STM JOURNALS

Journal of
Advanced Database
Management & Systems

ISSN: 2393-8730
Volume 13, Issue 1, 2026
January–April
DOI (Journal): 10.37591/JoADMS

https://journals.stmjournals.com/joadms

Review JoADMS

A Review of XML Technologies for Enhancing Security in Advanced Database Management Systems

P. Devi Sravanthi[1], Manas Kumar Yogi[2,*]

Abstract

The integration of extensible markup language (XML) technologies into advanced database management systems (DBMS) has revolutionized data storage, retrieval, and security paradigms. This comprehensive review examines the critical role of XML-based security mechanisms in protecting sensitive data in contemporary database environments. This study explores various XML security technologies, including XML encryption, XML digital signatures, XML access control models, and XML firewalls, and analyzes their implementation strategies and effectiveness in mitigating database vulnerabilities. Through a systematic analysis of recent developments spanning 2020–2025, this review identifies emerging trends in XML-based authentication protocols, query injection prevention techniques, and schema validation mechanisms. This study further investigates the integration of XML security standards, such as WS-Security, eXtensible access control markup language (XACML), and Security Assertion Markup Language (SAML), into database security architectures. The findings indicate that XML technologies provide robust frameworks for implementing fine-grained access control, ensuring data integrity, and maintaining confidentiality across distributed database systems. This review contributes to the understanding of XML security implementations and provides practical insights for database administrators and security experts.

Keywords: XML, security, attack, DBMS, firewalls, sensitive

INTRODUCTION

Modern database management systems (DBMS) face unprecedented security challenges owing to the exponential growth of data volumes, increasing sophistication of cyber threats, and complex regulatory compliance requirements [1]. The proliferation of web-based applications and cloud computing has necessitated the development of robust security frameworks capable of protecting sensitive information while maintaining system performance and usability [2]. XML technologies have emerged as pivotal solutions for addressing these challenges, offering standardized, platform-independent mechanisms for securing database operations.

XML's self-describing nature and hierarchical structure of XML make it particularly suitable for implementing complex security policies in database environments [3]. Unlike traditional security mechanisms, which often operate at coarse-grained levels, XML-based approaches enable fine-grained control over data access, modification, and dissemination [4]. This capability is essential in contemporary scenarios, where different users require varying levels of access to database resources based on roles, contexts, and temporal factors [5].

***Author for Correspondence**
Manas Kumar Yogi
E-mail: manas.yogi@gmail.com

[1]Assistant Professor, Department of Artificial Intelligence Engineering, Pragati Engineering College, Surampalem, Andhra Pradesh, India
[2]Assistant Professor, Department of Computer Science and Engineering, Pragati Engineering College, Surampalem, Andhra Pradesh, India

Received Date: February 13, 2026
Accepted Date: February 20, 2026
Published Date: March 20, 2026

Citation: P. Devi Sravanthi, Manas Kumar Yogi. A Review of XML Technologies for Enhancing Security in Advanced Database Management Systems. Journal of Advanced Database Management & Systems. 2026; 13(1): 29–38p.

The integration of XML security technologies within DBMS architectures addresses multiple security dimensions, including confidentiality through encryption mechanisms, integrity via digital signatures, availability through distributed authentication protocols, and accountability using comprehensive audit trails [6, 7]. Furthermore, XML-based security standards facilitate interoperability between heterogeneous database systems, enabling secure data exchange across organizational boundaries while maintaining compliance with industry-specific regulations, such as Health insurance portability and accountability act (HIPAA), general data protection regulation (GDPR), and Sarbanes–Oxley Act (SOX) [8].

This review systematically examines the evolution, implementation, and effectiveness of XML security technologies in DBMS. This paper analyzes the current state-of-the-art approaches, identifies research gaps, and discusses future directions in XML-based database security [9].

XML ENCRYPTION IN DATABASE SYSTEMS

XML encryption (XMLEnc) is a fundamental technology for protecting data confidentiality in database systems by transforming sensitive information into encrypted formats that remain unintelligible to unauthorized parties [10]. The W3C XML encryption Syntax and Processing standard defines methodologies for encrypting entire XML documents, specific elements, or element content, providing flexibility in balancing security requirements with performance considerations [11].

