Enhancing Privacy in Databases by Data-Layer

Sami Alharbi[1], Samer Atawneh[2], Hussein Al Bazar[3], Roxane Elias Mallouhy[4]
College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia[1,2]
College of Engineering, Al Yamamah University, Khobar, Saudi Arabia[3,4]

***Abstract*—This study addresses the growing challenge of enhancing privacy in enterprise database systems, where excessive privileges and shared service accounts often lead to unauthorized data access and insider threats. The study proposes a data-layer security framework that enforces fine-grained access control based on authenticated user identities, integrating role-based access control (RBAC) and the principle of least privilege (PLP) to protect sensitive information. The model restricts developer and administrative access strictly to authorized data objects, reducing exposure while maintaining operational efficiency. Drawing on established database security mechanisms, including authentication, authorization, and centralized identity management through Active Directory, the proposed framework ensures that all database interactions are executed under verified user credentials. The approach is implemented using Microsoft SQL Server within an enterprise environment and evaluated through controlled experiments conducted before and after deployment. Results demonstrate a significant reduction in unauthorized data retrieval without introducing noticeable performance overhead. The findings confirm that enforcing privacy at the data-layer provides an effective and scalable solution for securing sensitive data in modern database systems, strengthening accountability and mitigating risks associated with privilege misuse.**

***Keywords*—Database privacy; security model; access control; data protection; privacy enhancing technologies; database systems**

I. Introduction

The importance of privacy in database systems has become increasingly critical in today's digital landscape. As databases accumulate and utilize vast amounts of information, they serve as central repositories of sensitive and personal data. Safeguarding this information is a fundamental priority, as it forms the backbone of trust, security, and ethical data management across organizations. Key strategies such as access control, encryption, and data anonymization represent the core of privacy protection mechanisms within database systems. Access control ensures that only authorized users can access specific data, effectively minimizing the risk of unauthorized breaches. Encryption further reinforces confidentiality and integrity by converting data into secure formats that can be accessed only by users possessing the correct decryption keys. Meanwhile, data anonymization techniques modify or remove identifying attributes to prevent individual recognition, thereby enhancing privacy and compliance with data protection standards.

These mechanisms collectively play a crucial role in mitigating risks associated with both external cyberattacks and internal misuse. The growing emphasis on privacy protection is also reflected in the establishment of rigorous international frameworks, including the General Data Protection Regulation (GDRP) [1] and the National Institute of Standards and Technology (NIST) Special Publication 800-53, Revision 5 [2]. These frameworks highlight the importance of maintaining confidentiality [3], integrity [4], and accountability of information systems [5]. They provide an updated set of guidelines that encourage organizations to strengthen access control, ensure data resilience, and align their technological operations with legal and ethical standards. The challenge, however, lies in balancing the implementation of these stringent privacy measures with the need to sustain high levels of system performance and efficiency [6]. By integrating the guidance of GDPR and NIST standards, organizations can better align their security imperatives with database functionality, achieving equilibrium between compliance, accessibility, and performance.

On the other hand, the vulnerabilities of database systems are often exposed through real-world cases of data breaches. For instance, Timothy Young, a resident of Moorefield, Nebraska, who confessed to a serious violation of trust and security by committing wire fraud that resulted in the exposure of confidential data from his employer, a New Jersey-based analytics and risk assessment firm. This organization served a global clientele that included insurance, financial, and governmental entities. Young accessed the company's network without authorization, obtaining names, email addresses, phone numbers, and login credentials, which he attempted to sell. His actions underscore the serious consequences of inadequate data protection and the risks that arise from unauthorized internal access to sensitive databases. The case, investigated by the FBI and prosecuted by the U.S. Attorney's Office Cybercrime Unit in Newark, resulted in potential charges carrying a maximum prison sentence of twenty years and substantial financial penalties [7]. This incident highlights the urgent need for effective database privacy frameworks and the severe legal implications of neglecting such protocols. In addition to isolated cases, the overall frequency and scale of data compromises have risen sharply across industries [8]. The financial services sector, for example, experienced a near doubling of data breaches in the United States, increasing from 268 reported incidents in 2022 to 744 in 2023. As shown in Fig. 1, this rise illustrates how sectors that manage sensitive financial information have become prime targets for cyber threats. Such incidents not only compromise consumer confidence but also threaten the stability and reputation of institutions that rely on secure data handling [9]. The upward trend in data violations underscores the growing sophistication of cyber threats and the pressing demand for enhanced privacy mechanisms within database systems [10].

Nevertheless, during everyday maintenance and operational processes, developers often interact directly with live databases, sometimes with broad or unintended access privileges. This exposure creates potential privacy risks,

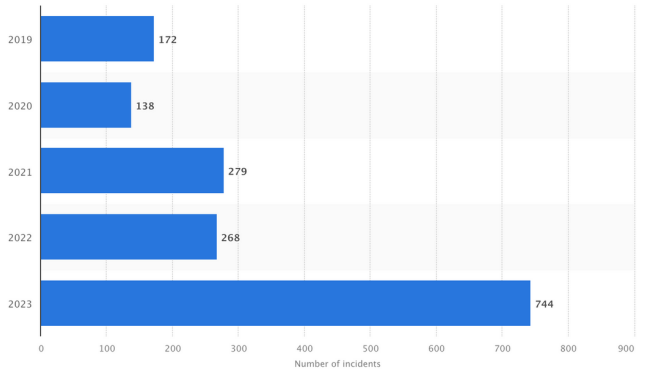

Fig. 1. Number of data violation cases in the United States (by statista [11]).

as unauthorized or excessive access may lead to data misuse or accidental disclosure of sensitive information. The challenge is to ensure that developers can perform their tasks efficiently while maintaining strict compliance with privacy and confidentiality requirements. Achieving a balance between operational flexibility and rigorous access control is essential to mitigating privacy risks and preserving database integrity [12].

The main objectives of this research are twofold. First, it seeks to enhance the protection of sensitive data within database systems by ensuring that users connect through properly authenticated and authorized accounts. Second, it aims to strengthen database privacy measures without compromising system performance. To achieve this, the study proposes a security model that integrates strict access controls, limiting developer access exclusively to the data necessary for their tasks. By reducing exposure and applying structured access rules, the model minimizes the risk of data leakage while maintaining efficient database operation. Consequently, the proposed work contributes to the broader field of database security and privacy engineering by offering a practical and scalable approach to safeguarding sensitive information at the data-layer. Its design emphasizes the balance between privacy protection and performance optimization, addressing one of the most persistent challenges in secure database management. By incorporating access control mechanisms guided by established privacy frameworks, this research provides a structured and adaptable solution that strengthens data protection while ensuring usability and efficiency in modern database environments. This work contributes to database security practice by introducing a data-layer access enforcement framework that explicitly binds database query execution to authenticated individual identities rather than shared service accounts. Unlike conventional role-based or attribute, based access control mechanisms that are typically enforced at the application or middleware layer, the proposed approach embeds identity, aware privilege enforcement directly at the database layer. The framework operationalizes the principle of least privilege through identity bound query execution and demonstrates its effectiveness through an enterprise scale implementation and empirical evaluation. In addition to the architectural contribution, the study provides experimental evidence that database layer identity enforcement can significantly reduce unauthorized access without introducing noticeable performance overhead.

