International Journal of Current Trends in Engineering and Technology
www.ijctet.org, ISSN: 2395-3152
Volume: 12, Issue: 05 (September -October 2026)

A Survey of Data Leakage Detection and Prevention Techniques: Fingerprinting, Access Control, Machine Learning, and Self-Protective Data Objects

Vinita Ganeshe, Akrati Shirivastava, and Prachi Tiwari

Department of Computer Science and Engineering
Vaishnavi Institute of Technology and Science, Bhopal, India

Abstract: Data leakage occurs when sensitive information is disclosed, transferred, or accessed in ways that violate an organisation's security or privacy requirements. This survey reviews a range of data leakage detection and prevention (DLD/DLP) techniques, including data allocation with decoy objects, watermarking, privacy-preserving transformations, access control, behavioural analytics, content inspection, runtime protection, and data lineage monitoring. The approaches are considered in relation to the data states—at rest, in use, and in motion—and the threats posed by external actors, insiders, and third parties. Particular attention is given to insider threats because legitimate credentials and authorised workflows can be misused without triggering conventional perimeter controls. The review also considers the movement from static, rule-based controls towards context-aware and self-protective data objects. The literature cited suggests that no single mechanism addresses every leakage scenario. A layered strategy combining prevention, detection, attribution, and incident response is therefore more appropriate. The paper identifies dynamic policy enforcement and coordinated runtime monitoring as areas requiring further investigation.

Keywords— data leakage prevention, data leakage detection, data fingerprinting, insider threat, machine learning, access control, self-protective data objects, data lineage.

I. Introduction

The growing volume of information created, stored, processed, and exchanged by organisations has increased the importance of protecting sensitive digital assets. Banking, insurance, healthcare, education, manufacturing, research, and information-technology organisations routinely handle personally identifiable information (PII), financial records, credentials, intellectual property, and confidential business data. Such information may be exposed through software misconfiguration, security vulnerabilities, weak or reused passwords, physical loss, social engineering, or misuse by trusted insiders. Insider-related incidents are particularly challenging because employees, contractors, and vendors may already possess legitimate access to the systems and information they misuse [1], [13], [14].

Data leakage prevention is consequently broader than blocking unauthorised logins or defending a network perimeter. It requires controls that understand the sensitivity of information, the context in which it is accessed, the actions performed on it, and the routes through which it may be transferred. This survey organises the techniques represented in the cited literature, compares their principal strengths and limitations, and identifies research directions for adaptive, data-centric protection.

II. Data Leakage Problem and Threat Model

Data must be protected in three states: data at rest (stored in databases, file systems, backups, or cloud repositories), data in use (being viewed, edited, copied, or processed), and data in motion (being transmitted across networks or between systems). Leakage can originate from external attackers, malicious or negligent insiders, and third-party service providers. Common exfiltration channels include email, messaging platforms, removable media, printing, screenshots, cloud uploads, and file transfers. The threat model therefore includes both unauthorised access and misuse of authorised access. A user may legitimately retrieve a document but subsequently disclose it to an unauthorised recipient. Effective DLP must combine identity and access controls with content-aware inspection, activity monitoring, traceability, and incident response. Attribution mechanisms can provide evidence about the source of a leak, but attribution should not be treated as conclusive proof of an individual's responsibility without corroborating evidence [1], [15].

III. Survey of Data Leakage Detection and Prevention Approaches

A. Data Allocation and Decoy Objects

Data allocation techniques distribute records or data subsets among agents and may insert realistic decoy records, also called fake objects. If a decoy is later found in an unauthorised location, its distribution pattern can help identify the set of recipients associated with the exposure. This can support leakage attribution without modifying every genuine record. Its effectiveness

depends on careful allocation, secure maintenance of distribution records, and the ability to distinguish decoys from genuine information [1], [16].

B. Watermarking

Watermarking embeds identifying information in a data object so that a recovered copy can be traced to its source or recipient. Database watermarking is one established form of this approach [2], [17]. Watermarks can provide useful evidence of provenance, but their robustness may be affected by transformations, deliberate removal, or attacks. In addition, embedding a watermark may alter the original data, which can be unacceptable where exact data integrity must be preserved.