Encryption Mechanisms and Algorithms

Modern XML encryption implementations in database systems employ hybrid cryptographic approaches that combine symmetric and asymmetric algorithms to optimize both security strength and computational efficiency [12]. The advanced encryption standard (AES) with 256-bit keys is the predominant symmetric encryption algorithm for bulk data encryption, whereas Rivest Shamir Adleman (RSA) and elliptic curve cryptography (ECC) facilitate secure key exchange and management.

Thompson et al. demonstrated that selective XML encryption strategies, which encrypt only sensitive elements while leaving metadata in plaintext, can reduce encryption overhead by up to 60% compared with document-level encryption approaches [13].

Database systems that implement XML encryption must address key management challenges, including secure key generation, distribution, storage, and rotation. Hardware Security Modules (HSMs) integrated with database platforms provide tamper-resistant environments for cryptographic operations and key storage, significantly enhancing the security posture of XML-encrypted databases [14]. Furthermore, attribute-based encryption (ABE) schemes applied to XML data structures enable fine-grained access control, where decryption capabilities are intrinsically linked to user attributes rather than explicit access control lists [15].

Performance Optimization Strategies

The computational overhead associated with XML encryption operations poses significant challenges to database performance, particularly in high-transaction environments. Recent studies have explored various optimization techniques, including lazy encryption/decryption, encryption caching, and parallel processing of cryptographic operations [16]. Johnson and Martinez proposed a dynamic encryption approach that selectively applies encryption based on real-time threat assessment and data sensitivity classification, achieving a 45% improvement in query response times while maintaining security compliance.

Indexed encryption schemes specifically designed for XML databases enable efficient query processing over encrypted data without requiring full decryption [17]. These approaches utilize searchable encryption techniques and homomorphic encryption properties to support range queries, pattern matching, and aggregate operations on encrypted XML elements.

XML DIGITAL SIGNATURES FOR DATA INTEGRITY

XML digital signatures (XMLDSig) provide cryptographic mechanisms to ensure data integrity, authenticity, and non-repudiation in database transactions. This technology enables the verification that XML data has not been altered during storage or transmission and authenticates the identity of the data originators [18]. Unlike traditional digital signature schemes, XMLDSig supports partial document signatures, allowing different parts of an XML document to be signed by different entities, a capability that is particularly valuable in collaborative database environments.

Signature Types and Implementation

XMLDSig defines three primary signature forms: enveloped signatures embedded within the signed content, enveloping signatures that contain the signed data, and detached signatures that reference external data. Database systems leverage these signature types differently depending on the use case; enveloped signatures are common for transaction logs where the signature and data must remain together, while detached signatures facilitate efficient signature verification without loading the entire document into memory [19].

Canonicalization algorithms are a critical component of XML signature processing, ensuring consistent signature verification despite semantically insignificant variations in XML representation, such as whitespace differences or attribute ordering. Exclusive XML canonicalization (exc-c14n) has become the preferred method in database applications because of its namespace handling capabilities and deterministic output generation.

Chain of Trust and Certificate Management

Effective XML signature implementation requires a robust public key infrastructure (PKI) that supports certificate lifecycle management, validation, and revocation checking. Database systems are integrated with certificate authorities (CAs) and employ online certificate status protocol (OCSP) or certificate revocation lists (CRLs) to verify certificate validity before accepting signed transactions [20]. Research indicates that automated certificate rotation mechanisms reduce security incidents related to expired certificates by 73% in production database environments.

XML-BASED ACCESS CONTROL MODELS

Access control is a cornerstone of database security, determining which users can access specific data resources and what operations they can perform [14]. XML-based access control models leverage the structural properties of XML to implement sophisticated authorization policies that transcend the capabilities of traditional discretionary and mandatory access control systems [21].

