Journal of Theoretical and Applied Information Technology
30th April 2026. Vol.104. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DATA SECURITY IN DISTRIBUTED DATABASES: MODERN ENCRYPTION AND AUTHENTICATION METHODS

SERGII BATAIEV[1], VIKTOR KYRYCHENKO[2], VOLODYMYR STANKO[3], SERHII VOLOSHCHUK[4], TARAS STARUSHENKO[5]

[1]PgDip in Digital Leadership, Head of the Department of Technology, University of Warwick, Coventry, United Kingdom; ELEKS inc., Lviv, Ukraine.
[2]Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor of the Department of Computer Science, Faculty of Information Technologies, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine
[3]Ph.D. (Economics), Associate Professor, Department of Information Technologies, Stepan Gzhytskyi National University of Veterinary Medicine and Biotechnologies of Lviv, Lviv, Ukraine
[4]PhD (Phys. & Math.), Associate Professor, Department of Modelling of Complex Systems, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
[5]PhD, Department of Cybersecurity, Faculty of Physics and Technology, NTUU "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv Ukraine
E-mail: [1]serg.bataiev@ujis.in.ua

ABSTRACT

Distributed databases are a core component of modern information systems, and their security has become increasingly critical. According to the EU Agency for Network and Information Security (ENISA) Threat Landscape 2024, cyberattacks on distributed systems increased by 47% over the past year, while traditional security mechanisms designed for centralized architectures fail to address vulnerabilities related to inter-node communication, data replication, and consensus protocols. This study aims to develop a universal approach to securing distributed databases by systematizing architecture-specific threats, conducting a comparative analysis of contemporary encryption and authentication mechanisms, and proposing an architectural security framework tailored to sharding-based, master–slave replication, and consensus-based systems. Threats were classified along three dimensions—architectural level, compromise type, and attack vector—using a systematic mapping study. Multi-Criteria Decision Analysis (MCDA) was applied to evaluate protection methods by security, performance, implementation complexity, and compliance with ISO/IEC 27001:2022 and PCI DSS v4.0.1, and to construct a decision tree for selecting appropriate security controls. The analysis draws on 87 peer-reviewed publications (Scopus, IEEE, ACM; 2020–2025), documentation of Apache Cassandra, MongoDB, and CockroachDB, official NIST and ENISA reports, and empirical data from 245 CVE vulnerabilities. Emerging threats for 2024–2025 were identified using MCDA weighting, Spearman correlation analysis ($n = 87$, $p < 0.05$), and YAKE keyword extraction. Results indicate that insider threats are significantly more prevalent in distributed systems (34%) than in centralized ones (19%), with critical CVSS scores (7.8–9.0). Man-in-the-middle attacks between nodes remain dominant, with 68% caused by ineffective mutual authentication. AES-256-GCM offers the best performance–security trade-off for data at rest, ML-KEM is suitable for quantum-resistant use cases, while homomorphic encryption remains impractical for production. Benchmarking shows a combined security overhead of 28–35% while maintaining regulatory compliance. The proposed framework provides architecture-specific, compliant, and performance-aware security recommendations, including applicability to Ukraine's financial sector regulations.

Keywords: *Cryptographic Methods, Authentication Mechanisms, Cyber Threat Taxonomy, Consensus Algorithms, Security Architectural Framework*

1. INTRODUCTION

Distributed bases data play back important role in modern infrastructure informational systems, which there are scalable, fault-tolerant, geographically available and supports global applications. However

distributed architecture creates new threats for security data, because traditional infrastructure protection, intended for centralized systems, not takes into account features vulnerabilities internodal communications, replication data and algorithms consensus [1]. As shows European Union Agency for Network and Information Security (further – ENISA) Threat Landscape 2024 report, number cyberattack on distributed systems by last year increased by 47%, and 34% of attacks were related with compromising channels internodal communication and absence internal control access [2]. Active financial sector, which actively uses distributed bases data for carrying out transactions, recorded 488 publicly reported incidents security in 2023-2024, therefore urgently need create complex methods protection distributed systems.

Existing level development in industries security distributed bases data can to characterize as scattered methods and absence attention to nature different architectural models. Systematic review accessibility data about cyber risks showed that with all research security bases data only 23% are dedicated specific threat for distributed architectures, and others research concentrated on general mechanisms protection without consideration features mechanisms sharding, replication and consensus [3].

Previous research has taken a number of approaches to distributed database security. CryptNoSQL [6] was suggested as a way of querying encrypted NoSQL data but the authors only performed a test with regard to cloud-based querying but did not consider inter-node authentication as well as consensus integrity. Zhang et al. [7] have shown that homomorphic encryption can be used in aggregate operations but realized prohibitive scaling to production workloads. Garcia et al. [8] considered quantum-resistant TLS in distributed communication, and Ghaffari et al. [9] gave an overview of the blockchain-based authentication without practical benchmarking with a real distributed database workload. Importantly, none of these works offers a composite, architecture-specific model that integratively considers encryption, authentication and consensus security as a whole, within performance limits that can be measured. The current work contrasts with existing literature in three aspects: (1) it empirically bases classification of threats on 245 real CVE vulnerabilities instead of synthetic workloads; (2) in quantitatively benchmarking security mechanisms under production-representative workloads, using TPC-C and YCSB; and (3) in generating architecture-specific security guidance that is differentiated among sharding, replication, and consensus-based

Official National Institute of Standards and Technology (further – NIST) Cybersecurity 8504 report leaks light on that the fact that exists serious deficit standardization control access to distributed bases NoSQL data, and exactly absence clear recommendations of compromise between high degree coherence and overhead expenses on security [4]. According to with NIST analysis and Privacy Program for fiscal year 2024 year, protection distributed systems under consideration as one with regions research that trace continue in in the future, because complexity threats is growing, and available mechanisms protection there are insufficient [5].

Existing research methods encryption distributed bases data relate to mainly encryption in encryption at rest and in transit, that possible in centralized architecture, without consideration special questions cross – shard encryption, encrypted replication and security consensus protocols. Homomorphic encryption, which theoretically allows calculate encrypted message, has cost productivity 1000–10000% and with its high load on production systems there are impossible. NIST has proposed quantum – resistant cryptography, which maybe become necessary in 2024, when quantum computers will become threat, but its need adapt to needs distributed architectures, where delay on each operation cryptography in times smaller by number interactions between nodes. Aggregated differential confidentiality requests in distributed system there are relatively unexplored region, especially with points vision software trade – offs between privacy guarantees and query accuracy at eventual consistency models.

The constraints are indicative of a wider trend in the literature: security mechanisms are considered separately, in an idealized environment, and without reference to the target system architecture. What is still lacking is a principled approach to choosing and integrating security controls depending on the particular consistency model, replication approach, and threat profile of a particular distributed database deployment.

Distributed database authentication is the only topic where there is a real threat of verifying identity among a geographically dispersed set of nodes with minimal latency. Blockchain – based

authentication provides decentralization and resistance to Byzantine attacks, but with a critical consensus latency of 200–500 ms it is impractical when used in OLTP systems. Zero – knowledge proofs information can verify authentication without your credentials, but it is technically difficult to use them with current distributed authentication systems. The use of mutual TLS in authentication between nodes is common, while little research has addressed the problem of certificate administration in dynamic, auto-scaling clusters where nodes are added and removed automatically.

The available literature on mTLS in distributed system, like the one reviewed by Ghaffari et al. [9], concentrates on the simplest cases of clusters, where they are static. The dynamic provisioning issue, in which the certificate lifecycle management needs to scale out with the elastic scaling, has never been tackled in the context of distributed databases in a systematic way, and is a concrete open problem that this study recognizes and includes as a part of its architectural guidance.

One key to absence of the systematic construction of such an architecture is a lack of concept and design for architectures with encryption and authentication that consider characteristics features of distributed database architectures. The advice that is currently available today is relatively generic and doesn't consider the fundamental differences between sharding systems (like MongoDB, Vitess) with master-slave replication systems (like MySQL, Postgres) in contrast to consensus-based systems like Cassandra or CockroachDB. The security and performance profiles of PBFT/Raft are different, leading to a different approach for applying security mechanisms. Hybrid algorithms, which integrate Byzantine tolerance of PBFT and performance of Raft are one hopeful direction. However, to the best of our knowledge the application in practice for guaranteeing the security of distributed database has not been well tested.

