify the compliance of actual actions with information security policy rules; on the other, the necessity of ensuring the immutability, traceability, and provability of the audit records themselves. Contemporary research confirms that formalized models for security policy specification and verification are critically important for distributed systems, as they are essential for maintaining consistent control over information flows, access rights, and data declassification procedures within multi-component environments [13].

Furthermore, the issue is compounded by the fact that traditional event logs do not inherently guarantee the authenticity of recorded actions nor provide proof that records have remained unaltered since their creation. Applied research in blockchain auditing highlights that classical approaches are limited by an unreliable evidentiary base, difficulties in obtaining corroborating data, high verification costs, and the low operational efficiency of audit procedures. Conversely, the core properties of blockchain-decentralization, immutability, traceability, and cryptographic linking of records-establish the foundation for a new class of verifiable audit mechanisms [14].

Nevertheless, migrating the entire volume of logs directly to a blockchain is not a universal solution. Practical studies on blockchain-based tamper-proof logging indicate that while a distributed ledger ensures integrity, immutability, and non-repudiation, the architectural choice remains a trade-off between transparency, performance, transaction costs, metadata volume, and scalability requirements [15]. Consequently, hybrid on-chain/off-chain approaches are becoming increasingly prevalent. In these models, comprehensive event logs or documents are stored off-chain, while the distributed ledger records only cryptographic fingerprints (hashes), timestamps, and service-level verification attributes [16]. The promise of this approach is further validated in applied domains where auditing is inextricably linked to access control and policy compliance. For instance, research on blockchain-enabled auditing of electronic health records (EHR) demonstrates that an immutable audit trail, combined with access-control policy verification, enables not only the recording of access events but also the assessment of their legitimacy regarding established Purpose-Based Access Control rules. Similar logic is found in modern review papers on blockchain-based access control, which emphasize that blockchain should be viewed not merely as a storage mechanism but as a trust infrastructure for transparent access control, authorization decision confirmation, and the preservation of an evidentiary history of operations [17].

The problem of regulatory compliance and accountability takes on special significance in environments processing personal or critical data. Recent studies indicate that blockchain and cryptographic mechanisms can provide the technical framework for compliance requirements, particularly regarding access control, operational provability, and data processing transparency in IIoT systems and related distributed environments [18]. However, even in these works, the focus remains largely on specific application domains or isolated aspects of compliance enforcement, while the development of a universal mathematical model and software architecture for the verifiable audit of information security policy compliance remains insufficiently addressed.

Thus, an analysis of current research suggests that the central scientific and applied problem lies in the absence of a holistic approach that simultaneously integrates:

1. A formalized representation of the audit event;
2. An information security policy model;
3. A mechanism for verifying event compliance with established rules;
4. A quantitative assessment of the audit circuit state;
5. Hybrid on-chain/off-chain storage;

6. Blockchain anchoring to ensure immutability, provability, and independent verification of audit records.

The resolution of this multifaceted problem is the primary objective of the subsequent research.

3. Results of the Study on the Verifiable Information Security Policy Compliance Audit System

3.1 Formalization of the Subject Area for Verifiable Information Security Policy Compliance Auditing

The formalization of the subject area for verifiable information security policy (ISP) compliance auditing is an essential prerequisite for developing a mathematical model for security event control and a software architecture for an evidentiary audit system. In contrast to traditional logging, where an event is viewed primarily as the fact of recording an action in a system log, the proposed approach interprets an audit event as a formalized information object. This object simultaneously reflects the execution of an operation, the conditions under which it occurred, the result of the ISP compliance check, and its inclusion in the subsequent cryptographic verification circuit.

In modern distributed information systems, auditing must function not only as a log repository but also as a formalized mechanism for establishing the fact of compliance or violation of the information security policy. This is necessitated by the fact that the mere presence of an event log does not guarantee the ability to unambiguously determine: which action was performed, who initiated it, against which resource, in what context, whether it adhered to established rules, and whether an evidentiary link between the primary record and its subsequent state in the log has been preserved. Consequently, the subject area of verifiable auditing must encompass not only subjects, objects, and access operations but also policies, event interpretation procedures, check results, log storage environments, and means of integrity confirmation.

From a methodological standpoint, it is appropriate to consider the subject area based on the principles of integrity, hierarchy, and functional connectivity.

- The principle of integrity dictates that the security policy, audit events, compliance results, logs, and blockchain anchors form a single functional circuit.
- The principle of hierarchy allows for the distinction of several levels of subject area representation: the event source level, the formalized audit event level, the policy rule level, the audit record storage level, and the cryptographic verification level.
- The principle of functional connectivity reflects the causal links between a subject's action, its execution conditions, the policy check result, and the subsequent inclusion of the event into the auditing and evidentiary circuits.

Within the proposed approach, the core entities of the subject area include: access subject, access object, operation, execution context, information security policy rule, audit event, compliance result, off-chain audit log, on-chain anchor record, and the integrity verification procedure.

Access subjects include users, administrators, system processes, application services, integration modules, software agents, and other active entities that initiate or support operations on information resources. Access objects include data, database records, files, API resources, configuration parameters, cryptographic keys, event logs, service messages, and other resources for which the security policy establishes rules for access, processing, and control.

An operation defines the type of action performed on an access object. Generally, these actions include reading, creating, modifying, deleting, transferring, exporting, approving, delegating access, changing configurations, starting a service, confirming a transaction, or any other functionally significant action. However, the operation alone is insufficient for determining its permissibility; therefore, the model introduces the execution context, which covers temporal parameters, request source, device type, network environment, interaction channel state,

authentication level, session information, trusted zone membership, and other attributes defining the conditions of the action.

