

An AI-Based Early Detection, Risk Assessment, and Decentralized Security Architecture for Preventing Question Paper Leakage in High-Stakes Competitive Examinations

Samiksha Sanjay Jadhav, Prof. Bangar Abhishek P., Dr. Anand Khatri A., Prof. Bhosale Sachin B.

Department of Computer Engineering
Jaihind College of Engineering, Pune, India

Abstract- Leakage of question papers before a high-stakes competitive examination damages fairness, wastes institutional resources, and erodes public trust. Most technical research on online-exam integrity, however, concentrates on what happens after the paper reaches the candidate. This review examines three recent IEEE Access studies that represent three complementary defensive ideas: blockchain-backed storage of exam answers and results, screenshot-based activity monitoring in bring-your-own-device (BYOD) exams, and a multi-stage agent that joins IP-address screening with a deep-learning behaviour classifier. Each study is summarised, compared on technique, evaluation scale, and reported outcomes, and critically assessed. The comparison shows that decentralized tamper-evidence, behavioural analytics, and risk scoring are individually mature enough to be useful, yet none of the reviewed systems protects the question paper itself during preparation, storage, and distribution. The review closes by identifying the resulting research gaps and outlining directions towards an integrated architecture that unites AI-based early detection, risk assessment, and decentralized security for preventing question paper leakage.

Keywords— Question paper leakage, examination security, blockchain, risk assessment, behavioural analytics, deep learning, high-stakes examinations

I. INTRODUCTION

High-stakes competitive examinations, such as university entrance tests and public-sector recruitment examinations, decide the academic and career paths of very large numbers of candidates. Because so much depends on the outcome, even a single leaked question paper can force a cancellation, a costly re-examination, and a lasting loss of public confidence. A question paper changes hands many times before it reaches the candidate: it is set, moderated, stored, printed or transmitted, and finally released at the centre. Each of these steps is a possible leakage point, and several of them are exposed to insiders who are trusted by design.

The shift towards computer-based and online testing has added a digital layer to this chain. Research on securing such tests has grown quickly, yet most of it addresses misconduct by candidates

once the paper is already in their hands, for example impersonation, collusion, searching for answers, or tampering with recorded results. Three recent IEEE Access studies represent this work well. Abdelsalam et al. [1] use blockchain to keep exam results tamper-proof, Sukmana et al. [2] capture smartphone screenshots to observe examinee behaviour in bring-your-own-device (BYOD) exams, and Tiong et al. [3] combine IP-address screening with a deep-learning behaviour classifier.

The problem is also a matter of public policy. In India, repeated leakage incidents in recruitment and entrance examinations led Parliament to enact the Public Examinations (Prevention of Unfair Means) Act, 2024 [4], which names unauthorised access to, or leakage of, a question paper or answer key as an unfair means, provides for stricter punishment of organised offences, and extends liability to the service providers that support examinations. Legal deterrence, however, acts largely after an offence has

been committed. Technical controls that make a leak difficult to carry out, or quick to notice, are the complementary line of defence, and they are the subject of this review.

This paper reviews these three studies with one question in mind: how far do current exam-integrity techniques help in detecting early, assessing the risk of, and preventing question paper leakage? The contributions are threefold. First, each study is summarised and critically examined. Second, the studies are compared on technique, evaluation evidence, and coverage of the examination lifecycle. Third, the research gaps that remain for leakage prevention are identified, and directions are proposed for an architecture that unites artificial intelligence (AI), risk assessment, and decentralized security.

The remainder of the paper is organised as follows. Section II explains the review methodology, Section III reviews the selected studies, Section IV compares them, Section V discusses challenges and gaps, Section VI proposes research directions, and Section VII concludes.

II. REVIEW METHODOLOGY

1. Scope and Review Questions

The review is guided by three questions. RQ1: which techniques do the selected studies use to protect exam integrity? RQ2: what evidence supports their effectiveness? RQ3: which stages of the examination lifecycle do they cover, and what remains uncovered with respect to question paper leakage?

