Credential Verification System Using ZK-SNARK: A Privacy-Preserving Blockchain Solution

Yashdeep Singh Negi
Dept. of Data Science & Business Systems
SRM Institute of Science and Technology
Chennai, India
yn6595@srmist.edu.in

Daksh Mamgain
Dept. of Data Science & Business Systems
SRM Institute of Science and Technology
Chennai, India
dm5565@srmist.edu.in

Samarth Ale
Dept. of Data Science & Business Systems
SRM Institute of Science and Technology
Chennai, India
sa8997@srmist.edu.in

***Abstract*—Traditional academic credential verification systems require students to disclose complete transcripts, marksheets, and personally identifiable information to employers, universities, and screening bodies. Although this approach enables verification, it exposes sensitive data that is often unnecessary for the verifier and increases the risk of privacy loss, misuse, and forgery. This paper presents ZKCert, a privacy-preserving credential verification system that combines blockchain-based credential anchoring with Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs). In the proposed framework, credential commitments are stored through a smart contract, while students generate cryptographic proofs that demonstrate eligibility conditions such as valid degree possession or minimum grade satisfaction without revealing exact academic records. A backend verification engine validates these proofs using snarkjs, and a web-based dashboard enables both proof generation and employer-side verification. The proposed design improves confidentiality, preserves trust through immutability, and supports scalable digital verification for use cases such as placements, scholarships, admissions, and other academic screening processes.**

***Index Terms*—credential verification, blockchain, zk-SNARK, privacy-preserving systems, Circom, snarkjs, smart contracts**

I. Introduction

Credential verification has become an essential component of modern academic and professional workflows because employers, scholarship agencies, and universities routinely need to validate a student's qualifications before allowing access to placements, admissions, or financial opportunities. In traditional systems, students are required to submit full academic records even when the verifier only needs a limited eligibility outcome, such as whether the candidate has obtained a valid degree or whether the candidate's cumulative grade point average satisfies a threshold. This creates a significant privacy problem because personal marks, subject-level scores, and additional details are exposed even when they are not relevant to the decision. At the same time, centralized verification systems depend on trust in document issuers and storage providers and remain vulnerable to tampering, forgery, and data leakage. Blockchain technology offers integrity and transparency, but many existing solutions still rely on full certificate disclosure during the final verification stage. Zero-knowledge proofs provide a more suitable alternative because they allow one party to prove the truth of a statement without revealing the underlying secret data. This paper presents ZKCert, a privacy-preserving credential verification framework that combines blockchain for immutable credential anchoring with zk-SNARKs for selective disclosure, thereby allowing students to prove academic eligibility without revealing exact marks or transcript contents while still enabling efficient employer-side validation.

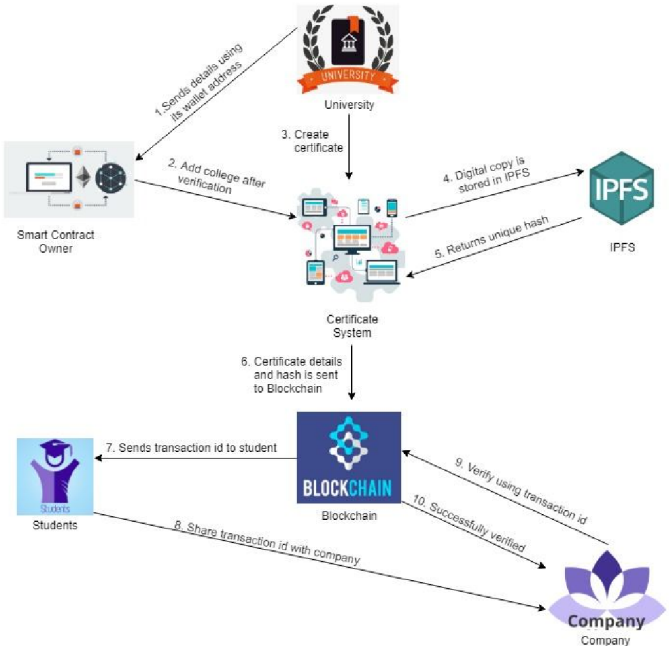

Fig. 1. Overall system architecture of ZKCert showing issuer, blockchain, student, backend verification, and employer modules.

