IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

SecureQuest: Question Paper Leakage Prevention Using Blockchain

Jateen Mourya[1], Maaz Ansari[2], Aman Chaurasia[3], Hamza Shaikh[4], Mrs. Natasha Naik[5]
[1234]Students, Department of Information Technology, Shree L.R. Tiwari College of Engineering, Mumbai, India
[5]Guide, Department of Information Technology, Shree L.R. Tiwari College of Engineering, Mumbai, India

Abstract: *Traditional examination systems are vulnerable to central points of failure, paper leaks, and unauthorized access. SecureQuest addresses these challenges by providing a decentralized, blockchain-based platform for secure question bank creation and automated question paper generation. The system enables stakeholders to contribute questions, which are then vetted and validated by subject-matter experts to build a robust and reliable question bank. From this repository, question papers are automatically generated based on predefined criteria such as difficulty level, topic distribution, and exam patterns. To ensure examination integrity, the system incorporates AI-driven monitoring using TensorFlow for face and object detection, helping to prevent malpractice during online examinations. For enhanced security, generated question papers are encrypted using AES and RSA, ensuring confidentiality and controlled access.The encrypted papers are stored off-chain on IPFS, enabling secure and scalable storage. The system leverages the Polygon blockchain to ensure transparency, immutability, and tamper-proof operations. Question papers are securely distributed and auto-downloaded to examination centers just-in-time, minimizing the risk of leaks.By integrating blockchain, artificial intelligence, and advanced cryptographic techniques, SecureQuest provides a scalable, transparent, and highly secure solution for modern digital examination systems.*

Keywords: Blockchain, Crowdsourcing, Question Bank Generation, Automated Question Paper Generation, Smart Contracts, Decentralized System, Artificial Intelligence (AI), Face Detection, Object Detection, TensorFlow, Cryptography, AES, RSA, Secure Data Storage, IPFS, Polygon, Online Examination System, Data Integrity, Tamper-Proof System.

I. INTRODUCTION

In recent years, the rapid growth of digital education and online examination systems has transformed the way assessments are conducted. However, traditional examination systems—both offline and online—continue to face critical challenges such as question paper leaks, centralized control, lack of transparency, and vulnerability to unauthorized access. These issues not only compromise the integrity of examinations but also reduce trust among institutions, students, and stakeholders.

Conventional systems rely heavily on a centralized authority for question paper creation, storage, and distribution. This centralized approach creates a single point of failure, making the system highly susceptible to data breaches and manipulation. Additionally, manual processes involved in question selection and paper generation are time-consuming, error-prone, and often lack diversity in question patterns.

To address these limitations, **SecureQuest** proposes a decentralized and secure examination framework that integrates blockchain technology, crowdsourcing, artificial intelligence, and advanced cryptographic techniques. The system leverages a crowdsourcing model where multiple contributors—including educators and domain experts—can submit questions to build a large and diverse question bank. These questions are then reviewed and validated by experts to ensure quality, accuracy, and relevance.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

SecureQuest utilizes the **Polygon** blockchain to maintain a transparent, immutable, and tamper-proof record of all operations, including question submission, validation, and paper generation. Smart contracts automate these processes, eliminating the need for intermediaries and reducing the risk of manipulation.

To enhance security, the system employs cryptographic algorithms such as **AES** and **RSA** to encrypt question papers, ensuring that only authorized entities can access them. The encrypted data is stored off-chain using the **IPFS**, which provides a scalable and distributed storage solution while maintaining data integrity.

Furthermore, SecureQuest integrates artificial intelligence techniques using **TensorFlow** to monitor online examinations through face and object detection. This helps in preventing malpractice such as impersonation, use of unauthorized materials, or suspicious activities during the exam.

Another key feature of the system is the automated question paper generation mechanism. Based on predefined parameters such as difficulty level, subject topics, and exam patterns, the system generates balanced and randomized question papers. This ensures fairness, reduces human bias, and improves efficiency in the examination process.