2. Selection of Studies

Three peer-reviewed research articles published in IEEE Access in 2024 and 2025 were chosen. Each (i) deals with the integrity of online or computer-based examinations, (ii) presents a concrete technical design rather than only a survey, and (iii) reports an evaluation with real or simulated examinees. Together they cover decentralized storage, on-device activity monitoring, and network-plus-behavioural analysis, which are the technology families most relevant to the topic. To place the findings in context, background works cited by these

studies are referenced where needed, and five further studies that target question paper protection were consulted at abstract level (Section III-D). This is a focused qualitative review of a small set of studies and not a systematic literature review, so its conclusions should be read within that scope.

Foundational material on blockchain, cryptography, anomaly detection, and insider threats is drawn from widely cited primary sources and surveys (Sections III-E to III-G). It is used only to explain the building blocks and not to extend the comparison. The reporting follows the general principle of stating selection criteria and limits openly, as recommended for software-engineering reviews [5], although the review does not claim to meet the full protocol of a systematic review.

3. Analysis Framework

Each study was examined for its problem statement, architecture, data and evaluation setup, reported results, and stated or apparent limitations. Numerical values quoted in this paper are those reported by the original authors. The mapping of studies to lifecycle stages and the identification of gaps are the reviewer's own interpretation.

III. LITERATURE REVIEW

1. Blockchain for Reliable Exam Results [1]

Abdelsalam et al. [1] begin from the observation that the Moodle learning management system (LMS) keeps exam data in a centralised MySQL database, where anyone with access can alter records. They therefore build a module that sends each completed exam to a smart contract on the Ethereum platform; a blockchain is a replicated ledger of hash-linked blocks [6]. The contract, written in Solidity, stores the student identifier, name, the score for each question, the final result, grade, completion date, and time taken. A mapping from wallet address to a Boolean flag allows only one submission per account. The authors also review earlier blockchain-based exam systems [7]–[9]. They note that one of these keeps student addresses in an ordinary database, where they can be altered [7], and that another uses ciphertext-policy attribute-based encryption (CP-

ABE) to restrict which teacher may set the questions of a test [9].

Identity is tied to a college-issued private key. The student imports this key into the MetaMask wallet, and the front end checks that the derived account belongs to the institution's private network before the exam opens. Submission requires the student to confirm a transaction in the wallet, which also pays the transaction fee. Data travels over HTTPS and is handled through the Web3.js library, and the Ganache tool provided a local test blockchain. An administrator dashboard reads results directly from the chain.

In a case study with 30 Business Information Systems students at Helwan University, the authors report that the data stored on the blockchain matched Moodle's own records exactly, with no loss. The design offers immutability, a built-in guard against resubmission, and key-based identity checks. Its weaknesses are also clear. The comparisons of transparency and failure probability are shown as bar charts without numeric values, and the authors themselves leave benchmarking of performance and scalability to future work. More importantly for this review, the system protects answers and results after the exam and does not address the confidentiality of the question paper before it.

2. Screenshot-Based Monitoring in BYOD Exams [2]

Sukmana et al. [2] study how to observe what an examinee does on a personal smartphone during a mobile LMS exam. They model a capture subsystem with the Systems Modeling Language (SysML) and implement it on Android using the Media Projection API, running as a foreground service inside the Edunex Mobile LMS. Screenshots are resized, compressed, and uploaded to MinIO object storage on the institution's private cloud. Three trigger designs were compared: a periodic timer, a user-interface change listener, and both combined, giving nine configurations tested in two modes, data collection and inference.

CPU, memory, energy, network, storage, screenshot count, and duplicate screenshots were profiled and

ranked with the TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) method using equal weights. A periodic timer at five-second intervals ranked first in both modes, with preference values of 0.6285 and 0.6352. Two simulated exam sessions with 45 students produced 4,510 screenshots, of which 3,422 showed exam screens and 1,088 did not. The non-exam screens fell into 17 categories, grouped as suspicious or dishonest. The most frequent dishonest actions were using a search engine (26 participants), opening web pages (16), and asking an AI assistant (15), a behaviour that has recently drawn research attention [10]. The authors also propose a cheating score that weights suspicious screens less than dishonest ones.

