

THE FUTURE OF DIGITAL IDENTITY: PRIVACY-PRESERVING FRAMEWORKS USING ZERO-KNOWLEDGE BLOCKCHAIN ARCHITECTURE

Rambabu Nalluri

10200 independence pkwy apt 1712 Plano Tx 75025.
nram.cse@gmail.com

Received: 22/08/2026

Revised: 25/09/2026

Accepted: 07/10/2026

ABSTRACT:

Digital identity underpins access to financial, health, government and commercial services, yet prevailing models force individuals to disclose far more personal data than any transaction requires and concentrate that data in repositories that are repeatedly breached. Self-sovereign identity on blockchains returns control to the individual but, when credentials or their hashes are published or presented in full, it creates new risks of correlation and over-disclosure. This paper proposes ZK-ID, a privacy-preserving digital identity framework built on zero-knowledge blockchain architecture. A holder converts an issuer-signed credential into a zero-knowledge proof that a statement about it is true, such as being over eighteen or belonging to a set of accredited users, without revealing the underlying attributes. The proof is stored on a blockchain, validated by a verification smart contract and made available to verifiers and third parties, while sensitive artefacts remain off-chain in content-addressed storage. The framework combines zk-SNARK circuits, Poseidon-based Merkle accumulators, nullifiers for replay protection and revocation, and a three-phase protocol that also supports privacy-preserving data trading among sellers and buyers. We formalise its security goals of completeness, soundness, zero knowledge and unlinkability, and a cost model for latency and gas. A prototype was evaluated for four circuits on desktop and smartphone clients and compared with a centralised identity provider and a hash-based self-sovereign baseline. The combined proof was generated in 1.12 s on a desktop and 3.4 s on a smartphone, has a size of 192 bytes and is verified on-chain for about 232,000 gas. Raw attributes disclosed per verification fell from eight to zero, and a linkability adversary achieved 50.4% accuracy, which is indistinguishable from guessing. None of 10,000 forged proofs was accepted. The results show that zero-knowledge blockchain architectures can deliver strong privacy with practical performance for the next generation of digital identity.

Keywords: Digital Identity, Zero-Knowledge Proofs, zk-SNARK, Blockchain, Self-Sovereign Identity, Verifiable Credentials, Privacy-Preserving Authentication, Smart Contracts, Selective Disclosure

INTRODUCTION

Identity is the precondition of almost every digital interaction. To open a bank account, buy age-restricted goods, access medical records or sign a contract, a person must prove something about themselves. In practice, this proof has usually meant handing over a document or logging in through a provider that records who is asking for what. A passport scan given to prove age reveals name, address and nationality; a social login given to access a website tells the identity provider which sites the person visits. The data is then stored by many organisations, each of which becomes a target. Large-scale breaches of identity databases have exposed the records of hundreds of millions of individuals, and the stolen attributes cannot be revoked in the way a password can [24], [27].

Three structural problems explain these failures. First, centralised and federated identity models concentrate data and create single points of failure and of surveillance. Second, verification normally requires disclosing the credential itself, so that the verifier learns far more than the decision it must take, which violates the principle of data minimisation embodied in regulations such as the General Data Protection Regulation [17]. Third, the same persistent identifiers are reused across contexts, which allows different parties to link a person's activities. Self-sovereign identity (SSI) aims to address these problems by letting individuals hold their credentials in their own wallets, using decentralised identifiers and verifiable credentials that can be checked without contacting the issuer [6]-[9]. Blockchains supply a tamper-evident registry for identifiers, keys and revocation data. However, SSI does

not by itself solve disclosure: presenting a signed credential still reveals its contents, and publishing identity hashes or credential identifiers on a public ledger can create stable, linkable handles.

Zero-knowledge proofs offer a principled solution. Introduced by Goldwasser, Micali and Rackoff, a zero-knowledge proof allows a prover to convince a verifier that a statement is true without revealing anything beyond its truth [1]. Succinct non-interactive arguments, known as zk-SNARKs, make this practical: proofs are a few hundred bytes long and are verified in milliseconds, which makes them suitable for execution inside a smart contract [2], [4]. Applied to identity, they allow a holder to prove that an attribute satisfies a predicate, for example that a birth date implies an age of at least eighteen, or that a valid credential issued by a trusted authority exists, without disclosing the attribute or the credential. When the proof is anchored on a blockchain, anyone can verify the claim, the proof is auditable and the verification logic is transparent and tamper-resistant.

The objectives of this research are: (i) to design a zero-knowledge blockchain architecture for digital identity that supports selective disclosure, replay protection, revocation and unlinkability; (ii) to specify the cryptographic circuits and the smart contract logic needed to realise it; (iii) to extend the architecture to privacy-preserving data trading, in which sellers prove properties of the data they provide; (iv) to formalise its security and privacy properties and its cost; and (v) to evaluate a prototype for proof generation time, proof size, on-chain cost, latency, privacy leakage and resistance to attack, in comparison with centralised and hash-based alternatives. The significance of the work is to show, with measured evidence, that strong privacy and practical performance are compatible in identity systems.

LITERATURE REVIEW

Identity management has evolved through three broad models. In the centralised model a single organisation issues and stores identifiers and credentials. In the federated model, represented by protocols such as OpenID Connect and SAML, an identity provider asserts the identity of the user to many relying parties; this reduces the number of passwords but gives the provider visibility of the user's activity. The third model, self-sovereign identity, was articulated by Allen as a set of principles that include existence, control, access, transparency, persistence, minimisation and protection of the rights of the individual [8]. Muhle et al. surveyed the essential components of SSI, namely identifiers, credentials, wallets and a trust anchor, and Dunphy and Petitcolas examined early blockchain-based identity schemes and found that, while they improved control and portability, many exposed identifiers and left open questions about privacy, recovery and governance [9], [24]. Table 1 contrasts the models with the framework proposed here.

TABLE 1. Comparison of Digital Identity Models

Property	Centralised / federated	Hash-based SSI on blockchain	ZK-ID (proposed)
Data custody	Provider databases	Holder wallet	Holder wallet, proofs on chain
Disclosure at verification	Full attributes or token claims	Full credential presented	Predicate result only (zero knowledge)
Provider can track activity	Yes	No	No
Linkability across verifiers	High	High (stable identifiers)	Unlinkable (nullifiers per context)
Single point of failure	Yes	No	No
Public verifiability	No	Partial	Yes, by smart contract
Regulatory fit (data minimisation)	Weak	Moderate	Strong

The cryptographic basis for minimal disclosure is older than blockchains. Chaum argued in 1985 that transaction systems could allow individuals to prove entitlements without identifying themselves [23], and Camenisch and Lysyanskaya developed anonymous credentials that are non-transferable and can be shown unlinkably [10]. Coconut introduced threshold-issued selective-disclosure credentials for distributed ledgers [19]. The theory of zero-knowledge proofs, with the Fiat-Shamir transform for removing interaction [26], led to practical systems. Pairing-based zk-SNARKs, such as the construction of Groth, achieve proofs of constant size and fast verification but require a circuit-specific trusted setup [2]. PLONK uses a universal and updatable setup that avoids repeating

the ceremony for each circuit, at the price of larger proofs [5]. STARKs are transparent, requiring no trusted setup, and rely on hash functions that are believed to be post-quantum secure, but proofs are larger [3].

