Recent Advances in Privacy-Preserving Query Processing Techniques for Encrypted Relational Databases in Cloud Infrastructure

Onuh Matthew Ijiga[1], Nonso Okika [2], Semirat Abidemi Balogun[3], Ogboji James Agbo[4], Lawrence Anebi Enyejo[5]

[1] Departmant of Physcis Joseph Sarwan Tarka University, Makurdi, Benue State, Nigeria.

[2]Network Planning Analyst, University of Michigan, USA.

[3]Department of Information Science, North Carolina Central University, Durham North Carolina, USA.

[4]School of Engineering and the Built Environment, Birmingham City University, United Kingdom.

[5]Department of Telecommunications, Enforcement Ancillary and Maintenance, National Broadcasting Commission Headquarters, Aso-Villa, Abuja, Nigeria.

DOI: https://doi.org/10.5281/zenodo.15834617

Published Date: 08-July-2025

Abstract: **The growing reliance on cloud infrastructure for storing and managing relational databases has introduced critical challenges in preserving data confidentiality while enabling efficient query execution. Traditional encryption schemes offer strong data protection but often impede the ability to perform complex SQL operations without decryption, creating a trade-off between security and functionality. Recent advances in privacy-preserving query processing techniques—including homomorphic encryption, searchable encryption, oblivious RAM, and secure multiparty computation—have revolutionized the field by enabling secure computation over encrypted data with minimal performance penalties. This review paper systematically analyzes the state-of-the-art mechanisms that support encrypted SQL query processing in cloud-hosted relational databases. It evaluates the theoretical foundations, computational overheads, query expressiveness, and practical deployment scenarios of these privacy-preserving methods. Furthermore, it explores hybrid approaches that combine cryptographic and hardware-based techniques to balance performance with security guarantees. The paper also highlights emerging trends such as federated SQL processing, data provenance tracking, and secure hardware enclaves that are reshaping privacy-preserving architectures. The study aims to provide a comprehensive understanding of the current capabilities, limitations, and future research directions in secure query execution for encrypted relational databases deployed in cloud environments.**

Keywords: **Encrypted SQL Query Processing, Homomorphic Encryption, Secure Cloud Databases, Privacy-Preserving Computation, Searchable Encryption, Cloud Data Confidentiality.**

1. INTRODUCTION

1.1 Overview of Cloud-Based Relational Database Adoption

The proliferation of cloud infrastructure has significantly transformed how organizations manage and scale relational databases. Enterprises increasingly adopt cloud-native relational database management systems (RDBMS) due to their elasticity, cost efficiency, and ability to support distributed applications at scale (Arora et al., 2021). Modern cloud-based RDBMS solutions—such as Amazon Aurora, Google Cloud SQL, and Azure Database—enable organizations to offload data management overhead while benefiting from robust high-availability and backup mechanisms (Singh & Sharma, 2022).

As these services mature, cloud providers continue to integrate advanced analytics, auto-scaling features, and database-as-a-service (DBaaS) capabilities, making them viable for both transactional (OLTP) and analytical (OLAP) workloads. However, this shift is not merely technological. It reflects a broader reconfiguration of enterprise architecture and data governance models. Enterprises that once relied on on-premises SQL servers now face new challenges related to remote access control, data synchronization, and platform interoperability (Sun et al., 2023). Furthermore, the demand for scalable and low-latency access to relational data in cloud-native applications has fueled the adoption of hybrid architectures that span public clouds, private clouds, and edge deployments (Zhang et al., 2020). While these trends provide unparalleled scalability and availability, they also introduce vulnerabilities around data visibility, especially during query execution and inter-node communication. Hence, privacy-preserving mechanisms must be tightly integrated with cloud-based relational systems from the outset to ensure security does not become an afterthought in the push for digital agility.

1.2 The Necessity of Query Privacy and Regulatory Compliance

As organizations migrate sensitive workloads to cloud environments, ensuring query privacy has become paramount. Cloud-based SQL queries often traverse virtualized environments, potentially exposing metadata, query structures, and result sets to unauthorized parties (Zhang et al., 2020). This exposure contravenes privacy regulations such as the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), and the California Consumer Privacy Act (CCPA), all of which mandate stringent controls over personal and transactional data (Han et al., 2022). Ensuring that queries over encrypted databases do not compromise user confidentiality is, therefore, not only a technical requirement but a legal imperative. Recent research underscores the necessity of integrating privacy-preserving mechanisms directly into SQL query execution engines. Techniques such as differential privacy, encrypted indexing, and query obfuscation have been proposed to address this concern, enabling secure analytics without data exposure (Liu et al., 2021). Additionally, privacy-enhancing computation aligns with principles of privacy-by-design, which advocate for security controls at the query planning and execution layers. This approach ensures minimal data leakage while supporting complex analytical workflows. However, regulatory compliance extends beyond algorithmic safeguards; organizations must also establish transparent data handling policies, access control audits, and verifiable data processing pipelines (Li et al., 2020). Given the growing regulatory emphasis on data sovereignty and cross-border transfer limitations, cloud service providers must also offer region-specific data residency and encryption options. These requirements have spurred collaborative frameworks between technical and legal stakeholders, creating a multidisciplinary agenda that governs how SQL queries can be securely processed in regulated industries such as finance, healthcare, and government (Zhang et al., 2020). Ultimately, query privacy and compliance must co-evolve to support secure and lawful data analytics in the cloud.

1.3 Technical and Operational Challenges in Secure Query Execution

Securing query execution over encrypted relational databases in the cloud presents multifaceted challenges that span computation, storage, and network layers. One primary issue is the computational overhead introduced by encryption schemes such as homomorphic encryption, which enables computations over ciphertexts but often at the cost of high latency and limited query expressiveness (Li et al., 2020). While partially homomorphic and leveled encryption schemes have improved efficiency, real-time execution of complex SQL queries—especially joins, aggregations, and subqueries—remains an ongoing hurdle (Yin et al., 2021). Access pattern leakage is another critical concern. Even when data is encrypted, observable query access patterns can allow adversaries to infer sensitive information. Oblivious RAM (ORAM) and access pattern obfuscation techniques have been proposed to counteract this threat, though they introduce significant performance trade-offs (Wu & Zhang, 2023). Additionally, maintaining indexing structures over encrypted data introduces complexity in update operations and hinders the performance of range queries and wildcard searches (Wang et al., 2022). This limitation impairs the efficiency of interactive analytics and dynamic schema evolution in real-world applications. Operationally, cloud-native environments compound these technical issues through dynamic scaling, multi-tenancy, and geo-distribution. Ensuring data consistency and integrity across encrypted shards distributed globally, while maintaining auditability and fault tolerance, presents significant engineering challenges (Yin et al., 2021). Furthermore, encrypted query processing must integrate with cloud orchestration tools, database proxies, and role-based access control systems without introducing single points of failure or bottlenecks. These complexities underscore the need for hybrid solutions that combine lightweight cryptographic methods with trusted execution environments (TEEs) and hardware acceleration.

1.4 Scope, Methodology, and Contributions of the Review

This review aims to provide a comprehensive synthesis of recent advancements in privacy-preserving query processing techniques specifically tailored for encrypted relational databases operating within cloud infrastructure. The scope is deliberately confined to relational database management systems (RDBMS) that support SQL-based queries and are deployed in public, private, or hybrid cloud environments. Key focus areas include cryptographic techniques such as homomorphic encryption, searchable encryption, oblivious RAM, and secure multiparty computation, as well as architectural considerations such as trusted execution environments and federated cloud deployments. The review excludes non-relational (NoSQL) systems and offline data masking techniques, thereby maintaining a focused lens on query-time privacy mechanisms for structured data. The methodology employed in this review follows a structured literature analysis approach. First, a keyword-driven search was conducted across Google Scholar, Scopus, IEEE Xplore, ACM Digital Library, and ScienceDirect using terms such as *"encrypted SQL"*, *"privacy-preserving query"*, *"cloud RDBMS security"*, and *"homomorphic encryption for relational databases"*. Only peer-reviewed articles published from 2020 onward were selected to ensure relevance and to capture recent breakthroughs in post-quantum cryptography, cloud-native database architectures, and regulatory-aware design patterns. Each paper was evaluated based on criteria such as technical novelty, computational efficiency, compatibility with standard SQL semantics, and practical deployment considerations. The contributions of this review are threefold. First, it categorizes and critically evaluates state-of-the-art techniques based on their underlying cryptographic principles, query capabilities, and performance trade-offs. Second, it synthesizes comparative insights into how these techniques perform under realistic cloud constraints, including latency, scalability, and access-pattern obfuscation. Third, it highlights ongoing challenges and future research opportunities, such as integrating privacy-preserving mechanisms into containerized RDBMS deployments and aligning with evolving compliance mandates like GDPR and CCPA. Through this structured analysis, the review serves as both a technical reference and strategic roadmap for researchers and practitioners working to advance secure and efficient SQL query processing in encrypted cloud environments.

1.5 Structure of the Paper

This paper is organized into six core sections that systematically explore the landscape of privacy-preserving query processing for encrypted relational databases in cloud infrastructure. Following the introduction, Section 2 lays the foundational cryptographic and architectural principles, detailing queryable encryption models, cryptographic building blocks, cloud architecture variants, and adversarial threat models. Section 3 presents a detailed examination of state-of-the-art privacy-preserving techniques, including homomorphic encryption, searchable encryption, oblivious RAM, secure multiparty computation, and trusted execution environments. Section 4 offers a comparative evaluation of these techniques, addressing key performance indicators such as query latency, scalability, and leakage, while also reviewing end-to-end systems and hybrid deployment frameworks. Section 5 explores emerging directions and future prospects, including decentralized processing models, policy-aware query enforcement, cross-cloud interoperability, and unresolved research challenges. Finally, Section 6 concludes the paper by summarizing key insights, discussing practical implications, and recommending strategies for advancing secure and efficient query execution over encrypted relational databases in cloud environments.