The key entity of the subject area is the information security policy, which, within this study, is treated as a set of formalized rules defining permissible and impermissible combinations of "subject–object–action–context," as well as requirements for registration, control, and subsequent event verification. Unlike a purely organizational understanding of policy, in a verifiable audit system, the policy must be suitable for algorithmic comparison with actual system events. This property enables the construction of a formalized mechanism for policy enforcement control rather than mere log accumulation.

The execution or attempted execution of a specific action results in an audit event, which is the central unit of analysis in the verifiable audit system. Unlike a standard system record, an audit event in the proposed approach must contain a sufficient set of attributes to unambiguously establish the action, its context, the associated policy rule, and the check result. Thus, an audit event performs a dual function: it records the operation and contains a formalized result of its interpretation relative to the ISP, serving as the base unit for subsequent cryptographic verification.

A critical characteristic of the subject area is the compliance result, which reflects the system's conclusion regarding the permissibility of the recorded action. In a general case, the result may have three states: compliant, non-compliant, or requires additional verification. The introduction of the third state is necessary for describing real-world distributed systems where the permissibility of an action may depend on incomplete, conflicting, or risk-oriented contexts. This approach provides greater model adequacy compared to rigid binary "allow/deny" logic.

Regarding storage organization and evidentiary weight, the subject area features a dual-circuit structure:

1. Off-chain circuit: Designed for storing full audit records containing detailed event attributes, policy check results, technical parameters, and service metadata.
2. On-chain circuit: Provides the fixation of cryptographic integrity proofs for record batches in the form of hashes, timestamps, digital signatures, and references to previous log states.

This separation allows for the combination of the performance and flexibility of off-chain storage with the immutability and independent verifiability of a blockchain environment.

The general functional logic of the subject area can be represented as a sequence of interconnected procedures: a subject initiates an operation on an access object; the action is performed within a defined context; an audit event is generated; the event is mapped against ISP rules; a compliance result is produced; the full event record is stored in the off-chain log; the cryptographic fingerprint of the audit record batch is anchored in the on-chain circuit; and finally, the capacity for independent verification of the log's integrity and evidentiary value is ensured.

Thus, the formalization of the subject area has identified the core entities, established their functional links, and justified the dual-circuit organization of auditing, which integrates operational event control with evidentiary blockchain anchoring. This formalization serves as the foundation for the subsequent development of the mathematical model of the audit event, the compliance verification function, and the blockchain anchoring model for audit records.

The structuring of the subject area is presented as a conceptual scheme (Figure 1), reflecting the relationships between the subject, object, operation, context, ISP, audit event, and the dual-circuit storage and verification system.

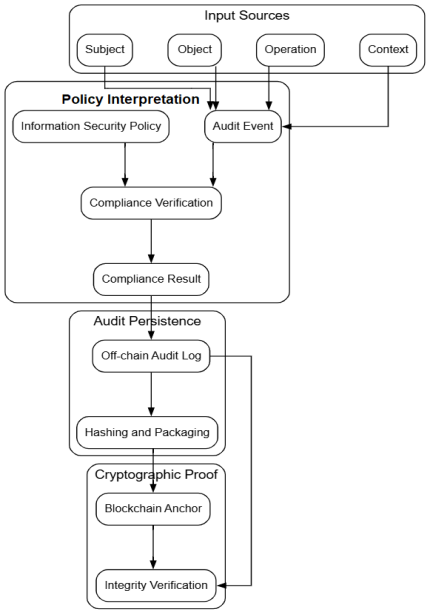

Figure 1: Conceptual model of the verifiable information security policy compliance audit system.

The conceptual model shown in Figure 1 demonstrates that the audit event is the system's central entity, as it is formed based on the interaction of the subject, object, operation, and context; interpreted according to ISP rules; stored in full within the off-chain log; and further secured by a blockchain anchor to confirm its integrity and evidentiary value.

3.2 Mathematical Model for Verifying Compliance of Audit Events with Information Security Policy

The developed formalization of the subject area provides a basis for the mathematical representation of audit events, information security policy rules, and compliance verification procedures. The necessity of such a model is determined by the fact that verifiable audit in distributed systems must provide not only event recording but also formalized establishment of the fact of compliance or violation of the policy at the level of specific actions on information resources [7, 8].

Within the proposed approach, an audit event is represented as an ordered tuple

$$e_i = \langle pid_i, cid_i, s_i, o_i, a_i, r_i, t_i, src_i, ctx_i \rangle,$$

where pid_i is the identifier of the policy rule with which the event is associated; cid_i is the identifier of the control procedure; s_i is the access subject; o_i is the access object; a_i is the action; r_i is the result of action execution; t_i is the event registration time; src_i is the event source; and ctx_i is the context of operation execution.

The information security policy is defined as a set of formalized rules

$$P = \{p_1, p_2, \ldots, p_m\},$$

where each rule p_1 specifies permissible or prohibited conditions for performing an action on an object by a certain subject in a given context. Formally, the rule can be represented as

$$p_j : \langle S_j, O_j, A_j, C_j \rangle \rightarrow D_j,$$

where S_j is the set of subjects, O_j is the set of objects, Aj is the set of permissible actions, Cj is the set of contextual conditions, and D_j $allow, deny, review$ is the policy decision.