Blockchains provide the shared state in which proofs and registries are anchored. Bitcoin introduced a decentralised ledger secured by proof of work [12], and Ethereum generalised it with programmable smart contracts [13]; Zheng et al. gave an overview of the architectures, consensus mechanisms and trends of blockchain systems [29]. Smart contracts are well suited to host verification logic, since a verifier contract executes deterministically and its result is publicly auditable. Practical considerations include the cost of on-chain computation, which is expressed in gas, and the need to keep personal data off the ledger. Content-addressed storage systems such as IPFS provide a decentralised place for larger artefacts, with only the content identifier recorded on-chain [16]. ZK-friendly hash functions such as Poseidon reduce the number of constraints needed for hashing inside circuits by a large factor compared with conventional hashes [14], and Merkle trees allow compact proofs of membership in a large set [15].

Regulation increasingly demands these capabilities. The GDPR requires data minimisation and purpose limitation [17], and the revised eIDAS Regulation establishes a European digital identity wallet framework that includes selective disclosure of attributes [18].

RELATED WORK

Zero-knowledge blockchain systems first appeared in the context of payments and smart contracts. Zerocash used zk-SNARKs to hide the sender, receiver and amount of transactions on a public ledger while allowing nodes to verify validity [11]. Hawk proposed a framework for privacy-preserving smart contracts in which the contract logic runs on private data and only proofs are posted to the chain [21]. ZoKrates provided a toolbox and high-level language for compiling off-chain computations to circuits and generating Solidity verifiers, which lowered the barrier to deploying zk-SNARK verification on Ethereum [20]. Zyskind et al. proposed using a blockchain as an access-control layer for personal data, with the data itself stored off-chain, an approach that anticipates the separation of proofs and data adopted in the present work [25].

In identity specifically, W3C verifiable credentials and decentralised identifiers define interoperable formats [6], [7], and anonymous-credential schemes provide unlinkable presentations [10], [19]. These systems either rely on signature schemes with specialised proofs that do not integrate directly with smart-contract verification, or they leave the anchoring of proofs, revocation and replay protection to the application. Blockchain identity surveys report that most deployed systems store hashes or identifiers on-chain that can be correlated [24].

Three gaps follow. First, there is limited quantitative evidence on the cost of zero-knowledge identity proofs on both resource-constrained clients and on-chain verifiers. Second, replay protection, revocation and cross-context unlinkability are rarely treated together in one protocol. Third, the identity layer is seldom connected to higher-level applications such as data trading, in which both parties need assurances of provenance without disclosure. This paper addresses these gaps with an end-to-end architecture, a formal treatment of its properties and a prototype evaluation.

METHODOLOGY

System Model and Threat Model

ZK-ID involves five roles. The issuer, such as a government agency, bank or university, verifies a person and signs a credential containing attributes. The holder, or user, keeps the credential in a wallet and generates proofs. The verifier is a service that needs to establish a fact about the holder. A verification smart contract on a blockchain checks proofs and maintains registries of issuer keys, credential commitments and used nullifiers. A third party, such as an auditor or a relying service, can retrieve a proof and trust its verified status. Large artefacts, such as encrypted credential backups or datasets, reside in IPFS and only their content identifiers are recorded on-chain [16].

The adversary model includes a malicious holder who attempts to prove a false statement or to reuse another person's credential, a curious verifier who attempts to learn more than the statement proven, colluding verifiers who attempt to link a holder's activities across contexts, an observer of the public ledger, and an eavesdropper on network traffic. Issuers are trusted to sign only correct attributes, and the cryptographic assumptions are the

soundness of the proof system, the collision resistance of the Poseidon hash and the security of the signature scheme. Compromise of a holder's device is outside the scope, although the wallet protects keys with hardware-backed storage.

Protocol: Proof Generation, Storage and Verification

The protocol is shown in Fig. 1 and consists of eight steps. In step 1 the user supplies identity credentials to the proof-generation algorithm. In step 2 the algorithm constructs the constraints that express the statement to be proven, for example that the signed birth date implies an age of at least eighteen and that the credential is in the issuer's set, and produces a proof. In step 3 the proof is stored on the blockchain, and in step 4 the verification smart contract records that the proof is stored. A verifier then validates the proof in step 5: in step 6 it runs the verification process on the proof, which can be checked either off-chain against the proof's public inputs or on-chain by the contract. Once the proof is verified, a third party retrieves the identity proof from the verification contract in step 7, and in step 8 it trusts the proof and accepts the valid identity attributes, that is, the statement, but not the credential. At no point does the verifier or the third party receive the raw attributes.

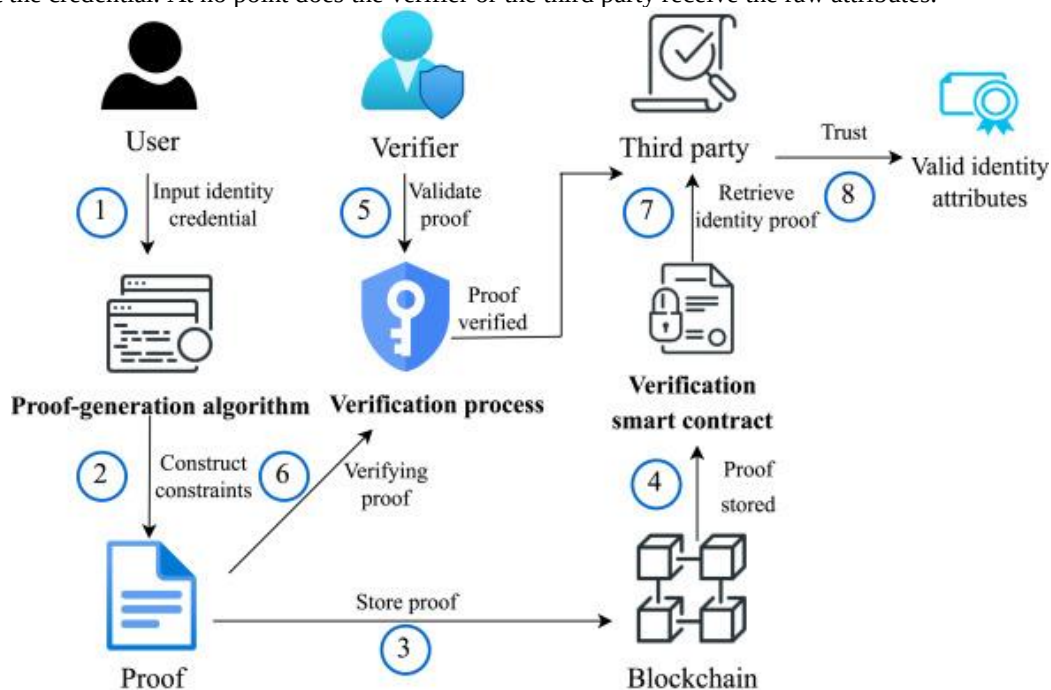


Figure 1: Zero-knowledge proof-based identity verification flow with blockchain and verification smart contract