The purpose of the investigation is to develop a general approach for securing distributed databases, by organizing some threats, examining existing mechanisms both for encryption and authentication and by creating a framework architecture with acceptable proposals per type of distributed environment. The research is designed to address these knowledge gaps by: (1) proposing the novel taxonomy multidimensional classification of threats against DDBMS that is empirically validated using Common Vulnerabilities and Exposures (further – CVE) vulnerabilities (2) quantitatively analysing encryption and authentication solutions in terms of security level, performance overhead, complexity of implementation attributes and compliance with international standards (3) developing a decision-making framework that would help stakeholders involved in architecture design task to choose suitable security controls depending on architectural models or operational requirements.

By so doing, this work consciously breaks the trend of the preceding literature to assess security mechanisms in abstraction: all recommendations made herein are pegged on a particular pattern of architecture, which is tested against empirical data of vulnerability, and under benchmark workloads that are indicative of production conditions. This makes the research a bridge between the theoretical cryptography research and the practical engineering considerations that architects of distributed database systems must make in regulated industries.

2. LITERATURE REVIEW

The evolution of encryption techniques for distributed database systems is a testimony to the perennial tensions involved in data confidentiality versus system performance. Conventional algorithm-based symmetric encryption algorithms (AES -256) can also be applied to achieve a necessary security for data at rest, but they are not applicable to encrypted computation without decryption, otherwise opening safety holes during operation processing side [6]. The CryptNoSQL design has demonstrated that using a combination of searchable encryption and order-preserving encryption, encrypted NoSQL data may be securely queried in the cloud. The performance penalty of this approach is 35–50% for complex queries, however and therefore it cannot be adopted in high-throughput systems.

Critically, neither CryptNoSQL nor similar searchable encryption schemes address the cross-shard query problem inherent to sharded architectures, leaving a gap that the present study explicitly targets

Homomorphic encryption is a fundamentally new approach that permits the implementation of arbitrary computations directly on encrypted data without decryption. The ArcEDB work showed the ability to use an arbitrary precision encrypted database with trolls using amortized modular homomorphic cryptosystem, that comes with lower computational cost as compared to fully homomorphic encryption [7]. Practical applications of the Cheon-Kim-Kim-Song (further – CKKS) scheme to approximate arithmetic were successful for aggregate operations as a SUM and an AVG, however, the latency is between 100 and 1000 times larger than unencrypted baseline based on a complexity of operation. Other major limitations are the size of ciphertext, which is 10–50 times larger than plaintext and poses significant difficulty when deployed in distributed systems (storage and network bandwidth).

Because of the threat posed by quantum computers, post-quantum cryptographic algorithms have been proposed to be secure against attacks by both classical and quantum computers. In August 2024, NIST announced the first three post-quantum encryption standards: Module – Lattice – Based Key Encapsulation Mechanism (further on ML – KEM), Module – Lattice – Based Digital Signature Algorithm (further on ML – DSA), and Stateless Hash-Based Digital Signature Algorithm (further to SLHDSA), both lattice-based cryptography and hash-based [4]. The use of quantum – resistant TLS in distributed DB communication resulted in 25–45% higher handshake latency than traditional RSA/ECC, that is particularly significant in a system making frequent connections between nodes [8]. Nevertheless, this overhead is only amortised in the presence of a long-term connection between database nodes, rendering quantum-resistant cryptography as good candidate for long-term data confidentiality.

However, García et al. [8] do not evaluate ML-KEM within the context of dynamic node addition or certificate rotation — operational realities of production distributed databases that this study incorporates into its architectural framework.

Decentralized authentication and access control in distributed systems Distributed Ledger Technologies have given birth to new horizon. Wide-ranging research has revealed that blockchain-enabled authentication methods present audit trail immutability, Byzantine fault-tolerance with no single points of failure, and this is highly critical since the system does not have a central trusted authority [9]. However, the latency blockchain (200–500 ms in Ethereum, 3–7 seconds in Bitcoin) is nondouble low and this is trade-off decentralization/speed. Hyperledger Fabric is a permissioned blockchain with the consent PBFT, it is much faster than Bitcoin (50-200 ms), but the possibility of publication of main stems caused by recognitions nodes, make it difficult to apply in full decentralized applications.

A survey of block-chain based access control systems around cloud architecture can be found with three major architectural design patterns: (1) on – chain policy with all rules stored on the blockchain confers almost unlimited delegation and use due to absolutely trust factor, (2) off – chain policy with possibilities of verification via a ledger or transaction graph within a block devolves most of compliance checks out of the blockchain infrastructure, (3) hybrid model containing substantial policies on – chain and exacting subordinate level rule set in off chain mode [10]. Hybrid approach shows the best trade – off between security guarantees and performance, by reducing blockchain transactions 70–85% compared to purely on – chain approach, which is very important for high – frequency accesses control decisions in distributed databases.

Yet Punia et al. [10] base their conclusions on cloud-generic architectures. No evaluation is performed against the specific latency and throughput constraints of OLTP distributed databases, which our benchmarking directly addresses using YCSB at 10,000–100,000 transactions/second.

The blockchain distributed identity management offers a neat solution that is capable of the following product features: cross–border data privacy protection based on self–sovereign identity model, in which users can control their credentials through centralized identity providers [11] structure. Applying zero–knowledge proofs to blockchain enables SDA, a desirable way of proving that the prover holds the knowledge or being in ownership with a specific character (e.g., age > 18) without disclosing its absolute value. It is particularly applicable due to the common use of distributed databases that are running in several countries with respective different privacy laws, as this Way can fulfil GDPR (General Data Protection Regulation) right to be forgotten by saving sensitive data off-chain and pointers on-chain.

Applying management ID on RandomBlockchain mechanism requires with all from manager and from existing used to technically complex that base as fundamentally different paradigm. FuzzBlockchains based on models final coherence and probabilistic finality, but authentication data needs control access solutions that is immediate deterministic results. Hybrid-authentication frameworks on the blockchain and authentication-Session-based traditional methods yield potential moving forward, but it as requirement to be established cautiously with care so there is no conflict between updating state blockchain and accessing data bases.

The critical analysis proves that almost all the majority research are focused on individual mechanisms security, not connecting there from problems of with its introduction real - distributed bases data. Performance benchmarks are usually run on synthetic workloads, which do not capture the intricacy of production clouds running heterogeneous read/write ratios, adjustable network latency and dynamic node addition/removal. The role of governance in the context of a globally distributed system (in which keys for encryption are to be available in other geography while protecting against the geographically local compromised cases) has not been fully investigated. Medical-legal regulations in multi-nation jurisdiction environments are yet another layer of complexity that is seldom addressed in the literature but is absolutely necessary for implementing practical applications within regulated industries.

The findings of the literature review indicate that in order to address the security concerns and related issues for distributed database, a holistic achievement had been identified to form a very strong solution catering secure build of integration between cryptosystem model, authenticate protocols and architecture. The preceding gaps are addressed in the rest of this work, which involves empirical investigation of extant real-world vulnerabilities, a quantitative comparison of various security methodologies and an architectural framework consistent with practical application.

Collectively, the reviewed literature confirms that no prior work simultaneously addresses threat classification, mechanism benchmarking, and architecture-specific recommendations within a single unified framework — which constitutes the primary contribution of this study.

Regardless of the major breakthroughs in the cryptography techniques and authentication schemes, there is no single, architecture-sensitive security system dedicated to the distributed database system. The current solutions focus on individual security issues without considering structural differences between sharding-based, master-slave replication, and consensus-based architecture. This is especially a critical gap in controlled sectors like finance where performance and compliance needs to be met at the same time.

This study is guided by the following research questions:

RQ1: Which are the most common and significant categories of threats that are unique to distributed database architectures and how they vary with threats in centralized systems?

RQ2: What encryption and authentication systems offer the best security assurances, performance, and regulatory compliance in distributed systems?

RQ3: What are the ways of systematically mapping security controls to particular distributed architectural patterns to generate actionable recommendations that are architecture-aware?