Specifically, this study makes the following contributions:

- Proposes a database-layer security framework that enforces identity-bound access control, eliminating reliance on shared service accounts.
- Demonstrates fine-grained enforcement of the principle of least privilege directly within the database execution context.
- Provides an enterprise-scale implementation integrating centralized identity management with database access control.
- Empirically evaluates the impact of the proposed framework, showing improved privacy enforcement without measurable performance degradation.

The remainder of this study proceeds from foundations to validation and implications. Section II establishes the conceptual framework for database privacy by defining the two complementary pillars access control and data anonymization—that anchor the rest of the work. Building on these foundations, Section III surveys the state-of-the-art across encryption, access control, masking, intrusion detection, confidential data management, IoT access control, and biometrics/pseudonyms, as well as privacy-preserving publishing, thereby identifying concrete gaps and design requirements that motivate our solution. Guided by these requirements, Section IV details the proposed methods and materials, including the system architecture and implementation choices that operationalize the framework. Section V then evaluates the approach empirically, reporting and discussing experimental results that test whether the framework delivers the intended security and performance properties. Finally, Section VI synthesizes the findings, outlines limitations, and charts directions for future research.

II. Conceptual Framework of Database Privacy Mechanisms

Safeguarding sensitive data within database systems requires a combination of mechanisms that prevent unauthorized access and ensure the confidentiality of stored information. Two fundamental approaches dominate this domain: access control [13] and data anonymization [14]. Each provides a distinct, but complementary layer of protection that collectively strengthens database privacy.

A. Access Control Mechanisms

Access control serves as the first line of defense in protecting databases from cyber threats, including sophisticated attacks such as spear phishing. These systems implement a structured framework of identification, authentication, and authorization processes to determine precisely who can access which data resources. Their primary objective is to restrict unauthorized entry and uphold data confidentiality [15].

Different access control models have evolved to address varying security requirements and operational contexts. Discretionary Access Control (DAC) allows data owners to define permissions, offering flexibility, but risking oversharing. Mandatory Access Control (MAC) enforces system-level policies based on data classification and user clearance, ensuring high security, but limited flexibility. Role-Based Access

Control (RBAC) assigns permissions to roles rather than individual users, simplifying management in large organizations. Attribute-Based Access Control (ABAC) applies fine-grained policies based on user, resource, or environmental attributes, supporting dynamic decisions. Finally, Rule-Based Access Control supplements other models by enforcing predefined administrator rules. These models collectively enable organizations to balance flexibility and security based on their specific data governance requirements. Table I summarizes the unique strengths and weaknesses of these five access control types [15].

TABLE I. OVERVIEW OF ACCESS CONTROL SYSTEMS

Type	Description	Strengths	Weaknesses
DAC	User-defined access rights	High flexibility; user-centric	Risk of accidental oversharing
MAC	System-enforced policies based on classification	High security; minimizes breaches	Rigid; less user flexibility
RBAC	Access based on user roles	Simplifies management; scalable	Requires precise role definition
ABAC	Decisions based on user, environment, and resource attributes	Highly flexible and dynamic	Complex policy management
Rule-Based	Access determined by specific rules	Adds extra security layers	Management complexity

By effectively implementing these mechanisms, organizations can protect sensitive information from unauthorized access, thereby mitigating the risk of data breaches and ensuring compliance with data protection regulations such as the General Data Protection Regulation (GDPR). As database technologies evolve and the volume of data grows, the importance of sophisticated access control systems becomes even more critical, underlining the need for ongoing research and development in this area to address emerging security challenges.

B. Data Anonymization Techniques

Beyond controlling access, privacy protection also requires anonymizing data to prevent re-identification. Data anonymization modifies sensitive attributes so that individuals cannot be uniquely identified, preserving utility while protecting privacy [16]. These methodologies ingeniously modify personal data into a state where individual identification becomes impossible, thus shielding sensitive information from unauthorized scrutiny or exploitation. Anonymization assumes a critical role when the utilization of data extends to analytical, research, or reporting endeavors, ensuring that such activities do not infringe upon individual privacy rights.

Databases employ several anonymization strategies without significantly compromising the data's analytical utility. Among these, data masking can be applied in either static or dynamic forms, offering flexibility in secure data access and use while maintaining confidentiality. Pseudonymization replaces private identifiers with fictitious values, allowing data association with its source without revealing actual identities; however, it may be reversible with additional contextual information. Generalization decreases precision by converting detailed data (e.g., exact ages) into broader categories, while data perturbation introduces small alterations or noise to obscure original values yet preserve statistical integrity.

Structured models such as k-anonymity ensure that each record is indistinguishable from at least k–1 others regarding identifying attributes, while l-diversity demands diversity of sensitive attributes within anonymized groups. Building further, t-closeness maintains the distribution of sensitive attributes close to that of the entire dataset, preventing inference attacks. Table II summarizes key anonymization techniques and their advantages and disadvantages [16].

TABLE II. DATA ANONYMIZATION TECHNIQUES: PROS AND CONS

Technique	Pros	Cons
Data Masking	Protects sensitive data while keeping it functional for operations.	May distort actual values, limiting analytical use.
Pseudonymization	Enables linking data to its source without revealing identities, maintaining a level of utility.	Potentially reversible if attackers gain access to auxiliary information, compromising privacy.
Generalization	Simplifies implementation and enhances privacy by reducing data precision.	Reduces utility for detailed analysis due to loss of specificity.
Data Perturbation	Preserves overall structure and integrity of datasets for aggregate analysis.	Alterations may affect accuracy of individual records or analyses.
k-Anonymity	Provides a quantifiable measure of privacy by ensuring indistinguishability among records.	Vulnerable to linkage attacks if an attacker possesses external information.
l-Diversity	Enhances protection against attribute disclosure by ensuring diversity within groups.	Difficult to implement effectively in datasets with high similarity among records.
t-Closeness	Strengthens defense against inference attacks by aligning attribute distributions with the overall dataset.	Balancing privacy and data utility can be complex and computationally demanding.

Implementing these anonymization techniques effectively enables organizations to maintain compliance with data privacy regulations while still leveraging data for analytical purposes. As privacy concerns and regulatory requirements evolve, the development and refinement of anonymization methodologies remain a central focus for researchers and practitioners in data security and privacy management.

