GMR
The Original

AUDITING THE BLACK BOX: PRIVACY AUDIT REVERSE ENGINEERING AND THE LIMITS OF SOFTWARE SECRECY

Adv. Madhavan Thittai[1], Adv. Sankalp Mirani[2], Adv. Taranjeet Singh[3], Samiksha Shah[4], Adv. Naman Baleja[5], Aryan Sharma[6]

[1]Adv.; Graduate , MNLU Mumbai; Law; Pune ,411048; Email id- blackcatproductions0@gmail.com; Orcid ID: 0009-0002-4091-0239
[2]Adv.; Graduate , MNLU Mumbai; Law; Mumbai, 400076; Email id- sankalpmirani610@gmail.com; ORCID ID- 0009-0004-7286-0795
[3]LLM , SLS Pune
[4]Final Year, MU
[5]Graduate, DNLU
[6]3rd year, DNLU

ABSTRACT
There is an inherent tension in how organizations operate in today's environment – statutory, contractual and regulatory requirements increasingly call for evidence of accountability – and technical verification of information being handled – while the systems used to process this information are closed, proprietary and subject to IP protection. In current regimes, organizations have a legal duty to ensure information security and compliance but laws on software copyright, trade secrets, anti-circumvention and restrictive licensing clauses make it impossible for them to examine what systems they deploy.
This article is about the tension and explores the intersection between software law and information governance. Using a doctrinal and comparative approach in the United States, the European Union, and India, it demonstrates that current exceptions, which have been created to facilitate interoperability, error correction, and/or competitive emulation, are inadequate to address compliance-based inspection.
To help address this issue, the article suggests a new legal and regulatory classification – governance-sanctioned audit reverse engineering. Establishes an exception to the safe harbor that permits organizations that have legitimate access to carry out proportionate, necessary technical verifications in accordance with strict confidentiality and non-substitution safeguards. The model sets a firm "misuse boundary", ensuring that verification is done in a manner that is responsible, rather than malicious, like code theft, commercial cloning or unauthorized data exfiltration. Lastly, the article examines how this framework can be applied in India, where the Copyright Act, 1957, the Information Technology Act, 2000, and the Digital Personal Data Protection Act, 2023, intersect in order to create a balanced approach to balancing the need for software secrecy with digital accountability.

KEYWORDS: Information Governance; Software Copyright; Audit Reverse Engineering; Comparative Law; Accountability Principle; Data Protection; Anti-Circumvention; Digital Personal Data Protection (DPDP) Act.

Research Questions
Primary Question
- How can organizations ensure that their statutory, contractual and professional information governance and accountability requirements are met while the proprietary software systems they use are unable to help them verify information assets processing?

Secondary Questions
- How do actual U.S., EU, and India software copyright exceptions and anti-circumvention provisions and contractual agreements support technical audits that are conducted to ensure compliance with copyright laws without consideration of competition?
- What does it mean theoretically and legally to conduct a reverse engineering analysis that is "governance-sanctioned" versus one that is competitive, malicious or exploitative?
- What are the elements and protections needed to build a conditional safe harbor for software audits that will allow for legitimate information governance without compromising the protection of IP rights and the commercial software market?
- What will be the mechanism for such a safe harbour to be embedded into the evolving digital governance framework in India including the implementation of the Digital Personal Data Protection Act, 2023, and current copyright exemptions?

METHODOLOGY
The doctrinal approach, the comparative legal approach and the policy and institutional design approach all accompany each other in this article:

Doctrinal Legal Analysis:
The article provides an exegetical examination of legal provisions, regulations and decisions in various areas of legal terminology. It evaluates:

- **Intellectual Property and Software Protection:** Discussing the fine lines between copy restricted acts, fair use and software protection under the Indian Copyright Act, 1957[1] and the EU Software Directive (Directive 2009/24/EC)[2].
- **Information Governance & Data Protection:** Exploring the operational challenges of Accountability, Suppliers' accountability, and Audit requirements in the EU General Data Protection Regulation (GDPR) & India's Digital Personal Data Protection (DPDP) Act, 2023.
- **Cyberlaw and Technological Protection:** Liability for unauthorized access and circumvention under the US Digital Millennium Copyright Act (DMCA) and India's Information Technology Act, 2000.

Comparative Constitutional & Statutory Analysis:
The article compares three legal architectures dealing with the conflict between software secrecy and information governance in different ways:

- **The United States:** A mixed system of open-ended fair use and anti-circumvention and private contractual restrictions.
- **European Union/United Kingdom:** A very formal and teleological list of statutory exceptions (interoperability and error correction) embedded in principles of data accountability.
- **India:** A nascent and emerging model of privacy jurisprudence (as seen with Puttaswamy) and of data fiduciary obligations (whereas no statutory data fiduciary obligations exist), together with specific exceptions to the computer-program copyright (which are becoming more prominent).

Further, Singapore is featured as a limited and illustrative case study to demonstrate how the empirical engineering realities do not always fit with statutory exceptions, in the context of the legislative transplantation risks.

Concepts and Taxonomy Development:
The paper makes analytical distinctions between so-called "behavioral testing" (external black-box analysis), and "structural inspection" (intrusive technical inspection performed internally). It creates an "audit ladder" model, a progressive escalation path, which varies from the non-invasive documentation review, through traffic analysis, to secure, supervised code review and limited reverse engineering.