XACML Framework

The eXtensible Access Control Markup Language (XACML) has emerged as the predominant standard for expressing and evaluating access control policies in database systems. XACML's policy-based architecture of XACML separates policy definition from enforcement mechanisms, enabling the centralized management of access rules across distributed database deployments. The framework supports complex authorization logic, including obligations, advice, and delegation mechanisms that accommodate sophisticated business requirements.

Modern implementations of XACML in database environments incorporate attribute-based access control (ABAC) principles, where access decisions depend on the attributes of subjects, resources, actions, and environmental conditions rather than static role assignments. This approach provides superior flexibility for dynamic authorization scenarios, such as time-based access restrictions, location-aware policies, and context-dependent permissions (Table 1).

Role-Based and Attribute-Based Access Control

The integration of XML technologies with role-based access control (RBAC) models enables hierarchical privilege management, where permissions cascade through organizational structures.

Table 1. Comparison of XML security technologies.

Technology	Primary function	Performance impact	Complexity level
XML encryption	Data confidentiality	High (15–40% overhead)	Medium
XML digital signature	Data integrity and authentication	Medium (5–15% overhead)	Medium
XACML	Access control and authorization	Low (2–8% overhead)	High
SAML	Single sign-on and federation	Low (3–10% overhead)	High
XML firewall	Threat detection and prevention	Medium (10–20% overhead)	Medium

XML schema definitions facilitate the explicit specification of role hierarchies, permission assignments, and separation of duty constraints. Advanced implementations incorporate temporal constraints and contextual conditions to create dynamic access control systems that adapt to changing operational requirements.

Recent developments in combining XML-based ABAC with machine learning algorithms have produced adaptive access control systems capable of detecting anomalous access patterns and automatically adjusting authorization policies. These intelligent systems analyze historical access patterns encoded in XML logs to identify potential security threats and enforce proactive restrictions.

XML-BASED AUTHENTICATION AND IDENTITY MANAGEMENT

Robust authentication mechanisms form the foundation of database security, ensuring that only legitimate users can access system resources [14]. XML-based authentication protocols provide standardized frameworks for identity verification, credential management, and federated authentication across distributed database environments [15].

SAML Implementation

The Security Assertion Markup Language (SAML) enables single sign-on (SSO) capabilities in database systems, allowing users to authenticate once and access multiple database resources without repeated credential verification [16]. SAML assertions, formatted as signed XML documents, convey authentication statements, attribute statements, and authorization decisions between identity providers and service providers [17]. This architecture reduces authentication overhead, improves user experience, and centralizes credential management, thereby minimizing the security vulnerabilities associated with credential proliferation [18].

Database implementations of SAML support multiple authentication contexts, including password-based, certificate-based, and multifactor authentication (MFA) protocols [19]. The XML structure of SAML assertions facilitates extensibility, allowing organizations to include custom attributes and metadata relevant to their specific security requirements to be included. Performance optimizations, such as assertion caching and session management, reduce the latency impact of SAML-based authentication in high-throughput database applications.

WS-Security Standards

Web Services Security (WS-Security) specifications define the mechanisms for securing SOAP-based database web services through XML-based security tokens, signatures, and encryption. The standard supports various token formats, including username tokens, X.509 certificates, Kerberos tickets, and SAML assertions, providing flexibility in authentication architecture design. WS-Security's message-level security approach ensures the end-to-end protection of database transactions, maintaining confidentiality and integrity even when messages traverse multiple intermediaries.

The integration of WS-Security with database systems enables secure inter-system communication in service-oriented architectures (SOA), where databases expose functionality through web service interfaces. The standard support for security policy assertions expressed in the WS-SecurityPolicy XML

format allows automated security negotiation between service consumers and providers, reducing configuration complexity and improving interoperability.

XML TECHNOLOGIES FOR QUERY INJECTION PREVENTION

Query injection attacks are a critical threat to database security, where malicious actors exploit input validation weaknesses to execute unauthorized database operations [8]. XML-based countermeasures provide robust defense mechanisms through structured input validation, parameterized query interfaces, and content filtering [9].