On usability, the system scored 80.8 on the UMUX-LITE questionnaire (grade A, 41 respondents). Compared with running the open-source ScreenLife tool beside the LMS, it used less memory (371.1 MB against 572.18 MB) and gave a higher frame rate (59.62 against 56.24 FPS). A binary exam versus non-exam classifier based on MobileNetV1 [11] reached a mean accuracy of 99.59% in five-fold cross-validation, slightly above a reconstructed baseline CNN at 99.39%. The authors note limits such as the Android-only scope, Indonesian multiple-choice content, privacy concerns (also studied in federated screen-activity tracking [12]), and heavy battery drain on an old device (34% in 30 minutes against 4% on newer phones). Two further points apply. The classifier separates exam interfaces from other app screens, a comparatively easy task, and classification into clean, suspicious, and dishonest classes is left for future work. The evidence gathered also mainly supports review after the event. The authors also link recording of exam content to the wider risk of content leakage, a risk raised earlier for proctoring systems [13].

3. IP Analysis and Behavioural Classification [3]

Tiong et al. [3] propose a multi-stage intelligent agent (multi-IA) for synchronous online assessments. Its first stage, the IP Detector, is rule-based. It compares a candidate's IP address with a stored list; if the address is already present or is highly similar (judged from the first and last two numbers), the invigilator is notified and a different question set is

assigned. The aim is to expose candidates who sit together or share a network.

The second stage, a Behavioural Monitor named DenseLSTM, combines long short-term memory (LSTM) [14] layers with dense connections [15]. Its input is a 23-element encoded vector: 20 elements mark each answer as correct or incorrect, and three describe the answering duration as long, normal, or short. The network labels a candidate as normal or suspected. It was trained on 192 records that mixed simulated misconduct with real data, and tested on 569 records from 301 candidates across three semesters at an anonymous university.

DenseLSTM reached accuracies of 91.96%, 88.78%, and 87.00% in the three semesters, with area-under-curve (AUC) values of 0.9561, 0.8997, and 0.8876, and it outperformed a multi-layer perceptron, a plain LSTM, and the instructors' own judgement. About 25–30% of candidates per semester were flagged as suspected. The authors acknowledge unreliable IP identification when candidates share a building, evasion through virtual private networks (VPN), connectivity problems, and a limited dataset. Two more observations apply. The input holds only correctness and timing, so it cannot show how the paper or the answers were obtained, and the labels appear to follow a behavioural definition adopted from earlier work [16], so the reported accuracy mainly measures agreement with that definition.

4. Related Work on Question Paper Protection

Beyond the three core studies, a smaller group of works targets the question paper itself. Islam et al. [17] propose a blockchain scheme for question sharing in which papers are encrypted in two phases using timestamps and hashes of earlier papers, and a smart contract with a timestamp-based lock prevents decryption before the permitted time. Rahman et al. [18] combine blockchain, digital signatures, hashing, and randomisation in a question paper management system that covers the path from paper creation to delivery. Yadav et al. [19] present a Truffle-based blockchain system that distributes PDF question papers through smart contracts, with role-based permissions and multi-signature authentication. Farhan et al. [20] describe

ASSURE-EXAM, which joins Ethereum smart contracts for access control, IPFS storage, and time-lock release, and report a working prototype with performance metrics. Majumdar [21] outlines a wider paradigm built on a permissioned blockchain, decentralized question sourcing to dilute insider risk, individualised papers generated through item response theory, and edge printing governed by a time lock. Only the abstracts of these five works were examined, so the descriptions are brief and their evaluations are not assessed here.

5. Foundational Building Blocks for Leakage Prevention

The blockchain used in [1] descends from the design introduced for Bitcoin [22], in which blocks of transactions are chained by cryptographic hashes and replicated across nodes. Ethereum generalised the idea by adding smart contracts that every node executes [23]. Two properties matter for exam security. A ledger is tamper-evident, because changing an old record changes its hash and breaks the chain, and Merkle trees [24] let many records be summarised by one root hash that is cheap to verify. At the same time, an open chain exposes data to all participants and adds cost and delay. Permissioned platforms such as Hyperledger Fabric [25], in which members are known, and consensus protocols that tolerate Byzantine faults [26] are therefore often preferred for institutional use. Surveys list scalability, privacy, and security as open issues of blockchain systems [27], and a catalogue of attacks on Ethereum contracts [28] shows that contract code can itself be a point of failure.

Blockchain has been proposed repeatedly for education. A European Commission report examined its potential for credentials and recognition [29], and EduCTX [30] proposed a blockchain-based platform for higher-education credits. These works concern records of achievement, which lie at the post-exam end of the lifecycle, and so they follow the same coverage pattern that Section IV-B finds in the reviewed studies.