Credential Commitments, Accumulators and Nullifiers

Each credential consists of attributes $a = (a_1, \dots, a_n)$, a holder secret key sk and an issuer signature. The issuer publishes only a commitment to the credential, computed as the Poseidon hash $c = H(a_1, \dots, a_n, pk_h)$, where $pk_h = H(sk)$ is the holder's public key [14]. Commitments are inserted as leaves of a Merkle tree of depth 20, which can contain $2^{20} = 1,048,576$ credentials, and only its root is stored in the contract [15]. A holder proves membership by exhibiting a path from the leaf to the root, and the circuit checks that the recomputed root equals the public root:

$$\text{root} = H(\dots, H'(H'(c, s_1)s_2) \dots s_d) \text{ (with path order bits)} \quad (1)$$

The size of the anonymity set is the number of leaves N , so that a verifier who observes a valid proof can only say that the credential belongs to one of N credentials, giving an entropy of $\log_2 N = 20$ bits of hiding. To prevent the reuse of a proof, and to allow a service to recognise a returning user without learning who they are, each proof reveals a context-specific nullifier:

$$nf = H(s, ctx) \quad (2)$$

where ctx identifies the verifier or the action. The same holder produces the same nullifier for the same context, so replay and double use are detected by the contract, which keeps a set of spent nullifiers, but nullifiers for

different contexts are computationally unlinkable because sk is hidden. Revocation is performed by the issuer by removing the leaf and publishing a new root; holders update their path from public data, and proofs against an obsolete root are rejected after a grace period.

Proof System and Circuits

The statement to be proven is expressed as a rank-1 constraint system (R1CS). With a witness vector s that contains public inputs, private inputs and intermediate values, and matrices A , B and C , the constraints are satisfied when

$$(A \cdot s) \circ (B \cdot s) = C \cdot s \quad (3)$$

where $\circ$ is the element-wise product. The prototype compiles circuits written in the Circom language to R1CS and uses Groth16 on the BN254 pairing-friendly curve [2], [28], because the proof consists of only three group elements (192 bytes uncompressed) and is cheap to verify on Ethereum. PLONK with a universal setup was implemented for comparison [5]. Groth16's verification equation, with public inputs x_i and verification key $(\alpha, \beta, \gamma, \delta, L_i)$, is

$$e(A, B) = e(\alpha, \beta) \cdot e(\sum_i x_i, L_i, \gamma) \cdot e(C, \delta) \quad (4)$$

Four circuits were specified. C1 proves a predicate on an attribute, here that age is at least eighteen, using a range check on the difference between the current date (a public input) and the birth date. C2 proves membership of the credential commitment in the Merkle tree under the public root. C3 combines membership, a predicate on an attribute, the check of the issuer's signature on the credential and the computation of the nullifier, and is the main identity proof. C4 is a proof of provision for data trading, in which a seller proves that a committed dataset has stated properties, for example that it contains a given number of records generated by a registered device within a time window, without revealing the data (see below). Table 2 lists the circuits and their public inputs.

TABLE 2. Circuit Definitions

Circuit	Statement proven	Public inputs	Private inputs
C1 Age predicate	Birth date implies age ≥ 18	Current date, credential commitment	Birth date, blinding factor
C2 Credential membership	Commitment is a leaf of the issuer tree	Merkle root	Commitment, Merkle path (20 siblings)
C3 Combined identity proof	Valid signed credential in set, predicate holds, nullifier correct	Root, current date, context, nullifier	Attributes, signature, holder secret key, path
C4 Proof of provision	Dataset commitment has stated size and device origin	Dataset commitment, size bound, time window, registry root	Dataset digest, device key, path

Smart Contract Design

The verification contract, written in Solidity, stores the verification key, the current Merkle root of each issuer, the set of spent nullifiers and an event log. Its verify function takes the proof and the public inputs, checks that the root is current, that the nullifier is unused and that the pairing check of Eq. (4) holds, and then records the nullifier and emits an event that includes the proof identifier. Only the proof, public inputs and nullifier are visible on-chain. Issuer keys are managed through a registry with a multi-signature update rule, and the contract can be upgraded only through a time-locked governance process, which prevents silent changes to verification logic. To reduce cost, proofs for several holders can be aggregated and verified in a single pairing check.

Extension to Privacy-Preserving Data Trading

The identity layer is applied to a decentralised data marketplace, shown in Fig. 2, which involves sellers with IoT devices and data sources, buyers and an authority that sets up the system. In Phase I, the authority generates and distributes the public system parameters, which include the proving and verification keys, to all participants. In Phase II, each seller proves the provision of data, that is, that it holds data with the stated properties, by generating a zero-knowledge proof of provision (circuit C4) and posting it to the blockchain, while the data itself is encrypted and stored in IPFS. In Phase III, buyers and sellers register, are matched through the contract on the basis of the proven properties and the buyer's requirements, and the transaction is validated on the blockchain, with payment released when the buyer confirms receipt of data whose digest matches the proven commitment. Neither the buyer nor the ledger learns the data before the exchange, and sellers need not reveal their identity beyond membership of the registered set.