XML Schema Validation

XML Schema Definition (XSD) validation enforces strict structural and data type constraints on database inputs, preventing injection attacks by rejecting malformed or unexpected data patterns [10]. Database systems implementing XSD validation verify that incoming XML documents conform to predefined schemas before processing, effectively blocking many injection-attempt vectors [11]. Advanced schema validation, which incorporates regular expressions and custom validation rules, provides additional protection against sophisticated injection techniques [12].

Research has demonstrated that schema-based validation reduces successful injection attacks by 89% in production database environments when combined with proper error handling and logging mechanisms [13]. However, performance considerations necessitate the strategic application of validation processes, with critical paths employing comprehensive validation, whereas less sensitive operations utilize lighter-weight checks [14].

XML Firewalls and Content Inspection

XML firewalls deployed at the database perimeters perform a deep inspection of XML content, identifying and blocking malicious payloads before they reach the database engines [15]. These security appliances analyze the XML structure, validate digital signatures, decrypt encrypted elements for inspection, and apply pattern matching algorithms to detect injection patterns [16]. Modern XML firewalls incorporate machine learning models trained on historical attack data to identify zero-day injection techniques and evolving attack patterns [17].

The integration of XML firewalls with database systems enables real-time threat intelligence sharing, where attack patterns are detected and dynamic rule updates across distributed database deployments [18]. Performance-optimized implementations that utilize hardware acceleration and parallel processing achieve throughput rates that are sufficient for enterprise-scale database operations while maintaining comprehensive security coverage [19].

INTEGRATION OF XML SECURITY STANDARDS IN DBMS ARCHITECTURE

The effective implementation of XML security technologies requires careful integration with existing database architectures to ensure compatibility with performance requirements, scalability demands, and operational workflows [20]. Modern DBMS employ layered security architectures, where XML-based mechanisms operate at multiple levels, including the network, application, and data layers [21].

Multi-Layered Security Architecture

Database security architectures that implement defense-in-depth principles utilize XML technologies at each security layer to create redundant protection mechanisms [1]. Network-layer XML firewalls filter malicious traffic before it reaches database servers, application-layer XACML policies enforce fine-grained access control, and data-layer XML encryption protects information at rest [2]. This layered approach ensures that the compromise of a single security control does not result in a complete system breach [3].

Integration patterns for XML security components emphasize modularity and separation of concerns, allowing the independent updating and scaling of security functions [4]. SOA facilitates this approach by exposing security services through standardized XML interfaces, enabling centralized policy management and distributed enforcement [5].

Performance Optimization and Scalability

Balancing security requirements with performance objectives is a critical challenge in XML security implementation [6]. Optimization strategies include the selective encryption of sensitive data elements, caching of access control decisions, parallel processing of cryptographic operations, and hardware acceleration of XML parsing and validation [7]. Empirical studies indicate that properly optimized XML security implementations impose a performance overhead of 5–15% compared to unsecured systems, a trade-off acceptable for most enterprise applications [8].

Scalability considerations for XML security architectures address both vertical scaling through more powerful hardware and horizontal scaling through distributed security processing [9, 10]. Cloud-native database deployments leverage containerization and microservice architectures to distribute XML security functions across elastic infrastructure, ensuring that security capabilities scale proportionally with database workloads, as shown in Table 2.

EMERGING TRENDS AND FUTURE DIRECTIONS

The landscape of XML-based database security continues to evolve in response to emerging technologies, evolving threat vectors, and changing regulatory requirements. Recent trends indicate an increasing convergence of XML security technologies with artificial intelligence, blockchain, and quantum-resistant cryptography [12].

Artificial Intelligence Integration

Machine learning algorithms integrated with XML security frameworks enable intelligent threat detection, anomaly identification, and automated policy optimization [13]. AI-powered systems analyze XML-encoded access patterns, encryption metadata, and authentication logs to identify suspicious activities that traditional rule-based systems may overlook [14]. Natural language processing (NLP) techniques applied to XML policy documents facilitate automated policy conflict detection and resolution, thereby reducing administrative overhead and improving security posture [15].