To determine whether an event complies with policy rules, the following compliance verification function is introduced

$$f_P(e_i)=\begin{cases}1, & \text{if } e_i \neq P,\\ 0, & \text{if } e_i \neq P.\end{cases}$$

Here, $e_i \neq P$ means that the event attributes satisfy at least one policy rule that allows the corresponding action in the given context. To account for events with uncertain or risky status, it is appropriate to use an extended classification function:

$$g_P(e_i)=\begin{cases}1, & \text{the event complies with the policy},\\ 0, & \text{the event does not comply with the policy},\\ -1, & \text{the event requires additional review}.\end{cases}$$

For quantitative assessment of the compliance level of a set of events with the policy, an integral indicator is introduced

$$C_{comp}=\frac{1}{N}\sum_{i=1}^{N} f_P(e_i),$$

where N is the number of verified audit events. The value $C_{comp}\in[0,1]$ reflects the proportion of events that comply with the information security policy rules.

Since individual events have different levels of criticality, it is advisable to use a weighted form of the indicator

$$C_{comp}^{(w)}=\frac{\sum_{i=1}^{N} w_i f_P(e_i)}{\sum_{i=1}^{N} w_i},$$

where w_i is the criticality weight of the event. The value of wi may be determined based on the resource class, operation type, subject privilege level, or object sensitivity.

To assess not only event compliance but also the security of the audit contour itself, an audit trust indicator is introduced

$$T_{audit}=\sum_{k=1}^{K}\lambda_k S_k,\ \sum_{k=1}^{K}\lambda_k=1,$$

where S_k are partial indicators of the audit contour state, and λ_k are their weight coefficients. The set S_k may include: S_{int}, record integrity; S_{auth}, authenticity of event sources; S_{ac}, correctness of access control; S_{aud}, completeness of logging; S_{api}, security of interaction channels; and S_{key}, manageability of key material.

Consequently, the mathematical model for verifying the compliance of audit events with the information security policy enables the formalization of audit events, policy rules, and comparison procedures, while facilitating a quantitative assessment of policy enforcement. This model serves as the foundational framework for establishing an evidentiary audit contour through blockchain anchoring. The structural scheme for verifying audit event compliance against the information security policy is illustrated in Figure 2. This scheme delineates the procedural sequence, including the extraction of event attributes, the identification of relevant policy rules, the comparison of events against established conditions, and the subsequent categorization of events into one of three distinct classes: compliant, non-compliant, or requiring additional review.

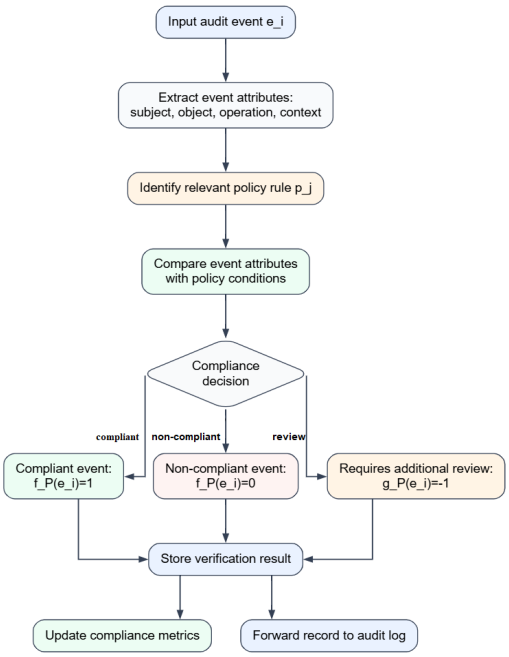

Figure 2: Structural scheme of compliance verification of an audit event against information security policy

As shown in Figure 2, the compliance decision is based on the formal comparison of the audit event attributes with the conditions defined by the relevant policy rule. The result of this comparison is further used to update compliance metrics and to forward the enriched audit record to the audit log. This ensures that each recorded event receives not only a descriptive representation but also an interpretable compliance status.

3.3 Model of Blockchain Anchoring of Audit Records

One of the key requirements for a verifiable audit system is to ensure immutability, evidential reliability, and independent verification of audit records after they have been recorded [19]. In traditional centralized logging systems, even the presence of backup mechanisms, access control, and cryptographic protection does not eliminate the risk of hidden modification, selective deletion, or retrospective correction of records. Therefore, modern studies consider blockchain as a trust infrastructure that enables tamper-evident logging, immutability of event history, traceability of changes, and cryptographically confirmed evidential reliability of audit.

At the same time, direct storage of the full content of audit logs in a blockchain is not feasible due to scalability limitations, increased latency, higher transaction costs, and the risk of disclosing service or confidential information. Therefore, this paper proposes a model of blockchain

anchoring of audit records, in which full event logs are stored in an off-chain environment, while only cryptographic fingerprints of event packages, timestamps, digital signatures, and links to previous records are fixed in the blockchain registry.

Let the set of audit events be represented as

$$e_i = \langle pid_i, cid_i, s_i, o_i, a_i, r_i, t_i, src_i, ctx_i \rangle,$$

where pid_i is the identifier of the policy rule with which the event is associated; cid_i is the identifier of the control procedure; s_i is the access subject; o_i is the access object; a_i is the action; r_i is the result of action execution; t_i is the event registration time; src_i is the event source; and ctx_i is the context of operation execution.

The information security policy is defined as a set of formalized rules

$$P = \{p_1, p_2, \ldots, p_m\},$$

where each rule p_j specifies permissible or prohibited conditions for performing an action on an object by a certain subject in a given context. Formally, the rule can be represented as

$$p_j : \langle S_j, O_j, A_j, C_j \rangle \rightarrow D_j,$$

where S_j is the set of subjects, O_j is the set of objects, A_j is the set of permissible actions, C_j is the set of contextual conditions, and $D_j \in \{allow, deny, review\}$ is the policy decision.