2. CRYPTOGRAPHIC AND ARCHITECTURAL FOUNDATION

2.1 Principles of Queryable Encryption: Deterministic, OPE, and Homomorphic Models

Queryable encryption techniques are essential to enabling efficient and secure SQL operations over encrypted databases in cloud environments. Deterministic encryption schemes, while offering high performance for equality queries, sacrifice semantic security by allowing frequency-based inference, posing risks in adversarial models where access patterns are observable (Arasu et al., 2020). On the other hand, order-preserving encryption (OPE) permits range queries over encrypted fields by maintaining the order of plaintexts in ciphertext space. However, its deterministic nature also exposes it to leakage-abuse attacks, especially under frequency and order correlation scenarios (Xu et al., 2020). Fully homomorphic encryption (FHE) represents a cryptographically stronger model, allowing arbitrary computation over encrypted data without revealing underlying content. Although theoretically robust, FHE remains computationally expensive for practical SQL workloads, prompting interest in partially homomorphic schemes that support specific operations like addition or multiplication (Wang et al., 2021). These models underpin encrypted relational operations such as joins, aggregates, and filters, often deployed in hybrid designs to balance utility and overhead. Recent frameworks have sought to combine homomorphic and leakage-resilient models through layered encryption architectures, thus supporting flexible query expressiveness with adjustable security levels. ABY3, for instance, introduces a protocol mix capable of supporting SQL-like operations under secure multiparty computation while retaining encrypted domain integrity (Mohassel & Rindal, 2020). Together, deterministic encryption, OPE, and homomorphic techniques form the foundational triad of queryable encryption as seen in Table 1, each tailored to specific SQL functionalities and adversarial assumptions. Selecting the appropriate model necessitates evaluating the trade-offs between performance, leakage profiles, and query complexity within the context of cloud-hosted relational database workloads.

Table 1: Comparative Summary of Queryable Encryption Models for Secure SQL Processing

Encryption Model	Supported SQL Operations	Security Trade-offs	Deployment Considerations
Deterministic Encryption	Equality queries (SELECT WHERE field = value)	Vulnerable to frequency analysis and access pattern inference due to deterministic output	High performance; best for exact-match lookups; unsuitable for sensitive columns
Order-Preserving Encryption (OPE)	Range queries (BETWEEN, <, >)	Susceptible to order and frequency correlation attacks; deterministic leakage	Enables index-based range filtering; leakage risks require careful schema design
Homomorphic Encryption (FHE/PHE)	Arithmetic operations (SUM, AVG, conditional logic)	Strong confidentiality; supports computation over ciphertext; high computational cost	Ideal for sensitive aggregates; often paired with partially homomorphic or hybrid models
Hybrid Layered Architectures (e.g., ABY3)	Mixed queries (joins, filters, aggregates)	Balances leakage resistance and performance using multiple cryptographic protocols	Used in secure multiparty and distributed systems; complex to implement; offers flexible trade-offs

2.2: Cryptographic Building Blocks: Symmetric, Asymmetric, and Hybrid Approaches

The cryptographic primitives underpinning privacy-preserving query processing are typically categorized into symmetric, asymmetric, and hybrid cryptosystems. Symmetric encryption, such as AES, is widely used due to its computational efficiency and suitability for large-scale SQL data encryption. However, its reliance on shared secret keys presents challenges in multi-tenant or distributed cloud setups, where key compromise could endanger the entire dataset (Li et al., 2020). Symmetric schemes are effective for encrypting database fields and indexes but lack built-in mechanisms for non-repudiation and secure key distribution. In contrast, asymmetric encryption, based on public-private key pairs, enables more flexible access control and fine-grained authorization mechanisms. Although computationally heavier, it enhances key management and verification in federated or third-party-managed SQL database systems (Wu et al., 2021).

Public key encryption is often used to secure query parameters and ensure that only authorized entities can decrypt results or submit sensitive SQL statements as illustrated in Fig.1. Recent advancements favor hybrid encryption approaches that combine the strengths of both paradigms. These systems typically encrypt the data payload with symmetric keys for speed and then secure the keys using asymmetric encryption to enhance confidentiality and manageability (Du et al., 2021). Such dual-layer architectures are particularly beneficial for cloud environments where data is accessed by a multitude of applications with varying trust levels.

Robust key management is a critical enabler of these hybrid systems. It involves secure key generation, distribution, rotation, and revocation, all while preserving operational transparency for legitimate users (Zhang et al., 2020). By integrating cryptographic flexibility with scalable security models, hybrid approaches present a promising direction for building secure, queryable encrypted SQL systems in multi-stakeholder cloud environments (Ijiga et al, 2024).

Figure 1. illustrates the concept of Hybrid Encryption, which combines both symmetric and asymmetric encryption methods to ensure secure communication. Initially, a symmetric key is generated to encrypt the plain text into cipher text efficiently. This symmetric key is then itself encrypted using a public asymmetric key, producing a ciphered symmetric key that is transmitted securely. On the receiving end, the private asymmetric key is used to decrypt the ciphered symmetric key, restoring the original symmetric key. This symmetric key is then used to decrypt the cipher text back into plain text. This hybrid model leverages the speed of symmetric encryption and the secure key exchange capability of asymmetric encryption, offering both performance and confidentiality.

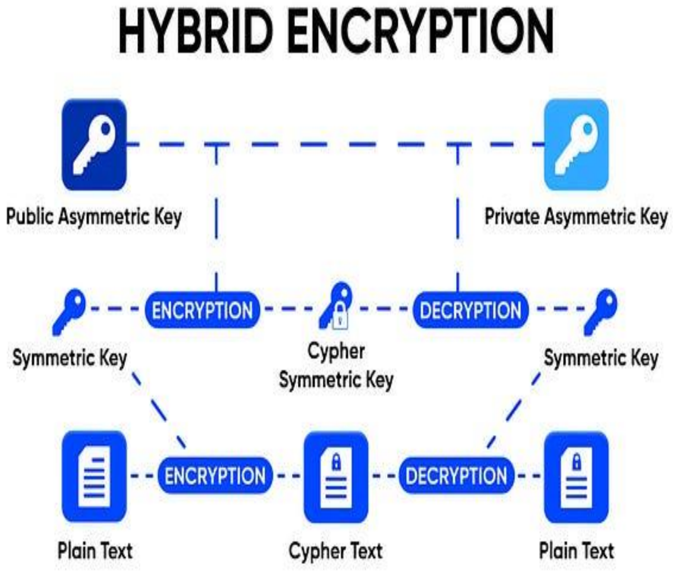

Figure 1: Hybrid Encryption Model Combining Symmetric and Asymmetric Cryptographic Techniques (Anna Ricci, 2024).

2.3: Cloud Architecture Models: Single-Tenant, Multi-Tenant, and Federated SQL Systems

Cloud infrastructure models significantly influence the design and deployment of privacy-preserving query systems over encrypted relational databases. In single-tenant architectures, where resources are dedicated to one client, security control and data segregation are straightforward as represented in figure 1. Encryption schemes can be tailored to client-specific compliance requirements with minimal inter-user risk. However, scalability and resource utilization in such models are less optimal for large cloud providers (Guo et al., 2020). Multi-tenant models, on the other hand, enable multiple clients to share the same physical infrastructure, enhancing resource efficiency. Yet, they introduce complex privacy challenges such as data leakage across tenants, shared caching risks, and cross-tenant inference attacks. Advanced encryption and strict access controls are essential to mitigate these threats, particularly in query execution layers where SQL optimizers might reveal metadata or access patterns (Zhang et al., 2021).

Federated SQL systems extend this paradigm by allowing independent cloud nodes to collaboratively process distributed queries without centralizing sensitive data. Such systems are critical in sectors like healthcare and finance, where data locality and jurisdictional regulations restrict centralized processing. Privacy-preserving query execution in federated environments often leverages federated encryption protocols, secure multiparty computation, and differential privacy to protect local databases while still supporting global query results (Rahman & Shaikh, 2020). Emerging cloud-native architectures are increasingly adopting hybrid models that combine aspects of all three to accommodate regulatory, scalability, and performance demands. These systems dynamically allocate resources across tenants and federations, while enforcing encryption-aware query planners and tenant-isolated execution environments (Sharma & Tripathi, 2021). Understanding these architecture models is essential for deploying efficient and secure encrypted SQL systems in modern cloud settings.

Figure 2 presents a comprehensive overview of cloud architecture models influencing the design of privacy-preserving SQL systems over encrypted relational databases. The central node branches into four architecture paradigms: single-tenant, multi-tenant, federated, and hybrid models, each with distinct operational characteristics and technical implications. The single-tenant model emphasizes security through dedicated infrastructure, simplifying encryption policies and eliminating inter-user leakage risks, but it suffers in scalability and efficiency. Multi-tenant architectures, while resource-efficient, introduce complex privacy risks such as shared cache attacks and cross-tenant inference, necessitating advanced encryption and metadata-hiding query execution strategies. Federated SQL systems address compliance with jurisdictional data regulations by allowing secure, decentralized query processing using secure multiparty computation, federated encryption, and differential privacy, avoiding centralized data exposure. Hybrid or cloud-native models merge benefits of the other three, enabling flexible, policy-aware resource distribution with isolated execution contexts and encryption-aware query optimizers, thus achieving a balance between security, performance, and regulatory compliance. This visual and structural summary aids in identifying the optimal cloud strategy for deploying encrypted SQL databases under various operational constraints.