II. Related Work

Blockchain-based trust systems were initially popularized through Bitcoin, where Nakamoto demonstrated that a distributed cryptographic ledger can maintain integrity without relying on a central authority [1]. Ethereum later extended

TABLE I
COMPARISON OF CONVENTIONAL VERIFICATION AND ZKCERT

Property	Traditional	ZKCert
Full transcript sharing	Yes	No
Selective disclosure	No	Yes
Tamper resistance	Medium	High
Issuer traceability	Limited	High
Privacy protection	Low	High
Automated verification	Partial	Yes

this idea by adding smart contract functionality, thereby allowing decentralized applications to manage logic as well as state [2]. These developments encouraged researchers to consider blockchain for educational credentials, digital certificates, and tamper-resistant verification systems, but most practical implementations still expose the complete certificate when verification is requested. In parallel, zero-knowledge proof systems evolved rapidly, and Groth proposed efficient pairing-based non-interactive arguments that improved proof compactness and verification efficiency [3]. Ben-Sasson et al. further advanced succinct non-interactive zero-knowledge systems and demonstrated that complex computational statements can be proven efficiently [4]. Although these cryptographic tools are highly promising, many existing academic credential solutions remain either purely theoretical or weakly integrated into real recruitment workflows. Most do not support minimal disclosure, employer dashboards, or policy-based academic filtering. ZKCert addresses this gap by combining blockchain-backed commitments, zk-SNARK eligibility proofs, and a usable front-end and back-end workflow that reflects actual placement, scholarship, and admission scenarios.

III. PROBLEM STATEMENT

Existing credential verification systems force students to reveal much more information than is necessary for a verifier to make a decision, and this overexposure becomes especially problematic in large-scale environments such as campus placement drives, scholarship portals, and admission platforms where the same transcript may be submitted repeatedly to multiple organizations. The central problem addressed in this work is how to verify academic eligibility conditions without exposing sensitive academic data while still preserving authenticity, integrity, and trust in the issuing institution. Current systems either depend on manual checking, which is slow and vulnerable to human error, or on centralized databases, which create single points of failure and raise questions of access control, tamper resistance, and long-term auditability. A privacy-preserving design must therefore achieve four goals simultaneously: it must allow trusted issuers to register credentials, permit students to prove statements about those credentials without exposing the hidden data, provide verifiers with fast and reliable outcomes, and maintain an immutable trail that prevents silent manipulation of verification records.

IV. OBJECTIVES

The primary objective of this project is to design and implement a working prototype of a credential verification system that leverages blockchain and zk-SNARKs to achieve selective disclosure in academic verification workflows. More specifically, the system aims to store credential commitments on-chain rather than raw academic records, generate proofs for eligibility policies such as degree validity or minimum CGPA, verify those proofs through a backend engine using snarkjs, and expose all major features through a simple user-facing dashboard suitable for students and employers. Another important objective is to ensure that the system remains understandable and practical at a minor-project scale, meaning that each component, including the smart contract, circuit design, API layer, and front-end interface, can be demonstrated clearly in a project review or research presentation. In addition, the work aims to highlight the novelty of combining privacy-preserving proof systems with real screening workflows, thereby moving beyond purely theoretical zero-knowledge examples and toward an actual deployable academic verification model.