To determine whether an event complies with policy rules, the following compliance verification function is introduced

$$f_P(e_i) = \begin{cases} 1, & \text{if } e_i \neq P, \\ 0, & \text{if } e_i \neq P. \end{cases}$$

Here, $e_i \neq P$ means that the event attributes satisfy at least one policy rule that allows the corresponding action in the given context. To account for events with uncertain or risky status, it is appropriate to use an extended classification function:

$$g_P(e_i) = \begin{cases} 1, & \text{the event complies with the policy}, \\ 0, & \text{the event does not comply with the policy}, \\ -1, & \text{the event requires additional review}. \end{cases}$$

For quantitative assessment of the compliance level of a set of events with the policy, an integral indicator is introduced

$$C_{comp} = \frac{1}{N} \sum_{i=1}^{N} f_P(e_i),$$

where N is the number of verified audit events. The value $C_{comp} \in [0,1]$ reflects the proportion of events that comply with the information security policy rules.

Since individual events have different levels of criticality, it is advisable to use a weighted form of the indicator

$$C_{comp}^{(w)} = \frac{\sum_{i=1}^{N} w_i f_P(e_i)}{\sum_{i=1}^{N} w_i},$$

where w_i is the criticality weight of the event. The value of w_i may be determined based on the resource class, operation type, subject privilege level, or object sensitivity.

To assess not only event compliance but also the security of the audit contour itself, an audit trust indicator is introduced

$$T_{audit} = \sum_{k=1}^{K} \lambda_k S_k, \quad \sum_{k=1}^{K} \lambda_k = 1,$$

where S_k are partial indicators of the audit contour state, and λ_k are their weight coefficients. The set S_k may include: S_{int}, record integrity; S_{auth}, authenticity of event sources; S_{ac}, correctness of access control; S_{aud}, completeness of logging; S_{api}, security of interaction channels; and S_{key}, manageability of key material.

Consequently, the mathematical model for verifying the compliance of audit events with the information security policy enables the formalization of audit events, policy rules, and comparison procedures, while facilitating a quantitative assessment of policy enforcement. This model serves as the foundational framework for establishing an evidentiary audit contour through blockchain anchoring. The structural scheme for verifying audit event compliance against the information security policy is illustrated in Figure 2. This scheme delineates the procedural sequence, including the extraction of event attributes, the identification of relevant policy rules, the comparison of events against established conditions, and the subsequent categorization of events into one of three distinct classes: compliant, non-compliant, or requiring additional review.

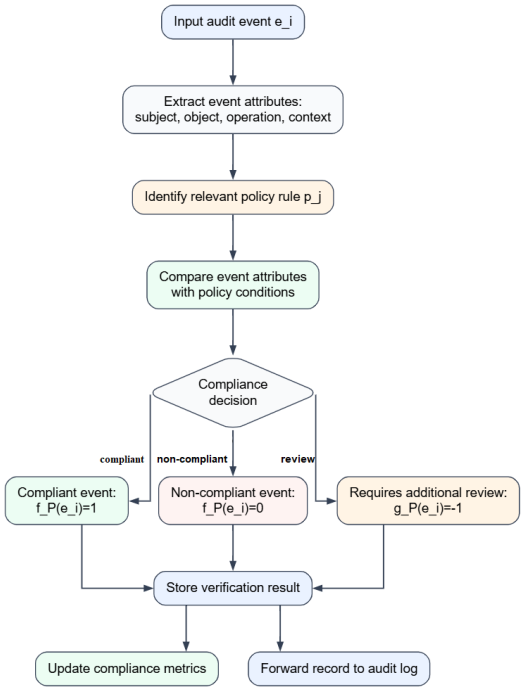

Figure 2: Structural scheme of compliance verification of an audit event against information security policy

As shown in Figure 2, the compliance decision is based on the formal comparison of the audit event attributes with the conditions defined by the relevant policy rule. The result of this comparison is further used to update compliance metrics and to forward the enriched audit record to the audit log. This ensures that each recorded event receives not only a descriptive representation but also an interpretable compliance status.

3.4 Conceptual Model and Software Architecture of the Verifiable Audit System

The developed formalization of the subject area, the mathematical compliance verification model, and the blockchain anchoring model provide the basis for building the conceptual model and software architecture of the verifiable audit system. Such an architecture should reflect not only the composition of the main components but also the logic of their functional interaction within a unified contour of audit record formation, verification, storage, and validation. Unlike traditional logging systems, where log files or database records are the final result of event registration, in the proposed approach an audit event passes through a sequence of stages: formation, normalization,

policy compliance verification, result recording, full storage in an off-chain environment, cryptographic aggregation, and anchoring in the blockchain contour.

Conceptually, the verifiable audit system consists of two interconnected functional contours: the operational audit contour and the cryptographic verification contour. The operational contour provides the collection of events from information system sources, their structured representation, verification of compliance with the information security policy, log formation, and analytical processing. The cryptographic verification contour is responsible for creating hash fingerprints of audit event packages, digital signing, blockchain anchoring, and subsequent independent verification of the immutability of recorded data. The system should include the following main functional modules: audit event sources; an event collection and normalization module; an information security policy compliance verification module; a compliance and audit trust indicator assessment module; an off-chain repository of full audit records; a hashing and package formation module; a blockchain anchoring module; and an integrity verification module with an audit interface. Audit event sources generate the primary information flow for the system. These include user subsystems, application services, authorization servers, API gateways, databases, SIEM/monitoring components, network devices, cryptographic modules, system logs, and other nodes that generate security events. The event collection and normalization module receives primary records from sources, transforms them into a unified formalized representation, and prepares them for further verification. At this stage, event attributes are extracted, brought to a standardized format, timestamps are synchronized, and the subject, object, operation, and context are identified.