III. LITERATURE REVIEW

Data security and privacy remain central challenges in modern digital environments, particularly as the volume, variety, and sensitivity of stored information continue to rise across sectors such as mobile communications [17], healthcare [18], IoT systems [19], and database-backed web applications [20]. Across the literature, researchers have proposed diverse mechanisms, including encryption, access control, anonymization, intrusion detection, logging frameworks, and biometric authentication, to mitigate risks and strengthen confidentiality, integrity, and availability of data. The following review synthesizes these contributions in a coherent progression, moving from encryption-based protections to access control mechanisms, to data masking, intrusion detection, confidential data management, IoT access control, cryptography, biometric and pseudonym-based protections, and finally privacy-preserving data publishing.

A. Encryption and Cryptographic Techniques for Data Protection

A foundational component of secure data management lies in the use of strong encryption and cryptographic methods. ASCII-based encryption techniques leverage character values

to obfuscate readable data, reinforcing protection during transmission. The incorporation of multiple keys as in the Coloring Tripartite Graph (CTG) approach [21] further enhances resistance to unauthorized decryption. Similarly, the Triple Key Security Algorithm (TKS) employs polyalphabetic substitution with three keys and XOR operations, providing robust security particularly in mobile communication systems [22]. Building on the need for multi-key mechanisms, Ibrahim et al. [23] highlight vulnerabilities in databases storing personal and work related information. Their ASCII based, three-key encryption formula secures text and numeric data, preserves data size, and ensures efficient encryption and decryption, while acknowledging the necessity of future comparisons with established cryptographic systems to validate performance.

Additionally, cryptography is essential in domains such as medical services. Oduor & Omariba [24] trace the historical evolution of security practices and demonstrate the role of cryptographic algorithms in safeguarding sensitive healthcare data, supported by real time IDE examples. However, limitations such as incompatibility with indexed data and key management vulnerabilities persist. Further work in the Internet of Medical Things [25] calls for lightweight, quantum-resistant, and AI-enhanced cryptographic algorithms to meet emerging threats.

B. Access Control Models and Dynamic Data Protection

While encryption secures raw data, access control ensures that only authorized users can interact with sensitive resources. Wu et al. [26] note that database-backed web applications frequently allocate full privileges to application-level accounts, violating the principle of least privilege (PLP). Their PDA framework enforces PLP by intercepting queries and applying fine-grained, query-specific privilege restrictions, achieving resistance to SQL injection and buggy query manipulations with minimal performance overhead. Estrela framework [27] complements this approach by separating policy specification from application logic, enabling contextual and API-specific access enforcement. On top of this, dynamic Data Masking (DDM) further strengthens access control by obscuring sensitive values at query time. Fotache et al. [28] integrate masking directly into the persistence layer, preventing unmasked data from leaving the database engine. Their experimental evaluations using TPC-H show minimal performance impact, encouraging future research on large-scale databases, customized masking functions, and privilege-based masking algorithms. In big data environments, BDMasker [29] demonstrates scalable DDM deployment, with performance fluctuations maintained within 3% during horizontal and vertical expansion.

Access control challenges extend to confidential data management, where Shan et al. [30] argue that excessive privileges and underutilized logging hinder security. Their confidentiality-level access control model applies role, domain, and row level filtering, supported by an ELK-based logging module to improve traceability and monitoring. The MLCAC model [31] advances this further through real-time policy adjustment, log-driven decision-making, and decentralized optimization, achieving 89.55% accuracy in automated policy generation. Furthermore, in the IoT environment, Jiang et al. [32] address limitations in auditability and privilege control by proposing CcBAC, a blockchain-based fine-grained access control model incorporating TEE and cryptocurrency-based authorization. Experimental findings and further elaboration by Wang [33] demonstrate strong performance under large-scale request loads, high auditability, and consistent authorization across distributed IoT systems.

In addition to access control and masking, preserving privacy during data publishing forms another essential aspect of dynamic data protection. Jayapradha et al. [34] highlight the limitations of traditional anonymization techniques, especially when handling multiple correlated sensitive attributes. Their Heap Bucketization-Anonymity (HBA) model integrates anatomization, k-anonymity, slicing, and heap bucketization to balance privacy and analytical utility. Using metrics such as the Normalized Certainty Penalty and KL-divergence, they show that HBA effectively mitigates background knowledge, membership, non-membership, quasi-identifier, and fingerprint correlation attacks. The authors also emphasize future directions, including support for dynamic and unstructured data as well as addressing 1:M microdata challenges, underscoring the evolving role of anonymization in comprehensive data protection systems.

C. Intrusion Detection and Biometric Protection Against Insider and External Threats

Even with strong encryption and access control, insider misuse remains a dominant cause of data breaches. Said & Mostafa [35] identify privileged account misuse as a critical vulnerability and propose an intrusion detection system based on Danger Theory and the Negative Selection Algorithm (NSA). By learning from previously detected intrusions, their hybrid system improves anomaly detection accuracy while reducing false positives and false negatives. Nonetheless, the authors emphasize the challenge of calibrating the hybrid immune model and maintaining system performance in cloud and mobile networks. NSA's strengths, and its limitations in scalability are further examined by Tosin & Gbenga [36], who note its adaptability but highlight computational concerns.

Into the bargain, biometric authentication adds another layer of defense, particularly for securing identity and preventing unauthorized database access. Abd Razak, Nazari, & Al-Dhaqm [37] demonstrate that unique identifiers stored in digital databases can expose users to eavesdropping and identity theft. They propose integrating palm vein recognition with pseudonym generation to anonymize records and reinforce privacy. Enhancing pseudonym generation remains a priority for improving protection, especially in cloud computing contexts. Complementary work [38] uses palm vein biohashing with near-infrared scanning, achieving an EER of 0 and protecting biometric templates from exposure during storage or transfer.

To provide a holistic comparison of the reviewed approaches, Table III summarizes the methodologies, benefits, and limitations of the most relevant studies discussed in this section.

D. Identified Research Gaps and Motivation

The reviewed literature demonstrates significant progress in database security through encryption techniques, access