V. PROPOSED SYSTEM

The proposed ZKCert system consists of five major components, namely the issuer, the blockchain layer, the student proof generator, the backend proof verification engine, and the verifier interface. The issuer is a trusted institution such as a university that validates the original academic record and computes a cryptographic commitment over the credential attributes. This commitment is stored on the blockchain through a smart contract, ensuring that the record can be checked later without allowing its silent modification. The student receives the relevant credential data and uses a zk-SNARK circuit to generate a proof for a required policy, such as whether the CGPA is greater than or equal to a threshold or whether a valid degree exists. The backend engine supports proof generation and verification through Circom and snarkjs and interacts with the blockchain to confirm that the public commitment being referenced actually exists. The verifier, typically an employer or institution, submits a proof and receives a simple TRUE or FALSE result indicating whether the student satisfies the required condition. In this way, the system guarantees minimal disclosure because the verifier never sees the raw marks while still obtaining a trustworthy cryptographic decision tied to an issuer-backed blockchain commitment.

$$C = H(ID \parallel Degree \parallel CGPA \parallel Salt) \tag{1}$$

$$\mathcal{P} : (C \text{ exists on-chain}) \wedge (CGPA \geq \tau) \wedge (Degree = Valid) \tag{2}$$

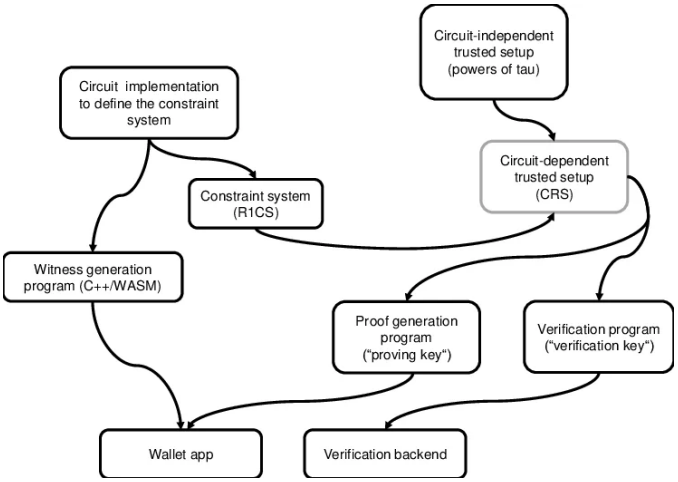

Fig. 2. zk-SNARK proving and verification workflow used in the proposed framework.

VI. System Workflow

The end-to-end system workflow begins when the issuing institution validates the student's academic information and generates a digital credential record containing the required attributes such as student identifier, degree status, and CGPA. A cryptographic commitment is then computed over these values together with a random salt and registered on the blockchain through the smart contract. After this, when a verification request arises, the student selects the relevant policy and uses the private witness values to generate a zk-SNARK proof. The generated proof is submitted to the verification engine, which first checks whether the associated commitment exists on-chain and then validates the proof against the public policy parameters. If both conditions hold, the employer-side dashboard displays a positive result, thereby confirming eligibility without exposing the underlying academic data. This workflow is particularly valuable because it allows the student to maintain privacy while still giving the verifier strong cryptographic assurance that the claimed condition is true and tied to a legitimate issuer-backed record.

VII. Methodology

The implementation begins with the creation of a Solidity smart contract that supports trusted issuer registration and commitment storage on a private Ethereum-compatible blockchain. For every verified student record, the issuer computes a commitment using a cryptographic hash over selected fields and a random salt so that the value stored on-chain reveals no direct academic information. After this, a Circom circuit is designed to encode two essential checks, namely that the witness values correspond to the public commitment and that the academic eligibility condition is satisfied. A trusted setup phase is then executed to generate the proving and verification keys required by the zk-SNARK protocol. The student uses the proving key and private witness values to generate a proof, while the backend verification engine accepts the proof, checks the corresponding commitment on-chain,

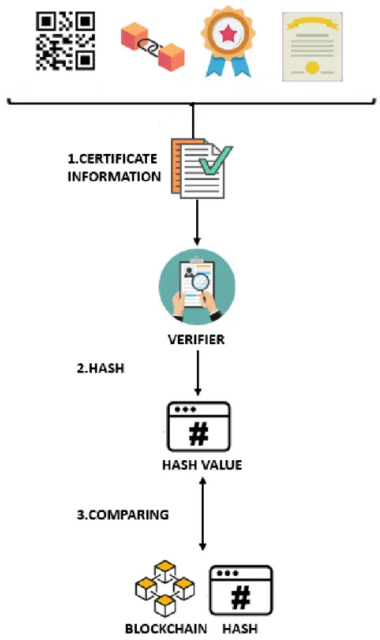

Fig. 3. Blockchain-based certificate verification process for hash-backed academic credentials.

and verifies the statement through snarkjs. A Node.js-based backend exposes proof submission and verification APIs, and a web interface allows the student to select a policy and generate a proof while the employer dashboard enables direct proof verification. The methodology therefore integrates blockchain immutability, zero-knowledge proof logic, smart contracts, and dashboard usability into a single end-to-end verification workflow.