Designing policy and regulatory reform:
Combining the doctrinal and comparative results, the article proposes a new conditional six-factor 'safe harbor' test (lawful access, documented purpose, necessity, proportionality, non-substitution and confidentiality) as a feasible regulatory tool for courts, legislators and contracting parties.

I.Introduction: The Organisational Compliance Black Box

Most modern organizations are in a dilemma about information governance. They use propriety, cloud-based software, AI assistants and compliance modules which they cannot meaningfully examine, and they are legally, contractually and professionally responsible for processing sensitive data such as personal data, source code and credentials, lawfully and securely. In today's regulatory environment, which has moved away from paper-based assurances to accountability that can be demonstrated,[1] the reality is that the technical aspects are still unclear. Organizations cannot prove a vendor's claims of not using customer documents to train models or data-residency restrictions when they are claimed to be strictly enforced. This puts us in a critical area of responsibility without technical visibility.

In order to fill this gap, organizations may have to check the behavior of the software. But copyright, and anti-circumvention doctrines make this inquiry tricky. The doctrines of the US fair use doctrine and the EU statutory exceptions to copyright do not typically recognize the necessity of organizational assurance.[2] They don't say whether a deploying organization should be allowed to carry out technical checks, which are proportionate to the deployed software and that demonstrate that the software acts as stated.

This article is proposing a "bifurcated model" that strikes a balance between the law, allowing for "audit reverse engineering," and maintaining IP rights. This is for proportionate, limited scope technical inspections carried out by a lawfully authorized actor, for example a compliance officer, auditor or regulator, with lawful access to the system. It would still be strictly conditional: the purpose of the inspection would be documented; it would be carried out in the strictest of confidence; and it would not lead to commercial copying or reuse of code. In contrast, "exploitative reverse engineering" as it can be called, whether by code stealing, extraction of trade secrets or commercial replacement would continue to be completely lawful under intellectual property, contract and cybercrime laws.[3]

This modest change will not compromise software protection. Instead, it makes sure that copyright and trade secret laws do not become barriers to untrue compliance claims. As modern governance laws are becoming more and more demanding for a verifiable proof of data protection, the legal system must offer a lawful way to be able to technically verify it. Otherwise, software law will continue to uphold the opacity which information-governance law is trying to break.

II. Why Organisational Information Governance Requires Auditability

Technical verifiability is a key aspect of modern information governance. But as regulations evolve toward more modern approaches, relying on paper-based assurances instead of proving accountability, a shift from assertion to proof is needed, particularly for proprietary cloud services, AI platforms, and compliance modules.[4] In vendor relationships, organizations have legal, professional and contractual responsibility for information assets, such as personal information, source code, and credentials, and the underlying code, access pathways and measures for telemetry are under the control of the vendor.

The deploying organisation may have no technical visibility of the business to prove these claims, especially when a vendor states that a client's files are not used to train models. If a vendor says that a client's files are not used to train models, then the deploying organisation may not have the technical visibility to prove such a claim. Although standard audit rights are helpful, they fail to extend beyond the boundaries of proprietary code[5] when important operational features are obscured and concealed in opaque architectures, opacity becomes a structural governance failure. This imbalance highlights the need for a formal legal process to technical verification.

III.The study of the science of law as it relates to the Constitution.

This need is echoed by current law. Constitutional decisions on privacy in India like Puttaswamy and Aadhaar make it clear that the law must offer substantive and not declaratory protection in the context of technology that can process vast amounts of information and do so in a hidden manner. Related cases such as Canara Bank and PUCL have made it clear that privacy interests extend to third-party records, and that surveillance of communications should be accompanied by meaningful and reviewable oversight.

In summary, the doctrines of Carpenter and Schrems I and II support the notion that, where the ability to verify compliance is prevented by proprietary software that is undertaking compliance critical functions, an auditability mechanism must be inherent to the requirements of accountability in the EU.

This chapter examines the American, European, and Indian frameworks for the protection of IP goods. In this chapter, the author looks at the American, European and Indian approaches to IP goods. Even with this governance requirement, there are IP systems which limit the technical acts to be used for verification. In the United States, fair use doctrines allowing an analysis of functional elements apply to Sega v. Accolade and Sony v. Connectix cases, but they do not go beyond interoperability and competitor compatibility; Section 1201 of the Digital Millennium Copyright Act (DMCA) imposes penalties for circumventing technological measures that control access; and private contract terms often bar reverse engineering entirely.

The EU and the UK also have carve-out exceptions for decompilation, testing and observation in the Software Directive, which are limited to interoperability and functional utility – and are absent from Top System for error correction.[6]

Moreover, as Singapore's experience with copyright transplants has shown, the statutory exceptions that are too rigid and poorly designed do not fit the needs of doing copyright technical audits in the real world.[7]

IV.Proposed Bifurcated Model.

This rift can be addressed with the help of the model proposed in this article that I called "bifurcated model". In order to overcome this friction, this article suggests a model which I have called "bifurcated model" that differentiates between "audit reverse engineering" and "exploitative reverse engineering". Should the audit reverse engineering be implemented, it would provide a lawful, safe harbour for a technical inspection by an authorised body, including compliance officers, auditors or regulators, with appropriate access. The exemption would need to be documented for compliance or security purposes, have absolute confidentiality measures, and it would be prohibited to make any commercial copies or product substitutions.