TABLE III. COMPARATIVE SUMMARY OF RELATED WORK

Author & Year	Methodology	Benefits	Limitations
[21]	Graph-based encryption using tripartite graph coloring with ASCII mapping	Enhanced mathematical security; Suitable for text-based protection; Resistant to brute-force attacks	Limited to ASCII; High computational complexity; Scalability concerns
[22]	Multi-round encryption using three independent keys	Protection against single-key attacks; Multiple encryption layers; Improved key flexibility	Higher computational load; Key synchronization challenges; Performance degradation over multiple rounds
[23]	Custom ASCII-based encryption algorithm with novel primitives for database security	Database-specific optimization; Protects structured data; Prevents unauthorized access	Non-standard; Limited cryptanalysis; Integration challenges with existing systems
[24]	Applied encryption, hashing, and digital signatures for medical data protection	HIPAA/GDPR support; Secure transmission; Confidentiality for medical records	Real-time performance issues; Healthcare key management complexity; Interoperability limits
[25]	Systematic review of symmetric, asymmetric, and lightweight cryptography for IoMT	Comprehensive IoMT landscape; Identifies effective schemes; Highlights attack vectors and mitigations	Review-based; No empirical validation; Resource constraints in IoT not fully explored
[26]	Application-driven privilege separation (PDA) using query interception	Automated least privilege enforcement; Prevents SQL injection; Low runtime overhead (8.13%)	Application modification required; Tested on limited systems; Generalizability concerns
[27]	Estrela contextual policy framework with API-level enforcement	Context-aware control; Works on legacy systems without DB changes; Fine-grained policies	Complex rules; Requires API–policy mapping; Limited application testing
[28]	TPC-H benchmark and ML-based performance analysis of dynamic data masking (DDM)	Minimal masking overhead; Effective for datasets up to 100GB; Data-driven masking strategy design	Oracle-only evaluation; Basic masking scenarios; Scalability beyond 100GB uncertain
[29]	SQL query rewriting with multi-engine unified security for DDM	Zero logic impact; Multi-engine scalability; 3% overhead; Preserves query intent	Complex rewriting; Limited encryption integration; End-to-end protection not covered
[30]	Hybrid RBAC/ABAC confidential data access control model	Flexible, granular policies; Supports separation of duties; Scalable for organizations	Role explosion; Attribute maintenance overhead; Susceptible to misconfiguration
[31]	Machine-learning context-aware access control (MLCAC) for insider threat detection	Behavioral anomaly detection; Adaptive authorization; Real-time mitigation	Requires large training datasets; False positives; Intensive computation
[32]	CcBAC hybrid IoT access control using blockchain audit logs + TEE enforcement	Tamper-proof logs; Secure execution; Suitable for distributed IoT environments	Blockchain scalability issues; TEE hardware dependency; Complex integration
[33]	Dynamic trust-based access control for IoT network boundaries	Granular trust evaluation; Adaptive policies; Prevents unauthorized IoT access	Trust metric definition; Continuous evaluation overhead; Scalability challenges
[34]	Heap Bucketization-Anonymity (HBA) combining anatomization, slicing, and k-anonymity	Resists background knowledge, membership, and fingerprint attacks; Balances privacy and utility	Not real-time; No live masking; Complex parameter tuning
[35]	Danger Theory + Negative Selection intrusion detection for insider threats	Self-learning; High detection coverage; Strong against privilege misuse; Adaptive behavior	Detection-only; Training required; Vulnerable to evolving attack patterns
[36]	Bio-inspired negative selection algorithm (NSA) for intrusion detection	Detects unknown attacks; Adaptable to new patterns; Low computational cost	High false positives; Detector generation complexity; No prevention capability
[37]	Palm-vein biometrics with pseudonym generation for identity anonymization	Strong authentication; Identity unlinkability; Dual-layer protection (biometric + pseudonym)	Biometric sensitivity concerns; Integration complexity; Limited analysis of encryption interactions
[38]	Vascular biometric recognition with enhanced liveness detection	Highly secure modality; Non-invasive capture; Suitable for high-security systems	Requires specialized hardware; Environmental sensitivity; Higher deployment cost

control models, anonymization strategies, and intrusion detection mechanisms. However, several limitations remain insufficiently addressed. Many existing approaches rely heavily on encryption or anonymization, which protect data content but do not prevent misuse by legitimately authenticated users with excessive privileges. Similarly, application-layer access control frameworks enforce authorization outside the database engine, leaving database-layer interactions vulnerable when shared service accounts or over-privileged credentials are employed.

Furthermore, while role-based and attribute-based access control models provide structured authorization, they are often implemented at the application or middleware level, limiting their effectiveness in mitigating insider threats and privilege escalation within the database itself. Existing solutions also lack direct enforcement of identity-bound query execution, where each database operation is explicitly tied to a verified individual account rather than a generic service identity. These gaps highlight the need for a database-layer privacy enforcement mechanism that integrates centralized identity management, enforces the principle of least privilege, and ensures that all database interactions are executed under authenticated and authorized user identities. Motivated by these limitations, the proposed approach introduces a data-layer security framework that directly couples Active Directory–based authentication with database access control, effectively reducing unauthorized data exposure while maintaining operational efficiency.

IV. METHODS AND MATERIALS

The methodology adopted in this research offers a clear and systematic explanation of the technical approach, detailing the phased execution workflow and the environment in which the security model was developed and evaluated. It maps the full operational cycle, from request initiation and identity verification to permission checking and final query execution, demonstrating how each stage contributes to enforcing strict access controls. The methodology is structured around three integrated phases: Request Initialization, Verification Process, and Execution and Response, as displayed in Fig. 2. Together, these phases represent the full lifecycle of a database interaction within a secure, access-controlled environment. The process begins with Query Initiation, where the developer issues a request to the database using SQL instructions intended to retrieve, insert, update, or delete data. This step forms the operational entry point, defining the intended action and the credentials through which it should be executed. Next, the system performs Identity Account Retrieval, extracting the account information from the domain controller to confirm the identity of the request initiator. This ensures that subsequent evaluation stages rely on verified and authenticated identity attributes.

Afterwards, a critical step in the workflow is Permission Verification, during which the system checks whether the authenticated user or service account is authorized to perform the requested action. This validation is executed by comparing the provided credentials to the records stored in the Active Directory, confirming exact matches before any database operation is permitted. Finally, the methodology concludes with Query Execution Based on Access Rights, where authorized SQL commands are securely executed. The database engine processes the approved instruction and returns the corresponding output. This structured cycle ensures that access is governed by strictly verified permissions, thereby strengthening privacy and reducing the risk of unauthorized operations.

To develop and evaluate the proposed security model, the study utilizes a comprehensive enterprise-grade technology stack that integrates development, authentication, and virtualization components in a structured manner. The implemen-

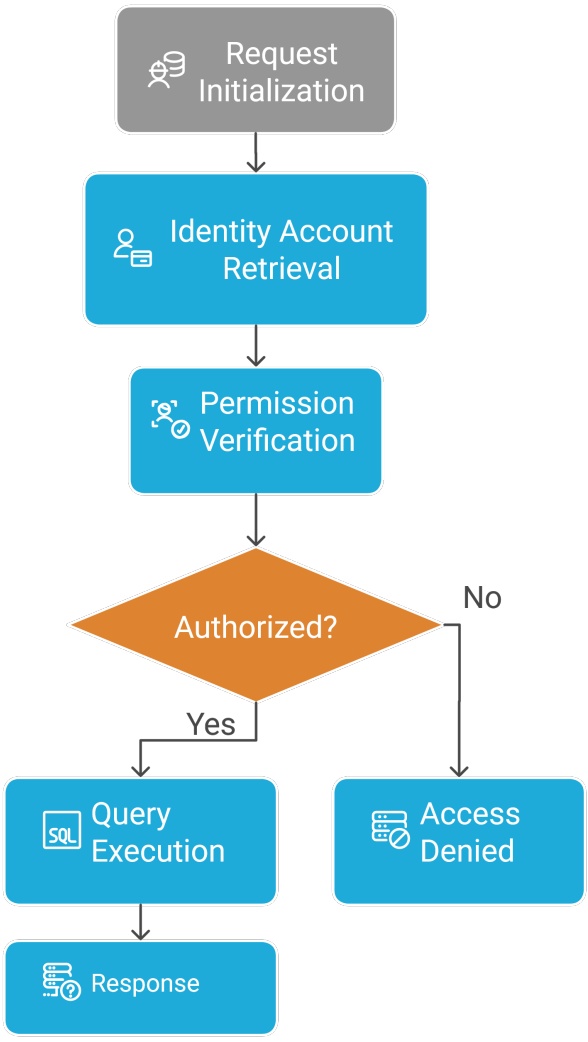

Fig. 2. Database access control flowchart.