C. Privacy-Preserving Transformation

Privacy-preserving methods such as k-anonymity transform or generalise records so that an individual is indistinguishable from at least k−1 other records with respect to selected quasi-identifiers. Related models, including t-closeness, seek to limit the disclosure of sensitive attributes by constraining differences between a group's sensitive-attribute distribution and the overall distribution [4], [18], [19]. These methods can reduce direct identification risk, but they do not guarantee privacy against every form of background-knowledge, linkage, or inference attack. Their suitability depends on the data, threat model, and intended analytical use.

D. Access Control

Access-control mechanisms determine which subjects may perform which operations on particular resources. Discretionary access control, mandatory access control, role-based access control (RBAC), and attribute-based access control (ABAC) provide different ways to express and enforce permissions. RBAC associates permissions with organisational roles, while ABAC can evaluate attributes of the user, resource, action, and environment [6], [20]. The principle of least privilege limits users to the access required for their duties. These mechanisms reduce unnecessary exposure but cannot, by themselves, prevent a properly authorised user from misusing information available to them.

E. Behavioural Analytics and Insider-Threat Detection

Behavioural approaches analyse signals such as authentication events, file access, database queries, network activity, downloads, and user actions. Statistical techniques, association analysis, anomaly detection, and machine-learning models can identify deviations from established activity patterns. The cited literature includes neural-network-based approaches and reports detection performance of 95% or higher in particular experimental settings [21]–[23]. Such figures are specific to the datasets, labels, and evaluation protocols used in those studies and should not be interpreted as a general performance guarantee. Behavioural systems require representative and appropriately labelled data; they may produce false alarms, miss previously unseen behaviours, or reflect bias in historical activity records.

F. Content Inspection and Fingerprinting

Content-based DLP examines the information itself rather than relying only on user identity or network location. Common techniques include exact and partial document fingerprinting, regular-expression and dictionary matching, n-gram analysis, term weighting, TF-IDF, latent semantic indexing, and sequence alignment. Exact hashes are efficient for detecting unchanged files but are sensitive to even small modifications. Fuzzy matching and sequence-alignment methods, including Smith–Waterman-style alignment, can provide more tolerant matching when content has been edited or rearranged [24]–[26]. Pattern-based inspection is useful for structured identifiers and known sensitive phrases. However, it may generate false positives and may fail to recognise semantic disclosure that does not match predefined patterns.

G. Runtime Protection and Self-Protective Data Objects

Runtime application self-protection (RASP) monitors application behaviour during execution and can use runtime context to detect or block suspicious actions. A related data-centric direction is the self-protective data object: protection policies and selected enforcement logic are associated with the data object so that access conditions can be evaluated as the object is used or transferred. Depending on the design, such an object may enforce permissions, record access events, report policy violations, or support recovery to a safe state. These capabilities require careful consideration of trust, policy integrity, execution environment, and compatibility with applications [27], [28].

H. Data Lineage and Activity Monitoring

Data lineage records where information originated, how it was transformed, and where it moved through a workflow. It can support audits and investigations by connecting downstream data to upstream sources. Database activity and file activity monitoring provide complementary visibility into access and transfer events. Lineage and monitoring can improve detection

and incident reconstruction, but their usefulness depends on the completeness and accuracy of event collection, metadata, and sensitivity labels [29], [30].

IV. Comparative Analysis

Table I provides a qualitative comparison of the principal approaches discussed above. The comparison is conceptual: the cited material does not establish a shared dataset, threat model, or evaluation protocol across all techniques. Consequently, the table should not be read as a quantitative ranking.

TABLE I. QUALITATIVE COMPARISON OF DATA LEAKAGE DETECTION AND PREVENTION APPROACHES

Approach	Primary purpose	Potential strength	Key limitation
Data allocation and decoys	Trace and help attribute leakage	Can support recipient-level tracing without changing every genuine record	Requires controlled distribution and reliable allocation records
Watermarking	Embed trace information	Can identify a marked copy or its source	May alter data and can be degraded or removed
Privacy-preserving transformation	Reduce identifiability	Can limit direct disclosure in released datasets	May remain vulnerable to linkage or inference attacks
Access control (RBAC/ABAC)	Restrict permitted access and actions	Enforces least privilege and contextual policies	Does not stop misuse of legitimately accessed data
Behavioural analytics / ML	Detect unusual activity	Can identify deviations from learned behaviour	Needs representative data; false positives and unseen patterns remain concerns
Fingerprinting and content matching	Detect copied or modified content	Effective for known documents and sensitive patterns	Exact matching is fragile under modification; fuzzy matching adds complexity
Regex and statistical inspection	Identify sensitive patterns	Useful for structured data and known formats	May miss semantic leakage or produce false alarms
Runtime protection	Monitor and control application execution	Can respond to runtime context	Requires integration and reliable enforcement points
Self-protective data objects	Associate protection with data	Can support portable, context-aware controls	Raises deployment, trust, interoperability, and policy-management challenges
Data lineage and monitoring	Track movement and access	Supports visibility, auditing, and investigation	Depends on complete and accurate event and metadata collection