The information security policy compliance verification module is the central element of the system. Its function is to compare the attributes of each event with the set of policy rules P, determine the admissibility or inadmissibility of the recorded action, and generate a formal verification result. It is at this level that the functions $f_P(e_i)$ and $g_P(e_i)$, described in subsection 3.2, are implemented. The compliance and audit trust indicator assessment module aggregates verification results and computes generalized characteristics of the audit contour state, including C_{comp}, $C^{(w)}_{comp}$, T_{audit}, C_{anch}, and $C^{(w)}_{anch}$. The presence of a separate analytical module makes it possible to use the system not only as a technical logging tool but also as an instrument for quantitative assessment of the level of compliance with information security policy. The off-chain repository of full audit records is intended to store detailed events together with context, policy verification results, service attributes, error logs, and references to related transactions. Considering the need for scalability and fast search, such a repository can be implemented using relational or document-oriented databases.

The hashing and package formation module aggregates full audit records into packages B_q, generates cryptographic fingerprints h_q, attaches service metadata, and prepares data for blockchain anchoring. The blockchain anchoring module publishes the generated records a_q in a permissioned or consortium blockchain environment. Within this module, smart contracts or specialized services for interaction with the blockchain network may be used. The use of smart contracts and permissioned blockchain components is consistent with practical blockchain engineering approaches described for Ethereum-based and enterprise blockchain systems [20, 21]. The integrity verification module and audit interface ensure verification of correspondence between off-chain packages and blockchain anchors, visualization of the audit contour state, report generation, support for analytical scenarios, and provision of an evidence base for internal or external audit.

Thus, the proposed conceptual model and software architecture of the verifiable audit system provide a structured representation of the audit event life cycle; integration of formal verification of compliance with information security policy with an evidence-based blockchain anchoring contour; separation of full event storage and cryptographic confirmation of integrity; and modularity suitable for software implementation in corporate and governmental information systems. The resulting architecture is presented in Figure 4 and reflects the interaction between

heterogeneous event sources, the collection and control layer, the persistence layer, the blockchain anchoring layer, and the verification and audit layer.

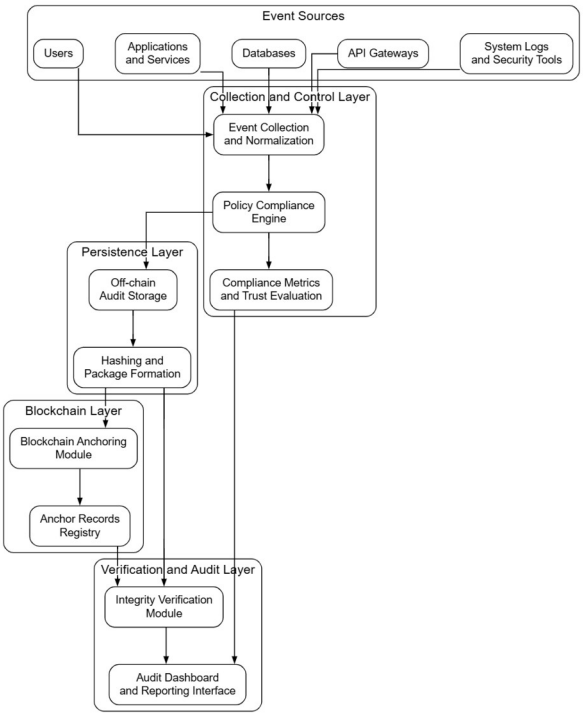

Figure 4: Conceptual model and software architecture of the verifiable audit system

As shown in Figure 4, audit events are collected from multiple sources, including users, applications and services, databases, API gateways, and system logs and security tools. At the collection and control layer, these events are normalized and processed by the policy compliance engine, which determines whether the recorded actions conform to the information security policy. The verified records are then stored in the off-chain audit repository, aggregated into packages, and prepared for blockchain anchoring. The blockchain layer generates and stores anchor records, while the verification and audit layer uses both off-chain packages and blockchain anchors to verify integrity and support evidence-based reporting.

4. Algorithmic and Software Implementation of the Verifiable Audit System

4.1 Algorithm of Operation of the Verifiable Audit System

The operation algorithm of the verifiable audit system for compliance with information security policy implements a sequence of procedures that ensure the transition from a primary security event to the formation of a verified audit record with cryptographically confirmed integrity. Unlike traditional logging, the proposed algorithm combines control of event compliance with the information security policy, storage of the full audit record in an off-chain environment, and anchoring of its evidential representation in the blockchain contour.

At the first stage, an event is received from an audit source. Such sources may include user subsystems, application services, authorization servers, databases, API gateways, system logs, and other components of a distributed information system. The obtained event undergoes a normalization procedure, within which its main attributes are extracted and unified: access subject, access object, operation, time characteristics, event source, and execution context. At this stage, the primary log record is converted into a formalized representation suitable for further automated processing.

At the second stage, an audit event e_i is formed from the normalized record and transmitted to the information security policy compliance verification module. At this stage, the relevant rule $p_j \in P$ is searched for, event attributes are compared with the set of admissibility conditions, and the compliance function $f_P(e_i)$ or the extended classification function $g_P(e_i)$ is computed. As a result, the event is assigned to one of three classes: compliant with the policy; non-compliant with the policy; or requiring additional review. Thus, already at the stage of primary processing, the audit event receives not only a descriptive but also an interpretive status.