tation begins with ASP.NET and C—supported by Visual Studio, as the core development framework used to build the access-control mechanisms and authentication logic. Microsoft SQL Server serves as the backend database system, providing scalable and secure data storage essential for testing various access scenarios. To support identity and permission management, Windows Server and its integrated domain controller constitute the backbone of authentication, ensuring centralized administration of user accounts and privilege verification. These components operate within a controlled virtualized environment created using VMware ESXi 6.7.0, hosted on a Dell PowerEdge R940 server equipped with an Intel(R) Xeon(R) Gold 5115 CPU, 80 logical processors, and multiple NICs, providing the computational capacity required for parallel and large-scale testing. This setup hosts 39 virtual machines, each configured to replicate real-world operational conditions in a reproducible manner. Active Directory operates on the DC01 server running Windows Server 2019 Standard Edition, with a functional level of Windows Server 2012 R2 to ensure compatibility with modern security protocols. And finally, firewall settings were carefully controlled to avoid external interference while preserving the integrity of testing procedures.

Consequently, the implementation of the proposed security model follows the methodology previously outlined and focuses on enforcing strict access controls and integrating authentication mechanisms within a Microsoft SQL Server environment using ASP.NET and C. The process begins by configuring user access requests and verifying the security policies that govern authentication and authorization within a Windows Server domain. This implementation is tested through controlled scenarios that observe system responses to both authorized and unauthorized attempts, ensuring that the enhanced security measures operate effectively without negatively impacting database performance. Hence, to support this implementation, the environment is designed as an interconnected framework in which the application, database, and Active Directory operate seamlessly to manage identity, permissions, and data flow. As illustrated in Fig. 3, this architecture enables continuous communication between components, allowing authentication and authorization checks to be performed before any database interaction occurs. This coordinated structure ensures that every user request is validated and processed according to predefined security rules, forming the operational backbone of the model.

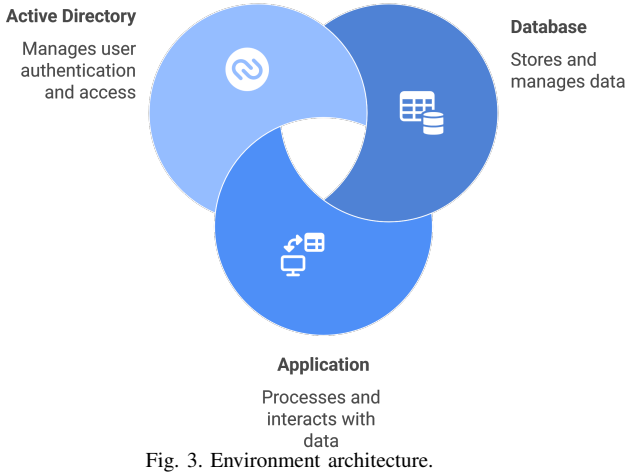

Fig. 3. Environment architecture.

Within this environment, Microsoft SQL Server 2016 hosts the CyberSecurity_CapstoneProject schema, which includes three essential tables. The Address table stores location data such as Building_No, StreetAddress, and FloorNo; the ContactInfo table contains communication fields including PhoneNumber, Email, POBOX, and SocialMediaAccount; and the EmployeeInfo table holds personal and professional information: FirstName, SecondName, FamilyName, DOB, PostalAddress, Occupation, CompanyName, Status, and associated audit fields like CreatedDate, CreatedBy, ModifiedDate, and ModifiedBy. These tables are interconnected to provide a complete representation of employee records, as shown in Fig. 4. Nevertheless, the SQL Server environment is organized in a standard hierarchical structure, where the schema and its tables appear under the database's "Tables" directory, alongside system tables, external resources, programmability components, and security configurations

The operational workflow begins with the initiation of SQL queries, where the developer formulates the required command and embeds the necessary account credentials within the connection string. The query prepared with its associated

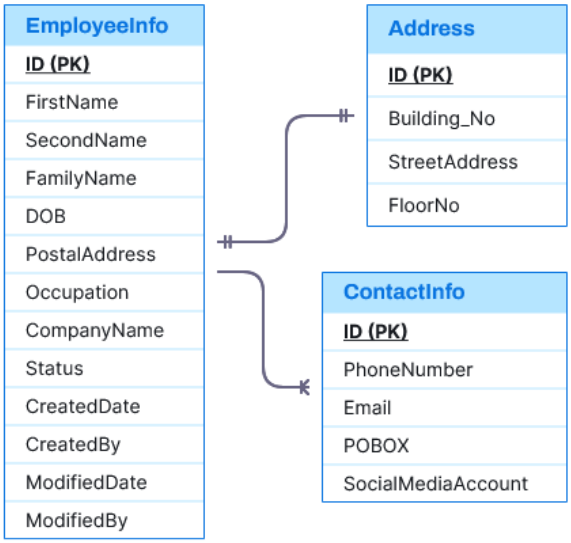

Fig. 4. Database ERD diagram.

identity information, is submitted to the system for execution:

```
string sQuery;
string sConnectionString =
    "Data Source=RACELDRT\\SQL2016;Initial
    ↪  Catalog=SANG_Weather_Prayer;"
    + "User ID=SQLTest;Password=123456789;";
sQuery = "SELECT [ID], [Building_No], [StreetAddress],
↪  [FloorNo] "
       + "FROM
       ↪  [CyberSecurity_CapstoneProject].[dbo].[Address];";
```

A complementary user interface, shown in Fig. 5, allows queries to be executed interactively, providing transparency during testing and making it possible to observe how the system handles different access attempts. And, after a query is issued, the system retrieves the corresponding account identity from Active Directory using C. This retrieval ensures that each request is tied to a verified domain identity before any authorization checks occur. Once the account details are obtained, they are used to determine the user's access level and validate whether the requested action aligns with the assigned permissions.

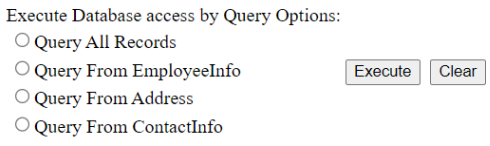

Fig. 5. Query options to execute on the database.