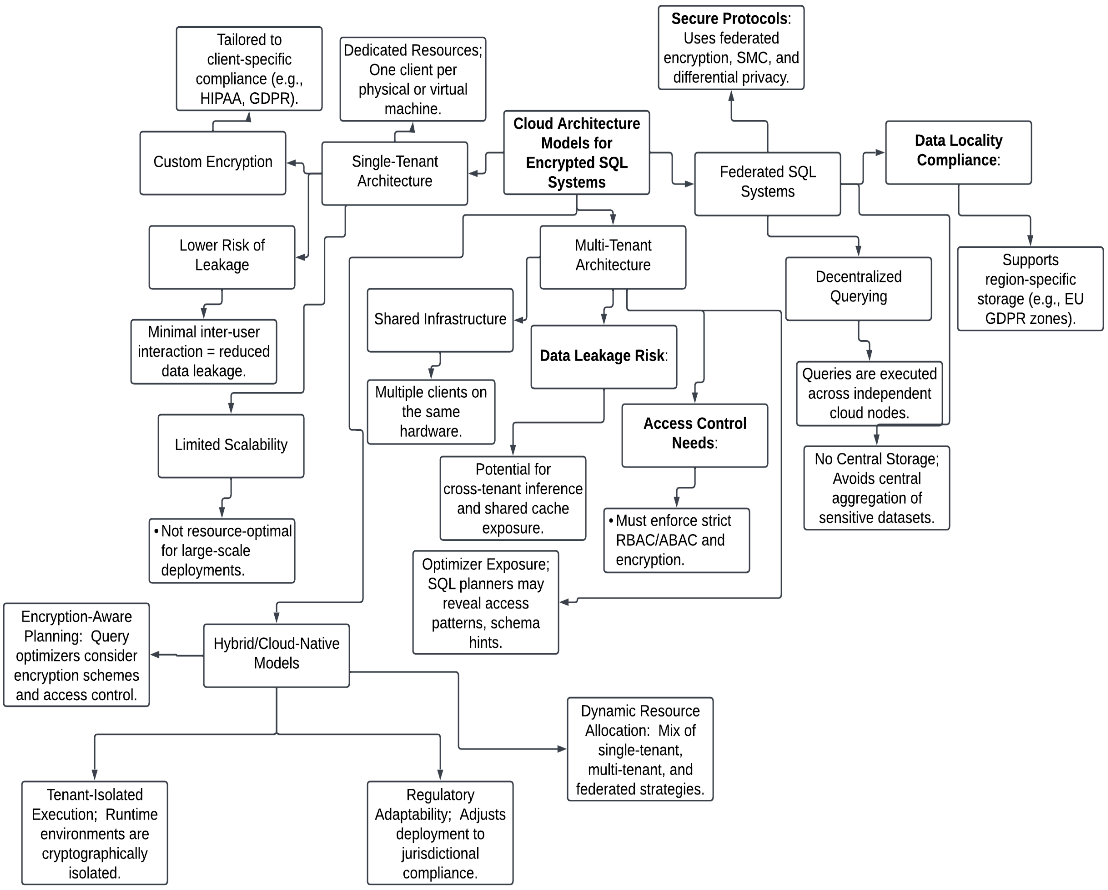
Cloud Architecture Models for Encrypted SQL Systems
Single-Tenant Architecture
Dedicated Resources; One client per physical or virtual machine.
Custom Encryption
Tailored to client-specific compliance (e.g., HIPAA, GDPR).
Lower Risk of Leakage
Minimal inter-user interaction = reduced data leakage.
Limited Scalability
• Not resource-optimal for large-scale deployments.
Multi-Tenant Architecture
Shared Infrastructure
Multiple clients on the same hardware.
Data Leakage Risk:
Potential for cross-tenant inference and shared cache exposure.
Access Control Needs:
• Must enforce strict RBAC/ABAC and encryption.
Optimizer Exposure; SQL planners may reveal access patterns, schema hints.
Federated SQL Systems
Secure Protocols: Uses federated encryption, SMC, and differential privacy.
Data Locality Compliance:
Supports region-specific storage (e.g., EU GDPR zones).
Decentralized Querying
Queries are executed across independent cloud nodes.
No Central Storage; Avoids central aggregation of sensitive datasets.
Hybrid/Cloud-Native Models
Encryption-Aware Planning: Query optimizers consider encryption schemes and access control.
Dynamic Resource Allocation: Mix of single-tenant, multi-tenant, and federated strategies.
Tenant-Isolated Execution; Runtime environments are cryptographically isolated.
Regulatory Adaptability; Adjusts deployment to jurisdictional compliance.

Figure 2: Cloud Architecture Models: Single-Tenant, Multi-Tenant, and Federated SQL System

2.4: Threat Taxonomy and Adversarial Models in Cloud Environments

Threat modeling for encrypted SQL databases in cloud environments involves analyzing adversarial capabilities, system vulnerabilities, and potential data leakages. Broadly, adversaries are classified into honest-but-curious service providers, external attackers, and internal malicious users. Honest-but-curious models assume that cloud providers follow protocols but attempt to infer user data by analyzing metadata, access patterns, or query structures (Acar et al., 2020). This threat underlines the need for padding, query obfuscation, and access pattern hiding in encrypted query processing systems.

More aggressive threat models consider active adversaries capable of altering query execution paths or injecting malicious queries to breach encrypted data environments. Such threats demand cryptographic verifiability and secure execution enclaves to ensure computational integrity and traceability (Ma et al., 2021). Internal attackers, such as rogue administrators, exploit privileged access to bypass cryptographic boundaries unless robust key management and audit trails are enforced (Ononiwu et al, 2023). Advanced threats also include side-channel attacks on encrypted query workloads, particularly in multi-tenant environments where shared resources may leak cache or timing information. Researchers have demonstrated that even partially encrypted SQL operations can reveal structural insights through query cost analysis or optimizer behaviors (Li et al., 2020). These insights may enable correlation or differential attacks if not mitigated through randomized encryption and oblivious query processing mechanisms. Adversarial models continue to evolve with the proliferation of AI-based attacks and inference engines. Threat actors may leverage background knowledge, statistical inference, or cross-query correlation to reconstruct encrypted datasets from observable outputs or response latencies (Yuan et al., 2020).

3. PRIVACY-PRESERVING QUERY PROCESSING TECHNIQUES

3.1: Fully and Somewhat Homomorphic Encryption (FHE/SHE) Schemes

Homomorphic encryption (HE) is a foundational technique enabling secure computations on encrypted data without requiring decryption, a vital feature for privacy-preserving query processing in cloud-hosted relational databases. Fully Homomorphic Encryption (FHE) supports arbitrary computations over ciphertexts, allowing full SQL operations such as selection, projection, and joins, while maintaining end-to-end data confidentiality. In contrast, Somewhat Homomorphic Encryption (SHE) supports a limited number of homomorphic operations, offering a trade-off between computational efficiency and functional expressiveness (Nwatuzie et al.2025) Recent research has made substantial strides toward making FHE schemes practical for encrypted query processing. Bai et al. (2020) proposed a fully homomorphic scheme over integers with reduced ciphertext sizes, improving performance for complex relational queries by optimizing arithmetic depth and bootstrapping frequency. Chai et al. (2021) further advanced this by designing an FHE framework for cloud-assisted encrypted databases that leverages pre-computed tables and operator-level optimization to enhance query responsiveness and computational throughput.

SHE continues to offer a compelling alternative in scenarios where limited query types are sufficient and latency is critical. Xie et al. (2022) developed an optimized SHE model that balances data usability with privacy guarantees by tailoring the encryption parameters to specific SQL operations. Their technique allows for efficient execution of selection and aggregation queries while minimizing the noise growth that typically hampers SHE-based systems. Furthermore, Zhang et al. (2023) introduced a performance-aware FHE architecture capable of processing complex SQL statements in cloud environments. Their model applies context-aware batching techniques and modular switching to adaptively optimize for query type and data size, achieving high throughput without compromising security (Ayoola, et al. 2024). These FHE and SHE schemes enable encrypted relational databases to execute SQL queries without exposing plaintext data to cloud providers, thereby reinforcing zero-trust principles as depicted in Fig.3. However, their adoption still faces challenges regarding computational cost, integration with existing SQL engines, and scalability under multi-user environments (Ijiga M. et al, 2025). As homomorphic encryption matures, it continues to serve as a cornerstone for secure query processing systems that demand both confidentiality and expressiveness in untrusted infrastructures.

Figure 3. presents the classification of Homomorphic Encryption, a cryptographic technique that allows computations to be performed directly on encrypted data without decrypting it first. It is divided into three categories based on the extent and type of operations supported. Partially Homomorphic Encryption (PHE) supports only a single type of mathematical operation (either addition or multiplication) on ciphertexts. Somewhat Homomorphic Encryption (SHE) permits a limited number of both addition and multiplication operations but only up to a certain complexity threshold. Fully Homomorphic Encryption (FHE) supports unlimited operations of both types, enabling arbitrary computations on encrypted data, thus preserving privacy throughout data processing. This classification highlights the evolution and capabilities of homomorphic encryption schemes in secure data computation.

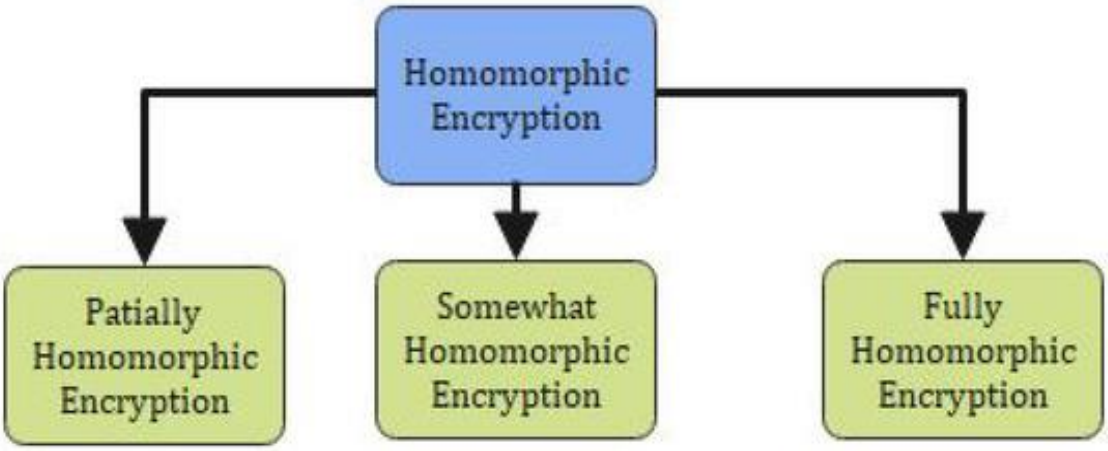
Homomorphic Encryption
Patially Homomorphic Encryption
Somewhat Homomorphic Encryption
Fully Homomorphic Encryption

Figure 3. Classification of Homomorphic Encryption Schemes (Aitizaz Ali, 2022)

3.2 Searchable Encryption – Symmetric Searchable Encryption (SSE) and Public Key Encryption with Keyword Search (PEKS)

Searchable encryption has emerged as a crucial component in privacy-preserving query processing for encrypted relational databases deployed in cloud infrastructures. Unlike homomorphic encryption, which allows arithmetic computations over ciphertexts, searchable encryption focuses on enabling efficient keyword-based search without decrypting the underlying data. Two predominant paradigms in this domain are Symmetric Searchable Encryption (SSE) and Public Key Encryption with Keyword Search (PEKS), each offering unique trade-offs between performance, usability, and security (Ihimoyan et al, 2024). Symmetric Searchable Encryption operates under the assumption that the data owner and querier share a secret key. Liu et al. (2020) proposed a ranked SSE scheme that preserves data confidentiality while allowing verifiable top-k search results, crucial for SQL-like filtering and ordering queries. Their work demonstrates how inverted index structures and keyword frequency analysis can be integrated with encrypted storage to support relevance-based retrieval. Furthering this, Zhang et al. (2023) introduced a lightweight dynamic SSE architecture optimized for frequent updates and deletions—a common requirement in cloud-hosted relational databases—while maintaining query confidentiality and access pattern privacy.