Confidentiality of a question paper rests on conventional cryptography and not on the ledger. The Advanced Encryption Standard [31] is the

standard symmetric cipher for protecting content, and SHA-3 hash functions [32] provide digests for integrity checks. Content-addressed storage in IPFS [33] names each file by its hash, so a fetched fragment can be verified against a recorded value. For release control, time-lock puzzles and timed-release cryptography [34] were introduced so that information cannot be read before a chosen time, and the time-based locks described in [17] and [20] apply this idea through smart contracts. A time condition alone does not stop a key holder from leaking the key early. Threshold secret sharing [35] splits a key so that no single custodian can recover it, and attribute-based encryption [36] lets an access policy, such as role or time window, be built into the ciphertext, as [9] already does for question setters.

6. AI Techniques for Early Detection and Insider-Threat Analysis

Early detection of leakage is largely an anomaly-detection problem, because confirmed leakage events are rare and labelled examples are scarce. The survey by Chandola et al. [37] groups anomaly-detection methods into classification-based, nearest-neighbour, clustering, statistical, and information-theoretic families, and stresses that the right choice depends on the data and on whether labels exist. Isolation forests [38] isolate unusual points by random partitioning without needing labels and are cheap to compute, which suits first-stage screening of access logs, in the same spirit as the rule-based first stage of [3]. Sequence models such as the LSTM [14] used in [3] can capture ordered events, and Tuor et al. [39] applied deep networks to unsupervised insider-threat detection in security-log streams.

Insider threats form a research area of their own. Homoliak et al. [40] survey taxonomies, models, and countermeasures, and the central difficulty they describe is that insiders already hold legitimate access, so perimeter defences alone are inadequate. This matches the situation of setters, moderators, and printers in the examination chain.

Because an alert may lead to action against a named official or candidate, scores should be explainable. Model-agnostic attribution methods such as SHAP

[41] can show which features drove a score. Privacy-preserving learning can reduce the exposure of personal data: federated learning [42] keeps raw data on local nodes, as in the screen-activity work of [12], and differential privacy [43] bounds what released statistics reveal about any one individual.

7. Online Proctoring and Academic-Integrity Research

Automated proctoring has been studied for some time. Atoum et al. [44] describe a multimedia-based system that combines several sensing streams to detect cheating in online exams, and empirical work on online assessment reports measurable levels of cheating [45], [46]. The security and ethical sides of proctoring have also been questioned, with Slusky [13] examining cybersecurity weaknesses and Coghlan et al. [47] discussing the “big brother” concern. Like the three core studies, this literature centres on candidates during the examination, which reinforces the finding that the upstream part of the chain is under-studied.

IV. COMPARATIVE ANALYSIS

Table I summarises the three studies, and Table II compares the capabilities that matter for preventing question paper leakage. The ratings in Table II are the reviewer’s assessment based on what each paper reports.

Table 1: Summary of the Reviewed Studies

Ref.	Technique and data	Evaluation and key result
[1]	Ethereum smart contract with MetaMask, integrated into Moodle	30-student case study; results matched Moodle data, no loss reported
[2]	Android screenshots every 5 s; TOPSIS tuning; MobileNetV1	45 students, 4,510 screenshots; usability 80.8; 99.59% binary accuracy
[3]	Rule-based IP Detector plus DenseLSTM on answer and timing vectors	301 candidates, 569 records; accuracy 87.00–91.96%; AUC 0.8876–0.9561

1. Techniques and Evidence

The studies rely on different kinds of data. Study [1] works with ledger transactions, study [2] with images of the screen, and study [3] with answer sequences and IP addresses. Their evaluation scales also differ: 30 students in [1], 45 students and 4,510 screenshots in [2], and 301 candidates with 569 records in [3]. Only [2] and [3] report standard measures such as accuracy, AUC, or resource consumption, while [1] relies on agreement between stored and original data. In all three, the test population was a few university classes, far from the scale of a national competitive examination.

2. Coverage of the Examination Lifecycle

Figure 1 maps each study to five stages of an examination: paper setting, storage and transit, pre-exam release, the exam itself, and the post-exam stage of results. All three concentrate on the exam and post-exam stages. The first three stages, where a leak typically begins, are not covered by any of them.