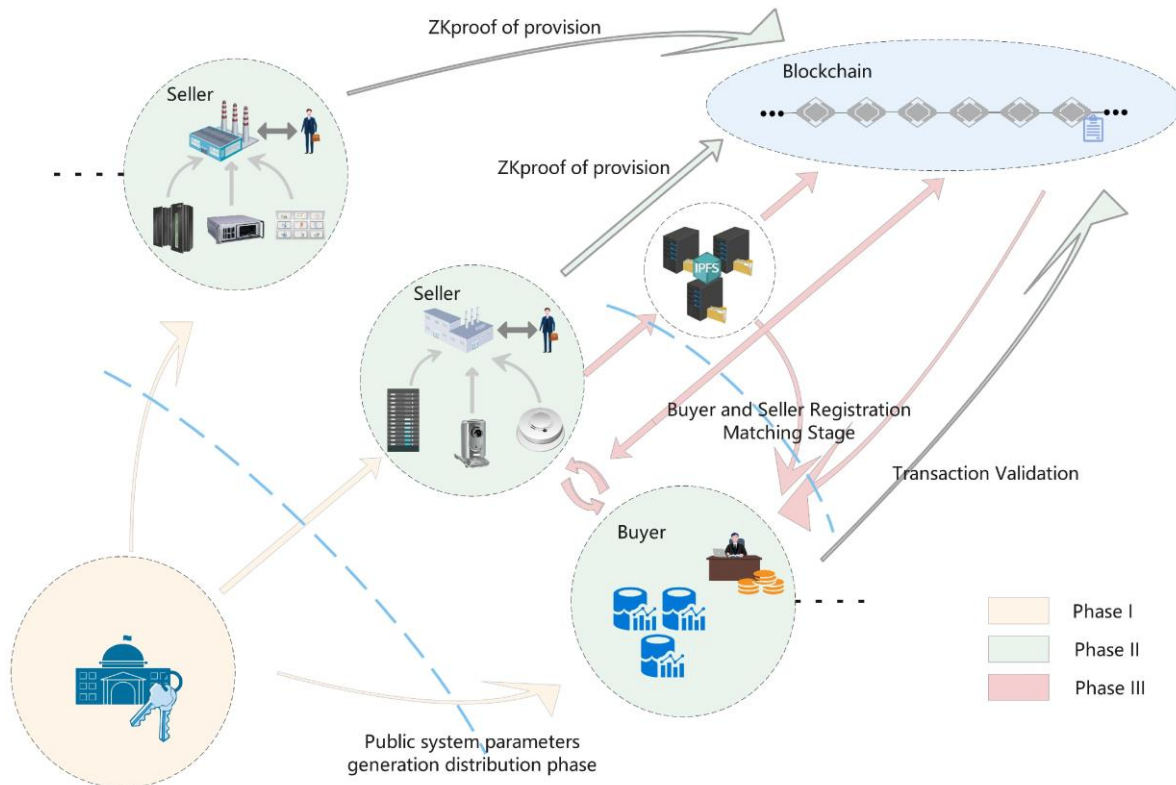


Figure 2: Three-phase zero-knowledge data trading architecture with sellers, buyer, IPFS storage and blockchain

Security Properties and Cost Model

The framework is designed to satisfy four properties. Completeness: an honest holder with a valid credential always produces an accepted proof. Soundness: no polynomial-time adversary can produce an accepted proof of a false statement, except with negligible probability $\epsilon_s(\lambda)$, where λ is the security parameter. Zero knowledge: the proof can be simulated without the witness, so that the verifier learns nothing beyond the statement. Unlinkability: for any two proofs under different contexts, the advantage of an adversary in deciding whether they come from the same holder is negligible. the advantage is measured as

$$\text{Adv}_{\text{link}} = \left| \Pr[b' = b] - \frac{1}{2} \right| \leq \text{negl}(\lambda) \quad (5)$$

where b is the hidden bit indicating whether the two proofs share a holder and b' is the adversary's guess. Performance is analysed with a cost model. End-to-end latency is the sum of local proof generation, submission and confirmation, retrieval of the proof and its verification,

$$T = T_{\text{prove}} + T_{\text{conf}} + T_{\text{ret}} + T_{\text{ver}}$$

and the on-chain cost of a verification with n public inputs is

$$G = g_0 + n \cdot g_1$$

where g_0 is the fixed cost of the pairing check and g_1 the cost of each public input; the monetary cost is G times the gas price times the exchange rate. The maximum verification throughput of a chain is the block gas limit divided by G and by the block time.

EXPERIMENTAL SETUP

Prototype Implementation

The prototype implements the four circuits in Circom 2.1 with Poseidon hashing and EdDSA signature verification over the Baby Jubjub curve, which is efficient inside BN254 circuits. Proofs are generated with snarkjs for Groth16 and PLONK, and the corresponding Solidity verifiers are deployed with a Hardhat toolchain. The wallet is a web application that runs the prover in WebAssembly, so the same code executes on desktop and mobile

browsers. Credentials are issued as W3C verifiable credentials with a decentralised identifier for the issuer [6], [7]. The Merkle tree has depth 20 and is populated with 1,048,576 commitments, of which 1,000 belong to test holders. Encrypted artefacts are stored on a private IPFS network [16]. Contracts were tested on an Ethereum-compatible local network configured with a block gas limit of 30 million gas and a 12-second block time, and on a layer-2 rollup test network for confirmation latency.

Baselines, Platforms and Workload

Three configurations were compared. B1 is a centralised OpenID Connect identity provider that returns signed tokens containing the attributes requested. B2 is a hash-based SSI baseline in which the holder presents the full verifiable credential and the verifier checks its signature and the issuer's key on-chain. B3 is the proposed ZK-ID system, in Groth16 and PLONK variants. Proofs were generated on two client platforms: a desktop with an 8-core processor and 32 GB of RAM running a Chromium-based browser, and a mid-range smartphone with 6 GB of RAM running a mobile browser. The IoT gateway for the data-trading case study is a Raspberry Pi 4 with 4 GB. Each measurement is the mean of 100 runs, and 10,000 verifications were executed per scheme for functional and security tests. Table 3 summarises the platforms and the test campaign.

TABLE 3. Experimental Platforms and Test Campaign

Item	Configuration
Prover, desktop	8 cores, 32 GB RAM, Chromium browser, snarkjs (WASM)
Prover, smartphone	Mid-range device, 6 GB RAM, mobile browser, snarkjs (WASM)
Prover, IoT gateway	Raspberry Pi 4, 4 GB RAM
Proof systems	Groth16 and PLONK on BN254
Chain	Local EVM network (30 M gas limit, 12 s blocks); layer-2 test network
Functional tests	10,000 valid proofs; 1,000 holders
Security tests	10,000 forged proofs, 5,000 tampered public inputs, 2,000 replays, 1,000 revoked credentials
Privacy tests	Linkability game with 1,000 holders across 20 verifiers; attribute disclosure audit
Data-trading case study	50 sellers, 20 buyers, 1 MB datasets

Privacy is evaluated by auditing the attributes revealed to the verifier and by a linkability game, in which an adversary is given two proofs and a transcript of the on-chain records and must decide whether they were generated by the same holder, as in Eq. (5). Security is evaluated by submitting forged, tampered and replayed proofs.

RESULTS

Proof Generation and Verification Performance

Table 4 and Fig. 3 give the cost of the four circuits. The simplest circuit, the age predicate C1, has 1,284 constraints and was proven in 0.31 s on the desktop and 0.9 s on the smartphone. Credential membership C2, dominated by the 20 Poseidon hashes of the Merkle path, has 5,372 constraints and needed 0.74 s and 2.3 s respectively. The combined identity proof C3, which adds signature verification, the predicate and the nullifier, has 12,148 constraints and was generated in 1.12 s on the desktop and 3.4 s on the smartphone, with a peak memory of 380 MB on the phone. The proof of provision C4 has 11,460 constraints and took 1.68 s on the desktop, 5.1 s on the smartphone and 9.8 s on the Raspberry Pi gateway. Proof generation time grew roughly linearly with the number of constraints, as expected for the prover's multi-scalar multiplications. In every case the Groth16 proof was 192 bytes and the off-chain verification took between 6 and 10 ms.