VIII. Smart Contract Design

The smart contract serves as the trust anchor of the proposed architecture because it stores the credential commitments and issuer identifiers in an immutable, queryable form. Instead of writing full academic documents to the blockchain, the contract only stores compact commitment values, which reduces on-chain data exposure and keeps the system efficient. The contract includes functions for registering issuers, adding commitments, checking whether a commitment exists, and linking commitments to an issuer identity so that employer-side verification can be tied to a trusted institutional source. This design ensures that if a student attempts to present a proof for a credential that was never issued or never registered, the commitment lookup will fail before proof verification can succeed. The smart contract therefore acts as both a registry of trust and a lightweight integrity layer for the rest of the verification workflow, and its role is central to preventing credential forgery and maintaining long-term auditability of issued records.

TABLE II
CORE MODULES OF THE PROPOSED SYSTEM

Module	Responsibility
Issuer Module	Validates academic record and creates commitment
Smart Contract	Stores commitments and issuer mapping
Student Module	Generates proof from private credential data
Backend Engine	Verifies proof and queries blockchain state
Employer Dashboard	Displays verification outcome to verifier

IX. CIRCUIT DESIGN AND PROOF GENERATION

The zero-knowledge circuit is implemented using Circom and is responsible for encoding the verification logic that must be satisfied before a proof can be accepted. At a minimum, the circuit takes as private witness values the student's actual credential attributes and the random salt, and as public inputs the expected commitment and the policy threshold. Inside the circuit, the commitment is recomputed to ensure consistency with the public on-chain value, and logical constraints are added to confirm that the eligibility rule, such as CGPA greater than or equal to the threshold, is satisfied. Once the circuit is compiled, trusted setup is performed to generate the proving and verification keys required by the zk-SNARK scheme, after which the student can generate succinct proofs for valid credentials. The resulting proofs are compact enough for efficient transmission and fast enough to support practical screening use cases, while the hidden witness values remain confidential throughout the process. This step is critical because it transforms the problem from document disclosure into statement verification, which is the main privacy-preserving principle behind the proposed system.

X. BACKEND AND FRONTEND IMPLEMENTATION

The backend of ZKCert is implemented using Node.js and acts as the orchestration layer between the smart contract, proof system, and user interfaces. It exposes endpoints for commitment lookup, proof submission, and proof validation, and internally relies on snarkjs to verify the zk-SNARK proofs against the verification key and the public inputs. The frontend is designed as a lightweight browser-based dashboard, with separate views for student-side proof generation and employer-side proof verification. On the student side, the interface allows the user to enter or load the credential values required for proof generation, select the policy that must be satisfied, and trigger the proof-creation process. On the employer side, the verifier can submit a proof, trigger blockchain-linked validation, and receive a concise TRUE or FALSE result together with issuer-related trust information. This separation of responsibilities keeps the cryptographic logic mostly hidden from the user while making the overall workflow easier to understand, demonstrate, and extend.