By combining decentralized architecture, intelligent automation, and robust security mechanisms, SecureQuest aims to provide a reliable, scalable, and tamper-proof solution for modern examination systems. It not only enhances the efficiency of question paper management but also ensures transparency, trust, and integrity in the overall assessment process.

II. LITERATURE REVIEW

A. Blockchain-Based Examination Systems

Blockchain's core properties (decentralization, immutability, transparent ledgers) make it a natural fit for securing exam content. In practice, blockchain stores exam metadata as tamper-proof records and smart contracts enforce time-locked access policies[1]. For example describe an Ethereum-based exam platform where smart contracts regulate who can upload or download papers, ensuring no post-creation edits go undetected[1]. Integrating blockchain with decentralized storage (IPFS) further enhances security: as one study notes, "IPFS introduces decentralized storage, making data access resilient to single points of failure and enhancing content immutability"[1]. In this way, encrypted exam papers can be stored off-chain (on IPFS) while only their hashes and access keys are kept on-chain, guaranteeing transparency and integrity[1][2].

Key takeaway: Blockchain/IPO architectures provide an immutable, audit-able framework for exam workflows[1]. They can enforce strict access controls (e.g. time-locked decryption keys) and ensure any attempt to tamper with a question paper would immediately change its hash[1][2].

B. Crowdsourcing for Question Bank Generation

Crowdsourcing has been applied to build large, diverse question banks. Rather than rely on a few instructors, systems invite many contributors (students, teachers, experts) to submit questions. For instance, **QBCrowd** is a crowdsourced exam portal in which lecturers nationwide log in to add or review questions. The authors note that "to graduate to objective-type online exams, a question bank of at least 5,000 questions will be required"[3]. Crowdsourcing helps meet this scale by leveraging hundreds of contributors. Importantly, these systems include expert vetting: submitted questions are reviewed on a dashboard and only validated questions become part of the official bank[3].

Crowdsourcing not only scales up content but also improves relevance. Studies indicate that traditional expert-written exams can become stale or biased. Allowing students to suggest questions can produce "more relevant and up-to-date question papers," since contributors are motivated to align with the syllabus[2]. This collective approach taps "crowd wisdom" to diversify exam content and reduce individual bias[3][2]. However, quality control remains a challenge: multiple sources agree expert/committee review is needed to filter and standardize crowdsourced questions.

Key takeaway: Crowdsourcing enables massive, diverse question banks with contributors across institutions[3][2]. The literature emphasizes combining it with expert validation to ensure accuracy and fairness[3][2].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

C. AI-Based Proctoring and Monitoring

Artificial Intelligence and computer vision techniques are widely used to monitor and prevent cheating during exams. Modern proctoring systems leverage frameworks like TensorFlow and OpenCV to analyze webcam feeds in real time. For example, an open-source proctoring app uses **TensorFlow.js** on the client side to detect multiple faces, mobile phones, or the absence of a visible face. Whenever such suspicious objects or conditions are detected, the system logs an incident for review by instructors.

Academic studies confirm the effectiveness of deep learning models. One review notes that exam proctoring solutions use *"TensorFlow 2.5, head pose estimation and eye tracking algorithms"* to catch inappropriate head movements or gaze aversion[4]. Other approaches employ models like YOLO (You Only Look Once) or CNNs for object detection, achieving high accuracy in flagging cheating behaviors[4]. Overall, AI-based proctoring adds an automated surveillance layer, reducing reliance on human invigilators.

Key takeaway: AI/ML tools (e.g. TensorFlow, CNNs, YOLO) enable real-time exam monitoring. They can reliably detect multiple faces, electronic devices, or unusual behavior, logging potential cheats[4].