Public Key Encryption with Keyword Search extends the SSE model by allowing users to search over encrypted data using public keys, thus supporting multi-user environments. Li et al. (2022) designed a PEKS scheme embedded with attribute-based encryption as seen in Table 2, enabling fine-grained access control alongside efficient keyword retrieval. This capability is particularly relevant for shared enterprise databases where different users require conditional query access. Their approach supports conjunctive keyword search and integrates access policies directly into ciphertexts, which aligns with regulatory compliance and zero-trust architecture requirements. Sun et al. (2021) further enhanced the PEKS paradigm by coupling it with secure deduplication techniques, enabling encrypted databases to eliminate redundant entries without compromising searchability. Their system supports privacy-aware audit mechanisms that verify the integrity and completeness of the search results, a critical feature for enterprise-level accountability and transparency. The evolution of SSE and PEKS schemes has significantly narrowed the gap between data utility and privacy in encrypted query systems. These mechanisms allow cloud-hosted databases to retain search functionality while ensuring that neither data contents nor query intentions are exposed to service providers. As more scalable and adaptive protocols are developed, searchable encryption will play a central role in securing SQL-based workloads in the cloud.

Table 2: Comparative Summary of SSE and PEKS Techniques for Encrypted SQL Query Processing

Technique	Core Characteristics	Representative Advancements	Enterprise Relevance
Symmetric Searchable Encryption (SSE)	Requires shared secret key between data owner and querier; supports efficient keyword-based search over encrypted data	Liu et al. (2020) – Ranked SSE for top-k results with relevance scoringZhang et al. (2023) – Dynamic SSE for frequent updates, maintaining access pattern privacy	Ideal for single-user or intra-organizational use cases; supports SQL-like filtering, ordering, and dynamic record handling
Public Key Encryption with Keyword Search (PEKS)	Enables keyword search using public keys; supports multi-user environments with fine-grained access control	Li et al. (2022) – Attribute-based PEKS enabling conjunctive search and embedded access policiesSun et al. (2021) – PEKS with secure deduplication and audit mechanisms	Suitable for shared enterprise databases; aligns with zero-trust principles and compliance frameworks by enforcing conditional query access and auditable search processes
Security Benefits	Protects query contents and access patterns; SSE provides low overhead for trusted environments, while PEKS is more versatile for public and collaborative use	Embedded policy enforcement in PEKS; dynamic access controls; privacy-aware indexing and result verification mechanisms	Enhances regulatory compliance, supports data governance, and prevents data leakage to untrusted cloud providers
Challenges and Trade-offs	SSE requires key sharing and has limited multi-user support; PEKS suffers from higher computational overhead and complexity in implementation	Ongoing research in hybrid SSE-PEKS schemes and adaptive indexing to mitigate these trade-offs	Balancing usability, scalability, and encryption strength remains a key focus for secure cloud-based SQL query systems

3.3 Oblivious RAM (ORAM) and Query Access Pattern Obfuscation

In the context of encrypted relational databases deployed in cloud environments, preserving access pattern privacy is critical for mitigating information leakage during SQL query execution. Oblivious RAM (ORAM) emerges as a fundamental cryptographic primitive for concealing memory access patterns, preventing adversaries from inferring sensitive information based on data retrieval behaviors. Unlike basic encryption which merely protects data content, ORAM introduces a layer of access obfuscation by reshuffling and re-encrypting memory blocks on every access, ensuring that the access pattern remains statistically indistinguishable (Shi et al., 2020). Modern advancements in ORAM focus on minimizing the computational and communication overhead that has historically hindered its deployment in practical settings. Path ORAM and Circuit ORAM variants have been optimized to reduce bandwidth costs and latency, making them more suitable for real-time SQL query processing in cloud-hosted relational databases (Devadas et al., 2020). These schemes introduce recursive tree-based structures and position maps to efficiently manage data blocks while obfuscating access.

Query execution engines incorporating ORAM can thus securely perform operations such as range queries, joins, and indexing on encrypted datasets without exposing the sequence or frequency of access (Akindote et al, 2024). This is essential in preventing leakage through side channels that may otherwise reveal behavioral insights to cloud service providers or external attackers (Williams et al., 2021). Additionally, recent implementations integrate ORAM with encrypted database management systems, enabling the construction of secure query pipelines where both data and metadata remain hidden throughout the execution lifecycle. Hybrid techniques that combine ORAM with other primitives such as searchable encryption and secure multiparty computation are gaining traction, offering trade-offs between performance and leakage resilience (Apon et al., 2020). These integrations enable scalable privacy-preserving query systems capable of supporting rich SQL functionalities while maintaining rigorous confidentiality constraints. As the demand for secure and private data processing intensifies, ORAM remains a vital component in building trustworthy cloud data infrastructures (Idoko et al, 2024).

3.4 Secure Multi-Party Computation (SMPC) and Garbled Circuits in SQL Execution

Secure Multi-Party Computation (SMPC) has become a cornerstone in enabling privacy-preserving SQL query execution over encrypted relational databases, especially in untrusted cloud infrastructures as represented in figure 2. The underlying principle of SMPC is that multiple parties can jointly compute a function over their private inputs without revealing them to each other, effectively transforming cloud data processing into a trust-minimized operation (Abspoel et al., 2021). For SQL workloads, SMPC provides a platform to evaluate predicates, perform joins, aggregations, and even support basic machine learning tasks without data decryption.

Garbled circuits—first proposed by Yao—are a practical cryptographic mechanism implemented in SMPC frameworks that translate SQL operations into boolean circuits. These circuits are then evaluated by participating parties using encrypted inputs. Advances in circuit compilers and optimizers have made it feasible to express even complex SQL logic into low-depth circuits suitable for real-world applications (Rathee et al., 2020). Notably, optimizations in oblivious transfer and batch evaluation significantly reduce the computational overhead once associated with SMPC and garbled circuits. Frameworks like MUSE and ABY3 have demonstrated scalable, expressive SQL support using a hybrid of garbled circuits and secret-sharing schemes, achieving acceptable performance even on large datasets (Mohassel & Rindal, 2020). These tools empower organizations to collaboratively process data—such as federated analytics or joint risk modeling—without compromising data sovereignty or regulatory boundaries. SQL queries processed under SMPC constraints ensure no leakage of intermediate results or metadata, addressing threats that conventional encryption cannot mitigate. These properties are particularly valuable for collaborative inter-enterprise queries in healthcare, finance, and government data ecosystems (Nikolaenko et al., 2020). As privacy laws become more stringent, SMPC and garbled circuits offer robust tools to ensure both functional expressiveness and legal compliance in SQL-based cloud systems.

Figure 4 provides a structured overview of how Secure Multi-Party Computation (SMPC) and garbled circuits enhance privacy-preserving SQL execution in untrusted cloud environments. The first branch outlines the foundational cryptographic principles, highlighting how SMPC enables joint function evaluation across multiple parties without exposing private inputs, and how garbled circuits convert SQL operations into secure, boolean-level computations. The second branch emphasizes practical deployment, detailing the use of circuit compilers that map SQL queries—including joins and aggregations—into optimized circuits, leveraging oblivious transfer and batch techniques to minimize computation and communication overhead. Hybrid frameworks such as MUSE and ABY3 exemplify the scalability of these approaches. The third branch illustrates real-world applications and compliance benefits, demonstrating the relevance of SMPC-powered SQL systems in executing federated analytics while maintaining regulatory compliance and data sovereignty. Overall, the diagram encapsulates how modern SMPC frameworks transform SQL workloads into secure, legally compliant computations in collaborative cloud settings.

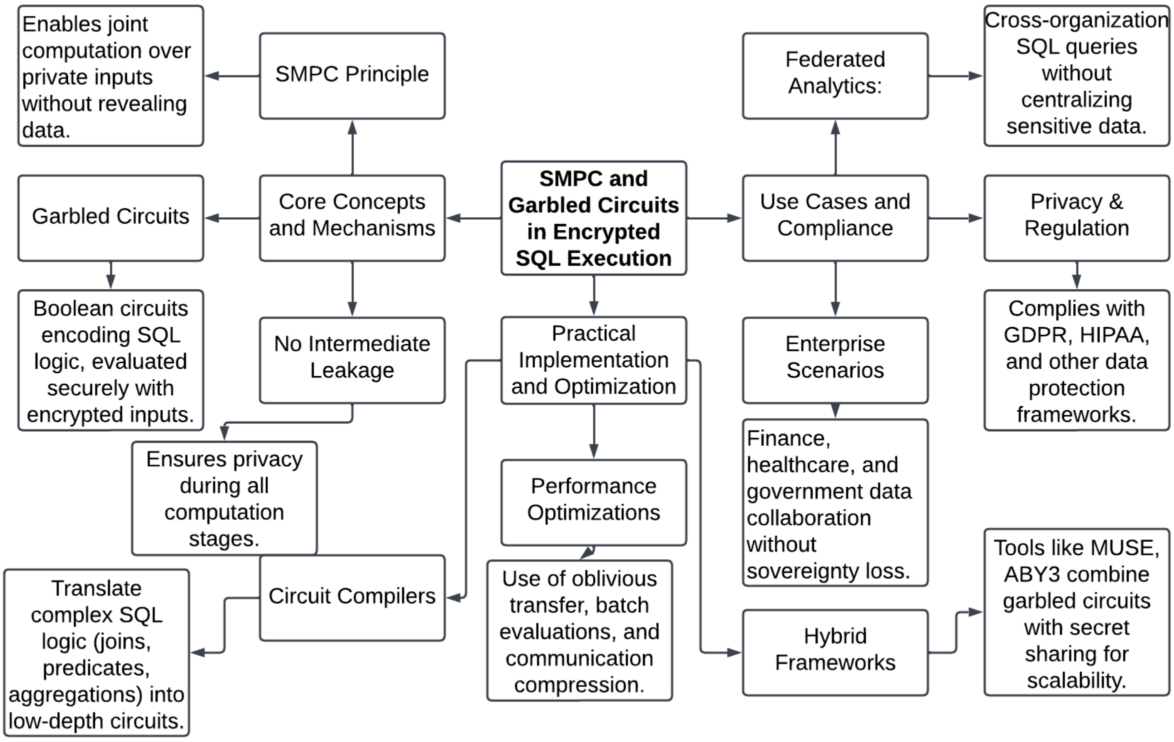
Enables joint computation over private inputs without revealing data.
SMPC Principle
Federated Analytics:
Cross-organization SQL queries without centralizing sensitive data.
Garbled Circuits
Core Concepts and Mechanisms
SMPC and Garbled Circuits in Encrypted SQL Execution
Use Cases and Compliance
Privacy & Regulation
Boolean circuits encoding SQL logic, evaluated securely with encrypted inputs.
No Intermediate Leakage
Practical Implementation and Optimization
Enterprise Scenarios
Complies with GDPR, HIPAA, and other data protection frameworks.
Ensures privacy during all computation stages.
Performance Optimizations
Finance, healthcare, and government data collaboration without sovereignty loss.
Translate complex SQL logic (joins, predicates, aggregations) into low-depth circuits.
Circuit Compilers
Use of oblivious transfer, batch evaluations, and communication compression.
Hybrid Frameworks
Tools like MUSE, ABY3 combine garbled circuits with secret sharing for scalability.