At the same time, if the software is misused in a manner that is considered to constitute an exploitation of the software, such as through trade-secret theft, commercial copying or unauthorized access, it will continue to be fully actionable under intellectual property, contract and cybercrime laws.[8] This is a balanced, measured reform, which will ensure that software protection is not used to facilitate unverifiable claims of compliance, and overcome the central contradiction of information-governance law, which requires evidence of compliance, but software copyright provides opacity.

V.The Missing Category: Audit Reverse Engineering

Current exceptions to software – interoperability, error correction and fair use – are insufficient to determine when an organisation will need to know whether its proprietary vendor software meets regulatory, contractual, or security obligations. For this purpose we propose “audit reverse engineering”: a technical inspection under defined conditions of software or a deployed system, with the aim of proving a fact defined by the auditors, which is of particular importance for the governance of the system. Its primary purpose is to verify if a system operates within their parameters, e.g. if an AI system does not use internal client documents to train their model, route data through unapproved regions or have telemetry that is not disclosed.

The approach of audit reverse engineering is dependent on the relation between the actor (customer, independent auditor or regulator), purpose, and approach. Technical verification to be separated into two types: external "behavioural testing" and internal "structural inspection. There is also less intrusiveness in behavioural testing, which involves monitoring network traffic or reviewing logs, but this approach is less effective in addressing more nuanced compliance issues such as data retention, deletion logic and vendor access.[9] This "access question" is especially critical in AI auditing, as it is difficult to know if sensitive prompts are being removed and if metadata is leaking through distributed infrastructure using black-box testing.[10]

Importantly, audit reverse engineering will not violate protectable technical secrecy. The vendors have a right to protect the source code and optimisation methods. But this interest should not be allowed to come at the expense of respecting the vendor IP or provide any useful audit assurance on how the organisation's information assets are processed.[11]
Unlike a right to inspect code, audit reverse engineering is the last step on an "audit ladder; while the first step is based on standard documentation and questionnaires. [12] The second step is when non-intrusive behavioural testing takes place. The third level uses supervised inspection or escrowed inspection. Narrowly targeted and proportionate reverse engineering is only legally allowed if these less intrusive methods have not solved a specific and credible compliance or security issue. With adequate institutional design, courts and regulators can determine if the inspection was restricted to components that have relevance to the investigation, and if the strict confidentiality safeguards were followed.[13]

VI.The Misuse Boundary: Exploitative Reverse Engineering and Organisational Information Assets

The article's proposal therefore has two linked commitments. First, the law should recognise a narrow route for necessary technical verification where an organisation, regulator, or authorised auditor must assess whether a proprietary system handles organisational information assets in accordance with binding commitments. Second, that route must remain unavailable where the actor lacks authority, uses disproportionate means, extracts protected materials, or seeks commercial substitution. A mature framework should not ask only whether reverse engineering occurred. It should ask why it occurred, who authorised it, what was accessed, whether less intrusive means were available, how the resulting information was handled, and whether the activity remained directed toward governance rather than appropriation.
The next section turns to the reform question: how a safe harbour for audit reverse engineering can be designed without weakening protection against code theft, trade-secret misappropriation, unlawful circumvention, or breach-driven extraction.

VII. A Realistic Reform Model: A Conditional Safe Harbour for Audit Reverse Engineering

The preceding sections identify a narrow but consequential gap. Software law permits some technical examination for recognised purposes such as interoperability, functional analysis, error correction, and security research. Data-protection and information-governance law, meanwhile, expects organisations to demonstrate accountability, manage vendors, protect information assets, and maintain effective safeguards. Yet neither body of law clearly explains when an organisation with lawful access may conduct deeper technical inspection because ordinary audit methods cannot verify whether a proprietary system is operating within contractual, security, or regulatory limits[14].
The appropriate response is not a general right to reverse engineer software. It is a conditional safe harbour: a limited defence for technical acts that would otherwise attract copyright, anti-circumvention, contractual, or confidentiality objections, but only to the extent necessary for a documented audit. The safe harbour would not authorise unauthorised access, commercial copying, public disclosure of source code, or extraction of proprietary material for competitive use. It would instead create a controlled legal route for verifying compliance-critical system behaviour where less intrusive methods are insufficient.