TABLE III
SAMPLE PERFORMANCE OBSERVATIONS

Records	Commitment (s)	Proof (s)	Verify (s)
10	0.8	2.1	0.5
25	1.7	5.0	1.1
50	3.2	10.3	2.6
100	6.1	21.7	5.2

XI. EXPERIMENTAL SETUP

The prototype was implemented using Solidity for smart contract development, Ganache or a local Ethereum-style environment for blockchain deployment, Circom for circuit construction, snarkjs for proof generation and proof verification, Node.js for the backend service layer, and a browser-based dashboard for front-end interaction. Synthetic academic credential records were used to simulate student transcripts, valid degrees, and threshold-based eligibility requirements. The experiments focused on validating system correctness, measuring approximate commitment generation, proof generation, and proof verification time, and checking whether invalid inputs were successfully rejected by the system. Testing was performed across multiple record sets of increasing size to observe whether proof generation remained practical for medium-scale verification scenarios such as college placement filtering or scholarship shortlisting. The goal of the experimental setup was not to provide industrial-scale benchmarking but to demonstrate that the proposed design can be implemented as a working and meaningful prototype in an academic project environment while still producing measurable results suitable for technical discussion.

XII. RESULTS

The prototype successfully demonstrated that privacy-preserving academic verification can be achieved without requiring verifiers to inspect raw transcripts or marksheets. The smart contract correctly stored commitment values and preserved issuer-linked traceability, while the Circom circuit and snarkjs toolchain generated valid proofs for correct witness data and rejected invalid or inconsistent inputs. The measured results show that proof verification is significantly faster than proof generation, which is advantageous in practical settings because many employers or verifiers may need to check proofs while the student only generates a proof once per policy condition. The end-to-end system also proved effective from a usability perspective, since the employer dashboard presented the verification result in a simple, comprehensible form without exposing any cryptographic complexity or academic details. These results indicate that the proposed architecture is not merely conceptually secure but also practically feasible within the scope of academic screening workflows.

XIII. SECURITY AND PRIVACY ANALYSIS

From a security perspective, ZKCert provides stronger assurances than conventional document-based verification because it separates the proof of eligibility from the disclosure of raw data. A malicious candidate cannot simply fabricate

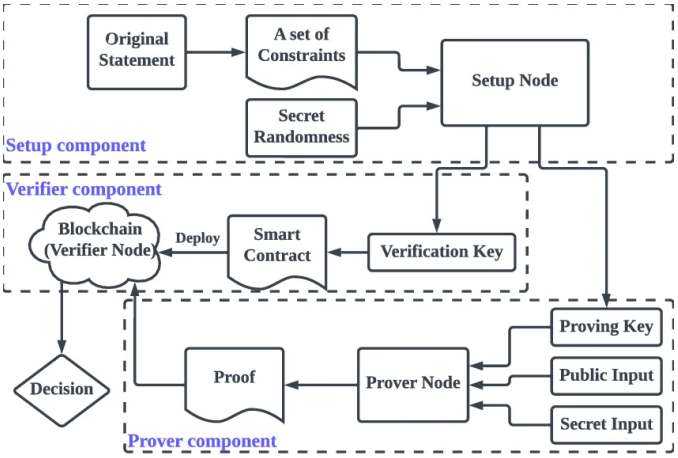

Fig. 4. General zk-SNARK flow illustrating prover, circuit, verification key, and blockchain-linked validation.

TABLE IV
THREATS AND MITIGATION IN ZKCERT

Threat	Mitigation
Forged credential	Proof must match issuer-anchored blockchain commitment
Excessive data exposure	Only policy satisfaction is revealed, not full transcript
Record tampering	On-chain commitments provide immutability
Verifier-side leakage	Verifier receives binary result instead of raw marks
Unauthorized trust claims	Issuer-linked registration limits accepted records

a transcript and claim compliance, since the corresponding proof must match an issuer-anchored on-chain commitment. A verifier cannot silently alter the credential state because the blockchain record is immutable once stored. A third party who intercepts the proof also learns very little about the underlying academic data because the proof only establishes the truth of the policy statement. This greatly reduces the risks associated with repeated transcript sharing, including data leakage, unauthorized reuse, and privacy erosion across multiple verification contexts. The system therefore offers a layered security model in which smart contracts protect record integrity, zk-SNARKs protect credential confidentiality, and the issuer-verifier linkage protects trust and traceability, making it suitable for privacy-conscious academic verification environments.