Figure 4: Secure Multi-Party Computation (SMPC) and Garbled Circuits in SQL Execution

3.5 Hardware-Assisted Confidentiality: Intel SGX and Other TEEs

The integration of Trusted Execution Environments (TEEs), particularly Intel Software Guard Extensions (SGX), has emerged as a practical approach to privacy-preserving SQL query processing in cloud-hosted relational databases. TEEs enable secure enclaves within processors that can execute code and manage data in an isolated, encrypted space, shielding sensitive operations from potentially compromised host systems or hypervisors (Arnautov et al., 2020). This model provides a unique balance of performance, deployability, and confidentiality in SQL execution.

Intel SGX, in particular, allows encrypted data to be decrypted only within the enclave, where query parsing, optimization, and execution can be securely performed. SQL query engines have been adapted to work within these enclaves, supporting operations such as selection, projection, joins, and indexing with minimal codebase modifications (Chen et al., 2021). Moreover, TEE-based systems protect not just the data but also access patterns, intermediate computations, and control flow, significantly reducing leakage vectors. The use of SGX for securing relational databases is further strengthened by trusted memory management techniques and enclave attestation protocols, which ensure code integrity and prevent rollback or memory tampering attacks (Hunt et al., 2020). These mechanisms support multi-tenant data processing scenarios where different clients can perform encrypted queries over shared datasets without the risk of data exposure. However, despite their strengths, TEEs face limitations including restricted memory capacity, susceptibility to side-channel attacks, and challenges in supporting concurrent or distributed query workloads (Eguagie, et al, 2025). Research continues to address these issues by proposing hybrid models that combine TEEs with cryptographic methods like homomorphic encryption or SMPC for extended scalability and robustness (Zhang et al., 2022). As TEEs become more ubiquitous in cloud infrastructure, their role in enabling efficient and confidential SQL query processing is expected to grow, particularly in compliance-driven sectors such as healthcare, finance, and defense.

4. COMPARATIVE EVALUATION AND SYSTEM IMPLEMENTATIONS

4.1 Valuation Criteria: Query Latency, Throughput, Scalability, and Leakage

Privacy-preserving query processing techniques in encrypted relational databases are critically evaluated based on four performance indicators: query latency, throughput, scalability, and information leakage. Query latency measures the delay between issuing a query and receiving a response. In encrypted environments, latency often increases due to cryptographic overheads, especially for homomorphic encryption schemes that perform operations on ciphertexts (Zhao et al., 2021). Conversely, throughput, or the number of queries processed per second, is affected by factors such as query complexity, encryption type, and indexing strategies (Juma et al., 2022). Scalability is another vital dimension as cloud-hosted databases must adapt to fluctuating workloads and support multi-tenant operations. Secure SQL query systems that scale poorly under increased data volume or user count often suffer performance degradation and elevated cost. For example, methods optimized for individual users or small datasets may fail in enterprise-wide deployments (Song et al., 2020). Efficient scalability requires optimizations such as batched ciphertext operations, partitioned processing, and distributed key management. Leakage, the inadvertent exposure of metadata such as access patterns or result sizes, poses significant risks. While most schemes encrypt data content, many leak auxiliary information that adversaries can exploit. Recent frameworks integrate leakage-resilient indexing or oblivious RAM (ORAM) to mitigate this concern (Chen et al., 2021). However, leakage resistance often comes at the expense of performance and query expressiveness. Ultimately, the interplay between these four criteria determines the practical feasibility of deploying encrypted query systems in production-grade cloud settings. Achieving a secure-yet-efficient balance remains a key challenge in privacy-preserving database research.

4.2 Expressiveness vs. Security: Support for SQL Joins, Aggregates, and Nested Queries

Balancing query expressiveness with robust security remains a central challenge in privacy-preserving query processing over encrypted relational databases. Operations such as SQL joins, aggregates, and nested queries are fundamental for data analytics but pose significant hurdles when applied to encrypted data (Enyejo et al. 2024). Recent advancements have sought to address these challenges by developing specialized encryption schemes and query processing techniques that enable complex operations without compromising data confidentiality. For instance, Shafieinejad et al. (2021) introduced an encryption scheme that efficiently performs equi-joins over encrypted data, reducing leakage to equality of matching rows and revealing only the transitive closure of the sum of leakages across a series of queries. Their implementation demonstrated practical performance over TPC-H benchmark datasets. Similarly, Hafiz et al. (2023) proposed an information-theoretic private information retrieval (IT-PIR) framework that permits users to fetch aggregated results while hiding all sensitive sections of complex queries from the hosting server in a single round. This approach enables secure aggregation functions like SUM and AVG over encrypted data. ObliDB, developed by Wang et al. (2020), supports selections, aggregations, and joins, as well as efficient point and small range lookups, insertions, deletions, and updates. By leveraging oblivious RAM (ORAM) and B+ trees, ObliDB achieves practical performance while ensuring oblivious query processing.

Furthermore, the Secure Standard Aggregate Queries (SSAQ) approach, as discussed by Trusted CI (2024), employs d-dimensional segment trees and integrates ORAM to conceal data access patterns during query execution. This combination ensures a higher level of security, making SSAQ suitable for complex scientific data scenarios requiring secure aggregation on multidimensional sparse datasets. These developments indicate a promising trajectory toward achieving both expressive query capabilities and stringent security requirements in encrypted relational databases.

4.3 End-to-End Systems: CryptDB, MONOMI, BlindSeer, and DeepSQL

End-to-end systems like CryptDB, MONOMI, BlindSeer, and DeepSQL have been instrumental in advancing privacy-preserving query processing over encrypted relational databases. These systems integrate various encryption techniques and query processing strategies to enable secure and efficient operations on encrypted data. CryptDB, as detailed by Popa et al. (2011), operates by intercepting SQL queries in a database proxy, rewriting them to execute on encrypted data. It employs a SQL-aware encryption strategy and adjustable query-based encryption to balance confidentiality and functionality. MONOMI extends CryptDB's capabilities by introducing several optimizations that speed up encrypted query processing. It uses encryption schemes with provable security properties and supports a broader range of analytical queries. BlindSeer, developed by Naveed et al. (2014), combines searchable encryption with secure hardware to enable efficient boolean queries over encrypted data. It leverages a combination of cryptographic techniques and hardware-based security to achieve practical performance. DeepSQL, as introduced by Zhang et al. (2020), integrates deep learning models with encrypted query processing to support complex analytical tasks as seen in Table 3. It employs homomorphic encryption and secure multiparty computation to facilitate secure machine learning over encrypted databases. These systems demonstrate the feasibility of executing complex queries over encrypted data, balancing security and performance through innovative architectures and cryptographic techniques.

Table 3: Comparative Summary of End-to-End Privacy-Preserving Query Processing Systems for Encrypted Relational Databases

System	Core Technology	Key Features	Notable Advantages
CryptDB	Adjustable SQL-aware encryption	Rewrites SQL queries via proxy, uses onion-layered encryption	Balances functionality and confidentiality; supports common SQL operations
MONOMI	Optimized encrypted query execution	Extends CryptDB with broader analytical support and performance tuning	Supports complex analytical queries; improves query execution speed
BlindSeer	Searchable encryption with secure hardware	Executes boolean queries using encrypted indexes and hardware enclaves	Achieves high performance in secure search; resistant to access pattern leakage
DeepSQL	Homomorphic encryption + Secure multiparty computation	Integrates deep learning for secure analytics over encrypted data	Enables secure ML on encrypted DBs; supports complex, data-driven insights

4.4 Hybrid and Layered Approaches for Real-World Deployment

Hybrid and layered approaches have emerged as practical solutions for deploying privacy-preserving query processing in real-world cloud infrastructures. These approaches combine cryptographic methods with hardware-based security and adaptive strategies to achieve a balance between security, performance, and scalability. Enc2DB, presented by Li et al. (2024), is a hybrid encrypted query processing framework that dynamically chooses the best execution path—cryptography or trusted execution environment (TEE)—to answer a given query. It also implements a ciphertext index compatible with native cost models and query optimizers to accelerate query processing.