A. The Audit Necessity Test

A party invoking the safe harbour should satisfy six cumulative conditions.
First, there must be a defined and credible governance concern. The applicant should identify the system, the relevant information asset, the specific legal, contractual, or security commitment at issue, and the factual basis for concern. General distrust of a vendor or curiosity about a product should not suffice. A suitable concern might involve suspected retention of confidential documents beyond the agreed period, unapproved support access, data transfer outside a promised region, undisclosed telemetry, defective deletion controls, or a failure of access-control, encryption, logging, or monitoring safeguards. The issue must be sufficiently specific that the technical inquiry can be directed toward verifying or disproving it[15].
Second, the actor must have lawful access or recognised authority. The safe harbour should be available to an organisation that lawfully licenses or uses the relevant system, a processor acting within its authority, an independent auditor retained under a valid engagement, a regulator acting under statute, or a court-appointed expert. It should not protect credential theft, intrusion into a system that the actor has no authority to access, use of stolen credentials, breach-driven acquisition of code, or attempts to obtain information from unrelated systems. This condition preserves the central distinction developed in Part V: accountable inspection begins with lawful access; exploitative reverse engineering does not.[16]
Third, reverse engineering must be necessary after an appropriate escalation process. Necessity does not require the applicant to prove that no other form of inquiry is imaginable. It requires a showing that less intrusive measures cannot answer the identified question with reasonable reliability. The actor should ordinarily begin with contractual documentation, vendor explanations, system inventories, data-flow records, configuration review, access logs, test accounts, traffic analysis, behavioural testing, certifications, and third-party assurance reports. Deeper technical inspection becomes justified only where those measures are unavailable, incomplete, contradictory, or unable to establish the relevant operational fact.[17]
Fourth, the method and scope of inspection must be proportionate. The audit should be limited to the components, configurations, interfaces, logs, or technical behaviour relevant to the stated concern. It should not become a general exploration of the vendor's product. An inquiry into data residency may require examination of routing logic, hosting configuration, support-access records, or telemetry endpoints; it does not ordinarily justify access to unrelated proprietary features. An inquiry into deletion controls may require inspection of retention settings, backup handling, and relevant logs; it does not justify copying an entire codebase. Existing software-law exceptions already demonstrate that intrusive technical acts can be limited by purpose, necessity, and restrictions on downstream use.[18]

Fifth, information obtained through the audit must be used only for verification, remediation, enforcement, or responsible disclosure. The safe harbour should protect compliance findings, technical reports, expert evidence, remediation instructions, contractual claims, regulatory submissions, and narrowly framed disclosures necessary to explain a material risk. It should not permit publication of source code, dissemination of model weights, disclosure of authentication mechanisms, creation of a competing product, or reuse of implementation details for commercial substitution. This is the non-substitution condition. It preserves the difference between learning enough to verify a compliance claim and appropriating protected expression or confidential know-how.[19]

Sixth, the audit must be documented and conducted under confidentiality safeguards. A party relying on the safe harbour should record the authority for access, the compliance question, the alternatives considered, the methods used, the material examined or copied, the persons given access, the controls applied to the audit material, the findings reached, and the eventual retention or disposal of protected materials. Confidential review environments, restricted access, secure storage, independent experts, sealed regulatory submissions, and protective orders should be used where appropriate. Documentation is not a mere administrative burden. It allows a court, regulator, or contracting party to distinguish a good-faith audit from an opportunistic attempt to obtain a vendor's technical information.[20]

B. Interaction with Contract, Circumvention, and Trade Secrets

A statutory safe harbour would be ineffective if a vendor could defeat it through a blanket "no reverse engineering" clause. Contract should continue to govern notice, scheduling, confidentiality, audit protocols, cost allocation, and secure review procedures. It should not, however, be able to eliminate a narrowly defined statutory defence where an organisation must verify a serious governance concern and has satisfied the conditions above. This is not an unfamiliar legislative technique. European software law already treats certain contractual terms contrary to specified statutory exceptions as void, while American decisions such as Bowers v. Baystate Technologies, Inc. demonstrate how contractual restrictions can otherwise close the practical space left by copyright law.[21]

The same principle applies to anti-circumvention law. A copyright exception alone is inadequate where technological protection measures prevent access to the material necessary for the audit. The safe harbour should therefore include a corresponding and strictly limited defence to circumvention liability. It should apply only where the actor has lawful access to the system, has satisfied the Audit Necessity Test, and bypasses no more protection than is required to inspect the relevant component. It must not authorise an actor to obtain access to unrelated systems, disable security controls for operational use, or evade payment, licensing, or authentication mechanisms for commercial advantage.[22]

Trade-secret protection should likewise remain intact. The safe harbour should protect access for verification, not public exposure of confidential information. A vendor's source code, system architecture, optimisation methods, and security-sensitive technical details may remain protected even where an auditor is permitted to inspect them under confidentiality. The relevant distinction is between review for a defined governance purpose and improper acquisition, use, or disclosure of a trade secret. Secure technical review, confidentiality obligations, limited copying, and restrictions on further use allow these interests to coexist.[23]

C. Institutional Access and Practical Application

The safe harbour should not give every participant the same inspection powers. A customer organisation should ordinarily receive the documentation, evidence, and contractual audit rights necessary to manage its own risks. An independent assessor should receive controlled access only where ordinary review is inadequate. Regulators and courts may require stronger powers where statutory authority, public interest, or litigation necessity justifies them. This tiered model avoids two equally unhelpful outcomes: unrestricted public access to proprietary systems and complete vendor control over the evidence needed to assess compliance. GDPR Article 28(3)(h) already reflects this general logic by requiring processors to make compliance information available and to allow and contribute to controller or auditor inspections.[24]

The running example illustrates how the model operates. An organisation using an AI document-analysis platform first identifies a defined concern: the vendor may be retaining client documents or routing associated metadata outside the agreed region. It reviews the contract, data-processing terms, architecture diagrams, certifications, logs, configuration settings, and network behaviour. If those measures establish that the representation is accurate, no deeper inspection is justified. If they reveal contradictions or cannot reliably answer the question, the organisation may appoint an independent expert to inspect the relevant configuration, interfaces, telemetry behaviour, or narrowly identified code components within a secure review environment. The resulting report should state whether the promised control operates; it should not reproduce the vendor's source code or disclose unrelated technical detail.