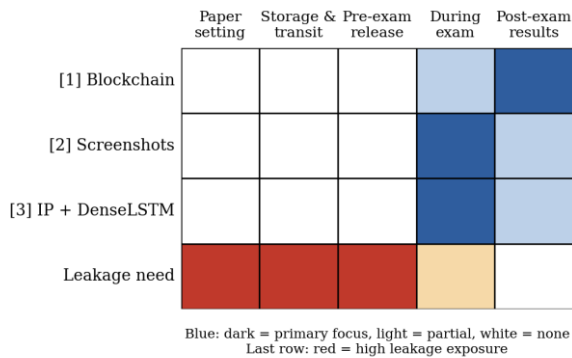


Fig.1. Examination lifecycle stages addressed by the reviewed studies (reviewer's mapping)

3. Complementary Strengths

The studies are nonetheless complementary. The blockchain in [1] supplies decentralized tamper-evidence and key-based identity. The screenshot capture and cheating score in [2] show how observed activity can be turned into a numerical risk indication. The multi-stage design in [3] shows that cheap rule-based checks can sit in front of a learned model. These ideas can be reused for leakage prevention if they are moved earlier in the examination chain.

Table 2: Capability Comparison Relevant to Question Paper Leakage

Capability	[1]	[2]	[3]
Tamper-evident record storage	Yes	No	No
Decentralized trust	Yes	No	No
Real-time behaviour observation	No	Yes	Partial
Risk scoring	No	Yes	Partial
Detects copying of exam content	No	Yes	No
Identity-based access control	Yes	No	Partial
Protects question paper before exam	No	No	No

Yes = reported; Partial = limited support; No = not covered.

4. Leakage Points and Candidate Controls

Table III links each stage of the examination chain to a typical leakage vector and to controls that the literature of Section III suggests. It is the reviewer's synthesis and shows where the reviewed techniques would have to be moved or supplemented. It is not an empirical result.

Table 3: Leakage Points and Candidate Technical Controls

Stage	Typical leakage vector	Candidate control
Paper setting and moderation	Setter or moderator shares or reuses draft questions	Role-based access, encrypted drafts, hash-chained audit log of every view and edit [9], [36]
Storage and transit	Copy taken from a shared drive, mail, or courier; one person holds the key	AES encryption, fragments on IPFS, hashes on a ledger, threshold key custody [31], [33], [35]
Pre-exam release	Early decryption or unauthorised access at the centre	Time-locked release, anomaly-scored access events [34], [38]
Printing and distribution	Photograph or soft copy leaves a	Multiple versions with fingerprinting,

	printing site or centre	edge printing [48], [21]
Exam and post-exam	Collusion, answer search, result tampering	Techniques reviewed in [1]–[3]

Controls are design candidates drawn from the cited literature and are not validated here.

The table shows that no single technique covers the chain. Cryptographic controls protect confidentiality at rest and in transit, time-based release addresses early access, behavioural analytics address misuse by authorised users, and fingerprinting supports traceability after a leak. The studies reviewed in Sections III-A to III-C occupy only the last row, which restates the central gap of this review.

V. RESEARCH CHALLENGES AND GAPS

1. No protection before the exam

None of the reviewed systems covers paper setting, storage, transit, or pre-exam release, although these stages are where leakage usually originates. Dedicated schemes are beginning to appear [17]–[21], but they lie outside the three studies compared here.

2. Integrity is not confidentiality

A blockchain makes records tamper-evident, but a replicated ledger is not secret by default. Placing sensitive content such as questions on a shared chain needs encryption and strict key management, which [1] does not discuss.

3. Insider threat

Leaks often involve people who legitimately hold the paper. The reviewed systems treat the candidate as the adversary and give little attention to setters, moderators, printers, or administrators.

4. Late detection

Study [2] mainly gathers evidence for later review, and [3] relies on answer and timing records, which exist only after candidates have already seen the paper.

5. Small, single-site, weakly labelled data

Evaluations involve one to three university classes, labels come from manual annotation or rule-based definitions, and no study supplies a dataset of real leakage incidents.

6. Scale, latency, and cost

Study [1] reports no benchmark, yet a national examination may involve very large numbers of candidates acting at once. Transaction fees and confirmation delay need study at that scale.

7. Evasion and privacy

On-device capture cannot see a second device, and IP checks can be bypassed with a VPN [3]. Screenshots and IP addresses are also personal data; study [2] blurs identifying content, but privacy-preserving design needs systematic treatment, and the ethics of exam supervision technologies are actively debated [47].