The subsequent stage focuses on permission verification, where the submitted credentials are compared with those stored in Active Directory. This comparison is implemented through C logic that ensures the credentials provided by the user exactly match the corresponding domain records before access is approved. Only when the account information fully aligns with the stored identity data does the system permit the request to proceed. This verification step strengthens the overall security posture by preventing unauthorized users from executing database commands.

Following successful authentication and authorization, the system proceeds to execute the SQL command using the verified account's privileges. A secure connection is established, after which the authorized operation is processed by the SQL Server engine. Whether the requested action involves retrieving, inserting, updating, or deleting data, the database completes the command and returns the corresponding response to the requester. By ensuring that all queries are executed strictly under validated permissions, the implementation guarantees that database interactions comply with the intended security controls and that unauthorized activity is effectively mitigated.

V. Results and Discussion

This section presents the experimental findings of the proposed access-control framework developed to strengthen database security. The implementation was conducted using Microsoft SQL Server integrated with Active Directory to enforce role-based authentication and manage user privileges effectively. The evaluation examined how the system mitigates unauthorized data access, preserves information confidentiality, and sustains operational efficiency under various workloads. The analysis focused on three core dimensions: the precision of access-control enforcement, the performance of query execution under restricted conditions, and the resilience of the database against security breaches. The discussion also interprets the practical implications of these findings for enterprise database protection and outlines areas where further optimization can enhance reliability and scalability in future implementations.

A. Privileges and User Accounts

The evaluation involved five user accounts: smharbi1, Ssmutawa, Salroumi, amathel, and maabkhet, representing different operational roles within the organization. Each user was assigned a specific level of access to three core database tables: EmployeeInfo, ContactInfo, and Address. These privileges were configured and verified within Microsoft SQL Server using Active Directory, based authentication to ensure that access rights were directly tied to domain-level credentials. Prior to applying the proposed access-control model, all user accounts were granted unrestricted access to the database through a shared service configuration. This approach violated the principle of least privilege and exposed sensitive information to unauthorized viewing and modification. After enforcing the model, each user's permissions were refined to reflect their functional responsibilities.

Administrative users retained full access to all datasets, while standard users were limited to only the tables necessary for their operational tasks. The analysis of user privileges revealed that the model successfully differentiated roles and applied restrictions accurately. For instance, smharbi1 maintained complete administrative rights, whereas Ssmutawa and amathel were limited to access specific data categories relevant to their roles. Salroumi and maabkhet had partial access aligned with departmental boundaries, demonstrating the granularity of the implemented policy. The overall distribution of user privileges is summarized in Table IV, which consolidates the effective permissions of each account across the three main database tables. The results confirm that the proposed model enforces precise and hierarchical access, reducing redundant permissions and minimizing the risk of unauthorized data exposure.

TABLE IV. USER ACCOUNT ACCESS TO DATABASE TABLES

User Account	EmployeeInfo	ContactInfo	Address
smharbi1	Yes	Yes	Yes
Ssmutawa			Yes
Salroumi		Yes	Yes
amathel		Yes	
maabkhet	Yes		

B. Comparative Evaluation of Access Control Before and After Implementation

The evaluation of this work, aimed to measure the effectiveness of the new mechanism in strengthening privilege enforcement, minimizing unauthorized access, and sustaining operational efficiency. The analysis focused on user interactions with the EmployeeInfo, Address, and ContactInfo tables to identify improvements in access accuracy and adherence to security policies. Before deploying the access-control model, all database users were authenticated through a shared service account (SQLTest), which bypassed role-based verification. Consequently, every account had unrestricted access to all data tables (EmployeeInfo, ContactInfo, and Address), regardless of job role. This violated the least-privilege principle and exposed sensitive information to potential misuse.

To illustrate this issue, a standard user was able to execute a query on the EmployeeInfo table and retrieve all records successfully. This behavior clearly reflects the absence of proper privilege enforcement, as the system made no distinction between administrator privileges and regular-user permissions. Such unrestricted access not only increased the risk of accidental data modification but also eliminated any meaningful user accountability. After applying the access-control model, authentication and authorization were tightly coupled through Active Directory and Microsoft SQL Server. Each query request was verified against the domain credentials of the requesting user before execution, ensuring that only authorized users could access their assigned tables. As shown in Fig. 6, the user Ssmutawa was permitted to access only the Address table, while Fig. 7 demonstrates that Salroumi could view records exclusively from the ContactInfo table. Attempts by these users to query unauthorized tables triggered explicit denial messages, confirming that privilege restrictions were properly enforced.

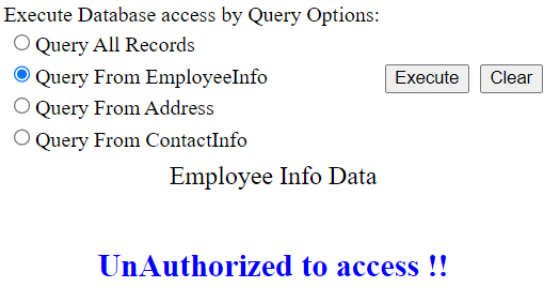

Fig. 6. Authorized access – Ssmutawa (Address table, Post-Implementation).

The evaluation confirmed that all privilege assignments were applied accurately and that no unauthorized user could access restricted information. Query execution remained efficient, indicating that the additional verification steps did not introduce noticeable performance overhead.

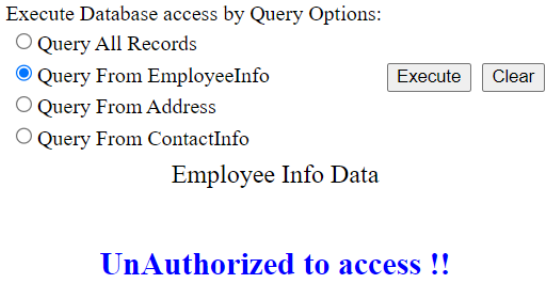

Fig. 7. Authorized access – Salroumi (ContactInfo table, Post-Implementation).

TABLE V. USER ACCESS PERMISSIONS ACROSS DATABASE TABLES

User Account	Emp-Info	ContactInfo	Address
smharbi1 (Admin)	Access	Access	Access
Ssmutawa	Denied	Denied	Access
Salroumi	Denied	Access	Access
amathel	Denied	Access	Denied
maabkhet	Access	Denied	Denied

C. Discussion

The comparative results reveal a complete transformation in database behavior following the model's deployment. Under the pre-implementation setup, every user had full access to all data. Post-implementation, access rights were strictly constrained according to defined roles. Administrative accounts retained full control, while standard users were limited to specific datasets required for their daily operations. This refinement achieved true least-privilege enforcement, improved accountability, and prevented privilege escalation. Furthermore, the integration with Active Directory ensured traceability, each query could be linked to a verified user identity, strengthening audit capabilities. The resulting user privileges after model implementation are summarized in Table V, which consolidates effective permissions across the main database tables.