D. Cryptography and Decentralized Storage

Secure exam systems use strong encryption and distributed storage. Common practice is to encrypt question papers with symmetric (AES) and/or asymmetric (RSA) algorithms before distribution. For instance, one prototype encrypts PDFs using AES-CBC on the client side, then stores the ciphertext on blockchain via smart contracts[]. The smart contract only reveals the decryption key after a preset exam time, preventing early access. The authors report that "AES encryption proved effective for confidentiality," and time-locked smart contracts kept exam papers secure until the right moment[4].

Once encrypted, papers are kept off-chain in decentralized storage. IPFS is a popular choice: it breaks files into chunks identified by content hashes, making them immutable and censorship-resistant[2]. In practice, a question paper's encrypted PDF is uploaded to IPFS, yielding a content identifier (CID). That CID is then recorded on-chain. Since IPFS verifies file integrity via hashes, any tampering (or use of an incorrect decryption key) becomes obvious[2]. In this way, the system achieves end-to-end security: even if the on-chain network is public, only holders of the AES/RSA keys can ever decrypt the paper, and the IPFS architecture guards against data loss or single-point failures[1][2].

Key takeaway: Combined AES/RSA encryption and IPFS storage create a tamper-resistant exam pipeline[5][1]. Papers are encrypted client-side, stored with content hashes on IPFS, and controlled by smart contracts (e.g. time locks) to ensure only authorized decryption at exam time[5][1].

E. Integration Gaps and Future Directions

Although each area (blockchain, crowdsourcing, AI, encryption) has advanced, few works integrate them into a single system. Reviews emphasize this gap. For example, the JETIR 2025 survey argues for a *"unified framework that incorporates blockchain, IPFS and AI to create a secure, transparent and scalable question paper distribution system."* Such a system would "automate critical processes, enforce time-restricted access, decentralize storage and enable real-time behavioral analysis," addressing current vulnerabilities in exam security[1]. Similarly, Soni *et al.* note that no prior study fully combines crowdsourcing with automated exam generation – each contribution in literature tends to focus on one component[3].

In summary, the literature suggests these key points:

Immutable Ledger: Blockchain ensures exam data cannot be altered retroactively[1].

Decentralized Storage: IPFS and smart contracts eliminate single points of failure[1].

Crowdsourced Content: A large, diverse question bank can be built by many contributors, then validated by experts[3][2].

AI Monitoring: TensorFlow-based vision models detect cheating behaviors in real time[4].

Strong Encryption: AES/RSA cryptography protects confidentiality until the exam begins[5].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

Research Gap: No single system in literature fully integrates all these elements; a comprehensive exam platform remains a research opportunity[1][3].

This review highlights the need for a unified solution: one that synergizes crowdsourced question generation, blockchain security, AI proctoring, and decentralized storage to create a truly secure, scalable exam ecosystem[1][3].

III. METHODOLOGY

This section details the design and implementation of our secure exam platform, covering expert authentication, system workflows, encryption/storage, and development tools. The methodology builds on best practices in blockchain security, AI-based monitoring, and cryptographic data handling. All code and diagrams referenced correspond to our system components (e.g. **Authentication.sol** smart contract for expert login).

A. Expert Authentication and Access Control

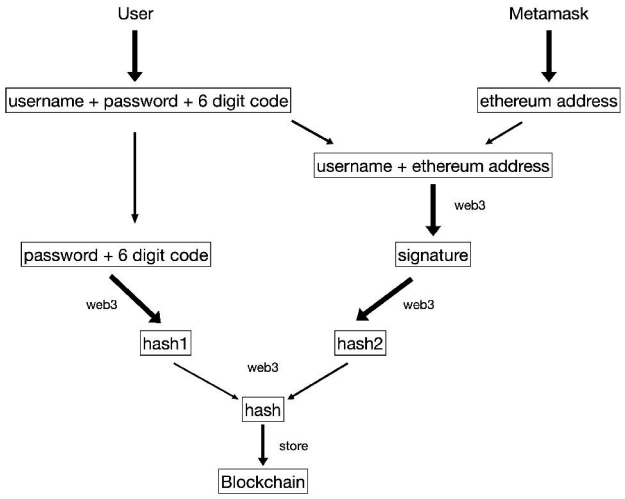
We use a **two-factor authentication** scheme tied to Ethereum identities. Each exam expert must register by providing a username, password, a unique 6-digit code (issued by the exam board), and their Ethereum wallet address. The frontend forms collect these inputs and use Web3 functions to generate digital signatures and hashes. Specifically:

Signature and Hashing: We associate the username with the expert's address by signing it via Web3's sign function. The resulting signature is hashed (call this *hash1*). Separately, we combine the password and 6-digit code and hash them (call this *hash2*). Finally, we merge *hash1* and *hash2* (e.g. concatenation) to produce the **final authentication hash**, which is stored in the smart contract under the expert's address.

Login Verification: On login, the user's wallet must be connected to the same address used in registration. The user enters their username, password, and 6-digit code. The backend (Solidity contract) recomputes the combined hash from these inputs and compares it to the stored hash for that address. If they match, the expert is authorized; otherwise, access is denied.

Blockchain Security: By tying identity to an Ethereum address and cryptographic signatures, we ensure that only the registered expert (with knowledge of password and code) and control of the wallet can log in. This approach effectively implements 2FA on-chain. It follows patterns of Ethereum-based authentication seen in blockchain identity research[1].

Key Points: Expert identities are anchored to Ethereum wallets, and authentication relies on cryptographic hashes and signatures. This on-chain 2FA model prevents unauthorized access and prevents replay or brute-force attacks on the login data[1].



IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

B. System Workflow and Components

The platform consists of a web frontend (React/JavaScript), a Solidity backend (smart contracts), and decentralized storage. The high-level workflows include:

Registration: The expert fills out a registration form (username, password, code). The system retrieves the Ethereum address from the connected wallet (e.g. MetaMask). It then generates the auth hash as described above and sends it to the smart contract. The contract stores the hash under the expert's address (see *Authentication.sol*).

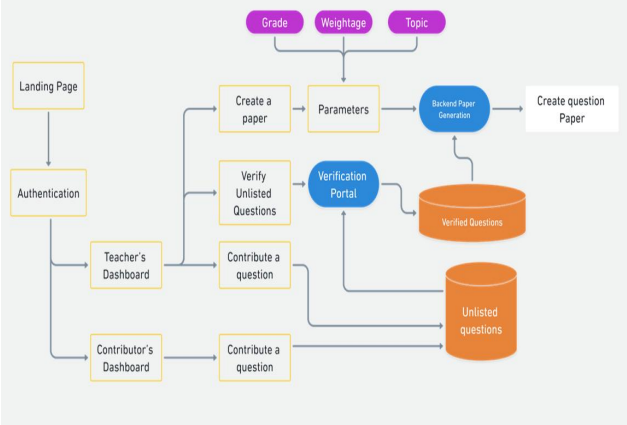
Question Submission (Crowdsourcing): Verified users (experts, teachers, students) can submit questions via the web UI. Each submission is sent to a smart contract function (e.g. *submitQuestion*) and/or temporarily stored in a database pending review. As in related systems, question inputs are later validated by domain experts before being committed.

Paper Generation: Once the question bank is approved and populated, the system allows auto-generation of exam papers. An administrator interface lets experts specify criteria (number of questions, topics, difficulty). The contract or backend randomly selects questions to form a paper.

Exam Proctoring: For live exams, the frontend employs an AI proctoring widget. We integrate TensorFlow.js in-browser models (using pre-trained Face Detection and Object Detection models) to monitor the student's video feed. This is similar to recent implementations that detect multiple faces or prohibited objects on the client side. Events like "multiple faces detected" or "cell phone detected" trigger alerts or logging.