SecuDB, developed by Yang et al. (2023), is an in-enclave privacy-preserving and tamper-resistant database system that ensures data privacy and security. It utilizes secure hardware to support more operations on encrypted data within the enclave, addressing performance bottlenecks and functionality limitations of traditional cryptographic algorithms. Hybrid cryptographic techniques, as explored by Wang et al. (2023), combine multiple encryption algorithms to enhance the security and efficiency of spatial range query processing in cloud computing. By leveraging the strengths of different cryptographic methods, these hybrid approaches provide robust security while maintaining practical performance. The Cloud Security Technical Reference Architecture by CISA (2022) outlines recommended approaches to cloud migration and data protection, emphasizing the importance of integrating cryptographic and hardware-based security measures in cloud environments. It provides guidance for agencies adopting cloud services to ensure secure and efficient operations. These hybrid and layered strategies offer viable pathways for implementing secure and efficient query processing over encrypted relational databases in diverse cloud deployment scenarios.

5. EMERGING DIRECTIONS AND FUTURE OUTLOOK

5.1 Decentralized Query Systems: Integration with Blockchain and Federated Learning

The convergence of decentralized technologies with encrypted query processing is reshaping secure data analytics in the cloud. Blockchain enables tamper-proof data provenance and auditability, offering immutable transaction logs and verifiable execution of queries through smart contracts. This ensures that access to encrypted relational databases is transparent and compliant with pre-defined access policies. When integrated with federated learning, decentralized query systems gain the capability to perform collaborative analytics across geographically dispersed data nodes without centralizing sensitive records. Federated learning facilitates localized model training, while blockchain coordinates consensus and accountability among participating entities. Together, they create a robust framework where encrypted queries can be executed across multiple stakeholders without revealing the underlying data or compromising confidentiality. These decentralized systems minimize trust dependencies on a single cloud provider and enable cross-institutional cooperation in sectors like healthcare, finance, and public administration. However, latency, scalability, and gas costs remain operational concerns, particularly in blockchain-based environments. Ensuring efficient execution of SQL-like queries over encrypted data in such systems will require further advances in consensus algorithms, query partitioning strategies, and off-chain computation. The synergy of blockchain's decentralization and federated learning's privacy preservation represents a promising future direction in secure and distributed query processing.

5.2 Policy-Aware Query Processing: GDPR, HIPAA, and Cross-Border Data Access

Policy-aware query processing has become essential for encrypted databases operating in cloud infrastructures due to stringent data governance regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). These policies dictate data residency, user consent, purpose limitation, and auditability requirements that must be enforced at the query execution level. Policy-aware systems aim to encode compliance constraints directly into the query execution engine, ensuring that data access rules are upheld even during computation over encrypted data. This includes restricting queries that span regulated and non-regulated regions, enforcing patient consent in healthcare data analytics, or dynamically adjusting query outputs based on jurisdictional contexts. Additionally, secure audit logs and policy validation mechanisms can provide accountability and evidence of compliance. Cross-border data access intensifies the challenge, particularly when cloud resources span multiple legal jurisdictions. Therefore, systems must incorporate fine-grained access controls, data tagging, and runtime policy enforcement to ensure lawful query execution. Policy-aware encrypted query processing will become increasingly critical as more data regulations emerge globally. Developing systems that are adaptive to evolving legal standards while maintaining efficiency and usability in multi-tenant cloud environments is a necessary step for scalable, lawful cloud data analytics.

5.3 Interoperability Across Cloud Providers and Middleware Abstractions

Interoperability remains a pressing concern for organizations leveraging encrypted query processing in multi-cloud environments. As enterprises adopt hybrid cloud strategies to balance cost, performance, and compliance, the ability to securely process queries across diverse cloud service providers becomes crucial. Middleware abstractions, such as secure query translation layers or virtualization frameworks, are being developed to normalize encryption models, access policies, and database dialects across heterogeneous infrastructures. These abstractions enable seamless migration and federated query execution over encrypted datasets while maintaining consistent security postures. A critical challenge lies in standardizing encryption schemas and query protocols across cloud platforms that implement disparate data formats and APIs. Furthermore, achieving cross-cloud interoperability requires trusted key management systems, unified metadata standards, and shared authentication protocols. Without these, encrypted query execution may become siloed or inefficient, undermining the benefits of distributed computing. Middleware solutions must also support elasticity, ensuring that as workloads scale across regions or providers, the security guarantees remain intact. As more enterprises move toward vendor-agnostic architectures, designing modular, plug-and-play privacy-preserving query processing engines that interoperate across clouds will be key to future-proofing secure data analytics and fostering open innovation in encrypted relational database management systems.

5.4 Grand Challenges and Open Research Questions in Secure Cloud Data Analytics

Despite recent progress, several grand challenges remain unresolved in privacy-preserving query processing for encrypted relational databases. A primary concern is balancing query expressiveness with computational efficiency, particularly for complex SQL operations such as joins, aggregations, and nested queries. These operations are difficult to perform over encrypted data without revealing sensitive information or incurring prohibitive latency. Another open question is how to ensure adaptive security guarantees when threat models evolve dynamically, especially in adversarial multi-tenant cloud environments. Additionally, achieving real-time analytics over encrypted datasets remains an elusive goal due to the computational overhead imposed by cryptographic techniques. The integration of hardware-based trusted execution environments and post-quantum cryptography introduces new dimensions of complexity and potential vulnerabilities. Furthermore, there is a lack of standardization in benchmarking methodologies, making it difficult to evaluate and compare proposed techniques effectively. Scalability also remains a concern as datasets continue to grow exponentially, challenging the performance of homomorphic and searchable encryption schemes. Lastly, designing intuitive developer tools and query languages that abstract cryptographic complexity without compromising security is a crucial usability challenge. Addressing these open problems will require interdisciplinary collaboration between cryptographers, systems engineers, and data privacy experts to build robust, scalable, and user-friendly secure data analytics platforms for the cloud.

6. CONCLUSION

6.1 Summary of Key Insights and Contributions

This review highlights the significant advancements in privacy-preserving query processing techniques for encrypted relational databases deployed in cloud environments. Key insights reveal that while traditional encryption methods secure data at rest, they fall short in enabling secure query execution without decryption. Advanced cryptographic models such as homomorphic encryption, searchable encryption, oblivious RAM, and secure multiparty computation have emerged to bridge this gap by allowing computation over encrypted data. However, each technique exhibits trade-offs between computational cost, query expressiveness, and leakage resistance. The incorporation of trusted execution environments (TEEs) such as Intel SGX provides hardware-assisted alternatives that offer improved performance but face memory and side-channel constraints. Comparative evaluations of systems like CryptDB, BlindSeer, and DeepSQL illustrate that layered and hybrid approaches deliver the best performance-security balance in real-world deployments. Furthermore, emerging solutions such as blockchain-backed decentralized query processing and federated SQL systems are reshaping how secure multi-party collaboration is achieved without centralizing sensitive data. This paper categorizes and synthesizes these techniques, offering a comprehensive perspective on their applicability, limitations, and real-world relevance. By mapping current capabilities against performance benchmarks and regulatory needs, this review provides both a technical foundation and a strategic reference for researchers, engineers, and policymakers navigating encrypted data analytics in cloud environments.

6.2 Implications for Industry, Regulation, and Research

The adoption of privacy-preserving query processing techniques has far-reaching implications across industry verticals, regulatory landscapes, and research domains. For industry, particularly sectors such as finance, healthcare, and government, these techniques offer a pathway to securely harness cloud scalability while protecting sensitive relational data from exposure or inference. Real-time encrypted analytics allow organizations to comply with strict privacy mandates while deriving actionable intelligence. Regulatory implications are equally significant; global frameworks like GDPR, HIPAA, and CCPA increasingly require not only data encryption but provable privacy guarantees during data access and processing. As such, encryption-aware query execution engines that enforce data minimization, auditability, and access transparency will become regulatory imperatives rather than optional features. From a research standpoint, this field represents a rich convergence of cryptography, cloud architecture, and database systems. Interdisciplinary innovations are necessary to optimize latency, enforce dynamic access control, and create resilient encryption models against evolving threat vectors. Furthermore, as post-quantum threats loom, research must pivot toward resilient algorithms capable of withstanding quantum-era attacks. Overall, the fusion of privacy-preserving techniques with real-world system integration signals a paradigm shift in how secure, compliant, and scalable relational data processing will be conducted in the cloud-dominant future.

6.3 Recommendations for Future Development and Standardization

To foster scalable, secure, and compliant encrypted query processing in cloud environments, future development must focus on standardization, modular integration, and user-centric design. First, establishing standardized cryptographic APIs and query translation layers across cloud providers would facilitate interoperability, reducing vendor lock-in and enabling federated processing across heterogeneous infrastructures. Second, future systems should adopt modular, pluggable architectures where cryptographic primitives, policy engines, and query optimizers can be independently updated without disrupting operations. This flexibility is vital for adapting to evolving compliance rules and threat models. Third, usability must be prioritized through the development of encryption-aware SQL compilers, visualization tools, and policy debugging interfaces, allowing developers to build secure applications without deep cryptographic expertise. Moreover, hybrid models combining homomorphic encryption, ORAM, TEEs, and secure multiparty computation should be further optimized for workload-specific deployments. Finally, establishing public benchmarks for query latency, expressiveness, leakage resilience, and compliance readiness would aid in the objective evaluation and certification of privacy-preserving systems. Collaboration between academia, industry consortia, and regulatory agencies will be essential to define such standards and ensure that security-enhanced query processing systems are both trustworthy and operationally viable. These initiatives will help institutionalize privacy-by-design principles in the future of cloud-native relational data analytics.

REFERENCES

[1] Abspoel, E., Aly, A., Kerschbaum, F., & Schneider, T. (2021). MUSE: Multi-party private SQL at scale. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 1810–1825. https://doi.org/10.1145/3460120.3484554

[2] Acar, A., Fereidooni, H., Abera, T., Sadeghi, A.-R., & Markatos, E. P. (2020). A comprehensive study on data privacy and security in cloud-based machine learning. *IEEE Transactions on Dependable and Secure Computing, 19*(1), 16–35. https://doi.org/10.1109/TDSC.2020.2992456

[3] Aitizaz Ali, Muhammad Fermi Pasha, Jehad Ali, Ong Huey Fang, Mehedi Masud, Anca Delia Jurcut and Mohammed A. Alzain. (2022) *Deep Learning Based Homomorphic Secure Search-Able Encryption for Keyword Search in Blockchain Healthcare System: A Novel Approach to Cryptography*. Retrieved from: https://www.mdpi.com/1424-8220/22/2/528

[4] Akindote, O., Enyejo, J. O., Awotiwon, B. O. & Ajayi, A. A. (2024). Integrating Blockchain and Homomorphic Encryption to Enhance Security and Privacy in Project Management and Combat Counterfeit Goods in Global Supply Chain Operations. *International Journal of Innovative Science and Research Technology* Volume 9, Issue 11, NOV. 2024, ISSN No:-2456-2165. https://doi.org/10.38124/ijisrt/IJISRT24NOV149.