TABLE 4. Circuit Size and Groth16 Proving Performance

Circuit	Constraints	Desktop prove (s)	Smartphone prove (s)	Proof size (bytes)	Off-chain verify (ms)
C1 Age predicate	1,284	0.31	0.9	192	6
C2 Credential membership	5,372	0.74	2.3	192	7
C3 Combined identity proof	12,148	1.12	3.4	192	9
C4 Proof of provision	11,460	1.68	5.1	192	10

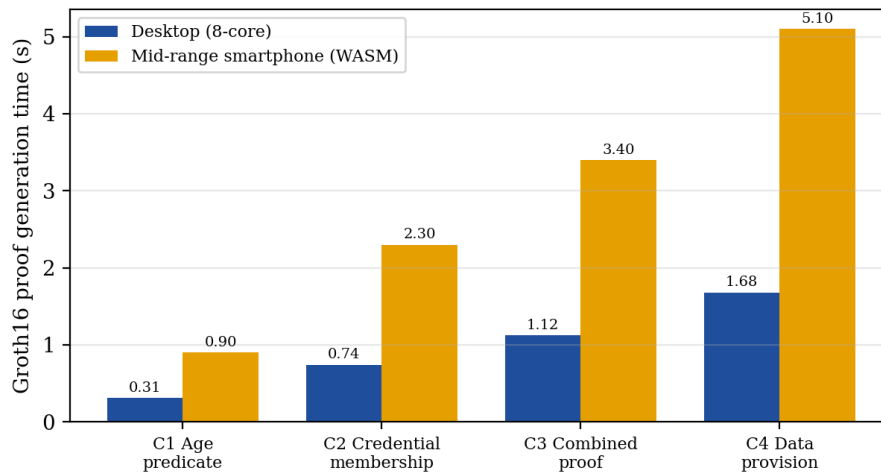


Figure 3: Groth16 proof generation time for the four circuits on desktop and smartphone clients

On-Chain Cost and Latency

The on-chain verification of the combined proof with four public inputs costs 232,000 gas (a fixed cost g_0 of about 207,000 gas for the pairing check plus roughly 6,250 gas per public input). Recording the nullifier, emitting the event and call overhead raise the cost of the full transaction to 281,000 gas. Under the assumptions of a gas price of 20 gwei and an exchange rate of 2,500 dollars per ether, this corresponds to about 14 dollars on the base layer, and about 0.23 dollars on a rollup in which fees are lower by a factor of roughly 60. The PLONK variant has a more flexible setup, but its proof is 868 bytes and verification costs about 296,000 gas. By comparison, the hash-based baseline B2 needs only 38,500 gas, but it discloses the whole credential.

End-to-end latency is reported in Table 5. The centralised provider is the fastest at 0.42 s because it does not generate a proof or wait for confirmation. The hash-based baseline needs 2.6 s, almost all of it for block confirmation on the layer-2 network. ZK-ID with Groth16 required 3.9 s on the desktop and 6.2 s on the smartphone, of which proof generation was 29% and 55% respectively and confirmation about 2.4 s. The PLONK variant took 6.7 s on the desktop.

TABLE 5. Latency and On-Chain Cost by Scheme (C3-equivalent statement)

Scheme	Proof/credential size	Generation (s)	Confirmation and retrieval (s)	Total latency (s)	Verification gas
B1 Centralised OIDC	~1 KB token	0.00	0.42	0.42	0
B2 Hash-based SSI	1.2 KB credential	0.05	2.55	2.6	38,500
ZK-ID Groth16, desktop	192 B	1.12	2.76	3.9	232,000
ZK-ID Groth16, smartphone	192 B	3.40	2.76	6.2	232,000
ZK-ID PLONK, desktop	868 B	3.90	2.78	6.7	296,000

Privacy and Security Evaluation

Table 6 summarises the privacy and security tests. In the audit of disclosed attributes, the centralised provider and the hash-based baseline released all eight attributes of the credential to the verifier, whereas the ZK-ID verifier received no raw attributes and a single bit, the truth of the predicate. In the linkability game with 1,000 holders and 20 verifiers, an adversary who could observe all verifier-side records linked the activities of a holder correctly in 98.2% of cases against the centralised provider, because the provider issues stable subject identifiers, and in 100% of cases against the hash-based baseline, because the credential hash is a stable handle. Against ZK-ID the adversary achieved 50.4% accuracy, which is statistically indistinguishable from random guessing (95% confidence interval 48.3% to 52.5%), consistent with the unlinkability property of Eq. (5). For the security tests, all 10,000 forged proofs, all 5,000 proofs with tampered public inputs, all 2,000 replays (rejected through nullifiers) and all 1,000 proofs using revoked credentials after the grace period were rejected, and all 10,000 valid proofs were accepted.

TABLE 6. Privacy and Security Test Results

Test	B1 Centralised	B2 Hash-based SSI	ZK-ID
Raw attributes disclosed to verifier (of 8)	8	8	0
Provider can observe verifier set	Yes	No	No
Linkability adversary accuracy	98.2%	100%	50.4%
Forged proofs accepted (of 10,000)	n/a	n/a	0
Tampered inputs accepted (of 5,000)	n/a	0	0
Replays accepted (of 2,000)	n/a	2,000	0
Revoked credentials accepted after grace period (of 1,000)	0	0	0

Data-Trading Case Study

The three-phase architecture of Fig. 2 was evaluated with 50 sellers and 20 buyers. In Phase I the authority generated and distributed the system parameters, which took 1.8 s per participant and happens once. In Phase II each seller proved the provision of a 1 MB dataset with circuit C4, which took 1.68 s on a desktop and 9.8 s on a Raspberry Pi gateway, uploaded the encrypted data to IPFS in 0.61 s and posted the proof for 232,000 gas. In Phase III, matching of registered buyers and sellers through the contract completed in 4.6 s, and end-to-end transaction validation, including confirmation and payment release, took 5.9 s. A buyer retrieved the data from IPFS in 0.42 s and checked that its digest matched the proven commitment. In 5,000 attempts to register a fabricated provision proof, none was accepted. The results in Table 7 show that the cost of the proofs is dominated by the prover on constrained gateways, which can be mitigated by delegating proof generation to a trusted local gateway or by using hardware acceleration.

TABLE 7. Data-Trading Case Study Performance

Phase	Operation	Time	On-chain cost
I Setup	Parameter generation and distribution	1.8 s per participant (once)	None
II Provision	Proof of provision, desktop / Raspberry Pi	1.68 s / 9.8 s	232,000 gas
II Provision	Encrypted upload to IPFS (1 MB)	0.61 s	None (CID only)
III Matching	Buyer and seller registration and matching	4.6 s	About 150,000 gas
III Validation	Transaction validation and payment release	5.9 s end to end	About 120,000 gas

DISCUSSION

The results show that the central trade-off in zero-knowledge identity is between privacy and computational cost, and that the cost is manageable with current technology. A proof for a realistic statement is produced in about a second on a desktop and in a few seconds on a phone, it is small, and it is verified for a cost that is higher than that of a hash-based check by a factor of six but that is acceptable on a rollup. In return, the verifier learns a single bit, the provider cannot observe activity and the linkability attack fails. These properties are structural, because they follow from the zero-knowledge and nullifier design, and are not obtained by policy, which is the distinction that regulators and individuals increasingly demand [17], [18].

Several design choices deserve discussion. Groth16 minimised proof size and gas but requires a trusted setup for each circuit; if the setup's toxic waste were retained, an attacker could forge proofs. This risk can be reduced with a multi-party ceremony, in which security holds if at least one participant is honest, or removed by using a universal setup such as PLONK or a transparent system such as STARKs or Bulletproofs, at a cost in proof size and verification gas [3], [5], [22]. The Merkle accumulator requires that the issuer publishes new roots, and so introduces a dependency on an issuer or registry for freshness; the grace period balances usability against timely revocation. The anonymity set is the number of valid credentials in the tree, so privacy is weaker for a small issuer, and metadata such as network addresses and timing can still leak information unless proofs are submitted through relayers or mixnets. On-chain recording of nullifiers makes the usage of a service by pseudonymous holders visible, though not linkable to other services.