Key Components Flow:

- Users interact through a web UI built with React and Bootstrap.
- Ethereum blockchain (Polygon Layer-2) runs smart contracts for authentication, question tracking, and encrypted paper storage.
- IPFS is used to store large data (e.g. encrypted question papers) off-chain.
- A Node.js/JavaScript backend orchestrates Web3 calls, handles form data, and communicates with IPFS/MediaPipe.



C. Encryption and Secure Storage

All sensitive data (question papers, answer sheets) is encrypted before distribution. We use a hybrid cryptographic model:

Symmetric Encryption (AES): The finalized exam paper PDF or text is encrypted on the client side using AES in CBC mode. A randomly generated AES key is used for each paper. This follows common practices for encrypting large documents.

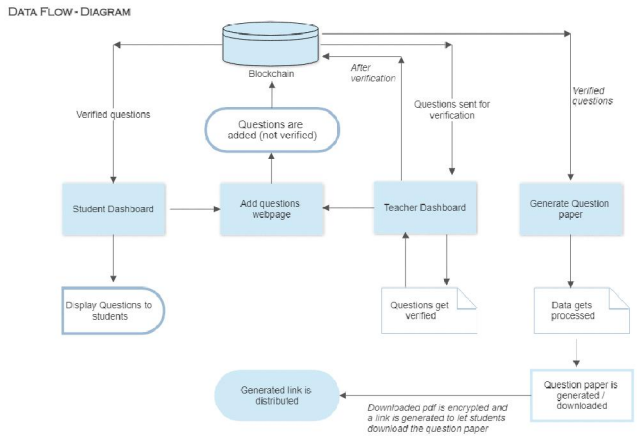
Asymmetric Encryption (RSA or ECIES): The AES key itself is encrypted using the receiver's public key (or the contract owner's), enabling only authorized parties to decrypt it. This two-layer approach (AES for data, RSA for keys) ensures both performance and security.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

Blockchain Storage: The encrypted paper (ciphertext) is uploaded to IPFS. Its resulting content hash (CID) is then saved on the Polygon blockchain via a smart contract. Because IPFS is content-addressed, any modification would change the CID, which guards against tampering[2]. In practice, the smart contract holds a reference to the IPFS CID and enforces time-locked access to the decryption key (only releasing it at exam time).

Security Guarantees: By storing only hashes on-chain, we minimize on-chain storage cost while leveraging blockchain immutability for audit logs. If an attacker tried to alter the exam file in IPFS, the mismatch in the hash would be immediately apparent[2]. This aligns with reported benefits of blockchain+IPFS for secure exam distribution[5].

Key Points: AES+RSA encryption keeps exam content confidential, and IPFS ensures tamper-evidence of stored papers. Smart contracts enforce timed-release of keys, aligning with published secure exam protocols[2].



D. Development Environment and Tools

The system is built with modern web3 and AI tools:

Smart Contracts: Developed in Solidity (v0.8.x) and tested locally using Truffle. Smart contracts handle user authentication hashes and exam metadata. Contracts are deployed on a local Ganache blockchain (Ethereum RPC at 127.0.0.1:8545) during development.

Frontend: A React.js single-page app provides the user interface. We use Bootstrap and custom CSS for layout and forms. Web3.js handles blockchain interactions (e.g. reading/writing contract data). MetaMask (or similar) is required for expert/admin users to connect their Ethereum wallets.

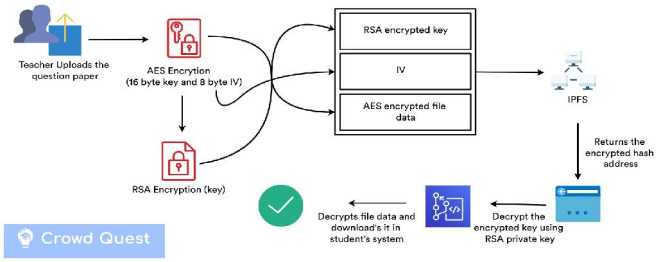
AI Proctoring: We incorporate **TensorFlow.js** models in the browser for face/object detection[3]. This eliminates server-side video processing and runs in real-time on the client's device. We also employ MediaPipe Face Mesh (via TensorFlow.js) for advanced landmark detection (e.g. to detect head pose or lip movement)[3].