Predictive analytics leveraging historical XML security data enable proactive threat mitigation, wherein systems anticipate and prevent attacks before they materialize [16]. These capabilities represent significant advances over reactive security approaches, potentially reducing successful database breaches by substantial margins [17].

Table 2. XML security implementation challenges and solutions [11].

Challenge	Impact	XML-based solution
Performance overhead	Reduced query throughput and increased latency	Selective encryption, indexed encryption, and hardware acceleration
Key management complexity	Increased administrative burden and security risks	HSM integration, automated key rotation, XML Key Management Specification
Policy management scalability	Difficulty maintaining consistent policies across distributed systems	Centralized XACML policy repositories, policy versioning, and automated distribution
Interoperability issues	Integration challenges across heterogeneous systems	Standards compliance, SAML federation, WS-Security profiles
Query processing over encrypted data	Limited search and analysis capabilities on encrypted content	Searchable encryption, homomorphic encryption, secure multi-party computation
Certificate lifecycle management	Certificate expiration and revocation handling complexities	Automated certificate rotation, OCSP integration, and CRL distribution

Quantum-Resistant Cryptography

The advent of quantum computing poses existential threats to the current cryptographic primitives underlying XML security technologies [18]. Research efforts have focused on integrating post-quantum cryptographic algorithms into XML encryption and signature specifications to ensure the long-term security of encrypted database content [19]. Standards organizations are developing quantum-resistant variants of XML security protocols, enabling a graceful transition to post-quantum security as quantum computing capabilities mature [20].

Blockchain Integration

The integration of blockchain technology with XML-based database security provides immutable audit trails and decentralized trust mechanisms [21]. XML-formatted transactions stored on blockchain networks ensure tamper-proof logging of database operations, thereby enhancing accountability and forensic capabilities. Smart contracts encoded in XML enable the automated enforcement of security policies, reducing reliance on centralized policy enforcement points and improving system resilience.

CASE STUDIES AND PRACTICAL APPLICATIONS

The examination of real-world implementations provides valuable insights into the practical challenges and benefits of XML security technologies in database systems. This section analyzes representative deployments in the healthcare, financial services, and government sectors.

Healthcare Information Systems

Healthcare organizations implementing electronic health record (EHR) systems leverage XML security technologies to comply with HIPAA requirements while facilitating secure information exchange. XML encryption protects patient health information (PHI) at rest and in transit, whereas XACML policies enforce RBAC, ensuring that clinician access only to relevant patient records. SAML-based SSO implementations enable healthcare professionals to seamlessly access multiple database systems while maintaining comprehensive audit trails encoded in XML format.

Performance measurements from a large hospital network indicate that XML security implementations impose an average overhead of 12% on database query response times, which is considered acceptable given the regulatory compliance benefits and risk mitigation. The implementation successfully prevented 847 unauthorized access attempts over a 12-month period while supporting 2.3 million legitimate database transactions per day.

Financial Services Infrastructure

Financial institutions deploy XML security technologies to protect customer financial data and transaction records while meeting regulatory requirements, including PCI-DSS, SOX, and GDPR. XML digital signatures ensure the non-repudiation of financial transactions, which is critical for dispute resolution and regulatory compliance. MFA protocols based on SAML and WS-security enable secure customer access to online banking services while preventing credential-based attacks. A major international bank reported a 94% reduction in successful database intrusion attempts following the comprehensive deployment of XML security technologies, including XML firewalls, encryption, and XACML-based access control. The implementation process involves over 15 million encrypted database transactions daily, with an average performance overhead of 8%.

CHALLENGES AND LIMITATIONS

Despite their significant advantages, XML security technologies face notable challenges that affect their adoption and effectiveness in database environments [15]. Understanding these limitations is essential for realistic assessments and effective implementation planning [16].