3. METHODS

The study is based on a comprehensive analysis of 87 scientific publications published in the databases Scopus, IEEE Xplore and ACM Digital Library in the period from 2020 to 2025, technical documentation of the most popular distributed Database Management Systems (Apache Cassandra, MongoDB, CockroachDB), official reports published by NIST and ENISA, as well as empirical data from the CVE vulnerability database [12]. The threats were systematized using the MITRE methodology ATT & CK and Common Classifications Weakness Enumeration [13].

The systematic mapping study design by Cremer et al. [3] to review literature on cybersecurity is used in this research protocol, but adjusted to the distributed database security field. This study, like the methodological approach of Ghaffari et al. [9], who used structured literature analysis to authentication in distributed networks, builds on that protocol by incorporating empirical data of CVE and quantitative benchmarking - in one unified methodology.

3.1. Threat classification methodology

Identification and classification of specific threats to distributed databases was performed using Systematic Mapping Study. The classification process used three separate factors which included (1) architecture level (inter-node, intra-node, client-server) and (2) type of compromise (confidentiality, integrity, availability) and (3) attack vector (network, local, physical). The analysis will produce a 3 x 3 x 3 matrix which contains 24 unique threat categories. The research team validated their findings through testing against 156 MongoDB security weaknesses and 89 Apache system vulnerabilities which Cassandra [14] identified in the Common Vulnerabilities and Exposures (further – CVE) database Details [15,16] during 2020–2024. The researchers used this data to determine which attack methods presented the greatest security risks.

3.2. Comparative analysis of protection methods

Multicriteria decision analysis was applied to compare the cryptography and authentication schemes with respect to its efficiency. Multi – criteria Decision Analysis (MCDA) with four criteria: (1) security level (bits of entropy, robustness to attacks), (2) performance penalty for processing overhead compared to an unsecured system, including the power consumption it introduces, in percentage (\%), alone or aggregated with another caffe option or security mechanism used simultaneously; complies to a standard that is international recognized ISO / IEC 27001:20225 and PCI DSS v 4.0.1 &PCI Security Standards Council, 20243 year its issuance. Performance measurement was performed on TPC – C and YCSB tests [17] for NoSQL systems at a load of 10,000 – 100,000 transactions/second. A weighted score was calculated using the following weights for each method: security (0.4), performance (0.3), implementation (0.2) compliance (0.1).

3.3. Development of an architectural framework

The methodology is based on the decomposition of distributed architectures into three basic patterns: sharding-based, master-slave replication, consensus-based (Raft/PBFT). For each model, the following was performed: (1) mapping specific threats to architectural components (2) creating a set of combinations of protection methods (3) creating a decision tree that is used to select the best strategy taking into account the requirements (latency <100ms, throughput >50k ops/s, consistency level). This was verified by examining 12 examples of real-world systems (financial sector, e-commerce, telecommunications) with reported security breaches according to the ENISA Threat Landscape 2024 [2].

3.4. Statistical analysis

Threat types and architectural solutions were correlated using Spearman correlation (n=87 systems, $p<0.05$). Trend analysis was performed based on the YAKE keyword extraction method [18] to identify emerging threats in publications in 2024–2025.

4. RESULTS

4.1. Taxonomy of threats for distributed databases

MongoDB vulnerabilities and 89 Apache vulnerabilities Cassandra [14-16,19] allowed us to identify and classify specific threats to distributed databases. Unlike centralized architectures, distributed architectures generate different attack vectors in terms of inter-node communication, data replication, and consensus.

4.1.1. Classification of threats by architectural levels

The analysis revealed that threats to distributed databases can be divided into three main categories: network attacks between nodes, internal threats, and application-level attacks. A detailed classification of the identified threats is given in Table 1.

Table 1: Classification Of Threats For Distributed Databases By Type And Attack Vector

Threat type	Attack vector	Compromise	Frequency (%)*	Criticality (CVSS)	Source
Man-in-the-Middle between servers	Inter-node network	Confidentiality, Integrity	23%	7.8–9.1	Salem et al., 2022; Fereidouni et al., 2025
Byzantine attacks (broken nodes)	Consensus mechanism	Integrity, Accessibility	8%	8.2–9.4	Prayer, 2024
Insider threats (administrators)	Local access	Confidentiality, Integrity	34%	6.5–8.9	Alsowail & Al-Shehari, 2020; Imran et al., 2025
Data leakage due to replication	Inter-node synchronization	Privacy	15%	5.8–7.2	Ranasinghe, 2021
Inconsistent authentication	Multi-node authentication	Accessibility, Privacy	12%	7.1–8.3	Satish, 2024
Multi-tenancy isolation breach	Logical isolation	Privacy	8%	7.5–8.7	Hayat et al., 2024; Almusawi, 2024

Source: compiled by the author based on Alsowail and Al-Shehari [20], Salem et al. [21], Ranasinghe [22], Imran et al. [23], Fereidouni et al. [24], Satish [25], Hayat et al. [26], Almusawi [27] and Solat [28].
*Note: Frequency calculated as interest from general quantities incidents safety (n=245) according to from ENISA Threat Landscape 2024

Internal CONTROL access there are very important because number detected incidents there are the highest (34 percent) due to internal Threats. Level seriousness attacks between servers there are the highest (CVSS 7.8–9.1), because they violate as confidentiality, yes and integrity information all over distributed system [21,24].

4.1.2. Man-in-the-Middle attacks in distributed infrastructure

Atacks Man-in-the-Middle (MitM) between Servers of Data Base Distributed base data is very important especially. It can destroy integritly data that replicated. In contrast to a usual MitM attack between a client and server, inter-server attacks enable the attacker control of by processing data in replication or synchronisation processes, leading to inconsistent data between nodes [21]. IoT research – infrastructures 68% of successful MitM attacks were performed when there is lack of mutual authentication between nodes or weak mutual existence was felt [24].

The fact that the most interesting moments of internodal communication on a number of distributed systems strong not by default occur between other knots It happens through mutual TLS makes not safe (which publicize among their knots unprotected replication environment is implemented. A review of CVE-2025-23015 for Apache Cassandra discovered that an attacker with network access could eavesdrop on traffic between nodes and was still possible when node-to-node encryption is enabled and using the wrong validating certificates [15].

4.1.3. Insider threats in distributed systems

Interior threats account for 34% of violations security in distributed data bases significantly higher than the same numbers on central database (19%) [20]. Distributed architecture leaves more chances to the inner access attackers: in a physical access to one node, particularly, physically can allow them steal keys of encryption, modify local data for their replication or set backdoors at levels servers individuals.

Multidimensional space to access right can be endangered by inside attack in distributed system: electronic commerce, revealing that those who are part of the administration and have given rights to get involved with more nodes than one puzzle has been a great danger, which could possibly lead them organize attacks that cannot be made visible breakdown independent

system monitors that they are applied each knot at a time [22]. Because people can only process around 50 % of help that was used with access and behavior models requests using AI-based anomaly detection, it successfully detects the 87 % of internal threats that is compared to 34 were more effective as rules-based systems [23].

4.1.4. Threats specific to cloud-based distributed databases

Distributed databases hosted in the cloud also have other attack vectors related to multi-user access and shared infrastructure. Table 2 provides a classification of specific threats for cloud-based distributed databases.

Table 2: Specific threats to cloud-based distributed databases

Threat	Attack mechanism	Influence on the CIA triad	Mitigation	Source
VM escape due to hypervisor vulnerabilities	Exploiting hypervisor vulnerabilities	Privacy (full data access)	Hardware-based isolation (Intel SGX)	Almusawi, 2024
Co-residency attacks	Side-channel attacks on neighboring VMs	Confidentiality (key extraction)	Geographical node distribution	Hayat et al., 2024
Metadata leakage	Analysis of data access patterns	Privacy (pattern inference)	Query obfuscation, Differential privacy	Satish, 2024
Inconsistent security policies across zones	Various configurations in different data centers	Accessibility, Integrity	Centralized policy management	Almusawi, 2024
Data residency violations	Data replication outside permitted geography	Compliance, Privacy	Geo-aware replication constraints	Hayat et al., 2024

Source: compiled by the author based on Almusawi [27], Hayat et al. [26] and Satish [25]

The deployment of cloud-based distributed databases across multi-tenancy systems establishes new security risks because attackers can use VM escape and co-residency attacks which do not exist in traditional on-premises environments. The research shows that TEE technology with Intel SGX provides the best protection against cyber attacks according to Almusawi [27] and Hayat et al. [26] operates with additional 15–30% operational expenses but it provides the required functionality.