8. Key custody and single points of trust

Encryption moves the leakage risk to the keys. Key-management guidance [49] treats generation, storage, use, and destruction as lifecycle processes that need explicit policy. A contract-enforced time lock does not stop a key holder from leaking the key, which motivates threshold custody [35].

9. Smart-contract and platform risk

Contract defects and misconfigured permissions can defeat the intended guarantees [28], and public chains add cost and confirmation delay [27]. Auditing, formal checks, and a permissioned deployment need evaluation for examination use.

10. Explainability and due process

Automated flags may wrongly implicate staff or candidates. Alerts should carry explanations [41] and pass through human review, in line with the ethical concerns raised for exam supervision [47].

11. Legal and regulatory alignment

Leakage is an offence under national law [4], and logs about identifiable staff and candidates are personal data under data-protection law [50]. An immutable ledger may conflict with erasure and

minimisation expectations, so what is written on chain must be chosen carefully.

VI. PROPOSED RESEARCH DIRECTIONS

Closing these gaps points towards a layered architecture in which the techniques reviewed here are moved to the earlier stages of the examination chain. The directions below are design concepts for future work; they are not validated by the reviewed papers. They are consistent with the leakage-specific schemes in Section III-D [17]–[21], which indicate that encrypted storage, time-lock release, and access control are feasible. Behavioural risk scoring of access events, however, is not described in the abstracts examined.

The architecture proposed here has four layers. A confidentiality layer encrypts and fragments papers, a release layer enforces time-locked and threshold-controlled key release, a monitoring layer scores access behaviour and raises alerts, and a governance layer records custody events on a ledger and manages privacy, audit, and legal compliance. Each layer corresponds to rows of Table III and to gaps listed in Section V.

1. Encrypted and Decentralized Paper Storage

Question papers can be encrypted with the Advanced Encryption Standard (AES-256) and stored as encrypted fragments on a distributed file system such as the InterPlanetary File System (IPFS), while their hashes are recorded on a blockchain for tamper-evidence. This extends the idea in [1] from results to questions. A time-lock mechanism can keep the decryption key unavailable until the scheduled start of the exam, so that no single custodian can open the paper early.

2. Early Detection and Risk Scoring

User and entity behaviour analytics (UEBA) can assign a risk score to each access to the paper, using features such as time of access, location, device, and frequency. This follows the logic of the cheating score in [2] and the two-stage design of [3], where fast rules screen events first and a learned model refines the result. Unusual events would raise alerts before the exam begins.

3. Multi-Versioning and Traceability

Issuing several versions of a paper, with small controlled differences, allows a leaked copy to be traced back to the version, and hence to the custody chain, from which it came. It also lowers the value of a leak, because the version a candidate will receive cannot be known in advance.

4. Key Management and Threshold Custody

Encryption moves the leakage risk to the keys. Following the lifecycle view of key management in [49], key generation, storage, rotation, and destruction should be governed by explicit policy. A secret-sharing scheme [35] can split the paper key among several custodians, for example members of the setting committee and an independent authority, so that a threshold of them must cooperate at the scheduled time and no single custodian, however senior, can open the paper alone. Attribute-based policies [36] can further restrict which roles may decrypt which parts of a paper. Such a design also limits the damage from one compromised insider, the main threat noted in Section V.

5. Privacy, Explainability, and Governance

Access logs and alerts concern identifiable people, so only hashes and pseudonymous identifiers, and not personal data, should be written to an immutable ledger. This limits conflict with data-protection law [50] while keeping the audit trail verifiable. Where learning is used, federated training [42] and differential privacy [43] can reduce exposure of raw data, and each alert should be accompanied by an explanation [41] so that human reviewers can judge it fairly. Governance rules should state who may act on an alert, so that no sanction follows from a score alone.

6. Evaluation Roadmap

A future study should test the architecture with simulated leak scenarios, including insider access, and report detection latency, false alarm rate, throughput under peak load, transaction cost, and privacy impact. Public benchmark datasets of access logs with labelled leakage events would allow fair comparison, and data minimisation should be applied to any behavioural data collected.