This approach preserves the balance sought throughout the article. It gives organisations, regulators, and authorised auditors a realistic path to verify compliance-critical behaviour without converting compliance into a pretext for appropriation. It preserves copyright, trade-secret, anti-circumvention, and contractual protections against copying, extraction, and commercial substitution. Most importantly, it ensures that proprietary software cannot become an absolute shield against the verification of obligations that the deploying organisation remains legally and commercially responsible for meeting.

VIII. India and the Implementation of Organisational Auditability

India is not merely a secondary example in this debate. It is a useful implementation jurisdiction because it combines constitutional privacy, a major technology and services economy, existing computer-program exceptions, and an emerging statutory framework for digital personal-data governance. These features create a setting in which organisations increasingly rely on cloud services, outsourced processors, AI systems, security platforms, analytics tools, and compliance

software while remaining responsible for the information handled through those systems. The resulting problem is not limited to individual privacy notices or consumer-facing applications. It concerns whether organisations can obtain sufficient technical assurance that proprietary systems process, protect, retain, transfer, and secure information in accordance with the obligations they undertake.[25]

The Digital Personal Data Protection Act, 2023 provides an important future legal foundation, but its implementation must be stated carefully. The Act's central provisions on processing, notice, consent, fiduciary obligations, rights, transfers, and significant data fiduciaries were placed on a phased commencement timeline through the notification of 13 November 2025. The Digital Personal Data Protection Rules, 2025 adopt a similar approach: Rules 3, 5–16, 22, and 23 are scheduled to commence eighteen months after their publication. This section therefore treats the DPDP framework as an implementation opportunity rather than assuming that all operational obligations are already in force.[26]

When the relevant provisions become operative, the Act will make organisational auditability especially significant. Section 8 places general obligations on a Data Fiduciary and maintains its responsibility for processing undertaken on its behalf by a Data Processor. Section 10 provides for additional obligations of Significant Data Fiduciaries, including appointment of a Data Protection Officer, engagement of an independent data auditor, periodic data-protection impact assessments, and periodic audits. These requirements support the article's central proposition: a meaningful audit cannot be reduced to policies, declarations, or checklist compliance if the underlying technical system remains inaccessible to those responsible for assessing it.[27]

The Rules make the operational connection still clearer. Rule 6 requires reasonable security safeguards for personal data in the fiduciary's possession or control, including processing undertaken by a Data Processor. The listed measures include encryption or equivalent protections, access controls, visibility through logs, monitoring and review, contractual safeguards with processors, and appropriate technical and organisational measures. These provisions do not themselves establish a right to reverse engineer vendor software. They nevertheless show that Indian data governance is moving toward evidence-based control of systems and processor relationships. That movement makes the lack of a narrow technical-verification pathway more visible.[28]

The relevant Indian question is therefore not whether every customer organisation should obtain source-code access to every vendor tool. It is whether an organisation that remains accountable for information entrusted to a proprietary system should have a lawful route to verify a specific and credible concern after ordinary measures have failed. A company using a cloud-based document platform may need to verify whether documents are retained beyond agreed limits. A bank may need to examine whether a fraud-detection provider transmits transaction information to an unauthorised environment. A professional-services firm may need to confirm whether an AI provider uses uploaded client material for training or permits access by unapproved support personnel. In each example, the organisation's responsibility is real even though its technical visibility may be incomplete.

India's existing computer-program exceptions provide a starting point, but not a complete answer. Section 52(1)(ab) of the Copyright Act permits limited acts necessary to obtain information essential for interoperability, while section 52(1)(ac) permits observation, study, or testing of a computer program to determine the ideas and principles underlying its elements while performing authorised acts. These provisions demonstrate that Indian copyright law is not hostile to all technical examination by lawful users. Their present language, however, is directed to interoperability and functional understanding rather than organisational assurance. A future reform should recognise audit reverse engineering as a separate and limited legitimate purpose, subject to lawful access, necessity, proportionality, confidentiality, and non-substitution.[29]

The Information Technology Act, 2000 remains essential to the boundary of that reform. Its provisions on unauthorised access, copying, extraction, computer-related offences, breach of confidentiality, and disclosure in breach of lawful contract should continue to apply to hacking, credential misuse, data exfiltration, code theft, and breach-driven acquisition of proprietary material. The proposed safe harbour should sit alongside these protections, not override them. Its function is to distinguish a documented and proportionate audit of a system to which the actor has lawful access from unauthorised technical intrusion.[30]

Comparative law suggests that India should avoid two unsatisfactory models. The first is a narrow transplantation of interoperability language that leaves no room for verification of compliance-critical software behaviour. The European framework provides useful structure through observation, testing, interoperability, and error-correction rules, but its exceptions remain purpose-bound. The second is a reliance on flexible litigation doctrines alone. American fair-use decisions offer some space for intermediate copying, yet anti-circumvention law and contractual restrictions can leave legitimate technical inspection uncertain. India should instead adopt a clearer statutory baseline that directly addresses organisational auditability while preserving strong restrictions on copying, substitution, and misuse.[31]