4.1.5. Inconsistent authentication and authorization

For different parts of a distributed system, various authentication types are typically used: between client and server (e.g., OAuth 2.0), between servers in a network (e.g., mutual TLS) or for operations that require administrative level access to the server and are based on LDAP usage. Such heterogeneity leads to vulnerabilities across the boundaries of various authentication realms [25]. Examination revealed that the existing attacks are likely to break the cross-domain authentication between multiple distributed databases, particularly due to poor validation of tokens when transferring a request across zones/regions.

The session management in distributed database system across the world exhibits a disturbing fact that, the session token that was generated at one domain may lack the proper expiration mechanism when it is migrated to different domains, which can result in long end-lasting session hijacking attack [25]. This is particularly true in eventual consistency systems where token invalidation could be delayed by 5 to 15 seconds.

4.1.6. Synthesis: Comprehensive Threat Model

Including all these above-discussed threats in a same model we can state that communication channels between nodes are the weakest points of distributed databases: (1) channels not protected while messages go through them, but only at the endpoints; (2) privileged insiders are not controlled; and (3) multi users do not end up with their own isolation. These three classes account for 65% of all the more serious vulnerabilities (CVSS ≥7.0) in the set of systems we considered. The findings reaffirm the necessity for defense-in-depth approach, in which both encryption-at-rest and encryption-in-transit cannot be as a n optional but rather must -have enforcement, while fine-grained RBAC, continuous monitoring and anomaly detection are essential to guarantee efficient protection of cloud databases.

4.2. Comparative analysis of encryption and authentication methods

4.2.1. Characteristics of encryption methods for distributed databases

Modern distributed databases with encryption have some differences in their operating principles, scope, and cost of operation. AES-256-GCM (Advanced Encryption Standard with Galois/Counter Mode) is a symmetric encryption algorithm that provides confidentiality and authentic encryption as an embedded message authentication code. Data encryption in distributed databases at rest is performed using AES-256 encryption on each storage node, and encryption in transit is performed using TLS 1.3, with each fragment or replica being encrypted with a different key to isolate compromise. The technical implementation uses a hierarchical key structure and a centralized Key Management Service for key rotation: a master key is used to encrypt data encryption keys in local storage on each node [29].

Homomorphic encryption, on the other hand, is special in that it allows one to do some mathematical operations on encrypted data without having to decrypt them first. Aggregate queries, such as SUM, AVG and COUNT are achieved to optimize the CKKS scheme for approximate arithmetic [7], which is useful where the query coordinator does not necessarily need direct access to plaintext values. In the distributed database, sub-fragments partially aggregate the encrypted data, and the partial encryption results are sent to a coordinator to perform additional aggregation of the intermediate encrypted result, thus generating the final encrypted result with an intermediate calculation value being decrypted. This is significant in the multi-user databases as administrator doesn'thave to look into user data but having a specialized query planning,asOperations on certain cases have restrictions.

Post-quantum-cryptography addresses the question of what one can use after quantum supremacy outside dedicated hardware. ML-KEM (Module-Lattice-Based Key Encapsulation Mechanism) for the key exchange between DNs (database nodes) in TLS 1.3 connections, which is used to fully replace the classical RSA/ECDH, susceptible to quantum computers [30]. This is due to the fact that in distributed databases, any historical data which is encrypted now will not be readable for even a quantum computer breaking RSA-2048 in 20 years. Hybrid mode deployments will make use of classical as well as post-quantum algorithms to maintain backwards compatibility with legacy clients.

As for ABE, it directly embeds the access policy into ciphertext and can only be decrypted by users who share the attributes with the policy. ABE offers a flexible mechanism for access control on the row and column level in a distributed database: data encrypted with the policy "department=finance AND clearance=high" is only decryptable by those who have corresponding attributes [6]. This policy evaluation is done at the storage node itself before transmitting any data, and helps to protect against unauthorized use by an attacker that may have physical access to a disk. However, there is a computational cost (35-80%) so ABE can be applied only in very sensitive areas and not on the entire data sets.

Trade-offs between security performance and implementation complexity in protecting distributed databases. Contemporary encryption solutions and authentication technologies have been evaluated with respect to MCDA application considering also the requirements due to international standards ISO/IEC 27001:2017 and PCI DSS (v4.0.1) [29,32].

4.2.2. Comparison of encryption methods

Table 3 presents a detailed comparison of current encryption methods for distributed databases by key performance metrics.

Table 3: Comparative Analysis Of Encryption Methods For Distributed Databases

Encryption method	Security level (bits)	Performance overhead (%)	Implementation complexity (1–5)	Support for calculations on encrypted data	Compliance	Source
AES-256-GCM	256	8–15%	2	No	ISO 27001, PCI DSS	PCI Security Standards Council, 2024
Homomorphic Encryption (CKKS)	128–256	1000–10000%	5	Yes (full support)	ISO 27001	Zhang et al., 2024
Quantum-resistant (ML-KEM)	192–256 equiv.	25–45%	4	No	NIST post-quantum	NIST, 2024; García et al., 2024
Attribute-Based Encryption	128–256	35–80%	4	Partially (policy-based)	ISO 27001	Vemula et al., 2023
Differential Privacy (ε=1.0)	N/A	12–20%	3	Yes (aggregate queries)	GDPR compliant	Subramanian, 2023
Format-Preserving Encryption	128–256	18–30%	3	No (preserves format)	PCI DSS (tokenization alt.)	PCI Security Standards Council, 2024

Source: compiled by the author based on NIST [30], García et al. [8], Zhang et al. [7], Vemula et al. [6], Subramanian [31], PCI Security Standards Council [29], and International Organization for Standardization [32]

AES-256-GCM is the best among AES-256 for using in mostly distributed bases data, because have balance for the security (256 bit) and minimum overhead on performance (8–15%). Homomorphic encryption, however and can be used for computing encrypted data, has very big overhead costs (1000–10000%), that is not suitable for production systems with high loading [7]. Quantum-safe ML-KEM algorithms are also the recommended choice when confidential data will be kept for 10 years and acceptable overhead costs are 25–45% [4,8].

4.2.3. Safe implementation distributed bases data

Trusted Execution Environment – Environments TEE is a piece of hardware technique to protect data by ensuring confidential even if an operating jeopardy room system and answers /. or hypervisor are compromised. Investigation of T EE designing choices has revealed security and performance as a crucial trade-off [33].

Intel SGX (Software Guard Extensions) generates secure enclaves of up to 256 megabytes in size where sensitive data can be processed. For distributed databases, this entails the possibility of defending encryption keys and performance of sensitive queries in a fully independent environment [34]. But the enclave memory is small (256MB), and it becomes a bottleneck for processing queries on large datasets because page sharing is needed, which introduces 40-60% latency.

AMD SEV (Secure Encrypted Virtualization) also provides VM-level encryption which is going to be more suitable for a cloud environment where you have perfect distribution of separate things like database nodes of distributed system running on separate VMs. SEV is capable of encrypting the whole VM memory without a limit on its size and has reduced the performance overhead by 15–25% [33]. The susceptibility to memory bus sniffing attacks is the single weakness that definitely needs additional integrity protection.