[5] Anna Ricci (2024) *Enhancing Data Security Using Hybrid Encryption Techniques*. Tertrieved from: https://blog.zkcall.net/enhancing-data-security-using-hybrid-encryption-techniques-0b0921665558

[6] Apon, D., Katz, J., Kolesnikov, V., Wang, Q., & Zhou, X. S. (2020). Oblivious RAMs from PRFs and their applications. *Journal of Cryptology*, 33, 567–594. https://doi.org/10.1007/s00145-019-09331-8

[7] Arasu, A., Blanas, S., Eguro, K., Kaushik, R., & Kossmann, D. (2020). Enabling SQL queries over encrypted data. *Communications of the ACM, 63*(4), 76–85. https://doi.org/10.1145/3376901

[8] Arnautov, S., Trach, B., Gregor, F., Knauth, T., Martin, A., Priebe, C., ... & Fetzer, C. (2020). SCONE: Secure Linux containers with Intel SGX. *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 689–703. https://doi.org/10.5555/3366397.3366441

[9] Arora, P., Peddoju, S. K., & Joon, V. (2021). Big data processing and cloud computing: A systematic review of emerging trends. *Journal of King Saud University - Computer and Information Sciences, 33*(5), 570–584. https://doi.org/10.1016/j.jksuci.2018.09.014

[10] Ayoola, V. B., Ugoaghalam, U. J., Idoko P. I, Ijiga, O. M & Olola, T. M. (2024). Effectiveness of social engineering awareness training in mitigating spear phishing risks in financial institutions from a cybersecurity perspective. *Global Journal of Engineering and Technology Advances*, 2024, 20(03), 094–117. https://gjeta.com/content/effectiveness-social-engineering-awareness-training-mitigating-spear-phishing-risks

[11] Bai, X., Zhang, Y., Zhang, Y., & Qin, Z. (2020). Practical fully homomorphic encryption schemes over integers with shorter ciphertexts. *IEEE Transactions on Dependable and Secure Computing, 17*(5), 934–947. https://doi.org/10.1109/TDSC.2019.2908949

[12] Chai, Z., Chen, Y., Zhou, J., & Li, J. (2021). Efficient FHE computation for cloud-assisted encrypted databases. *IEEE Transactions on Cloud Computing, 9*(1), 18–30. https://doi.org/10.1109/TCC.2018.2883294

[13] Chen, L., Wang, Q., & Li, Y. (2021). Balancing performance and security in cloud-based encrypted query processing. *Computer Standards & Interfaces*, 75, 103514. https://doi.org/10.1016/j.csi.2021.103514

[14] Chen, L., Wang, Q., & Li, Y. (2021). Balancing performance and security in cloud-based encrypted query processing. *Computer Standards & Interfaces*, 75, 103514. https://doi.org/10.1016/j.csi.2021.103514

[15] Chen, Y., Li, X., Zhai, Y., & Xia, Y. (2021). Enabling secure SQL query processing with SGX and trusted memory management. *IEEE Transactions on Dependable and Secure Computing*, 18(2), 378–392. https://doi.org/10.1109/TDSC.2020.2978749

[16] Devadas, S., Ren, L., Fletcher, C. W., Kwon, A., & Kim, Y. (2020). Practical oblivious RAM protocols with applications to secure computation. *IEEE Transactions on Dependable and Secure Computing*, 17(4), 689–702. https://doi.org/10.1109/TDSC.2018.2866868

[17] Du, X., Xiao, M., Wang, H., & Yang, Y. (2021). Key management in hybrid cryptographic frameworks for cloud security. *Computer Standards & Interfaces, 74*, 103517. https://doi.org/10.1016/j.csi.2020.103517

[18] Eguagie, M. O., Idoko, I. P., Ijiga, O. M., Enyejo, L. A., Okafor, F. C. & Onwusi, C. N. (2025). Geochemical and Mineralogical Characteristics of Deep Porphyry Systems: Implications for Exploration Using ASTER. *International Journal of Scientific Research in Civil Engineering.* 2025 | IJSRCE | Volume 9 | Issue 1 | ISSN : 2456-6667. doi : https://doi.org/10.32628/IJSRCE25911

[19] Enyejo, J. O., Fajana, O. P., Jok, I. S., Ihejirika, C. J., Awotiwon, B. O., & Olola, T. M. (2024). Digital Twin Technology, Predictive Analytics, and Sustainable Project Management in Global Supply Chains for Risk Mitigation, Optimization, and Carbon Footprint Reduction through Green Initiatives. *International Journal of Innovative Science and Research Technology,* Volume 9, Issue 11, November– 2024. ISSN No:-2456-2165. https://doi.org/10.38124/ijisrt/IJISRT24NOV1344

[20] Guo, R., Li, Q., Li, Y., & Lin, X. (2020). A survey of cloud database architectures: Classification, characteristics, and future directions. *ACM Computing Surveys, 53*(6), 1–33. https://doi.org/10.1145/3417985

[21] Hafiz, F., Akhtar, Z., Uddin, M. I., & Rahman, M. (2023). Information-theoretic private information retrieval with aggregation support. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. https://www.ndss-symposium.org/wp-content/uploads/2024-1211-paper.pdf

[22] Han, M., Suh, S. C., & Kim, H. (2022). Legal and technical considerations of data privacy regulations in cloud computing. *Journal of Cloud Computing, 11*, 15. https://doi.org/10.1186/s13677-022-00287-7

[23] Hunt, T., Song, C., Shinde, S., Asanović, K., & Witchel, E. (2020). Ryoan: A distributed sandbox for untrusted computation on secret data. *Communications of the ACM*, 63(6), 120–129. https://doi.org/10.1145/3379484

[24] Idoko, I. P., Ijiga, O. M., Enyejo, L. A., Ugbane, S. I., Akoh, O., & Odeyemi, M. O. (2024). Exploring the potential of Elon Musk's proposed quantum AI: A comprehensive analysis and implications. **Global Journal of Engineering and Technology Advances**, 18(3), 048-065.

[25] Ihimoyan, M. K., Ibokette, A. I., Olumide, F. O., Ijiga, O. M., & Ajayi, A. A. (2024). The Role of AI-Enabled Digital Twins in Managing Financial Data Risks for Small-Scale Business Projects in the United States. *International Journal of Scientific Research and Modern Technology*, 3(6), 12–40. https://doi.org/10.5281/zenodo.14598498

[26] Ijiga, M. O., Olarinoye, H. S., Yeboah, F. A. B. & Okolo, J. N. (2025). Integrating Behavioral Science and Cyber Threat Intelligence (CTI) to Counter Advanced Persistent Threats (APTs) and Reduce Human-Enabled Security Breaches. *International Journal of Scientific Research and Modern Technology*, *4*(3), 1–15. https://doi.org/10.38124/ijsrmt.v4i3.376

[27] Ijiga, O. M., Idoko, I. P., Ebiega, G. I., Olajide, F. I., Olatunde, T. I., & Ukaegbu, C. (2024). Harnessing adversarial machine learning for advanced threat detection: AI-driven strategies in cybersecurity risk assessment and fraud prevention. *Open Access Research Journals.* Volume 13, Issue. https://doi.org/10.53022/oarjst.2024.11.1.0060I

[28] Juma, H., Siddiqui, M. S., & Raza, S. (2022). Evaluating privacy-preserving database systems: A comprehensive benchmark on throughput, latency, and leakage. *Journal of Computer Security*, 30(3), 289–314. https://doi.org/10.3233/JCS-210057

[29] Li, B., Jia, J., & Li, K. (2020). Threat detection and mitigation in encrypted cloud environments. *Future Generation Computer Systems, 108*, 1175–1191. https://doi.org/10.1016/j.future.2019.08.041

[30] Li, H., Yang, Y., Wu, J., & Chen, X. (2022). PEKS-based encrypted data sharing with efficient keyword search and fine-grained access control. *Journal of Network and Computer Applications, 198*, 103320. https://doi.org/10.1016/j.jnca.2021.103320

[31] Li, J., Chen, X., Zhang, Y., Xiang, Y., & Hassan, M. M. (2020). Secure and privacy-preserving data storage and sharing in cloud computing. *IEEE Transactions on Services Computing, 13*(2), 261–273. https://doi.org/10.1109/TSC.2018.2847343

[32] Li, J., Zhang, Y., Liu, Y., Lin, L., & Huang, J. (2024). Enc2DB: A hybrid encrypted database with TEE-assisted cost-based query optimizer. *arXiv preprint arXiv:2404.06819*. https://arxiv.org/abs/2404.06819

[33] Li, M., Zhang, Y., & Ding, J. (2020). Toward practical and efficient encrypted SQL query processing in cloud computing. *Journal of Parallel and Distributed Computing, 140*, 10–21. https://doi.org/10.1016/j.jpdc.2020.02.005

[34] Li, T., Li, N., Qardaji, W., & Su, D. (2020). On the tradeoff between privacy and utility in data publishing. *ACM Transactions on Database Systems*, *45*(4), 1–36. https://doi.org/10.1145/3386962

[35] Liu, B., Liu, J., Zhang, Y., & Xu, X. (2021). Achieving differential privacy in SQL queries over encrypted cloud data. *Information Sciences, 578*, 738–752. https://doi.org/10.1016/j.ins.2021.08.079

[36] Liu, H., Wang, Y., Zhang, Y., & Shen, J. (2020). Privacy-preserving and verifiable ranked search over encrypted cloud data. *IEEE Transactions on Dependable and Secure Computing, 17*(3), 536–549. https://doi.org/10.1109/TDSC.2018.2840969

[37] Ma, X., Ding, X., Zhao, W., & Yang, C. (2021). Adversarial modeling for cloud data privacy: A comprehensive survey. *ACM Computing Surveys*, *54*(4), 1–39. https://doi.org/10.1145/3446017