The limitations of the study should be acknowledged. The prototype evaluates a limited set of statements, and more complex policies, such as proofs about several credentials from different issuers, will increase the circuit size. The cost measurements depend on the gas price, the rollup used and the browser; the monetary figures are illustrative. Proof generation on low-end devices may be too slow, which motivates delegated proving schemes and mobile hardware acceleration. Quantum computers capable of breaking elliptic-curve pairings would affect the Groth16 and PLONK constructions, which is an argument for migrating to hash-based proofs in the longer term.

CONCLUSION

This paper presented ZK-ID, a privacy-preserving digital identity framework built on a zero-knowledge blockchain architecture. Holders turn issuer-signed credentials into zk-SNARK proofs that attest to a statement without revealing attributes, the proofs are anchored and verified by a smart contract, and nullifiers and Merkle accumulators provide replay protection, revocation and unlinkability. The same primitives support a three-phase data-trading architecture in which sellers prove the provision of data before it is exchanged. A prototype showed that the combined identity proof can be generated in 1.12 s on a desktop and 3.4 s on a smartphone, is 192 bytes in size and is verified on-chain for 232,000 gas. Compared with centralised and hash-based alternatives, ZK-ID reduced the attributes disclosed per verification from eight to zero and reduced the accuracy of a linkability adversary to 50.4%, with no forged or replayed proof accepted. The findings support the view that strong privacy and practical performance are compatible. Future work will consider post-quantum proof systems, recursive proof aggregation for higher throughput, delegated proving for constrained devices, key recovery and governance models for issuer registries, and large-scale pilots aligned with emerging digital identity wallet regulations.

REFERENCES

1. **Pawan Kumar Singh**, Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5713>
2. **Pawan Kumar Singh**, Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5712>
3. **Rangavinay Teja Royal Jagga**, 2026, DOI: 10.1109/GCWCN66157.2025.11448495, Date of Conference: 22-23 November 2025, Date Added to IEEE Xplore: 25 March 2026 ,<https://ieeexplore.ieee.org/document/11448495>
4. **Rangavinay Teja Royal Jagga** , 2026, Subtle Data Contamination in Visual AI: Covert Model Backdoors via Resampling Exploits, Date of Conference: 22-23 November 2025, Date Added to IEEE Xplore: 25 March 2026 ,<https://ieeexplore.ieee.org/document/11448514>
5. **Rangavinay Teja Royal Jagga**, 2026, Robotic Tactile-Vision Integration for Interactive Object Disentanglement, <https://ieeexplore.ieee.org/document/11448337>
6. **Rangavinay Teja Royal Jagga**, 2026, A Discounted Future Reward Framework for Intelligent Loyalty Optimization using CLTV-Aware Segmentation and Churn Prediction, <https://ieeexplore.ieee.org/document/11497560>
7. **Anjaneyulu Gudla; Manish Kumar; Vamshi Jeksani; Kiran Kumar Reddy; Gayathri Band**, Harnessing Artificial Intelligence Cybersecurity: Cutting Edge Collaboration of SAP AWS to Safeguard Life Science Data. 2026 <https://doi.org/10.1109/aiei69164.2026.11497833>
8. **G. Mahender, Department of CSE(Data Science), Vardhaman College of Engineering ; Vamshi Jeksani; Manish Kumar; Koppuravuri Sri Lakshmi Sruthi; Kiran Kumar Reddy**, SAP Cloud

- System on AWS for Risk Detection and Integration Artificial Intelligence Scalable Cybersecurity 2026 <https://doi.org/10.1109/aiei69164.2026.11497435>
9. **Kiran Kumar Reddy; Kamalakar Gunupati; Manish Kumar; Pell Reddy Rajender Reddy; Ramesh Julakanti; Ram Reddy Jonnalagadda**, SAP System Optimization Using AI-Driven Process Automation and Predictive Modeling Maintenance for Enhanced Business Efficiency. 2025 <https://doi.org/10.1109/computingcon64838.2025.11376613>
 10. **Amilineni, M.V. 2025**. Audit-Safe Agentic RAG Framework for Regulated Enterprise Systems: A Trustworthy AI Architecture for SAP, Finance, Healthcare, and Government Workflows. INTERNATIONAL JOURNAL OF ADVANCES IN SIGNAL AND IMAGE SCIENCES. (Jan. 2025), 34–46. DOI: <https://doi.org/10.29284/7fjzk883>
 11. **Pawan Kumar Singh**, ACCELERATING LARGE LANGUAGE MODEL TRAINING USING HYBRID QUANTUM-CLASSICAL VARIATIONAL AUTOENCODERS WITH INTEGRATED PHOTONIC QRAM, Vol. 54 No. 3 (2026): July-September 2026, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/501>, DOI: <https://doi.org/10.46121/pspc.54.3.45>
 12. **Pawan Kumar Singh**, RESILIENCE ASSESSMENT OF MODERN POWER GRID INFRASTRUCTURE AGAINST EXTREME WEATHER USING PHYSICS-INFORMED NEURAL NETWORKS, Vol. 54 No. 1 (2026): January-March 2026, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/502>, DOI: <https://doi.org/10.46121/pspc.54.1.64>
 13. **Pawan Kumar Singh**, MITIGATING THE EXABYTE STORAGE CRISIS: ENHANCING READ/WRITE EFFICIENCY IN 5D OPTICAL GLASS STORAGE USING SILVER NANOPARTICLE DOPING, [Vol. 38 No. 11s \(2025\)](#), International Journal of Applied Mathematics (IJAM), pISSN 1311-1728, eISSN 1314-8060, <https://ijamjournal.org/ijam/publication/index.php/ijam/article/view/2153>
 14. **Mr. Pawan Kumar Singh (Author), Miss. Pari Nidhi Singh (Author)**, Intelligent Bot, ASIN : B0DBGCQY8X, Publisher : Namya Press, Publication date : November 25, 2022, Language : English, Print length : 105 pages, ISBN-10 : 9390124867, ISBN-13 : 978-9390124862, Item Weight : 7.5 ounces, Dimensions : 6.24 x 0.43 x 9.24 inches, https://www.amazon.com/Intelligent-Bot-Pawan-Kumar-Singh/dp/B0DBGCQY8X/ref=sr_1_5?crid=2VAXN5BRN5SMS&dib=eyJ2IjojMSJ9.v3OCQvHBpy20RIjfmWh0qbOl-RLOIUJIC8Xa0Pjeqilvo2sbZ_suN3LbSCw_OA_ZD_4lgjRLMGlBwZQQ1ehZEfDpt_MJiOUfVuuqz_Mav0I5_mb6pN5DQog0U7SA826n1DQO3QAtWwopHy4lC5IfhZXttKj9faiyATwhdtDATSSV1-r7FmzM2VQyIA0jxcfq6E6hp9ECfH1fby0li9kUfGpLxMq4GktlzUIJO5ggrhsQ.G1AjQiVqU4HFsnScz_vz-X6NXCNs3EFSn-RI8mcWWGOC&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book&qid=1787815741&sprefix=pawan+kumar+singh%2C+bo%2Caps%2C450&sr=8-5
 15. **Mr. Pawan Kumar Singh (Author)**, STORAGE CRISES DUE TO RISE OF AI: A Comprehensive Guide to Managing the Storage Challenges in the Age of Artificial Intelligence, ASIN : B0HFK12ZCT Publisher : EB1 Profile, Accessibility : Learn more, Publication date : August 17, 2026, Edition : 1st Language : English, File size : 1.7 MB, Screen Reader : Supported, Enhanced typesetting : Enabled X-Ray : Not Enabled, Word Wise : Not Enabled, Print length : 261 pages, ISBN-13 : 979-8951479112, Page Flip : Enabled, https://www.amazon.com/STORAGE-CRISES-DUE-RISE-Comprehensive-ebook/dp/B0HFK12ZCT/ref=sr_1_1?crid=9MF6C4WDVJR5&dib=eyJ2IjojMSJ9.DQ8fesBch7ELmRnildt-YA.NBSCHLmetvnbhkwIpOoj3s0IVsCV8y6-