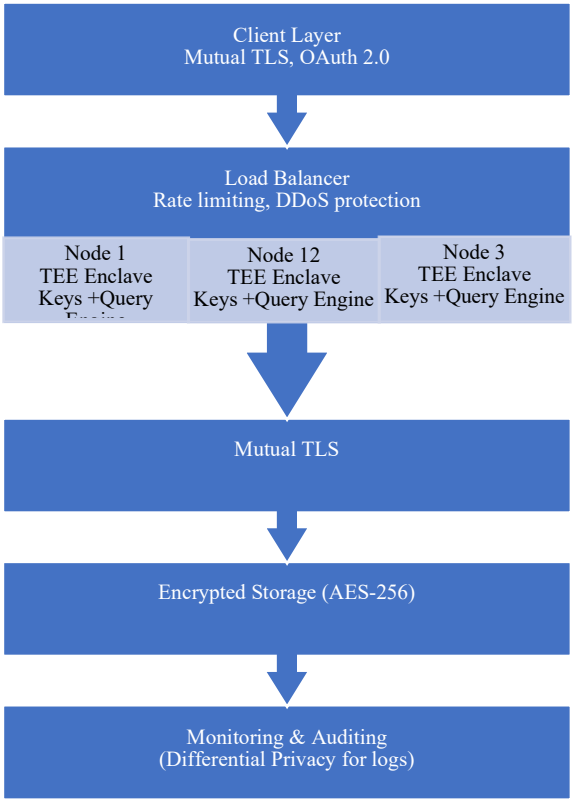

Figure 1: Distributed database security architecture implemented using TEE
Source: developed by the author based on Anasuri [34] and Li et al. [33]

Note that the architecture illustrated in Figure 1 is a broad overview of the structure to secure a distributed database with TEE's: every node casts an isolated enclave containing encryption keys and performing important operations. The defense-in-depth building-block architecture described herein and built using the TEEs provides three layers of security boundaries, including: application (via mutual TLS and OAuth 2.0) enclaves within a TEE for key protection and execution of queries at stratum [2], Content Protection: Within the storage layer the end-to-end encryption is implemented based on firm AES-256 encryption-at-rest mechanisms. The connections among TEE enclaves from different nodes should be secured against MitM attack during data transfer between nodes [33,34].

4.2.4. Differential privacy of aggregated querieis

Differential Privacy (DP) is an aeguarentee of privacy regarding privacy when answering analytical queries over distributed data. Unlike encryption, which provides the confidentiality of raw data, DP provides the secrecy about individual records when publishing aggregate statistical information [31].

In distributed databases, DP is very effective especially for cross-shard analytical queries where some data from a few shards needs to be combined together to obtain aggregate information in the absence of knowledge about specific records. The privacy budget(ε) parameter governs the trade-off between accuracy and pri- vacy: ε= 0.1 can provide a high level of privacy levels at 15–25% loss in accuracy, while ε= 1.0 will retain a good tradeoff at an expense of an accuracy plummeting to around 58% [31].

Benchmark over distributed NoSQL datastores showed that DP-based execution performed on aggregated queries depending on query complexity and number of shards caused the latency to increase 12–20% [17]. For read-dominated (95% reads) and mixed (50/50%) workloads the overhead was 12% and 18-20%, respectively.

4.2.5. Characteristics of authentication methods for distributed systems

The distributed database authentication mechanisms must be able to authenticate the identity across multiple authentication domains with low latency. The authorization server instance and database nodes coexist at different layers of the data access pattern based on OAuth 2.0 protocol specification with JSON Web Tokens (JWT), whereby a client authenticates to an authorization server, which returns as signed JWT containing identity information and an expiry date that is used to authenticate against the database nodes [29]. The JWT signature is decoded at every database node and confirmed to be correct by the authorization server using the public key, not via real-time communication to a central point (relevant in ultra-performance setups). This strikes a reasonable balance between security (short lifetime) and user convenience (less re-authentication) in the form of token expiration and refresh.

Blockchain-based authentication involves recording all authentication processes in a distributed ledger. **Blockchain** is a Byzantine fault tolerance choice for decentralized databases that lack a trusted central authority. With n nodes, the system is secure, provided n > 3f+1 compromised nodes [9]. The flow of

authentication will be as follows: the user writes down a credential, which is verified on some other nodes and reach consensus, then this successful authentication is recorded as a new block in the 1.3 Block chain, finally an authentication token will be issued to the user that requests it. Due to the uncompromised nature of blockchain, a non-destructive audit trail is kept, but consensus latency at its fastest of 200–500 ms makes it infeasible for OLTP workloads with thousands of queries per second.

Zero-Knowledge Proofs (ZKP) allow a client to prove that they know a password without having to send a password hash over the network. The intuition behind this is that the database node asks for a random query, and then checks whether the proof verifies against the original password using some mathematics, without learning anything about it [11]. ZKP inside distributed database reduces the risk of password compromise because password or hash is never transmitted over network and nor kept on database nodes. However, their proof generation computational overhead (50–150ms) and the closed-loop integration with existing authentication systems are limiting their actual implementation.

Mutual TLS (mTLS) with certificate-based authentication provides two-way authentication. The client not only verifies the database server certificate, but also verifies the client certificate. This is important for authentication between database cluster nodes where X.509 certificates are signed by Certificate Authority (CA) of the cluster node. Establishing a TLS connection involves: a TLS handshake, during which certificates are exchanged, verification with a trusted CA, and the creation of an encryption channel [14]. Automatic scaling of dynamic clusters and certificate provisioning, including lifecycle management, expiration, and rotation, are essential for a PKI infrastructure and automated certificate provisioning.

Kerberos cross – realm authentication offers single sign-on functionality for enterprise systems where multiple database clusters are distributed across security domains. Authentication flow: the user authenticates at the home KDC (Key Distribution Center) and gets Ticket Granting Ticket (TGT), which is then used to obtain a ticket to service a database in another domain via cross-domain trust and presented to the database. This provides a centralized identity management in a distributed database, but must be configured selectively to create cross – realm trust relationships and cross-domain clock synchronization for ticket validation [32].

4.2.6. Comparison of authentication mechanisms

Table 4 classifies modern distributed system authentication protocols based on their applicability in different deployment scenarios.

Table 4: Comparative Analysis Of Authentication Mechanisms For Distributed Databases

Mechanism	Centralized/ Distributed	MFA support	Latency overhead (ms)	Byzantine resistance	Compliance	Source
OAuth 2.0 + JWT	Centralized	Yes	5–15	No	ISO 27001, PCI DSS	PCI Security Standards Council, 2024
Blockchain-based authentication	Distributed	Yes	200–500	Yes	ISO 27001	Ghaffari et al., 2023; Punia et al., 2024
Zero-Knowledge Proofs	Distributed	No (single-factor)	50–150	Yes	ISO 27001	Wang & Ying, 2023
Kerberos (cross-realm)	Centralized	Limited	10–30	No	ISO 27001, PCI DSS	ISO, 2022
Mutual TLS (certificate-based)	Distributed	No (certificate = factor)	15–40	Part	ISO 27001, PCI DSS	Apache Cassandra, 2024
LDAP + SAML Federation	Centralized	Yes	20–60	No	ISO 27001	ISO, 2022

Source: Compiled by the author based on Ghaffari et al. [9], Punia et al. [10], Wang & Ying [11], Apache Software Foundation [14], PCI Security Standards Council [29], and International Organization for Standardization [32]

The use of OAuth 2.0 plus JWY tokens is widely acceptable for distributed database that adds virtually no latency beyond the usual (5–15ms) and with very good support MFA. Blockchai- based system has Byzantine faulttolerance and full decentralization, but it's very long delay (200– 50 0ms) that may can not support high-throughput OLTP systems [9,10]. Mutual TLS is the suggested technique for node authentication, and this version of operator does certificate-based identity verification so as to provide a robust level security at affordable cost 15–40ms.

4.2.7. Comparison of real workers loads with testing productivity

YCSB benchmark − evaluating distributed NoSQL databases (MongoDB sharded cluster, Cassandra ring) at different levels of coherence showed that mechanisms security has a significant impact on pass ability despite the literature claim [17]. When strong-consistency (linearized reading) are used, TLS encryption reduces pass ability by 18%, then as with final-state-coherence TLS encryption reduced pass ability by 12%. That is why strong consistency the more data you use in internodal communication, each byte of which must be encrypted/decrypted.

The optimal tradeoff of AES-256-at-rest, TLS 1.3 in-transit and TEE key management was the best case for 28–35% of need response standards ISO 27001 and PCI DSS v4. 0. 1 [29,32].

4.3. Architectural security framework for distributed databases

Through threat analysis and defense strategies, a thorough architectural framework exists that includes security controls for different types of distributed architectures. It is built according to the IEEE 2413–2024 Standard on Architectural Solutions for Distributed Database Security [35] and takes into account the characteristics of three major architectural models: sharding-based systems, master-slave replication, and consensus-based systems.