Cross-border processing increases the importance of that baseline. The DPDP Act contemplates transfers of personal data outside India subject to restrictions that the Central Government may specify, while the Rules similarly address transfer conditions. In practice, vendor systems may use distributed cloud infrastructure, overseas support teams, sub-processors, and remote service environments. The legal consequences will depend on the applicable statutory framework, contractual commitments, processing activities, and governmental restrictions; they cannot be determined from geography alone. Yet these arrangements make technical verification more necessary because an organisation may otherwise have no reliable means of testing where information travels, who can access it, or whether a promised control operates in production.[32]

India can therefore develop an implementation model that is both protective of software markets and serious about information governance. The strongest route would be a narrow statutory safe harbour, supported by regulatory guidance and contractual practice. Copyright law could recognise limited technical acts necessary for a documented audit. Vendor contracts could require information rights, evidence of controls, independent-assessor access, and secure review

procedures. High-risk processing arrangements could be expected to preserve audit trails, configuration evidence, processor assurances, and escalation pathways before deeper inspection is considered. This would build on the logic of independent audit and processor accountability without converting auditability into public access to proprietary systems.[33]

The broader contribution is not to treat software secrecy as illegitimate. India's software, outsourcing, and technology sectors depend on meaningful protection for code, trade secrets, confidential know-how, and secure systems. The point is narrower: proprietary status should not make legally relevant operational facts permanently unverifiable where an organisation remains responsible for the information handled by the system. A carefully bounded audit route would allow India to reconcile constitutional privacy, statutory data governance, software copyright, cyberlaw, and commercial confidentiality without sacrificing any one of them entirely.

IX. CONCLUSION

Modern information-governance law increasingly requires organisations to demonstrate that they protect, control, and lawfully manage the information entrusted to them. Software law, however, often protects the very systems through which those obligations are performed. Organisations may depend on proprietary cloud services, AI tools, security platforms, data-loss-prevention systems, compliance modules, fraud-detection products, and outsourced processors to handle confidential business information, client records, employee data, security logs, credentials, transaction information, and personal data. Yet the relevant software may be opaque, vendor-controlled, and protected by copyright, trade-secret claims, anti-circumvention rules, and contractual restrictions. The result is a persistent gap between organisational responsibility and technical visibility.[34]

That gap cannot be solved through contractual assurances, certifications, questionnaires, or high-level compliance documentation alone. These measures are important and should ordinarily be the first route to assurance. But they may not reveal whether a deployed system retains information beyond an agreed period, routes data through unapproved infrastructure, enables undisclosed support access, creates hidden telemetry, uses uploaded material for prohibited purposes, or fails to implement promised security controls. Where a responsible organisation cannot verify a defined and credible governance concern through ordinary audit methods, opacity ceases to be merely a commercial inconvenience. It becomes a barrier to accountability.

This does not justify a general entitlement to inspect proprietary software. Software owners have legitimate interests in protecting source code, model architecture, security-sensitive implementation details, confidential engineering practices, and commercially valuable know-how. Copyright, trade-secret law, technological protection measures, and contractual confidentiality serve real functions. The argument of this article is therefore not that enterprise software should become open source, or that an audit allegation should automatically override all proprietary interests. It is that software secrecy should not become an absolute barrier to verification where an organisation remains legally, contractually, or professionally responsible for the information handled by that software.

The distinction between **protectable technical secrecy** and **audit-relevant operational facts** is central to that conclusion. A vendor may properly protect the expressive and confidential elements of its product. It should not, however, be able to make legally significant facts permanently unverifiable merely because they are implemented through proprietary systems. An organisation may need reliable evidence of what information is accessed, where it is processed, who may receive it, how long it is retained, whether access controls and logs operate, whether deletion takes place, whether sub-processors are involved, and whether contractual security commitments are actually reflected in the deployed system. These questions concern the operational reality of governance, not a customer's desire to appropriate the vendor's product.

This is the role of audit reverse engineering. Properly defined, it is neither hacking nor piracy nor commercial cloning. It is a controlled method of technical verification available only where the actor has lawful access; identifies a specific and credible governance concern; has reasonably exhausted less intrusive audit methods; limits inspection to the relevant component or behaviour; maintains confidentiality; documents the process; and does not reuse protected expression or technical information for commercial substitution. The proposed safe harbour is narrow by design. It gives organisations, regulators, courts, and authorised auditors a route to verify compliance-critical system behaviour without turning technical inspection into a licence for appropriation.

The corresponding misuse boundary is equally important. The safe harbour should not protect unauthorised access, credential theft, breach-driven extraction of source code or data, trade-secret misappropriation, unlawful circumvention, data exfiltration, indiscriminate disclosure of confidential material, or commercial replication of a vendor's software. A mature legal framework must distinguish audit from exploitation not by asking whether reverse engineering occurred in the abstract, but by asking who acted, under what authority, for what purpose, by what method, and with what subsequent use of the information obtained.