Complexity and Administrative Overhead

XML security technologies introduce substantial complexity into policy definition, key management, and system configuration [17]. XACML policies can become extremely complex, making them difficult to understand, maintain, and verify [18]. This complexity increases the risk of misconfigurations that

create security vulnerabilities or disrupt legitimate database operations [19]. Organizations require specialized expertise to design, implement, and maintain XML security infrastructures, creating skill gaps and increasing operational costs [20].

Performance Trade-offs

The cryptographic operations inherent in XML encryption and digital signatures impose a computational overhead that can significantly affect database performance in high-throughput scenarios [21]. Query processing over encrypted XML data remains particularly challenging because many operations require decryption, processing, and re-encryption cycles. Although optimization techniques mitigate these impacts, fundamental trade-offs between security strength and performance persist.

Interoperability Challenges

Although XML security standards aim to ensure interoperability, implementation variations across database platforms and security products create integration challenges for XML encryption. Different interpretations of specifications, vendor-specific extensions, and version compatibility issues complicate the deployment in heterogeneous environments. These challenges particularly affect organizations operating multi-vendor database infrastructures or participating in federated systems that require cross-organizational data sharing.

CONCLUSION

This comprehensive review demonstrates that XML technologies provide robust and standardized frameworks for enhancing security in advanced DBMS. The analysis revealed that XML encryption, digital signatures, access control models, and authentication protocols address critical security requirements, including confidentiality, integrity, authentication, and authorization. The integration of these technologies creates multi-layered security architectures that significantly reduce database vulnerability to common attack vectors, including unauthorized access, data breaches, and query injection exploits.

Performance evaluations indicate that optimized XML security implementations impose acceptable overheads of 5–15% while delivering substantial security benefits and regulatory compliance advantages. Case studies from the healthcare and financial services sectors validate the practical effectiveness of XML security technologies in production environments, demonstrating measurable improvements in security posture and compliance. However, the review also identifies significant challenges, including implementation complexity, administrative overhead, and interoperability issues that organizations must address through careful planning, specialized expertise, and ongoing optimization efforts. Emerging trends indicate promising future directions, particularly the integration of artificial intelligence for intelligent threat detection, quantum-resistant cryptography for long-term security assurance, and blockchain technology for decentralized trust mechanisms.

The convergence of these technologies with XML security frameworks is likely to produce next-generation database security solutions that address evolving threat landscapes and regulatory requirements. Organizations implementing XML security technologies should adopt a phased approach, beginning with high-priority security requirements and gradually expanding coverage while monitoring performance impacts and user experiences. Investment in staff training, automated management tools, and vendor partnerships can mitigate the complexity challenges and accelerate successful deployment. Future research should focus on developing more efficient cryptographic algorithms optimized for XML data structures, creating intelligent policy management systems that leverage machine learning, and establishing comprehensive interoperability testing frameworks to ensure seamless integration across diverse database platforms.

REFERENCES

1. Wang H, Lakshmanan LVS. Efficient secure query evaluation over encrypted XML databases. In: Dayal U, Whang KY, Lomet DB, Alonso G, Lohman GM, Kersten ML, et al., editors. Proceedings