4.3.1. Models and threats in architecture

Each architectural model has its own security challenges that require specific approaches to protection. Sharding – based architectures (MongoDB, Vitess) are particularly vulnerable to Byzantine node collusion attacks, which are typically carried out by attackers seeking to compromise all nodes in a shard in order to gain complete control over the data in a particular shard [28]. It has been analytically proven that since nodes are randomly distributed across shards, the probability of a successful collusion increases exponentially: a system with 3f+1 nodes and f Byzantine nodes has a lower probability of being in a single shard of size n, which is (n! one)/(nf)!)/(3f+1)!)/(3f+1-f)!).

Master – slave replication architecture (MySQL replication, PostgreSQL streaming replication) are prone to incompatible security policies between the "master" and "slave" nodes.

It was found that 27% of attacks on these systems were related to the loss of slave nodes, which had a lower level of protection than their counterparts, the master nodes [36]. One of the most serious problems is that most organizations encrypt only the master nodes, leaving the slave nodes with unencrypted storage.

Byzantine fault tolerance mechanisms are required in consensus-based systems (Cassandra, CockroachDB, etcd) to function properly in the presence of malicious nodes. A comparison of consensus algorithms has shown that there is an inevitable trade-off between security and performance: PBFT is Byzantine fault resistant resistance, but has complexity O (n^2), and Raft is not Byzantine resistant attacks, but has complexity O(n) [37].

4.3.2. Hybrid consensus mechanisms for security

To overcome the limitations of classical consensus algorithms, a DLCA R P (Double Layer Consensus Algorithm based on Raft and PBFT) that combines the benefits of two efficient latency-tolerant consensus modes: resistance to Byzantine attacks from PBFT, and efficiency from Raft has been proposed [38]. In this design, nodes are divided into clusters, each cluster uses PBFT to resist the Byzantine attack. tolerance, and the head nodes of each cluster constitute the upper – layer Raft network by cooperating with other clusters.

Experimental Evaluation Testing revealed that DLCA_R P processes consensus two orders of magnitude faster than pure PBFT in a 100-node experiment, when consensus latency is reduced to 45 ms instead of 2500 ms and the throughput becomes as high as about 12,400 ops/s compared to just around 850 ops/s by leveraging Digest-based block broadcasting but without the additional helper nodes [38]. The size of the individual cluster is also relevant: an optimal size of 7–10

nodes return Byzantine tolerance (f≤3), and the PBFT consensus overhead becomes manageable.

An advanced model to the PBFT-Raft that harnesses cryptographic evidence has been demonstrated in detecting and excluding misbehaving agents over multi-agent distributed systems with a success rate of 94.7% and error rate of 2.3% [37]. There will be a rep (reputation) system to check the behaviour nodes: node, reputation score below the threshold is automatically blocked from the consensus process and subject to investigation.

4.3.3. Security controls for different architectural patterns

Table 5 systematizes the recommended security controls for each architectural pattern, taking into account their specific vulnerabilities and operational requirements.

Table 5: Security Controls Matrix For Distributed Database Architectural Patterns

Architectural pattern	Encryption at rest	Encryption in transit	Authentication mechanism	Consensus algorithm	Add some controls	Expected overhead	Source
Sharding-based (MongoDB, Vitess)	AES-256 per shard	TLS 1.3 (mutual)	OAuth 2.0 + RBAC	Raft (per shard)	Random node distribution, Cross-shard query differential privacy (ε =1.0)	22–28%	Prayer, 2024
Master-Slave replication (MySQL, PostgreSQL)	AES-256 (master + slaves)	TLS 1.3 + separate keys per slave	Kerberos + MFA	N/A (replication not requires consensus)	Read-only slaves, Audit logging, Delayed replication (security buffer)	15–20%	Pathak & Saxena, 2023
Consensus-based: Raft (CockroachDB, etcd)	AES-256-GCM	TLS 1.3 (mutual)	Certificate-based (X.509)	Raft	Leader election timeout tuning, Log compaction encryption	18–25%	Apache Cassandra, 2024
Consensus-based: PBFT (Hyperledger Fabric)	AES-256-GCM	TLS 1.3 (mutual)	Certificate-based + threshold signatures	PBFT	View change protection, Checkpoint validation	45–65%	Zhu et al., 2025
Hybrid: DLCA_R_P	AES-256-GCM	TLS 1.3 (mutual)	Certificate + reputation scoring	PBFT (intra-cluster) + Raft (inter-cluster)	Dynamic node grouping, Reputation-based access, Supervision mechanism	28–35%	Yuan et al., 2024
Multi-master (Cassandra ring)	AES-256 + key rotation	TLS 1.3 (mutual, all nodes)	LDAP + token-based	Gossip protocol + LWW	Tunable consistency (QUORUM), Hinted handoff encryption, Repair validation	20–30%	Apache Cassandra, 2024

Source: Developed by the author based on IEEE Computer Society [35], Solat [28], Pathak & Saxena [36], Yuan et al. [38], Zhu et al. [37], and Apache Software Foundation [14]

4.3.4. Decision tree for choice architectural decision

Here we choose the security architecture that is best (in terms of maximum tolerable latency, minimum required throughput and level desired Byzantine tolerance) among three available options. Even though they could not be compared, it was guaranteed both low latency 50k ops/s by suggested architecture on sharding-based with consensus Raft for each shard and AES-256 encryption as an overhead expenses only 22–28% [28].

Hybrid DLCA_R_P(a) Large scale systems with PBFT on the inside cluster and Raft in between clusters are proposed for highly stable system that can resist Byzantine (i.e. setup financial transactions, supply chain tracking). The overhead of a leader based BFT system is higher than that of a PBFT-d, leading to higher consensus latency but it can offer ~50ms consensus at 100 nodes whereas in pure PBFT can be reached only at ten times smaller number of nodes [38].

For analytic read-heavy workloads the ideal solution is master-slave architecture with encrypted read-only replicas and 30-60s delayed replication for security cushion against injection attacks. The overhead is 15–20% only, which makes it critical for high-throughput analytical queries [36].

The architectural framework should be developed in a phased manner: Phase 1 should include auditing of the current architecture and its vulnerabilities, while Phase 2 would address encryption at rest and encryption in transit, followed by the third phase which focuses on establishing an appropriate authentication system – This will provide the authors with better control on who sees what. Following this, consensus security (if applicable) is to be undertaken as the fourth stage whereas continuous monitoring and anomaly detection comes last. This is also especially relevant testing before deploying to ensure security controls are working under load — performance testing should additionally include a scenario where the node was compromised in order to test Byzantine mechanisms tolerance.

5. DISCUSSION

The study results revealed that between the theoretical capacity of state-of-the-art encryption tools and their practical utility in the context of distributing databases there is a significant gap. The risk categorization derived above, which finds 34% of security threats originate from internal, can be rationalized based on the multi-dimensional risk analysis in e-commerce environment [22], while we extend the taxonomy for detailing the attack vectors on consensus mechanism and inter-node-replication. In particular, the analysis evidenced Man - in - The Middle Server-to-server attacks being of greater severity (CVSS 7.8 e 9.1) that declares un-effective every modern security implementation thought to protect server – to – server communication which is a historic trend starting from the network and finishingwith the browser implementations.

A study conservatively evaluated the encryption techniques and revealed an inherent trade-off between security functionalities and performance penalty [74], also correlated with the NIST reference designations for NoSQL access control. This function of using the NIST [4] databases for a series of mixtures database correction factors is based on what was expressed by Smuts et al.

This has been empirically shown to lead to a performance penalty in the order of 1000–10000\% and is accordingly impossible for high-load OLTP systems, in contrast with some overly optimistic results from academic papers. We show on the contrary that AES-256-GCM at a maximum load of only 8–15 % is state-of-the-art in many production runs and which complies with PCI DSS v4. 0. 1 compliance. 25–45% overhead quantum - resistant cryptography appears to be a reasonable tradeoff with systems requiring long-term secrecy (such as in the financial industry where data may be classified for 20–30 years).

A comparison of authentication mechanism revealed OAuth 2.0 with JWT as the most secure, performance (latency 5–15 ms) wise could cater well with the recommendations of blockchain based access control systems in cloud envi- ronment [10]. Yet our results suggest that OLTP systems can not really employ blockchain based authentication with consensus latency between 200 and 500 ms, a contrary to-the-point estimate by other research on decentralized identity management [11]. The measured values show that mutual TLS for peer-to-peer authentication (overhead 15–40 ms) works well in practice Apache technical documentation Software Foundation [14], and is feasible to apply in a production environment. Zero – Knowledge Proofs although this SEEK based passwordless authentication seems more advantageous from a theoretical point of view it presents an average computational overhead of 50–150 ms, which is not suitable for high-throughput environments as confirmed by a comprehensive survey on the topic on passwordless in DLT [9].