A threat model should be fixed in advance, with at least three adversaries: an external attacker, a malicious insider with legitimate access, and colluding custodians. For each, the evaluation should report detection latency, true-alarm and false-alarm rates of the risk score (with unsupervised detectors such as isolation forests [38] as baselines), throughput and confirmation delay under peak load, cost per paper, time to recover a key after custodian failure, and the share of leaked copies correctly traced to their version.

VII. CONCLUSION AND FUTURE SCOPE

This review examined three recent studies on exam integrity: blockchain-backed result storage [1], smartphone screenshot monitoring [2], and IP analysis combined with a DenseLSTM behaviour classifier [3]. Each technique works well within its own scope, and together they show that tamper-evident records, activity observation, and learned risk indicators are practical. However, all three act at or after the examination itself, and none protects the question paper while it is prepared, stored, and distributed, which is where leakage occurs.

Future work should therefore build and test an integrated architecture that combines encrypted decentralized storage, time-locked release, risk scoring of access behaviour, and multi-versioning, evaluate it with realistic insider scenarios at national-exam scale, and treat privacy as a design requirement. Extending the comparison to a larger, systematically selected body of literature is another natural next step, as is a full-text comparison with the leakage-specific schemes [17]–[21].

This review has limitations. It examines three studies in depth and five more only at abstract level, the lifecycle mapping and capability ratings are subjective, and the legal discussion refers to one national framework [4]. The proposed architecture is a design concept that has not been implemented or tested here. Its feasibility rests on the cited building blocks and on schemes such as [17]–[21], and it should be confirmed through experiments.

REFERENCES

1. M. Abdelsalam, M. Shokry, and A. M. Idrees, "A proposed model for improving the reliability of online exam results using blockchain," *IEEE Access*, vol. 12, pp. 7719–7733, 2024, doi: 10.1109/ACCESS.2023.3304995.
2. Y. Sukmana, I. G. B. B. Nugraha, and Y. Rosmansyah, "Evaluating trustworthiness in BYOD online exams via smartphone screenshots," *IEEE Access*, vol. 13, pp. 216161–216189, 2025, doi: 10.1109/ACCESS.2025.3646711.
3. L. C. O. Tiong, Y. Lee, K. L. Lim, and H. J. Lee, "Advancing online assessment integrity: Integrated misconduct detection via Internet protocol analysis and behavioral classification," *IEEE Access*, vol. 12, pp. 106056–106069, 2024, doi: 10.1109/ACCESS.2024.3434608.
4. Government of India, "The Public Examinations (Prevention of Unfair Means) Act, 2024," Act No. 1 of 2024, *Gazette of India*, Feb. 2024.
5. [5] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele Univ. and Univ. Durham, U.K., Tech. Rep. EBSE-2007-01, 2007.
6. Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Jun. 2017, pp. 557–564.
7. A. Jain, A. K. Tripathi, N. Chandra, and P. Chinnasamy, "Smart contract enabled online examination system based in blockchain network," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2021, pp. 1–7.
8. J. Rooksby and K. Dimitrov, "Trustless education? A blockchain system for university grades," *Ubiquity, J. Pervasive Media*, vol. 6, no. 1, pp. 83–88, Nov. 2019, doi: 10.1386/ubiq_00010_1.
9. S. J. Pee, E. S. Kang, J. G. Song, and J. W. Jang, "Online test and management system using blockchain network," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2019, pp. 269–272, doi: 10.23919/ICACT.2019.8701891.
10. D. R. E. Cotton, P. A. Cotton, and J. R. Shipway, "Chatting and cheating: Ensuring academic integrity in the era of ChatGPT," *Innov. Educ.*