At the third stage, an extended audit record is formed, including the primary event attributes, the policy rule identifier, the compliance verification result, and service metadata. The formed record is stored in the off-chain audit log and simultaneously used to update integral indicators of the audit contour state, in particular C_{comp}, $C^{(w)}_{comp}$ and T_{audit}. As a result, the system provides not only event recording but also accumulation of quantitative characteristics necessary for assessing the level of information security policy enforcement.

At the fourth stage, after a specified number of events has been accumulated or a defined time interval has expired, a package of audit records B_q is formed. For this package, a cryptographic fingerprint $h_q=H(B_q \| meta_q)$ is computed, after which an anchor record a_q is generated, containing the package identifier, hash fingerprint, timestamp, digital signature, and link to the previous anchor record. This approach makes it possible to move from storing isolated events to forming a consistent evidential chain of audit states.

At the fifth stage, the generated anchor record is transmitted to the blockchain contour, where it is fixed as an element of the audit evidence chain. At the same time, the full content of the package remains in the off-chain repository, which ensures system scalability and reduces the load on the blockchain network. Thus, the separation of the content and evidential levels of audit is implemented: full information is stored in a high-performance environment, while its cryptographic confirmation is stored in a distributed immutable ledger.

At the final stage, an independent integrity verification procedure is performed. For this purpose, the package B_q is retrieved from the off-chain log, its hash fingerprint is recomputed, and the function $V(B_q, a_q)$ is verified by comparing the calculated value with the data recorded in the blockchain anchor. In the case of a match, the immutability of the audit record package is confirmed; in the case of a mismatch, an integrity violation or a break in the evidential chain is recorded. Such a procedure allows the system to be used not only for internal monitoring but also for independent audit and evidential confirmation of log reliability.

Thus, the proposed algorithm provides functional integration of operational event control, formalized verification of compliance with information security policy, off-chain logging, and blockchain confirmation of integrity, which forms a methodological and applied basis for implementing a verifiable audit system in distributed information environments. The generalized scheme of the verifiable audit system operation is presented in Figure 5.

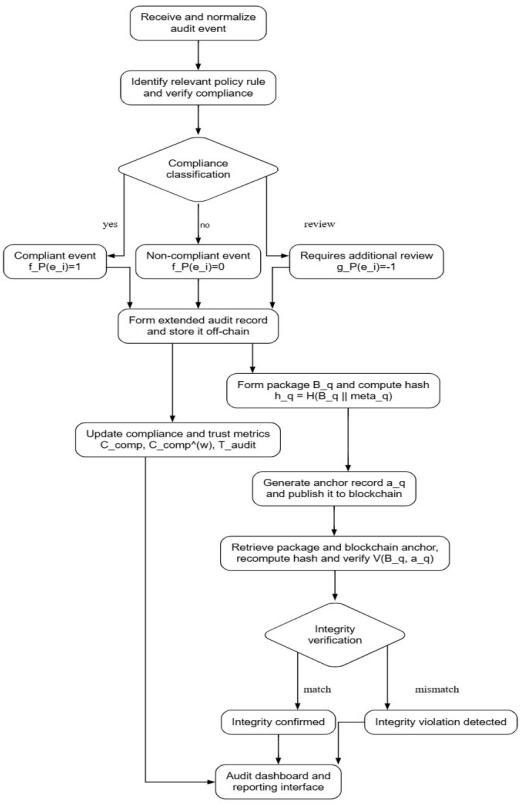

Figure 5: Algorithm of the verifiable audit system operation

As shown in Figure 5, the proposed algorithm combines two logically connected processes: operational verification of audit events and cryptographic confirmation of the integrity of audit records. Each event is first interpreted according to the relevant information security policy rule and assigned a compliance status. Then the extended audit record is stored off-chain, included in a package, hashed, and anchored in the blockchain contour. The subsequent verification procedure compares the recalculated package hash with the corresponding blockchain anchor, which makes it possible to confirm the integrity of the audit record or detect its violation.

4.2 Software Implementation of the Verifiable Audit System

The software implementation of the verifiable audit system for compliance with information security policy is based on a modular construction principle and involves separating functionally complete components, each of which implements a specific part of the proposed model. This approach ensures flexibility, scalability, controllability of updates, and the possibility of integrating the system with heterogeneous event sources and external information security services. Within the software implementation, the following modules should be distinguished: event collection module, normalization module, policy compliance verification module, indicator calculation module, off-chain storage module, packaging and hashing module, blockchain anchoring module, and integrity verification module. The event collection module receives records from heterogeneous audit sources, including application services, access logs, API calls, transaction logs, and system messages. The normalization module converts events into a single formalized representation, unifying attributes, time formats, identifiers of subjects and objects, and contextual parameters.

The compliance verification module compares event attributes with the set of rules P, determines the relevant policy rule, and generates the compliance status. It is in this module that the logic for computing $f_P(e_i)$ and $g_P(e_i)$ is implemented, and the interpretation result is generated as part of the extended audit record. As a result, the audit system performs not only the function of

event recording but also the function of formal policy compliance control. The indicator calculation module aggregates verification results and calculates quantitative characteristics of the audit contour state. In particular, it computes the integral compliance indicator C_{comp}, its weighted form $C^{(w)}_{comp,}$ the audit trust indicator T_{audit}, as well as the confirmed integrity indicators C_{anch} and $C^{(w)}_{anch}$. The presence of such a module allows the system to be used not only as a logging tool but also as an instrument for quantitative assessment of compliance with information security policy.