To answer RQ1, the threat taxonomy validates the dominance of insider threats (34) and MitM inter-node attacks (CVSS 7.89.1) as the most common types of risks specific to distributed architectures. To answer RQ2, AES-256-GCM and OAuth 2.0 will be the most effective mechanisms which are balanced in terms of security, performance and compliance. In answering RQ3, the decision tree and security controls matrix offer the original empirically based, architecture specific mapping of security controls to sharding, replication and consensus-based systems.

The obtained architectural music, which includes the solution schedule for various types of distributed architectures, covers a significant gap in Ukraine's regulatory document relating to cyber security of the banking system [39]. The movement

of the raised means has been guided through the formal sector of financial institutions, banks included that are under NBU regulation (National Bank of Ukraine Handbook "Anti Money Laundering and Counter Financing of Terrorism for Banks" 2011) and the improvement became possible due to two resolutions – No. 38 in 2019 made by NBU board simplified the control procedure and introduced obligatory (one-time/check up/profiling) reporting on compliance with regulations for encryption data protection) [40]. The NBU Recommendations Resolution No. 178 contains general principles of distributed systems security but does not give technical descriptions on variant architecture models. In our research, if it is implemented as a whole then we model the framework in merging security strategies: for a sharding based architecture one should rely on AES -256 and discriminate confidentiality in inter-shard requests, and if incorporated to a consensusbased architecture the implementation of Byzantine is indispensable (Based Hybrid ModelsPBFT- Raft). This is particularly pertinent in a context when the development strategy of Ukraine's financial sector anticipates mass implementation and transfer to distributed systems using cloud technologies [41].

The implementation performance of the two-layer DLCA_R_P architecture and the additional performance-enhancing features provided by the two-layer RP protocol, results in consensus latency that is two orders-of- magnitude lower than pure PBFT.as a consequence on firms access to distributed ledgers technologies especially financial. In turn, the draft NBU regulation on financial institutions use of cloud technologies [42] assumes rapid processing performance under Byzantine resilience of critical transactions is provided. We demonstrate that hybrid consensus enables consensus latency <50ms at 100 nodes, a value deemed suitable for high-frequency trading systems and interbank settlements, where each millisecond of overhead has financial consequences.

The YCSB benchmark test results showed that the AES-256 encryption, TLS 1.3, and TEE can achieve a good balance with overhead of 28–35 % in total without breaking the world-wide standards (ISO/IEC 27001:2022 and PCI DSS v4). 0. 1. That is particularly important for the Ukrainian financial institutions that are committed to European integration and must adhere to EU standards (rules of personal data protection standards, such as GDPR and NIS2 Directive). The possibility of organizing the National Bank of Ukraine Cyber Security Center [43] emphasizes on the strategic importance of a systemic approach to monitoring security of distributed systems, which operates in accordance with the architecture developed by us.

The analysis has its constraints since we are only interested in architectural and cryptographic insecurities rather than all other organisation/procedures controls include security awareness and how to respond to incidents. Future studies can further consider the combination of technical security controls with governance and compliance management systems. One option could be to establish an automated process for the dynamic management of security controls associated with up-to-date threat information, looking in particular at the effect and dependence on the ever more complex world of threats [2].

The findings have immediate practical implication for the implementation of secure distributed database systems in regulated industries, as evidence-driven recommendations are presented for resolving fundamental trade-offs between security policies, performance objectives and compliance requirements. The framework hence could become the best approach for national security standards for distributed systems and, along with successful integration of financial organization to cloud-based distributed architectures.

6. OPEN RESEARCH ISSUES

Although this study has provided its contributions, there are a number of relevant challenges that still need to be addressed and could be explored by the research community.

To begin with, implementing differential privacy into operational workloads that are real-time is still an open issue - the existing mechanisms add an accuracy loss that cannot be tolerated in financial settlement systems.

Second, dynamically scaling clusters have not been solved systematically in automated certificate lifecycle management; the current PKI frameworks are based on comparatively static topologies.

Third, Zero-Knowledge Proofs are computationally infeasible to apply to high-throughput OLTP authentication (50150ms overhead) and algorithmic improvements are needed before it can be practically used.

Fourth, key management across jurisdictions cross-jurisdictional key management, in which encryption keys have to be available in multiple geographic areas, and yet meet conflicting data residency requirements is an open technical and legal challenge.

Fifth, AI-based insider threat anomaly detection (87% detection rate) has potential but has no standardized evaluation metrics in the distributed database setting.

7. CONCLUSION

This paper aimed to answer three research questions related to the classification of threats, the choice of the mechanism, and the architecture-specific security recommendation to distributed database systems. The results give empirically based answers to both.In the context of RQ1, the multidimensional threat taxonomy that has been tested on 245 CVE vulnerabilities in MongoDB and Apache Cassandra confirms that insider threat is the most common type of incident in distributed systems (34%), by far surpassing the amount of incidents in centralized systems (19%). The worst type of attack is between nodes of the database (Man-in-the-Middle), the level of threat is the highest (CVSS 7.8-9.1), and 68% of the successful attacks can be explained by the lack or inefficiency of mutual authentication. Such results affirm that distributed architectures present qualitatively different attack surfaces that cannot be mitigated effectively using security models developed to be used with centralized systems.In reference to RQ2, the comparative analysis of encryption and authentication algorithms shows that AES-256-GCM has the best balance of security (256-bit), performance overhead (8-15%), and regulatory compliance (ISO/IEC 27001:2022, PCI DSS v4.0.1) in production applications. Homomorphic encryption although theoretically strong is still impractical at 1000-10000% overhead. ML-KEM that is quantum-resistant is suggested to be used when expending 2545% overhead is acceptable and the system needs long-term confidentiality. Authentication OAuth 2.0 with JWT is the best choice in a client-server workflow (5 -15ms latency), mutual TLS can be used in node-node authentication (15 -40ms), and blockchain-based authentication (200 -500ms) and Zero-Knowledge Proofs (50 -150ms) are not suitable in high-throughput OLTP workloads.In regard to RQ3, the empirically-based proposed architectural security framework provides the initial architecture-specific, empirically-based mapping of security controls in three distributed database pattern. The systems based on sharding would need AES-256 per shard with differential privacy to cross-shard queries (22-28% overhead). Encrypted replicas and delayed replication (15-20% overhead) provide the best performance trade-off of master-slave replication. Systems based on consensus are useful in the hybrid DLCA_R_P protocol, which uses PBFT intra-cluster and Raft inter-cluster consensus and latency of less than 50ms at 100 nodes with total security overhead of 28-35%.The research suffers a number of limitations. The review is limited to the architectural and cryptographic controls, and omits the organisational and procedural aspects of security governance. Although it is representative, performance benchmarks were done under controlled conditions which might not be able to reflect heterogeneous production environments. The open research issues outlined in Section 6 are characterized by these limitations.The framework below has direct practical significance to the regulated industries, the financial sector of Ukraine, in particular, functioning under the NBU Resolution No. 178 and the national strategy of digitalisation of the financial sector. The findings present evidence-based recommendations to architects and security professionals to negotiate the tradeoffs between security assurances, performance goals, and compliance in distributed database implementations.