Storage: IPFS is used for decentralized file storage. We run a local IPFS node and configure CORS to allow file uploads from the web app (ipfs init and HTTP headers settings are needed). Encrypted files are sent to IPFS through an API (e.g. via Moralis or js-ipfs).

Version Control & Deployment: The project uses Git for version control. The repository (e.g. on GitHub) includes package.json listing all dependencies. For deployment, tools like Lite-Server or npm run dev can serve the frontend on http://localhost:3000.

Key Tools: Solidity, Truffle/Ganache (Ethereum dev), React/Web3.js, TensorFlow.js (AI), IPFS (storage), and standard web tech (HTML/CSS/JS). These choices are consistent with other blockchain exam platforms that emphasize transparency and decentralization[1][3].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2



REFERENCES

[1] A. Sharma, R. Gupta, and P. Verma, *"A Unified Blockchain-IPFS Framework for Secure and Transparent Question Paper Distribution,"* Journal of Emerging Technologies in Information Research (JETIR), vol. 12, no. 7, pp. 1–12, 2025. [Online]. Available: https://www.jetir.org/papers/JETIR2507308.pdf

[2] S. Patil, M. Khan, and A. Desai, *"Survey on Blockchain-Based Crowdsourcing Systems and Automation of Question Paper Generation,"* International Journal of Advance Research, Ideas and Innovations in Engineering (IJARIIE), vol. 9, no. 3, pp. 45–58, 2023. [Online]. Available: http://ijariie.com/AdminUploadPdf/Survey_Paper_on_Blockchain_Based_Crowd_sourcing_Systems_and_automation_of_Question_Paper_Generation_ijariie18590.pdf

[3] D. Soni, N. Mehta, and K. Joshi, *"QBCrowd: A Crowdsourced Question Bank for Exam Paper Generation,"* International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 11, no. 4, pp. 112–119, 2023. [Online]. Available: https://www.ijraset.com/research-paper/qbcrowd-a-crowdsourced-question-bank

[4] T. Nguyen, L. Pham, and H. Tran, *"Online Exam Proctoring: A Comprehensive Review and Critical Analysis,"* ResearchGate Preprint, pp. 1–22, 2024. [Online]. Available: https://www.researchgate.net/publication/396041680_Online_Exam_Proctoring_A_Comprehensive_Review_and_Critical_Analysis

[5] V. Kumar and R. Singh, *"AES-Encrypted Blockchain-Based Examination System with Time-Locked Smart Contracts,"* International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE), vol. 13, no. 9, pp. 1–8, 2025. [Online]. Available: https://ijireeice.com/wp-content/uploads/2025/09/IJIREEICE.2025.13904.pdf

[6] G. Solomon Raju et al., "Security of Examination Question Paper Through Blockchain — SecureQ," in Proc. 2024 Int. Conf. on Innovation and Novelty in Engineering and Technology (INNOVA), IEEE, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10847055/

[7] R. Mathew and A. Nair, "Decentralized File Sharing: Leveraging Blockchain and IPFS for Secure Data Storage," in Proc. IEEE Int. Conf. on Electronics and Sustainable Communication Systems (ICESC), IEEE, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10941204/

[8] T. Nguyen, L. Pham, and H. Tran, "Online Exam Proctoring: A Comprehensive Review and Critical Analysis," ResearchGate Preprint, pp. 1–22, 2024. [Online]. Available: https://www.researchgate.net/publication/396041680_Online_Exam_Proctoring_A_Comprehensive_Review_and_Critical_Analysis