The off-chain storage module is responsible for accumulating full audit records in a structured form suitable for search, analytical processing, filtering, and incident investigation. Its structure should include logical entities for audit events, policy rules, audit record packages, packaging metadata, and verification results. This organization ensures a coherent representation of the full event life cycle — from the moment of its occurrence to the stage of evidence-based anchoring. The packaging and hashing module forms packages B_q, computes cryptographic fingerprints h_q, and prepares service metadata for blockchain anchoring. The blockchain anchoring module creates and publishes anchor records a_q in a permissioned blockchain environment, while the integrity verification module checks correspondence between off-chain packages and on-chain records by recomputing hashes, comparing package identifiers, verifying digital signatures, and generating a conclusion about the state of log integrity. From a software perspective, the system can be implemented as a multi-level service-oriented architecture in which interaction between modules is carried out through APIs or internal service interfaces. For application logic, it is advisable to use a high-level programming language, cryptographic libraries, and a database management system for off-chain storage. For the blockchain contour, it is appropriate to use a permissioned or consortium blockchain platform, which makes it possible to provide controlled access to the ledger, support smart contracts, and confirm the immutability of audit records. From the access-control perspective, the architecture is also aligned with the zero trust approach, according to which access decisions should be based on continuous verification, contextual attributes, least privilege, and explicit trust evaluation [22]. The logic of software implementation involves the formation of three main groups of data: extended audit records, blockchain anchor records, and integrity verification results. Extended audit records contain the attributes of the primary event, the identifier of the relevant policy rule, the compliance status, contextual parameters, and service metadata. Blockchain anchor records contain only the evidential representation of audit packages, namely package identifiers, cryptographic hash fingerprints, timestamps, digital signatures, and references to previous anchor records. Integrity verification results reflect the correspondence between the off-chain audit package and the data fixed in the blockchain contour.

Such organization makes it possible to separate the operational level of audit from the evidential level of integrity confirmation. Full audit records remain available in the off-chain storage for search, filtering, analytical processing, and incident investigation, while the blockchain contour stores only compact cryptographic evidence. This reduces the load on the blockchain network and avoids placing the full content of audit logs in the distributed ledger, while preserving the possibility of independent verification of record immutability.

Thus, the software implementation of the verifiable audit system provides practical implementation of the proposed mathematical compliance verification model and the blockchain anchoring model, while the modular architecture creates prerequisites for its use in systems where access controllability, accountability of subject actions, and evidential reliability of event logs are critical.

4.3 Example of Audit Record Verification

To demonstrate the operation of the proposed approach, let us consider an example of verifying an audit record package formed in the off-chain audit log. Let the package B_q contain a set of normalized audit events, each of which has previously been checked for compliance with the

information security policy using the functions $f_P(e_i)$ or $g_P(e_i)$. After the completion of a specified accumulation interval or after reaching the defined event threshold, the package is supplemented with service metadata meta_q. Such metadata may include the package identifier, creation timestamp, number of events, identifier of the previous package, and technical parameters of the logging source. On the basis of the formed package, the system computes the cryptographic fingerprint
$h_q = H(B_q \parallel meta_q)$ and generates the anchor record $a_q = <id_q, h_q, t_q, sig_q, prev_q>$. The full content of the package remains in the off-chain repository, while only its evidential representation is transferred to the blockchain contour. This evidential representation includes the hash fingerprint, timestamp, digital signature, and reference to the previous anchor. Therefore, the system does not duplicate the full audit log in the blockchain, but fixes the cryptographic state of the off-chain log at a specific moment in time.

The verification procedure is performed by retrieving the package B_q from the off-chain storage, recomputing its hash fingerprint, and comparing the obtained value with the hash fixed in the corresponding anchor record a_q. If the condition $H(B_q \parallel meta_q) = h_q$ is satisfied and the digital signature sig_q is correct, the verification function $V(B_q, a_q)$ returns 1, confirming the integrity of the audit package. Otherwise, the function returns 0, and the system records an integrity violation or a break in the evidential chain. This example demonstrates that the evidential reliability of audit is achieved not by transferring the full content of logs to the blockchain, but by cryptographically anchoring the states of the off-chain audit log. As a result, the system preserves the possibility of fast search, analytical processing, and incident investigation within the full audit repository, while the blockchain contour acts as an independent mechanism for confirming the immutability and sequence of audit records. Thus, the described verification procedure confirms the practical applicability of the proposed approach for systems in which access controllability, accountability of subject actions, evidential reliability of logs, and the possibility of independent integrity verification are critical.

5. Conclusions

The paper develops a mathematical model and software architecture of verifiable audit of compliance with information security policy based on blockchain anchoring. The proposed approach is aimed at solving the problem of insufficient evidential reliability of traditional centralized event logs, which do not always provide the possibility of independently verifying record integrity after their formation. The subject area of verifiable audit of compliance with information security policy is formalized. Within it, an audit event is considered not only as a logging fact but as a structured information object containing action attributes, execution context, policy verification result, and relation to the evidential contour. Such formalization made it possible to distinguish the operational, control, and evidential levels of audit.

A mathematical model for verifying compliance of audit events with information security policy is developed. Within the model, the event is represented as an ordered tuple, the policy as a set of formalized rules, and the control procedure as a compliance function that allows events to be classified as compliant with the policy, non-compliant, or requiring additional review. An integral compliance indicator C_{comp}, its weighted form $C^{(w)}_{comp}$, and an audit trust indicator T_{audit} are proposed. A model of blockchain anchoring of audit records is proposed. In this model, full event logs are stored in an off-chain environment, while only cryptographic fingerprints of packages, timestamps, digital signatures, and links to previous records are fixed in the blockchain contour. This model ensures cryptographically confirmed integrity, non-repudiation, traceability, and the possibility of independent log verification without placing the full content of events in the blockchain. The conceptual model and software architecture of the verifiable audit system are developed. The architecture includes modules for event collection and normalization, policy compliance

verification, indicator calculation, off-chain storage, packaging and hashing, blockchain anchoring, and integrity verification. The operation algorithm of the system and the logic of software implementation of its main modules are described.