To contextualize these improvements, it is important to compare them directly with the conditions observed in Experiment One. In Experiment One, all user accounts (smharbi1, Ssmutawa, Salroumi, amathel, maabkhet) were able to retrieve data from the EmployeeInfo, ContactInfo, and Address tables. This unrestricted access was attributed to the use of the service account (SQLTest), which possessed elevated privileges. Consequently, this setup allowed all user accounts to bypass access control restrictions, thereby compromising the principle of least privilege (PLP) and exposing the database to potential security threats. On the other hand, experiment Two demonstrated a marked improvement in access control and data security. The newly implemented access control model was designed to enforce specific access rights tailored to each user account based on their roles and responsibilities within the organization. This granular approach to access control ensured that only authorized user accounts could retrieve data from the designated tables, thereby mitigating the risk of unauthorized

TABLE VI. COMPARISON OF USER ACCESS BEFORE AND AFTER MODEL IMPLEMENTATION

User Account	Emp Info (Before)	Emp Info (After)	Contact Info (Before)	Contact Info (After)	Address (Before)	Address (After)
smharbi1	Yes	Yes	Yes	Yes	Yes	Yes
Ssmutawa	Yes	No	Yes	No	Yes	Yes
Salroumi	Yes	Yes	Yes	Yes	Yes	Yes
amathel	Yes	No	Yes	Yes	Yes	No
maabkhet	Yes	Yes	Yes	No	Yes	No

data access. For example, the account smharbi1 retained full access to all tables due to the comprehensive permissions associated with the user's role. On the other hand, accounts such as Ssmutawa and Salroumi faced restricted access, preventing them from retrieving data from certain tables, thereby enhancing the overall security of the database system. The differences in access capabilities before and after implementing the access control model are illustrated in Table VI. These visuals depict the restricted access observed in Experiment Two, showcasing the model's effectiveness in enforcing data security protocols and preventing unauthorized access.

These findings underscore the effectiveness of the new access control model in significantly enhancing database security by strictly enforcing user-specific access privileges and preventing unauthorized data access. The systematic comparison of the two experiments highlights the critical role of robust access control mechanisms in safeguarding sensitive information within database environments. When compared to the works of Wu et al. [26] and Shan et al. [30], the results of this study align with the growing emphasis on enforcing the principle of least privilege at the data source level. While Wu et al.'s PDA framework focused on privilege separation through application-driven enforcement, the model in this research extends the concept by embedding identity verification directly at the database layer, combining role-based control with domain authentication. Similarly, Shan et al. emphasized multi-layer filtering and the integration of logging for confidentiality assurance.

The proposed model complements this by providing traceable and verifiable access through Active Directory integration, strengthening accountability for every query execution. Furthermore, unlike the encryption-based approach by Ibrahim et al. [23], which primarily addressed data confidentiality through encoding mechanisms, this work enhances security through behavioral access enforcement restricting who can interact with the data in real time. The combined analysis across both experiments demonstrates that implementing access verification at the database layer provides a more scalable and flexible solution compared to pure encryption or masking techniques discussed in related studies. Overall, the proposed model advances the state of access control research by bridging theoretical models with practical enterprise deployment. It achieves fine-grained privilege management, aligns with established security frameworks, and complements existing literature by demonstrating that database-layer access verification can achieve both operational efficiency and strong data protection.

VI. CONCLUSION

This research addressed the critical challenge of strengthening privacy and access control in enterprise database systems, particularly in operational environments where shared service accounts and over-privileged credentials expose sensitive information to unauthorized access. Such configurations undermine accountability and violate the principle of least privilege, increasing the risk of insider misuse and accidental data disclosure. To address these issues, this study proposed a data-layer security framework that enforces fine-grained access control by binding every database operation directly to authenticated user identities. The proposed framework integrates centralized identity management with database access enforcement to ensure that all SQL operations are executed under verified and authorized user credentials. By eliminating reliance on generic service accounts and enforcing role-based permissions at the data-layer, the model provides stronger control over who can access specific database objects. The implementation demonstrated how authentication and authorization mechanisms can be tightly coupled with database operations to enhance privacy without disrupting normal development or administrative workflows.

The framework was implemented in an enterprise-grade environment using Microsoft SQL Server integrated with centralized identity services and evaluated through controlled experiments conducted before and after deployment. Experimental results clearly demonstrated a significant reduction in unauthorized data access, improved enforcement of user-specific privileges, and enhanced accountability across database operations. Importantly, these security improvements were achieved without introducing noticeable performance overhead, indicating that stronger privacy controls can be implemented while maintaining system efficiency and usability. Despite the effectiveness of the proposed approach, certain limitations were identified. The framework depends on centralized identity infrastructure, which may introduce single points of dependency in some deployment scenarios. Additionally, integrating the model into legacy database environments may require additional configuration and administrative effort. Addressing these challenges are essential to ensure broader applicability and long-term maintainability.

Future work will focus on extending the framework to improve resilience and adaptability in dynamic enterprise environments. Potential directions include introducing redundancy mechanisms for identity services, developing automated tools for privilege management, and exploring adaptive access control strategies that adjust permissions based on user behavior and operational context. These enhancements aim to further strengthen database privacy, scalability, and robustness in the face of evolving security threats. Collectively, this study establishes that enforcing access control at the database layer through identity-bound execution is a practical and effective mechanism for improving privacy, accountability, and compliance with the principle of least privilege in enterprise database systems.

REFERENCES

[1] European Parliament and Council of the European Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing

of personal data and on the free movement of such data (general data protection regulation)," Official Journal of the European Union, L119, 1–88, 2016, accessed: November 2025. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[2] J. T. Force, "Security and privacy controls for information systems and organizations," National Institute of Standards and Technology, Tech. Rep., 2020.

[3] P. Shojaei, E. Vlahu-Gjorgievska, and Y.-W. Chow, "Security and privacy of technologies in health information systems: A systematic literature review," *Computers*, vol. 13, no. 2, p. 41, 2024.

[4] M. S. Nasir, H. Khan, A. Qureshi, A. Rafiq, and T. Rasheed, "Ethical aspects in cyber security maintaining data integrity and protection: A review," *Spectrum of engineering sciences*, vol. 2, no. 3, pp. 420–454, 2024.

[5] R. Purnamasari, A. I. Hasanudin, R. Zulfikar, and H. Yazid, "Do internal control and information systems drive sustainable rural development in indonesia?" *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 1, p. 100242, 2024.

[6] E. Pina, J. Ramos, H. Jorge, P. Váz, J. Silva, C. Wanzeller, M. Abbasi, and P. Martins, "Data privacy and ethical considerations in database management," *Journal of Cybersecurity and Privacy*, vol. 4, no. 3, pp. 494–517, 2024.

[7] U.S. Department of Justice. (2020, June) Nebraska man admits stealing and selling his employer's confidential information. Press release. Accessed: November 2025. [Online]. Available: https://www.justice.gov/usao-nj/pr/nebraska-man-admits-stealing-and-selling-his-employer-s-confidential-information

[8] A. S. George, T. Baskar, and P. B. Srikaanth, "Cyber threats to critical infrastructure: assessing vulnerabilities across key sectors," *Partners Universal International Innovation Journal*, vol. 2, no. 1, pp. 51–75, 2024.