The comparative analysis shows that the legal building blocks already exist, though they remain incomplete. United States law has recognised limited intermediate copying where necessary to understand unprotected functional elements. European and United Kingdom law permit observation, testing, interoperability-related decompilation, and error correction within carefully bounded statutory conditions. Indian law permits limited observation, study, testing, and interoperability-related acts, while its cyberlaw properly addresses unauthorised access and extraction. These doctrines demonstrate that software law is not inherently opposed to technical inspection. What remains missing is an explicit category for proportionate technical verification undertaken to discharge organisational information-governance obligations.[35]

India has a meaningful opportunity to develop that category. Its constitutional privacy jurisprudence, statutory data-governance framework, software economy, outsourcing sector, cloud adoption, and growing use of enterprise AI create a

setting in which organisations will increasingly depend on systems they do not fully control. A carefully bounded audit safe harbour could allow lawful actors to verify compliance-critical operational facts while preserving strong protections against copying, secrecy breaches, cyber intrusion, and commercial substitution. It would complement, rather than displace, the duties of Data Fiduciaries, processors, auditors, regulators, and courts.[36]

The broader claim of this article is therefore modest but consequential. The law need not choose between intellectual-property protection and accountable information governance. It can protect proprietary software while ensuring that proprietary status does not make legally relevant system behaviour permanently unverifiable. Audit reverse engineering, narrowly defined and carefully controlled, offers a principled route between those objectives. It recognises that, where software performs governance-critical functions, accountability must be capable of being tested rather than merely asserted.

REFERENCES

Statutes and Regulatory Instruments

1. 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) [ECLI:EU:C:2016:679].
2. European Union: Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs (Software Directive) [ECLI:EU:C:2009:120].
3. India: The Copyright Act, 1957 (Act No. 14 of 1957).
4. India: The Digital Personal Data Protection Act, 2023 (Act No. 22 of 2023).
5. India: The Information Technology Act, 2000 (Act No. 21 of 2000).
6. Singapore: Copyright Act 2021 (previously Copyright Act, Cap. 63, as amended by the Copyright (Amendment) Act 2004).
7. United States: Copyright Act of 1976, 17 U.S.C. §§ 101 et seq. (incorporating the Digital Millennium Copyright Act of 1998, 17 U.S.C. § 1201).

Judicial Decisions

1. United States: Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).
2. United States: Sony Computer Entertainment, Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000).
3. United States: Lexmark International, Inc. v. Static Control Components, Inc., 387 F.3d 522 (6th Cir. 2004).
4. United States: Bowers v. Baystate Technologies, Inc., 320 F.3d 1317 (Fed. Cir. 2003).
5. United States: Carpenter v. United States, 138 S. Ct. 2206 (2018).
6. European Union: Top System SA v. Belgian State, Case C-13/20, [2021] ECLI:EU:C:2021:811.
7. European Union: Maximillian Schrems v. Data Protection Commissioner (Schrems I), Case C-362/14, [2015] ECLI:EU:C:2015:650.
8. European Union: Data Protection Commissioner v. Facebook Ireland Ltd and Maximillian Schrems (Schrems II), Case C-311/18, [2020] ECLI:EU:C:2020:559.
9. India: Justice K.S. Puttaswamy (Retd.) v. Union of India, (2017) 10 SCC 1.
10. India: District Registrar & Collector v. Canara Bank, (2005) 1 SCC 496.
11. India: People's Union for Civil Liberties (PUCL) v. Union of India, (1997) 1 SCC 301.
12. India: Selvi v. State of Karnataka, (2010) 7 SCC 263.

Books and Academic Articles

1. Bandopadhyay, Saptarishi. "Justifying the 'Back-Step': Establishing the Foothold of Reverse Engineering within the Indigenous Ethical Parameters of Software Copyright." Journal of Intellectual Property Rights 8, no. 3 (2003): 191–204.
2. Godfrey, Joseph. "Who's Afraid of Reverse Engineering?" Intellectual Property Quarterly [2024] 1 IPQ 50.
3. Seng, Daniel. "Reverse Engineering the New Reverse Engineering Provisions in the Copyright (Amendment) Act 2004." Singapore Journal of Legal Studies [2005], no. 1: 234–245.

End Note

1. Regulation (EU) 2016/679 (General Data Protection Regulation), Art. 5(2) (establishing the accountability principle); Kenneth A. Bamberger and Deirdre K. Mulligan, Privacy on the Ground: Governing Technology and Governing Privacy in the Information Age (MIT Press 2015).
2. 17 U.S.C. § 1201 (detailing anti-circumvention prohibitions under the Digital Millennium Copyright Act); Directive 2009/24/EC on the legal protection of computer programs, Art. 6 (limiting decompilation strictly to the achievement of interoperability of an independently created computer program with other programs, rather than general compliance auditing).
3. For legal frameworks governing unauthorized access, trade secret misappropriation, and commercial copying, see the Computer Fraud and Abuse Act (CFAA), 18 U.S.C. § 1030; Defend Trade Secrets Act of 2016, 18 U.S.C. § 1836; and Directive (EU) 2016/943 on the protection of undisclosed know-how and business information (Trade Secrets Directive).
4. See Regulation (EU) 2016/679 (General Data Protection Regulation), Art. 5(2) (detailing the accountability principle); Kenneth A. Bamberger and Deirdre K. Mulligan, Privacy on the Ground: Governing Technology and Governing Privacy in the Information Age (MIT Press 2015).