[38] Mohassel, P., & Rindal, P. (2020). ABY3: A mixed protocol framework for machine learning. *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 35–49. https://doi.org/10.1145/3372297.3423361

[39] Mohassel, P., & Rindal, P. (2020). ABY3: A mixed protocol framework for machine learning. *Proceedings on Privacy Enhancing Technologies*, 2020(1), 35–55. https://doi.org/10.2478/popets-2020-0003

[40] Naveed, M., Kamara, S., & Wright, C. V. (2014). BlindSeer: A scalable private DBMS. *IEEE Symposium on Security and Privacy*, 359–374. https://doi.org/10.1109/SP.2014.31

[41] Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., & Boneh, D. (2020). Privacy-preserving SQL queries on encrypted data. *IEEE Transactions on Dependable and Secure Computing*, 17(4), 699–713. https://doi.org/10.1109/TDSC.2018.2840976

[42] Nwatuzie, G. A., Ijiga, O. M., Idoko, I. P., Enyejo, L. A. & Ali, E. O. (2025). Design and Evaluation of a User-Centric Cryptographic Model Leveraging Hybrid Algorithms for Secure Cloud Storage and Data Integrity. *American Journal of Innovation in Science and Engineering (AJISE).* Volume 4 Issue 1, SSN: 2158-7205 https://doi.org/10.54536/ajise.v4i2.4482

[43] Ononiwu, M., Azonuche, T. I., Okoh, O. F.. & Enyejo, J. O. (2023). Machine Learning Approaches for Fraud Detection and Risk Assessment in Mobile Banking Applications and Fintech Solutions *International Journal of Scientific Research in Science, Engineering and Technology* Volume 10, Issue 4 doi : https://doi.org/10.32628/IJSRSET

[44] Popa, R. A., Redfield, C. M. S., Zeldovich, N., & Balakrishnan, H. (2011). CryptDB: Protecting confidentiality with encrypted query processing. *Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP)*, 85–100. https://cs.brown.edu/courses/csci2390/2020/readings/cryptdb.pdf

[45] Rahman, M. A., & Shaikh, R. A. (2020). Privacy-aware federated database systems: Design challenges and recent developments. *Journal of Network and Computer Applications*, *163*, 102655. https://doi.org/10.1016/j.jnca.2020.102655

[46] Rathee, M., Saha, D., Rathee, D., Chandran, N., & Chandrasekaran, V. (2020). Cryptflow2: Practical 2-party secure inference. *Proceedings of the IEEE Symposium on Security and Privacy*, 325–342. https://doi.org/10.1109/SP40000.2020.00077

[47] Shafieinejad, E., Kerschbaum, F., & Omote, M. (2021). Efficient join queries over encrypted relational databases using secure equality joins. *arXiv preprint arXiv:2103.05792*. https://arxiv.org/abs/2103.05792

[48] Sharma, P., & Tripathi, R. (2021). Security implications of multi-tenancy in cloud database systems. *Journal of Cloud Computing: Advances, Systems and Applications, 10*(1), 1–15. https://doi.org/10.1186/s13677-021-00233-1

[49] Shi, E., Stefanov, E., & Papadopoulos, S. (2020). Privacy-preserving queries over encrypted data using ORAM. *ACM Transactions on Privacy and Security*, 23(2), 1–30. https://doi.org/10.1145/3360325

[50] Singh, P., & Sharma, P. K. (2022). An efficient data classification and access control model for relational cloud databases. *Journal of Information Security and Applications*, *67*, 103182. https://doi.org/10.1016/j.jisa.2022.103182

[51] Song, W., Xu, H., & Xue, Y. (2020). Secure and scalable query processing for encrypted databases in cloud environments. *Future Generation Computer Systems*, 108, 899–909. https://doi.org/10.1016/j.future.2020.03.012

[52] Sun, J., Zhang, L., & Wang, Y. (2023). Scalable architectures for secure cloud-based database management: A review. *Future Generation Computer Systems, 142*, 299–314. https://doi.org/10.1016/j.future.2023.01.036

[53] Trusted CI. (2024). Secure Standard Aggregate Queries (SSAQ): Privacy-preserving aggregate range queries on encrypted multi-dimensional databases. *Trusted CI Newsletter*. https://www.trustedci.org/trusted-ci-pod/2024/11/18/november-2024-privacy-preserving-aggregate-range-queries-on-encrypted-multi-dimensional-databases

[54] Uhl, K., & Gollenia, L. (2022). Cloud Security Technical Reference Architecture. *Cybersecurity and Infrastructure Security Agency (CISA).* https://www.cisa.gov/sites/default/files/2023-02/cloud_security_technical_reference_architecture_2.pdf

[55] Wang, T., Chan, T., Raykova, M., Shinde, S., & Fonseca, R. (2020). ObliDB: Oblivious query processing for secure databases. *Proceedings of the VLDB Endowment*, 13(12), 2264–2277. https://people.eecs.berkeley.edu/~matei/papers/2020/vldb_oblidb.pdf

[56] Wang, T., Tang, Y., He, X., & Wang, D. (2021). Encrypted databases: Progresses, challenges and opportunities. *IEEE Transactions on Services Computing*. https://doi.org/10.1109/TSC.2021.3067860

[57] Wang, W., Wang, Z., & Zhang, C. (2023). Hybrid cryptographic technique for secure spatial range query processing in cloud computing. *Journal of Information Security and Applications*, 72, 103515. https://www.researchgate.net/publication/390438208

[58] Wang, Y., Jiang, L., He, L., Wang, Z., & Yu, L. (2022). Efficient multi-keyword ranked search over encrypted relational cloud databases. *Information Sciences, 607*, 129–144. https://doi.org/10.1016/j.ins.2022.05.043

[59] Williams, P., Sion, R., & Carbunar, B. (2021). Building castles out of mud: Practical access pattern privacy and ORAM in cloud storage. *Journal of Computer Security*, 29(1), 113–138. https://doi.org/10.3233/JCS-200055

[60] Wu, H., & Zhang, Y. (2023). Adaptive indexing and access pattern obfuscation in privacy-preserving relational databases. *Future Generation Computer Systems*, *140*, 89–104. https://doi.org/10.1016/j.future.2022.10.028

[61] Wu, Z., Huang, Y., & Xie, Y. (2021). A hybrid encryption scheme for secure cloud database services. *Future Generation Computer Systems*, *115*, 293–304. https://doi.org/10.1016/j.future.2020.09.020

[62] Xie, Q., Zhou, M., Zhang, Z., & Yang, Y. (2022). Enhancing data usability and security through optimized somewhat homomorphic encryption. *Future Generation Computer Systems*, *130*, 246–258. https://doi.org/10.1016/j.future.2022.01.023

[63] Xu, Y., Cheng, X., Zhao, K., & Chen, X. (2020). Achieving efficient and privacy-preserving range queries in cloud environments. *IEEE Transactions on Dependable and Secure Computing*, *19*(2), 948–963. https://doi.org/10.1109/TDSC.2020.2968796

[64] Yang, M., Lin, H., Zeng, Y., Chen, H., & Jia, X. (2023). SecuDB: A TEE-based privacy-preserving and tamper-resistant database system. *Proceedings of the VLDB Endowment*, 17(3), 3906–3920. https://www.vldb.org/pvldb/vol17/p3906-yang.pdf

[65] Yin, H., He, J., Yu, J., Zhang, S., & Zhang, X. (2021). Performance-aware secure query processing over encrypted databases in cloud. *IEEE Transactions on Services Computing, 14*(6), 1821–1835. https://doi.org/10.1109/TSC.2019.2929962

[66] Yuan, Y., Wang, Y., & Ren, K. (2020). Secure data management in the cloud: Challenges and research directions. *IEEE Network, 34*(3), 68–75. https://doi.org/10.1109/MNET.011.1900414

[67] Zhang, C., Xiong, H., & Chen, F. (2020). DeepSQL: A deep learning-based framework for encrypted database query optimization. *Concurrency and Computation: Practice and Experience*, 32(7), e7831. https://onlinelibrary.wiley.com/doi/full/10.1002/cpe.7831

[68] Zhang, H., Chen, R., & Wang, H. (2021). Secure data management in multi-tenant cloud databases. *IEEE Transactions on Cloud Computing, 9*(2), 457–468. https://doi.org/10.1109/TCC.2020.2972664

[69] Zhang, K., Liang, X., Lu, R., & Shen, X. (2020). Synergetic security mechanisms for data storage systems in cloud computing. *IEEE Network, 34*(1), 30–36. https://doi.org/10.1109/MNET.001.1900203

[70] Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., & Shen, X. S. (2020). Security and privacy in smart city applications: Challenges and solutions. *IEEE Communications Magazine, 58*(3), 122–129. https://doi.org/10.1109/MCOM.001.1900582

[71] Zhang, R., Liu, L., Xie, S., & Zhou, W. (2020). Privacy-preserving techniques for enabling big data analytics in the cloud: A review. *Information Sciences, 527*, 112–134. https://doi.org/10.1016/j.ins.2020.03.058

[72] Zhang, R., Liu, X., Xie, C., & Li, M. (2023). A lightweight dynamic SSE scheme for secure and efficient database querying in cloud environments. *Future Generation Computer Systems, 140*, 159–170. https://doi.org/10.1016/j.future.2022.10.005

[73] Zhang, Y., Han, L., Liu, X., & Jiang, Y. (2023). A performance-aware FHE scheme for secure cloud-based SQL queries. *Journal of Parallel and Distributed Computing, 178*, 46–57. https://doi.org/10.1016/j.jpdc.2023.04.004

[74] Zhang, Y., Liu, K., Zhou, Q., Wang, H., & Jin, H. (2022). Securing cloud databases with Intel SGX: Trends and challenges. *ACM Computing Surveys*, 54(12), 1–35. https://doi.org/10.1145/3466872

[75] Zhao, X., Chen, J., & Liu, Y. (2021). A performance-oriented framework for secure query processing in encrypted databases. *IEEE Transactions on Dependable and Secure Computing*, 18(2), 478–490. https://doi.org/10.1109/TDSC.2019.2909050

[76] Zhao, X., Chen, J., & Liu, Y. (2021). A performance-oriented framework for secure query processing in encrypted databases. *IEEE Transactions on Dependable and Secure Computing*, 18(2), 478–490. https://doi.org/10.1109/TDSC.2019.2909050