[9] X. Wu and W. Bao, "Research on the design of a blockchain logistics information platform based on reputation proof consensus algorithm," *Procedia Computer Science*, vol. 262, pp. 973–981, 2025.

[10] Q. Razi, R. Piyush, A. Chakrabarti, A. Singh, V. Hassija, and G. Chalapathi, "Enhancing data privacy: A comprehensive survey of privacy-enabling technologies," *IEEE Access*, 2025.

[11] Statista Research Department. (2025) Number of data loss incidents in the u.s. financial sector. Accessed: November 2025. [Online]. Available: https://www.statista.com/statistics/1318486/us-number-of-data-loss-incidents-in-financial-sector/

[12] R. J. Ramniklal, "Database security and integrity: Ensuring reliable and secure data management," *Mosaic of Ideas: Multidisciplinary Reflections*, vol. 73, 2024.

[13] D. Sargiotis, "Data security and privacy: Protecting sensitive information," in *Data governance: a guide*. Springer, 2024, pp. 217–245.

[14] A. Gadotti, L. Rocher, F. Houssiau, A.-M. Creţu, and Y.-A. De Montjoye, "Anonymization: The imperfect science of using data while preserving privacy," *Science advances*, vol. 10, no. 29, p. eadn7053, 2024.

[15] N. Kashmar, M. Adda, M. Atieh, and H. Ibrahim, "A review of access control metamodels," *Procedia Computer Science*, vol. 184, pp. 445–452, 2021.

[16] Z. El Ouazzani and H. El Bakkali, "A classification of non-cryptographic anonymization techniques ensuring privacy in big data," *International Journal of Communication Networks and Information Security*, vol. 12, no. 1, pp. 142–152, 2020.

[17] Ş. Erdal, F. Karakoç, and E. Özdemir, "A survey on security and privacy aspects and solutions for federated learning in mobile communication networks," *ITU Journal of Wireless Communications and Cybersecurity*, vol. 1, no. 1, pp. 29–40, 2024.

[18] O. Popoola, M. Rodrigues, J. Marchang, A. Shenfield, A. Ikpehai, and J. Popoola, "A critical literature review of security and privacy in smart home healthcare schemes adopting iot & blockchain: problems, challenges and solutions," *Blockchain: Research and Applications*, vol. 5, no. 2, p. 100178, 2024.

[19] S. Rani, M. Shabaz, A. K. Dutta, E. A. Ahmed *et al.*, "Enhancing privacy and security in iot-based smart grid system using encryption-based fog computing," *Alexandria engineering journal*, vol. 102, pp. 66–74, 2024.

[20] Y. Huang, C. Shi, J. Lu, H. Li, H. Meng, and L. Li, "Detecting broken object-level authorization vulnerabilities in database-backed applications," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 2934–2948.

[21] S. NAGARAJAN, "An ascii value based data encryption using coloring tripartite graph," *CONTEMPORARY MATHEMATICS*, pp. 2113–2130, 2025.

[22] M. Akram, M. W. Iqbal, S. A. Ali, M. U. Ashraf, K. Alsubhi, and H. M. Aljahdali, "Triple key security algorithm against single key attack on multiple rounds." *Computers, Materials & Continua*, vol. 72, no. 3, 2022.

[23] S. Ibrahım, A. Zengin, S. Hızal, A. S. Akhter, and C. Altunkaya, "A novel data encryption algorithm to ensure database security," *Acta Infologica*, vol. 7, no. 1, pp. 1–16, 2023.

[24] X. F. Oduor and Z. B. Omariba, "Application of cryptography in enhancing privacy of personal data in medical services," *Int J Commun Inf Technol*, vol. 3, no. 1, pp. 16–21, 2022.

[25] W. Robert, A. Denis, A. Thomas, A. Samuel, S. P. Kabiito, Z. Morish, and G. Ali, "A comprehensive review on cryptographic techniques for securing internet of medical things: A state-of-the-art, applications, security attacks, mitigation measures, and future research direction," *Mesopotamian Journal of Artificial Intelligence in Healthcare*, vol. 2024, pp. 135–169, 2024.

[26] H. Wu, Z. Yu, D. Huang, H. Zhang, and W. Han, "Automated enforcement of the principle of least privilege over data source access," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 510–517.

[27] A. Bichhawat, M. Fredrikson, J. Yang, and A. Trehan, "Contextual and granular policy enforcement in database-backed applications," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020, pp. 432–444.

[28] M. Fotache, A. Munteanu, C. Strîmbei, and I. Hrubaru, "Framework for the assessment of data masking performance penalties in sql database servers. case study: Oracle," *IEEE Access*, vol. 11, pp. 18 520–18 541, 2023.

[29] Y. Tu, J. Niu, D. Wang, H. Gao, J. Xu, and K. Hong, "Bdmasker: Dynamic data protection system for open big data environment." *International Journal of Software & Informatics*, vol. 13, no. 1, 2023.

[30] L. Shan, H. Zhou, D. Hong, Q. Dong, Y. Wang, and S. Song, "Application of access control model for confidential data," *Procedia Computer Science*, vol. 192, pp. 3865–3874, 2021.

[31] L. Xiao, A. Yu, H. Wang, L. Zhao, and D. Meng, "Mlcac: Dynamic authorization and intelligent decision-making towards insider threats," in *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2024, pp. 407–412.

[32] W. Jiang, E. Li, W. Zhou, Y. Yang, and T. Luo, "Iot access control model based on blockchain and trusted execution environment," *Processes*, vol. 11, no. 3, p. 723, 2023.

[33] J. Wang, C. Liu, G. Zhu, X. Liu, and B. Xiao, "Fine-grained trusted control methods for iot boundary access," *Scalable Computing: Practice and Experience*, vol. 26, no. 1, pp. 180–190, 2025.

[34] "Heap bucketization anonymity—an efficient privacy-preserving data publishing model for multiple sensitive attributes," *IEEE Access*, vol. 10, pp. 28 773 – 28 791, 2022. [Online]. Available: https://ieeexplore.ieee.org/ielx7/6287639/9668973/09732456.pdf

[35] W. Said and A. M. Mostafa, "Towards a hybrid immune algorithm based on danger theory for database security," *IEEE Access*, vol. 8, pp. 145 332–145 362, 2020.

[36] S.-I. T. Tosin and J. R. Gbenga, "Negative selection algorithm based intrusion detection model," in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*. IEEE, 2020, pp. 202–206.

[37] S. Abd Razak, N. H. M. Nazari, and A. Al-Dhaqm, "Data anonymization using pseudonym system to preserve data privacy," *IEEE access*, vol. 8, pp. 43 256–43 264, 2020.

[38] C. Humphry, S. R. R. Pushparaj, and N. Ratha, "Secure vascular biometric recognition," in *2023 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*. IEEE, 2023, pp. 1–5.