XIV. Discussion

Compared with traditional transcript-based verification, the proposed system provides stronger privacy protection, better tamper resistance, and more meaningful selective disclosure. It is particularly useful in situations where the verifier only needs to know whether a requirement is satisfied, such as whether a student has crossed a CGPA threshold or possesses a valid degree, rather than needing to study the entire transcript. The architecture is also conceptually extensible because the same proof-based approach can be adapted to other conditions such as subject-wise prerequisites, percentage cutoffs, or certificate validity periods. At the same time, the system still inherits some limitations common to zk-SNARK-based applications, including dependence on a trusted setup phase and a comparatively higher computational cost for proof generation. These costs are acceptable in many verification-heavy environments because proof verification remains relatively fast, but they still represent a practical consideration for wider deployment. Overall, the discussion suggests that the value of ZKCert lies not only in its security benefits but also in its ability to align privacy-preserving cryptography with recognizable real-world recruitment and academic workflows.

XV. Applications

The proposed system can support a wide range of real-world academic verification scenarios in which trust, automation, and privacy are simultaneously important. In campus placements, employers can verify whether a student satisfies minimum academic thresholds without requesting full transcripts from every candidate. In scholarship filtering, committees can check eligibility based on policy conditions while minimizing the handling of private educational records. In university admissions, candidates can prove prior degree validity or marks-based eligibility without repeatedly sending raw documents to multiple institutions. Similar ideas can also be extended to competitive examinations, fellowship applications, and digital certificate issuance systems where a simple decision outcome is sufficient for screening. Because the system separates policy satisfaction from full academic disclosure, it is especially attractive for high-volume processes in which repeated document handling would otherwise increase operational overhead and privacy risk.

XVI. Limitations and Future Work

Although the proposed system demonstrates the feasibility of privacy-preserving credential verification, it remains a prototype with several limitations that should be acknowledged. First, the evaluation was conducted on synthetic datasets and moderate-scale workloads rather than on large real-world issuer databases. Second, the current proof logic focuses on relatively simple policy conditions such as threshold checks and degree validity, whereas real institutional workflows may require more complex academic constraints, revocation models, and cross-institution interoperability. Third, the trusted setup step used in many zk-SNARK pipelines can be seen as a deployment concern if not handled transparently and securely. Future work can address these limitations by introducing richer circuit logic, supporting certificate revocation and expiration, integrating decentralized identity standards, evaluating performance on larger datasets, and exploring more scalable or transparent proof systems. Such extensions would move the system closer to deployment in operational academic and professional environments.

XVII. Conclusion

This paper presented ZKCert, a privacy-preserving credential verification system that combines blockchain-backed commitment storage with zk-SNARK-based selective disclosure to solve the problem of unnecessary transcript exposure in academic and professional verification workflows. The proposed design allows a student to prove academic eligibility conditions without revealing the exact grades or transcript values, while the blockchain ensures immutability and issuer-linked trust. The prototype demonstrates that smart contracts, Circom circuits, snarkjs verification, and dashboard-based interaction can be integrated into a practical end-to-end solution for placements, scholarships, and admissions. The results show that the system improves privacy, strengthens integrity, and preserves verification usability, thereby establishing that privacy and verifiability can coexist effectively in modern digital credential ecosystems. The work therefore contributes both a functional prototype and a concrete argument that privacy-preserving verification should play a larger role in future academic credential infrastructure.

Acknowledgment

The authors thank the project supervisor and faculty advisors for their guidance and support throughout the design and implementation of this work.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[2] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014.
[3] J. Groth, "On the size of pairing-based non-interactive arguments," in *Advances in Cryptology – EUROCRYPT*, 2016, pp. 305–326.
[4] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza, "SNARKs for C: Verifying program executions succinctly and in zero knowledge," in *Advances in Cryptology – CRYPTO*, 2013, pp. 90–108.
[5] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, 2014.
[6] A. Biryukov and D. Feher, "Zero-knowledge proof systems in practical applications: A survey," *IEEE Access*, vol. 10, pp. 112233–112250, 2022.
[7] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," in *Proc. NDSS*, 2018.