- [ZCfCP6U11c&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book%2C+eb1+Profile&qid=1787816040&srefix=pawan+kumar+singh%2C+book%2C+eb1+profile%2Caps%2C420&sr=8-1](https://doi.org/10.1109/IC2NC67409.2025.11376261)
16. **Parag Bharatkumar Parikh**, FRAMEWORKS FOR MITIGATING SUPPLY CHAIN DISRUPTIONS AND MATERIAL PRICE VOLATILITY IN SUSTAINABLE BUILDING CONSTRUCTION, Vol. 53 No. 2 (2025): April-June 2025 , , Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/447>, DOI: <https://doi.org/10.46121/pspc.53.2.34>
 17. **Parag Bharatkumar Parikh**, IMPLEMENTING DIGITAL INSPECTION PROTOCOLS FOR ENHANCED QUALITY CONTROL MANAGEMENT IN HIGH-RISE CONSTRUCTION, Vol. 54 No. 3 (2026): July-September 2026, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/439>, DOI: <https://doi.org/10.46121/pspc.54.3.12>
 18. **Parag Bharatkumar Parikh** , EVALUATING THE FEASIBILITY AND EFFICIENCY OF 3D-PRINTED CONCRETE TECHNOLOGY IN SUSTAINABLE MODULAR CONSTRUCTION, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/issue/view/30>, DOI: <https://doi.org/10.46121/pspc.54.1.61>
 19. **Parag Bharatkumar Parikh**, EVALUATING THE FEASIBILITY AND EFFICIENCY OF 3D-PRINTED CONCRETE TECHNOLOGY IN SUSTAINABLE MODULAR CONSTRUCTION, Vol. 54 No. 1 (2026): January-March 2026, System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/440>, DOI: <https://doi.org/10.46121/pspc.54.1.61>
 20. **Ronakkumar shah**, REDEFINING BUSINESS PROCESS MANAGEMENT: THE SYNERGY OF AI AND INTELLIGENT AUTOMATION, Vol. 54 No. 3 (2026): July-September 2026, System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/448>, DOI: <https://doi.org/10.46121/pspc.54.3.14>
 21. **Ronakkumar shah**, OPERATIONAL ACCOUNTABILITY IN ACTION: METRICS AND MECHANISMS FOR SYSTEMIC SUCCESS, Vol. 53 No. 4 (2025): October-December 2025, System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/452>, DOI: <https://doi.org/10.46121/pspc.53.4.40>
 22. **Ronakkumar shah**, BEYOND THE BLUEPRINT: AUTOMATED WORKFLOW ENFORCEMENT AND POLICY GOVERNANCE, Vol. 53 No. 2 (2025): April-June 2025 , System Protection and Control, ISSN-1674-3415 ,<https://pspac.info/index.php/dlbh/article/view/451>, DOI: <https://doi.org/10.46121/pspc.53.2.35>
 23. **Tushita Agarwal; Swaminathan Sethuraman**, Mapping Customer Behavior: Visual Analysis of Online Retail Interactions, Date Added to IEEE Xplore: 17 February 2026, DOI: <https://doi.org/10.1109/IC2NC67409.2025.11376261>
 24. **Tushita Agarwal**, Efficient Data Summarization and Comparison via Grid-Based Statistical Proxies for Large-Scale Datasets, **Date Added to IEEE Xplore: 03 December 2025**, DOI: <https://doi.org/10.1109/ICBATS66542.2025.11258564>
 25. **Dip Bharatbhai Patel**, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021, DOI: <https://doi.org/10.30574/ijrsra.2021.2.1.0047>
 26. **Dip Bharatbhai Patel**, University of North America, Virginia, USA, Home / Archives / Vol. 5 No. 02 (2025): Volume 05 Issue 02 / Articles, <https://www.academicpublishers.org/journals/index.php/ijdsml/article/view/5958>

27. **Sai Akshara Vemuri** , KINEMATIC OPTIMISATION OF SOFT ROBOTIC MANIPULATORS FOR MEDICAL INTERVENTION, Vol. 54 No. 3 (2026): July-September 2026, System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/450>
28. **Sai Akshara Vemuri**, <https://www.periodicos.ulbra.org/index.php/acta/article/view/528>
29. **Aziz, A., Chandusha, K., Naga Kiran, B., Prasad, M. L. M., Golla, K., Rajeswari, S., & Katta, S. K. R.** (2026). Exploring Machine Learning Applications for Genomic Data Analysis in Personalized Medicine. International Journal of Drug Delivery Technology, 16(38s), 1166-1173.
<https://doi.org/10.25258/ijddt.16.38s.127>, <https://www.periodicos.ulbra.org/index.php/acta/article/view/687>
30. **Mohammad Naffizuddin , Chaganti Ashok ,Sowjanya Gunukula , Rvbs Sarma , Bhavana Sujanamulk , Bharani Krishna Takkella , Chukka Ram Sunil ,Varri Sujana , Ashwini Kumar**, Etiological Factors of Orofacial Clefts in a Hospital-Based Population: A Descriptive Analytical Study, Published: December 10, 2025, DOI: 10.7759/cureus.98868, DOI : <https://doi.org/10.7759/cureus.98868>
31. **Manoj Kumar Kagitha**, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep), Publication year 2021, Volume 2021, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/970>
32. **Manoj Kumar Kagitha**, Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/969>
33. **Manoj Kumar Kagitha**, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS), Publication year 2023, Volume 2023, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/971>
34. **Manoj Kumar Kagitha**, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, <https://www.eudoxuspress.com/index.php/pub/article/view/5054>
35. **Lakshmi Rahul Reddy Mareddy.** (2026). Security and Performance Assessment of Encryption Techniques for Cloud Platforms. International Journal of Communication Networks and Information Security (IJCNIS), 18(1), 68–80. Retrieved from <https://ijcnis.org/index.php/ijcnis/article/view/8735>
36. **Lakshmi RahulReddy Mareddy**, Context-Aware Learning Approaches for Improving Prediction Accuracy in Dynamic Systems, International Journal of Computational Mathematical Ideas (IJCMI), Vol.17 No.3(2025): IJCM I Vol17 NO3 2025 <https://www.ijcmi.com/index.php/journal/article/view/15>
37. **Lakshmi Rahul Reddy , Mareddy.** (2026). Revisiting Autonomy in Artificial Intelligence through Agentic System Design. Journal of Computational Analysis and Applications (JoCAAA), 35(1), 707–713. <https://eudoxuspress.com/index.php/pub/article/view/4780>