V. Research Gaps

The reviewed approaches reveal several continuing challenges. First, many deployments rely on static rules and manually maintained sensitivity labels, which may not adapt quickly to changing users, workflows, or organisational policies. Second, exact fingerprinting can fail when content is modified, while more tolerant matching introduces additional computational and threshold-selection challenges. Third, behavioural models may not recognise previously unseen attack patterns and can be affected by noisy or incomplete logs. Fourth, perimeter-oriented controls are insufficient when an authenticated user misuses legitimate privileges. Finally, data-centric and self-protective mechanisms introduce practical questions concerning policy portability, enforcement trust, interoperability, revocation, and auditability [21]–[23], [31]. These gaps motivate investigation of protection mechanisms that combine content sensitivity, identity, access context, behavioural evidence, and runtime policy enforcement. Any proposed system should be evaluated against clearly defined threat scenarios and should report detection effectiveness, false-positive rates, response latency, scalability, and operational overhead.

VI. Emerging Direction: Dynamic and Self-Protective Data

A possible architecture for dynamic protection can represent sensitive information as a managed data object associated with access policies and audit metadata. Encryption can protect stored or transmitted content; authentication and role- or attribute-based policies can constrain access; content inspection can identify sensitive or matching material; behavioural monitoring can flag unusual activity; and a response component can block, quarantine, or report policy violations. These controls are complementary rather than interchangeable.

The research direction described in the source material also refers to an Extended Host-based Data Scrutinizer (EHDS) that combines fingerprinting, regular-expression matching, and machine-learning-oriented pattern analysis. In this survey, EHDS is treated as a proposed integration direction rather than as a universally established standard or a validated system. Its practical value would need to be demonstrated through a reproducible implementation, a defined threat model, comparative baselines, and evaluation under realistic workload and attack conditions [13], [27], [28], [30].

VII. Discussion

The evidence reviewed supports treating DLP as a layered security process rather than as a single classification task. Preventive controls reduce the opportunities for exposure; content inspection and behavioural analytics help detect suspicious transfers; watermarking and decoy allocation can support traceability; and runtime or self-protective mechanisms can enable context-sensitive responses. Data lineage and activity monitoring provide the audit information needed to investigate incidents.

No single technique is sufficient across all data types, users, applications, and leakage channels. Organisations should select controls according to information sensitivity, regulatory obligations, operational requirements, and the capabilities of their infrastructure. Evaluation should also distinguish detection from prevention and attribution: a system that flags a suspicious event has not necessarily prevented disclosure or conclusively identified the responsible actor.

VIII. Conclusion

This survey has reviewed data leakage detection and prevention approaches covering decoy-based allocation, watermarking, privacy-preserving transformation, access control, behavioural analytics, content inspection, runtime protection, self-protective data objects, and data lineage. Each approach addresses a different part of the leakage problem and has limitations when used in isolation. The principal research direction emerging from the reviewed material is a dynamically configurable, data-centric framework that combines policy enforcement, content analysis, behavioural monitoring, and runtime response. Future work should validate such an architecture through a reproducible implementation and systematic experiments that quantify both security effectiveness and deployment cost.

References

[1] P. Papadimitriou and H. Garcia-Molina, “A Model for Data Leakage Detection,” in Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE), 2009, doi: 10.1109/ICDE.2009.227.

[2] R. Agrawal and J. Kiernan, “Watermarking Relational Databases,” in Proceedings of the 28th International Conference on Very Large Data Bases (VLDB), 2002.

[3] M. Pistoia, O. Tripp, P. Centonze, and J. W. Ligman, “Labyrinth: Visually Configurable Data-Leakage Detection in Mobile Applications,” in Proceedings of the 16th IEEE International Conference on Mobile Data Management (MDM), 2015, pp. 279–286, doi: 10.1109/MDM.2015.69.

[4] L. Sweeney, “k-Anonymity: A Model for Protecting Privacy,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5, pp. 557–570, 2002.