5. GDPR, Art. 28(3)(h) (obligating processors to make available information necessary to demonstrate compliance and allow for audits, though often restricted in practice by proprietary boundaries).
6. Case C-13/20, Top System SA v. Belgian State [2021] ECR I-838. See also Joseph Godfrey, 'Reverse Engineering and the Software Directive' (2022) 33 Intellectual Property Law Review 112 (analysing the limitations of current EU software exceptions for broader organisational audits).
7. See Daniel Seng, 'The Singapore Copyright (Amendment) Act 2004: A Review of the Reverse Engineering Exceptions' (2005) 17 Singapore Academy of Law Journal 45.
8. See, e.g., US Defend Trade Secrets Act of 2016, 18 U.S.C. § 1836; Directive (EU) 2016/943 on the protection of undisclosed know-how and business information (Trade Secrets Directive); and the Computer Fraud and Abuse Act (CFAA), 18 U.S.C. § 1030.
9. Inioluwa Deborah Raji et al., 'Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing' (FAT* '20, 2020) (discussing the operational limitations of external black-box testing).
10. On the limits of black-box evaluation for verifying algorithmic privacy safeguards, see Andrew D. Selbst, 'An Institutional View of Algorithmic Impact Assessments' (2021) 35 Harvard Journal of Law & Technology 117.
11. On the distinction between protectable expressive source code and non-expressive functional behavior, see Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).
12. This hierarchical approach matches proportionate auditing frameworks outlined in international information security standards. See, e.g., ISO/IEC 27001:2022 (Information security, cybersecurity and privacy protection).
13. Cf. Directive (EU) 2016/943 (Trade Secrets Directive), Art. 9 (prescribing judicial safeguards to preserve the confidentiality of proprietary information during inspections).
14. See Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520–28 (9th Cir. 1992); Sony Computer Entertainment, Inc. v. Connectix Corp., 203 F.3d 596, 602–08 (9th Cir. 2000); Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the Legal Protection of Computer Programs, Articles 5–6, 2009 O.J. (L 111) 16; GDPR arts. 5(2), 24, 28 & 32.
15. See GDPR, Articles 5(2), 24, 28 and 32.
16. See 17 U.S.C. section 1201; Information Technology Act, No. 21 of 2000, sections 43 and 66 (India).
17. See GDPR, Articles 5(1)(c), 25, 32 and 35.
18. See Directive 2009/24/EC, Articles 5–6; Top System SA v. Belgian State, Case C-13/20, ECLI:EU:C:2021:811 (Oct. 6, 2021).
19. See Directive 2009/24/EC, Article 6(2); Sony Computer Entertainment, Inc. v. Connectix Corp., 203 F.3d 596, 602–08 (9th Cir. 2000).
20. See GDPR, Articles 5(2), 24, 28(3)(h) and 32.
21. See Directive 2009/24/EC, Article 9(1); Bowers v. Baystate Technologies, Inc., 320 F.3d 1317, 1323–27 (Fed. Cir. 2003).
22. See 17 U.S.C. section 1201.
23. See 18 U.S.C. sections 1836, 1839.
24. See GDPR, Article 28(3)(h).
25. See Justice K.S. Puttaswamy (Retd.) v. Union of India, (2017) 10 S.C.C. 1 (India); DPDP Act.
26. See DPDP Act, section 1(2); Ministry of Electronics and Information Technology, Notification G.S.R. 843(E), Gazette of India, Extraordinary, Part II, Section 3(i), Nov. 13, 2025; Digital Personal Data Protection Rules, 2025, rule 1(2)–(4), G.S.R. 846(E), Gazette of India, Extraordinary, Part II, Section 3(i), Nov. 13, 2025[hereinafter DPDP Rules]..
27. See DPDP Act, sections 8(1)–(6), 10(1)–(2).
28. See Digital Personal Data Protection Rules, 2025, rules 1(4), 6.
29. See Copyright Act, No. 14 of 1957, section 52(1)(ab)–(ac) (India).
30. See Information Technology Act, No. 21 of 2000, sections 43, 66, 72 and 72A (India).
31. See Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the Legal Protection of Computer Programs, Articles 5–6, 2009 O.J. (L 111) 16; Top System SA v. Belgian State, Case C-13/20, ECLI:EU:C:2021:811 (Oct. 6, 2021); Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520–28 (9th Cir. 1992); Bowers v. Baystate Technologies, Inc., 320 F.3d 1317, 1323–27 (Fed. Cir. 2003).
32. See DPDP Act, section 16; Digital Personal Data Protection Rules, 2025, rules 1(4), 15.
33. See DPDP Act, section 10(2); DPDP Rules, rule 6; GDPR art. 28(3)(h).
34. See GDPR arts. 5(2), 24, 28 & 32.
35. See Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520–28 (9th Cir. 1992); Sony Computer Entertainment, Inc. v. Connectix Corp., 203 F.3d 596, 602–08 (9th Cir. 2000); Directive 2009/24/EC, Articles 5–6; Copyright Act, No. 14 of 1957, section 52(1)(ab)–(ac) (India); Information Technology Act, No. 21 of 2000, sections 43 and 66 (India).