- Teaching Int.*, vol. 61, no. 2, pp. 228–239, Mar. 2024.
11. A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
 12. D. Mistry, M. F. Mridha, M. Safran, S. Alfarhood, A. K. Saha, and D. Che, "Privacy-preserving on-screen activity tracking and classification in e-learning using federated learning," *IEEE Access*, vol. 11, pp. 79315–79329, 2023.
 13. L. Slusky, "Cybersecurity of online proctoring systems," *J. Int. Technol. Inf. Manage.*, vol. 29, no. 1, pp. 56–83, Jan. 2020.
 14. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
 15. G. Huang, Z. Liu, G. Pleiss, L. van der Maaten, and K. Q. Weinberger, "Convolutional networks with dense connectivity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8704–8716, Dec. 2022.
 16. A. Ullah, H. Xiao, and T. Barker, "A dynamic profile questions approach to mitigate impersonation in online examinations," *J. Grid Comput.*, vol. 17, no. 2, pp. 209–223, Jun. 2019.
 17. A. Islam, M. F. Kader, and S. Y. Shin, "BSSSQS: A blockchain based smart and secured scheme for question sharing in the smart education system," 2018, *arXiv:1812.03917*.
 18. S. M. T. Rahman, M. N. Iftekhhar, R. Alam, S. M. R. Gafur, and D. Nandi, "Secured question paper management system," *AIUB J. Sci. Eng.*, vol. 20, no. 1, pp. 25–32, Apr. 2021, doi: 10.53799/ajse.v20i1.121.
 19. G. Hemanth Kumar Yadav, K. Adithya, T. Ehshanulla, M. Mokshith Srinivas Yadav, and Y. Gowrishankar, "Blockchain-powered question paper leakage prevention system," SciTePress, 2025.
 20. M. Farhan, M. Rajapakshe, R. B. Sulaiman, and U. Butt, "Assure-exam: A blockchain-based assurance framework for secure and transparent exam paper distribution," in *Proc. 2025 Annu. Comput. Security Appl. Conf. Workshops (ACSAC Workshops)*, 2025.
 21. P. Majumdar, "Architecting a cryptographically secure, AI-augmented paradigm for high-stakes educational assessments," *Am. J. Comput. Sci. Technol.*, 2026, doi: 10.11648/j.ajcst.20260902.14.
 22. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
 23. G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, 2014.
 24. R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology—CRYPTO '87* (Lecture Notes in Computer Science, vol. 293). Berlin, Germany: Springer, 1988, pp. 369–378.
 25. E. Androulaki et al., "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Apr. 2018, Art. no. 30.
 26. M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Design Implement. (OSDI)*, Feb. 1999, pp. 173–186.
 27. Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
 28. N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SoK)," in *Proc. Int. Conf. Princ. Security Trust (POST)*, 2017, pp. 164–186.
 29. A. Grech and A. F. Camilleri, "Blockchain in education," Publications Office of the European Union, Luxembourg, JRC Sci. Policy Rep. EUR 28778 EN, 2017.
 30. M. Turkanović, M. Hölbl, K. Košič, M. Heričko, and A. Kamišalić, "EduCTX: A blockchain-based higher education credit platform," *IEEE Access*, vol. 6, pp. 5112–5127, 2018.
 31. National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication 197, Nov. 2001.
 32. National Institute of Standards and Technology, "SHA-3 standard: Permutation-based hash and extendable-output functions," Federal Information Processing Standards Publication 202, Aug. 2015.
 33. J. Benet, "IPFS—Content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*.

34. R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," *Lab. Comput. Sci.*, Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. MIT/LCS/TR-684, Feb. 1996.
35. A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.
36. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy (SP)*, May 2007, pp. 321–334.
37. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.
38. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2008, pp. 413–422.
39. A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," in *Proc. AAAI Workshops*, 2017.
40. I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight into insiders and IT: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1–40, Mar. 2019.
41. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 4765–4774.
42. Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Art. no. 12, Feb. 2019.
43. C. Dwork, "Differential privacy," in *Proc. 33rd Int. Colloq. Automata, Lang., Programming (ICALP)*, 2006, pp. 1–12.
44. Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1609–1624, Jul. 2017.
45. S. Dendir and R. S. Maxwell, "Cheating in online courses: Evidence from online proctoring," *Comput. Hum. Behav. Rep.*, vol. 2, Aug. 2020, Art. no. 100033.
46. E. Bilen and A. Matros, "Online cheating amid COVID-19," *J. Econ. Behav. Organ.*, vol. 182, pp. 196–211, Feb. 2021.
47. S. Coghlan, T. Miller, and J. Paterson, "Good proctor or 'big brother'? Ethics of online exam supervision technologies," *Philos. Technol.*, vol. 34, no. 4, pp. 1581–1606, Dec. 2021.
48. D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897–1905, Sep. 1998.
49. E. Barker, "Recommendation for key management: Part 1—General," NIST Special Publication 800-57 Part 1, Rev. 5, May 2020.
50. Government of India, "The Digital Personal Data Protection Act, 2023," Act No. 22 of 2023, *Gazette of India*, Aug. 2023.