The practical value of the proposed approach lies in its applicability to the development of secure audit subsystems in corporate, departmental, and governmental information systems, where the integrity of event logs, accountability of user actions, access controllability, and evidential reliability of verification results are critical. Further research should focus on experimental evaluation of the performance of the proposed architecture under different event packaging parameters, blockchain network node counts, and audit flow intensities.

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] V. Susukailo, S. Vasylyshyn, I. Opirskyy, V. Buriachok & O. Riabchun. Cybercrimes investigation via honeypots in cloud environments. Pap. present. CEUR Workshop Proc., 2021, 2923, 91-96.

[2] Y. Martseniuk, A. Partyka, O. Harasymchuk, V. Cherevyk, N. Dovzhenko. Research of the Centralized Configuration Repository Efficiency for Secure Cloud Service Infrastructure Management, 2025, CEUR Workshop Proc., 3991, pp. 260-274.

[3] ISO/IEC 27001:2022, Information security, cybersecurity and privacy protection. Information security management systems. Requirements, ISO, 2022. https://www.iso.org/standard/27001

[4] National Institute of Standards and Technology, NIST Cybersecur. Framew. (CSF) 2.0, NIST CSWP 29 (2024). doi:10.6028/NIST.CSWP.29.

[5] K. Kent, M. Souppaya, Guide to Computer Security Log Management, NIST Spec. Publ., 800-92 (2006). https://csrc.nist.gov/pubs/sp/800/92/final.

[6] Joint Task Force, Security and Privacy Controls for Information Systems and Organizations, NIST Spec. Publ., 800-53 Rev. 5 (2020). doi:10.6028/NIST.SP.800-53r5.

[7] L. Alevizos, Automated cybersecurity compliance and threat response using AI, blockchain and smart contracts, Int. J. Inf. Technol., 17 (2025), 767-781. doi:10.1007/s41870-024-02324-9.

[8] F. Wolf, P. Müller, Verifiable Security Policies for Distributed Systems, in: Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2024. doi:10.1145/3658644.3690303.

[9] F. Ullah, S. Zhang, K. Ullah, F. A. Khan, M. Alazab, Blockchain-enabled EHR access auditing: Enhancing data security through immutable audit trails, Heliyon 10 (2024). https://www.sciencedirect.com/science/article/pii/S2405844024104380.

[10] J. Morillo Reina, T. Mateo Sanguino, Decentralized and Secure Blockchain Solution for Tamper-Proof Logging Events, Future Internet 17(3) (2025) 108. doi:10.3390/fi17030108.

[11] Y. Zhang, Y. Han, K. Song, Auditing in the blockchain: a literature review, Frontiers in Blockchain 8 (2025). doi:10.3389/fbloc.2025.1549729.

[12] A. Tawfik, A. Al-Ahwal, A. Tag Eldien, H. Zayed, Blockchain-based access control and privacy preservation in healthcare: a comprehensive survey, Clust. Comput., 28 (2025) 529. doi:10.1007/s10586-025-05308-x.

[13] A. Isazade, A. Malik, M. B. Alshawki, Blockchain-Enabled GDPR Compliance Enforcement for IIoT Data Access, J. Cybersecur. Priv., 5(4) (2025) 84. doi:10.3390/jcp5040084.

[14] V. Balatska, I. Opirskyy, N. Slobodian, Blockchain for enhancing transparency and trust in government registries, in: Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II 2024), CEUR Workshop Proc., Vol. 3826, Kyiv, Ukraine, 2024, pp. 50-59.

[15] V. Balatska, V. Poberezhnyk, I. Opirskyy, Utilizing blockchain technologies for ensuring the confidentiality and security of personal data in compliance with GDPR, in: Cyber Security and Data Protection (CSDP 2024), CEUR Workshop Proc., Vol. 3800, Lviv, Ukraine, 2024, pp. 70- 80.
[16] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain Technology Overview, NIST Interagency Report 8202, Natl. Inst. Stand. Technol., 2018. doi:10.6028/NIST.IR.8202.
[17] V. Hu, D. Kuhn, D. Ferraiolo, J. Voas, Blockchain for Access Control Systems, NIST Interagency Report 8403, Natl. Inst. Stand. Technol., 2022. doi:10.6028/NIST.IR.8403.
[18] I. Opirskyy, V. Poberezhnyk, V. Balatska, Method of Ensuring Data Integrity and Authenticity based on the Integration of Blockchain Technology, in: Digital Economy Concepts and Technologies (DECaT 2025), CEUR Workshop Proc., Vol. 4029, Kyiv, Ukraine, 2025, pp. 96-109.
[19] V. Balatska, V. Poberezhnyk, I. Opirskyy, Method for Ensuring the Reliability and Security of Personal Data in Blockchain Systems of State Registers, in: Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2025), CEUR Workshop Proc., Vol. 3991, Kyiv, Ukraine, 2025, pp. 293-310.
[20] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princet. Univ. Press, Princeton, 2016.
[21] A. Antonopoulos, G. Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, Sebastopol, 2018.
[22] S. Rose, O. Borchert, S. Mitchell, S. Connelly, Zero Trust Architecture, NIST Special Publication 800-207, Natl. Inst. Stand. Technol., 2020. doi:10.6028/NIST.SP.800-207.