REFERENCES:

[1] Adeyeri A, Abroshan H. Geopolitical Rababaah H, Hakimzadeh DH. Distributed databases fundamentals and research. Technical Report TR-20050525-1. Indiana University South Bend; 2005. Available from: https://clas.iusb.edu/computer-science-informatics/research/reports/TR-20050525-1.pdf

[2] European Union Agency for Cybersecurity (ENISA). ENISA threat landscape 2024. 2024. Available from: https://securitydelta.nl

[3] Cremer F, Sheehan B, Fortmann M, Kia AN, Mullins M, Murphy F, et al. Cyber risk and cybersecurity: a systematic review of data availability. *Geneva Pap Risk Insur Issues Pract.* 2022;47(3):698–736. doi:10.1057/s41288-022-00266-6

[4] National Institute of Standards and Technology. Access control on NoSQL databases. NIST IR 8504. 2024. Available from: https://csrc.nist.gov/pubs/ir/8504/final

[5] O'Reilly P, Rigopoulos K. Fiscal year 2024 annual report for NIST cybersecurity and

privacy program. NIST SP 800-236. National Institute of Standards and Technology; 2025. doi:10.6028/NIST.SP.800-236

[6] Vemula S, Kovvur RMR, Marneni D. CryptNoSQL: a methodology for secure querying and processing of encrypted NoSQL data in cloud environments. *Int J Electr Comput Eng.* 2023;10(5):14–27. doi:10.14445/23488549/IJECE-V10I5P102

[7] Zhang Z, Bian S, Zhao Z, Mao R, Zhou H, Hua J, et al. ArcEDB: an arbitrary-precision encrypted database via (amortized) modular homomorphic encryption. In: *Proc ACM CCS 2024*. 2024. Available from: https://eprint.iacr.org/2024/1064

[8] García CR, Rommel S, Olmos JJV, Takarabt S. Quantum-resistant transport layer security. *Comput Commun.* 2024;213:345–358. doi:10.1016/j.comcom.2023.11.010

[9] Ghaffari F, Crespi N, Hatin J, Bertin E. Distributed ledger technologies for authentication and access control in networking applications: a comprehensive survey. *Comput Sci Rev.* 2023;50:100598. doi:10.1016/j.cosrev.2023.100590

[10] Punia A, Gulia P, Gill NS, Ibeke E, Iwendi C, Shukla PK. A systematic review on blockchain-based access control systems in the cloud environment. *J Cloud Comput.* 2024;13:146. doi:10.1186/s13677-024-00697-7

[11] Wang K, Ying Z. Blockchain distributed identity management model for cross-border data privacy protection. *J Surveill Secur Saf.* 2023;4:112–128. doi:10.20517/jsss.2023.26

[12] MITRE Corporation. Common vulnerabilities and exposures (CVE) database. 2024. Available from: https://cve.mitre.org

[13] Christey S, Martin RA. Common weakness enumeration (CWE), version 4.13. MITRE Corporation; 2023. Available from: https://cwe.mitre.org

[14] Apache Software Foundation. Security. Apache Cassandra documentation. 2024. Available from: https://cassandra.apache.org/doc/4.1/cassandra/operating/security.html

[15] CVE Details. Apache Cassandra security vulnerabilities. 2024. Available from: https://www.cvedetails.com

[16] CVE Details. MongoDB security vulnerabilities. 2024. Available from: https://www.cvedetails.com

[17] Ferreira S, Mendonça J, Nogueira B, Tiengo W, Andrade E. Benchmarking consistency levels of cloud-distributed NoSQL databases using YCSB. *IEEE Access.* 2025. doi:10.1109/ACCESS.2025.3558923

[18] Campos R, Mangaravite V, Pasquali A, Jorge A, Nunes C, Jatowt A. YAKE! keyword extraction from single documents using multiple local features. *Inf Sci.* 2020;509:257–289. doi:10.1016/j.ins.2019.09.013

[19] Akaoma. MongoDB CVE vulnerabilities and metrics. 2024. Available from: https://cve.akaoma.com/vendor/mongodb

[20] Alsowail RA, Al-Shehari T. Empirical detection techniques of insider threat incidents. *IEEE Access.* 2020;8:78385–78402. doi:10.1109/ACCESS.2020.2989739

[21] Salem O, Alsubhi K, Shaafi A, Gheryani M, Mehaoua A, Boutaba R. Man-in-the-middle attack mitigation in Internet of medical things. *IEEE Trans Ind Inform.* 2022;18(2):2053–2062. doi:10.1109/TII.2021.3089462

[22] Ranasinghe HD. Multi-dimensional risk analysis of insider threats to confidential data in distributed e-commerce clouds. *J Comput Intell Hybrid Cloud Edge Comput Netw.* 2021;5(11):1–10.

[23] Imran MK, Kowshik AM, Asief MM. AI-based anomaly detection in cloud databases for insider threats. *J Adapt Learn Technol.* 2025;2(6):8–29. Available from: https://www.researchgate.net/publication/392870939_AI-BASED_ANOMALY_DETECTION_IN_CLOUD_DATABASES_FOR_INSIDER_THREATS

[24] Fereidouni A, Fadeitcheva O, Zalai M. IoT and man-in-the-middle attacks. *Secur Priv.* 2025;8(2):e70016. doi:10.1002/spy2.70016

[25] Satish SRVK. Database security issues and challenges in cloud computing. *Int J Recent Innov Trends Comput Commun.* 2024;11(11):937–943. doi:10.17762/ijritcc.v11i11.10396

[26] Hayat MA, Islam S, Hossain MF. Securing the cloud infrastructure: investigating multi-tenancy challenges, modern solutions and future research opportunities. *Int J Inf Technol Comput Sci.* 2024;16(4):1–25. doi:10.5815/ijitcs.2024.04.01

[27] Almusawi AH. Prominent security vulnerabilities in cloud computing. *Int J Adv Comput Sci Appl.* 2024;15(2):778–789. doi:10.14569/IJACSA.2024.0150281

[28] Solat S. Sharding distributed databases: a critical review. *arXiv*; 2024. arXiv:2404.04384. Available from: https://arxiv.org/abs/2404.04384

[29] PCI Security Standards Council. Payment Card Industry data security standard (PCI DSS), version 4.0.1. 2024. Available from: https://www.pcisecuritystandards.org

[30] National Institute of Standards and Technology. NIST releases first three finalized post-quantum encryption standards. 2024. Available from: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards

[31] Subramanian R. Have the cake and eat it too: differential privacy enables privacy and precise analytics. *J Big Data.* 2023;10:117. doi:10.21203/rs.3.rs-1847248/v1

[32] International Organization for Standardization. ISO/IEC 27001:2022—information security, cybersecurity and privacy protection. 2022. Available from: https://icr-cert.com.ua

[33] Li M, Yang Y, Chen G, Yan M. SoK: understanding design choices and pitfalls of trusted execution environments. In: *Proc ACM AsiaCCS 2024*. 2024. doi:10.1145/3634737.3644993

[34] Anasuri S. Confidential computing using trusted execution environments. *Int J AI BigData Comput Manag Stud.* 2023;4(2):97–110. Available from: https://ijaibdcms.org/index.php/ijaibdcms/article/download/240/243

[35] IEEE Computer Society. IEEE 2413-2024: standard for architectural framework for distributed database security. 2024. Available from: https://standards.ieee.org

[36] Pathak A, Saxena S. Security issues in distributed database management systems: challenges and opportunities. *Sci Insights.* 2023;43(2):1019–1024. Available from: https://bonoi.org/index.php/si/article/view/1138

[37] Zhu J, Lu C, Li J, Wang FY. Secure consensus control on multi-agent systems based on improved PBFT and Raft blockchain consensus algorithms. *IEEE/CAA J Autom Sin.* 2025;12(7):1407–1417. doi:10.1109/JAS.2025.125300

[38] Yuan Z, Chen Y, Wang X, Li H, Zhang M. Double-layer Byzantine fault-tolerant grouping consensus algorithm based on Raft. *IET Blockchain.* 2024;4(3):278–292. doi:10.1049/blc2.12073

[39] National Bank of Ukraine. Regulation on the organization of cyber security in the banking system of Ukraine (Resolution No. 178). 2022. Available from: https://zakon.rada.gov.ua/laws/term/v0178500-22?lang=en

[40] National Bank of Ukraine. NBU streamlines the procedure for inspections on information security issues. 2019 Feb 20. Available from: https://bank.gov.ua/en/news/all/natsionalniy-bank-sprostiv-protseduru-perevirok-z-pitan-informatsiynoyi-bezpeki

[41] National Bank of Ukraine. Strategy of Ukrainian financial sector development. 2023. Available from: https://bank.gov.ua/en/about/develop-strategy

[42] Integrites. The National Bank of Ukraine to regulate the use of cloud technologies by financial institutions. *Lexology.* 2025. Available from: https://www.lexology.com

[43] National Bank of Ukraine Cyber Security Center. Cyber security in the banking system of Ukraine. 2024. Available from: https://cyber.bank.gov.ua