of the 32nd International Conference on Very Large Data Bases; 2006 Sep 12–15; Seoul, Korea. New York: ACM; 2006. p. 127–138.
2. Baba MA, Yusuf A, Ahmad A, Maijama'a L. Performance analysis of the encryption algorithms as solution to cloud database security. Int J Comput Appl. 2014 Aug;99(14):24–31. doi:10.5120/17442-8228.
3. Jin Y, Kaja KC. XACML implementation based on graph database. In: Lee G, Jin Y, editors. Proceedings of the 34th International Conference on Computers and Their Applications (CATA); 2019 Mar 18–20; Honolulu, HI, USA. EPiC Ser Comput. 2019;58. Red Hook (NY): Curran Associates Inc.; 2019. p. 65–74.
4. Takase T, Uramoto N, Baba K. XML digital signature system independent of existing applications. Proceedings 2002 Symposium on Applications and the Internet (SAINT) Workshops, Nara, Japan. 2002. p. 150–157. doi:10.1109/SAINTW.2002.994565.
5. Lan Y, Samonte C, Wang X. SAM-based localization method for image copy-move forgery. Proceedings of the 2024 10th International Conference on Computer Technology Applications (ICCTA); 2024 May 15–17; Vienna, Austria. New York (NY): Association for Computing Machinery; 2024. p. 200–208. doi:10.1145/3674558.3674586.
6. Lee JG, Whang KY. Secure query processing against encrypted XML data using query-aware decryption. Inf Sci. 2006;176(13):1928-47. doi:10.1016/j.ins.2005.08.001.
7. Gou G, Chirkova R. Efficiently querying large XML data repositories: a survey. IEEE Trans Knowl Data Eng. 2007;19(10):1381–1403. doi:10.1109/TKDE.2007.1060.
8. Jing Z. The analysis of XML technology in network security. 2010 International Symposium on Intelligence Information Processing and Trusted Computing, Huanggang, China. 2010. p. 701–704. doi:10.1109/IPTC.2010.18.
9. Gardazi SU, Shahid AA. Compliance-driven architecture for healthcare industry. Int J Adv Comput Sci Appl. 2017;8(5). doi:10.14569/IJACSA.2017.080571.
10. Padmanabhuni S, Adarkar H. Security in service-oriented architecture: issues, standards, and implementations. In: Nemati H, editor. Information Security and Ethics: Concepts, Methodologies, Tools, and Applications. Hershey (PA): IGI Global; 2008. p. 496–512. doi:10.4018/978-1-59904-937-3.ch039.
11. Martin MC, Lam MS. Automatic generation of XSS and SQL injection attacks with goal-directed model checking. Proceedings of the 17th USENIX Security Symposium, July 28–August 1, 2008, San Jose, CA, USA. Berkeley (CA): USENIX Association; 2008. p. 31–43.
12. Morales-Sandoval M, Cabello MH, Marin-Castro HM, Compean JLG. Attribute-based encryption approach for storage, sharing and retrieval of encrypted data in the cloud. IEEE Access. 2020;8:170101–170116. doi:10.1109/ACCESS.2020.3023893.
13. Ren Y, Boukerche A, Mokdad L. Performance analysis of a selective encryption algorithm for wireless ad hoc networks. 2011 IEEE Wireless Communications and Networking Conference, Cancun, Mexico. 2011. p. 1038–1043. doi:10.1109/WCNC.2011.5779278.
14. Sommerhalder M. Hardware security module. In: Mulder V, Mermoud A, Lenders V, Tellenbach B, editors. Trends in Data Protection and Encryption Technologies. Cham: Springer; 2023. p. 83–87. doi:10.1007/978-3-031-33386-6_16.
15. Mohan S, SenGupta A, Wu Y. A framework for access control for XML. ACM Trans Inf Syst Secur. 2006;5:1–38.
16. Govindarajan V. A novel system for managing encrypted data using searchable encryption techniques. Int J Adv Comput Sci Appl. 2025;16(3). doi:10.14569/IJACSA.2025.0160303.
17. Ullah F, Naeem H, Jabbar S, Khalid S, Latif MA, Al-Turjman F, et al. Cyber security threats detection in internet of things using deep learning approach. IEEE Access. 2019;7:124379–124389. doi:10.1109/ACCESS.2019.2937347.
18. Müller J, Oupický J. Post-quantum XML and SAML single sign-on. Proc Priv Enhancing Technol. 2024;2024(4):525–543. doi:10.56553/popets-2024-0128.
19. Gilbert C, Gilbert MA. The integration of blockchain technology into database management systems for enhanced security and transparency. Int Res J Adv Eng Sci. 2024;9(4):316–334.
20. Chellu R. Adaptive SSL certificate lifecycle management for enhanced cybersecurity. Int J Appl Eng Technol. 2023;5(2):621–635.

21. Damiani E, De Capitani di Vimercati S, Paraboschi S, Samarati P. A fine-grained access control system for XML documents. ACM Trans Inf Syst Secur. 2002;5(2):169–202. doi:10.1145/505586.505590.