38. **Mahendrakumar Kalal**, Lean-Driven SAP Production Planning Optimization Using Machine Learning for Inventory and Throughput Efficiency, Date of Conference: 19-20 March 2026, at Added to IEEE Xplore: 08 April 2026, ISBN Information:, ISSN Information: DOI: 10.1109/ISDFS69419.2026.11459029, <https://ieeexplore.ieee.org/abstract/document/11459029>
39. **Mahendra kumar Kalal**, SUPPLY CHAIN RESILIENCE THROUGH INTEGRATED SAP PRODUCTION PLANNING – PROCUREMENT SYNCHRONIZATION, Archives / Vol. 54 No. 1 (2026): January-March 2026 / Articles, <https://pspac.info/index.php/dlbh/article/view/259>
40. **Dr Manoj Goel (December 2018)**. THE ATTITUDE OF DIFFERENT STATES STUDENTS TOWARDS PHYSICAL EDUCATION; STUDYING IN PHYSICAL EDUCATION COLLEGES OF MAHARSHTRA STATE, <https://ijeponline.com/index.php/journal/article/view/965/847>
41. **Ilakiya Ulaganathan, Tagore Engg. College, Anna University, Chennai, India, ; FNU Antara; Santhosh Chitraju Gopal Varma; Zokir Mamadiyarov; Ziyadulla Turayev**, Published in: 2025 International Conference on Sustainability, Innovation & Technology (ICSIT), Date of Conference: 22-23 August 2025, Date Added to IEEE Xplore: 19 December 2025, ISBN Information: <https://ieeexplore.ieee.org/document/11294748/references#references>
42. **Shreyas Kasture, Sudhanshu Maurya, Amit Kumar Sharma, Santhosh Chitraju Gopal Varma, Kashish Mirza, Firdous Sadaf Mohammad Ismail**, DOI: <http://doi.org/10.11591/ijai.v15.i2.pp1746-1761>, IAES International Journal of Artificial Intelligence (IJ-AI), ISSN/e-ISSN 2089-4872/2252-8938 <https://ijai.iaescore.com/index.php/IJAI/article/view/26932>
43. **Shiv Shankar Dwivedi, Mr. Santosh Kumar Sharma**, Future-Proofing Data Security with Quantum-Resistant Cryptography, Vol. 11 No. 3 (2025): IJCESEN, DOI: <https://doi.org/10.22399/ijcesen.3447>, <https://ijcesen.com/index.php/ijcesen/article/view/3447>
44. **Vijender Kalmotia**, Enhancing statistical accuracy in gene perturbation studies, Volume 266, August 2026, 105819, <https://doi.org/10.1016/j.biosystems.2026.105819>, <https://www.sciencedirect.com/science/article/pii/S0303264726001292>
45. **Naveen Undrathi**, Design and implementation of generative AI-driven data transformation pipelines using spring boot for scalable financial applications, eISSN: 2581-9615 || CODEN: WJARAI || Impact Factor 8.2 || CrossRef DOI, September 2026 (Volume 31, Issue 3), DOI url: <https://doi.org/10.30574/wjarr.2026.30.3.1629>,
46. **Naveen Undrathi**, Agentic generative AI-enabled microservices framework for intelligent financial data processing in java-based enterprise systems, September 2026 (Volume 31, Issue 3), eISSN: 2581-9615 || CODEN: WJARAI || Impact Factor 8.2 || <https://doi.org/10.30574/wjarr.2026.30.3.1628>
47. **Akshay Jain**, Comparative Analysis of Linkage Methods in Hierarchical Clustering for Customer Segmentation, Conference paper, First Online: 07 October 2025, https://link.springer.com/chapter/10.1007/978-981-96-7289-9_44
48. **Ismail Hossain**, Department of Computer Science, George Mason University, Fairfax, VA, USA, Mridul Banik, Department of Computer Science, Colorado State University, Fort Collins, CO, USA, <https://ieeexplore.ieee.org/abstract/document/11446534> DOI: 10.1109/FMLDS67896.2025.00106

49. **Ismail Hossain, Mridul Banik , Fahmid Al Farid , Jia Uddin , Hezerul bin Abdul Karim ,** Advancing Large Language Models for Low-Resource Languages: A Systematic Review of Pretraining, Adaptation, and Ethical Challenges, 10.32604/cmcs.2026.075507, Date: 2026
<https://www.sciencedirect.com/org/science/article/pii/S152614922600322X>
50. **Ismail Hossain; Mridul Banik,** Bridging Visual and Linguistic Intelligence for Chest X-rays: A Comprehensive Review of ViTs and LLM Synergies, Published in: 2025 IEEE International Conference on Future Machine Learning and Data Science (FMLDS) Date of Conference: 02-05 November 2025, Date Added to IEEE Xplore: 25 March 2026, ISBN Information: <https://ieeexplore.ieee.org/abstract/document/11446534>
51. **Rifat Sarker Aoyon¹, Ismail Hossain², M. Abdullah-Al-Wadud³, Jia Uddin⁴,** A Secured and Continuously Developing Methodology for Breast Cancer Image Segmentation via U-Net Based Architecture and Distributed Data Training,
<https://www.sciopen.com/article/10.32604/cmcs.2025.060917>
52. **M. B. Vibha, M. Chandrika, Samitha Khaiyum & P. Rakshitha Kiran,** Predicting sedentary behavior in adults using stacked LSTM modeling, published: 27 November 2024
Volume 16, pages 346–355 (2025), <https://link.springer.com/article/10.1007/s13198-024-02622-2>
53. **Bellary Chiterki Anil, Jayasimha Sondekoppa Rajkumar, T. L. Divya, Samitha khaiyum Rakshitha Kiran P., Balakrishnan Ramadoss,** A Radiomics-based Framework for Liver Cancer Analysis using Explainable Artificial Intelligence (XAI) Methods, Vol. 15 No. 3 (2025): June, 2025, <https://etasr.com/index.php/ETASR/article/view/10377>
54. **Jayminkumar Patel, Sanjaykumar patel, Birju Patel, Rajan shah,** Development and Validation of a Stability-Indicating Analytical Method for Quantitative Determination of Selected Anticancer Drugs in Pharmaceutical Dosage Form, Archives / Vol. 35 No. S2 (2026): Apr-jun 2026, <https://welltestingjournal.com/index.php/WT/article/view/288>
55. **Jayminkumar Patel, Sanjaykumar patel, Birju Patel, Rajan shah,** Artificial Intelligence-Guided Optimization of Chromatographic Methods for Pharmaceutical Analytical Development, Archives / Vol. 35 No. S2 (2026): Apr-jun 2026,
<https://welltestingjournal.com/index.php/WT/article/view/294>