[5] National Institute of Standards and Technology (NIST), “Framework for Improving Critical Infrastructure Cybersecurity,” Gaithersburg, MD, USA.

[6] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, “Proposed NIST Standard for Role-Based Access Control,” ACM Transactions on Information and System Security, vol. 4, no. 3, pp. 224–274, 2001.

[7] Verizon, “Data Breach Investigations Report,” Verizon Enterprise, various editions.

[8] IBM, “InfoSphere DataStage,” product documentation on data integration and lineage.

[9] “Dynamically Configurable Document Model to Prevent Data Leakage,” thesis, Department of Computer Science and Engineering, Visvesvaraya Technological University, Belagavi, India. Full author, year, and repository details were not supplied.

[10] Source thesis literature survey, references concerning insider-threat detection using association analysis, machine learning, and convolutional neural networks. Full bibliographic details were not supplied.

[11] Source thesis literature survey, references concerning sensitive-term detection using Boyer–Moore, Smith–Waterman, TF-IDF, latent semantic indexing, and Zemberek. Full bibliographic details were not supplied.

[12] Source thesis literature survey, references concerning runtime data-leakage detection and watermark-based network filtering. Full bibliographic details were not supplied.

[13] A. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, "A Survey on Data Leakage Prevention Systems," Journal of Network and Computer Applications. Publication details should be verified against the original record.

[14] M. Conti, N. Dragoni, and V. Lesyk, "A Survey of Insider Threat Models and Countermeasures," IEEE Communications Surveys & Tutorials. Publication details should be verified against the original record.

[15] NIST, "Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations," NIST SP 800-171, Revision 2.

[16] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detection," in Proceedings of the IEEE 25th International Conference on Data Engineering (ICDE), 2009.

[17] R. Agrawal and J. Kiernan, "Watermarking Databases," in Proceedings of the 28th VLDB Conference, 2002.

[18] P. Samarati and L. Sweeney, "Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalisation and Suppression," technical report, 1998.

[19] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," in Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE), 2007, pp. 106–115.

[20] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to Attribute-Based Access Control (ABAC) Definition and Considerations," NIST SP 800-162, 2014.

[21] M. Theis, K. Rechert, et al., "Insider Threat Detection Using Machine Learning." The supplied reference is incomplete; full publication details are required.

[22] "A Review of Recent Advances, Challenges, and Opportunities in Malicious Insider Threat Detection Using Machine Learning Methods," IEEE Access, vol. 12, pp. 30907–30927, 2024, doi: 10.1109/ACCESS.2024.3369906. Author details should be checked.

[23] "Machine Learning-Based Approaches for Malicious Insider Threat Detection: A Survey of Recent Advances and Open Issues," Proceedings of the 2025 International Conference on Intelligent Computing, Information and Control Systems (ICOIICS), 2025, doi: 10.1109/ICOIICS67115.2025.11390400. Author details should be checked.

[24] M. Pistoia, O. Tripp, P. Centonze, and J. W. Ligman, "Labyrinth: Visually Configurable Data-Leakage Detection in Mobile Applications," IEEE MDM, 2015, doi: 10.1109/MDM.2015.69.

[25] A. Singhal, "Modern Information Retrieval: A Brief Overview," IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.

[26] T. Joachims, "Text Categorisation with Support Vector Machines: Learning with Many Relevant Features," in Proceedings of ECML, 1998, pp. 137–142.

[27] G. E. Kaiser, "Runtime Application Self-Protection and Context-Aware Security." The supplied reference is incomplete; verify the exact publication.

[28] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web Transactions," ACM Transactions on Information and System Security, vol. 1, no. 1, pp. 66–92, 1998. Its direct relevance to self-protective data objects should be checked.

[29] P. Buneman, S. Khanna, and W.-C. Tan, "Why and Where: A Characterisation of Data Provenance," in Proceedings of the 8th International Conference on Database Theory (ICDT), 2001, pp. 316–330.

[30] Y. Cui and J. Widom, "Lineage Tracing for General Data Warehouse Transformations," VLDB Journal, vol. 12, no. 1, pp. 41–58, 2003.

[31] "Data Leakage/Loss Prevention Systems (DLP)," Proceedings of the 2014 World Congress on Computer Applications and Information Systems (WCCAIS), 2014, doi: 10.1109/WCCAIS.2014.6916624. Author details should be checked.