Journal of Contemporary Science and Technology Management
Volume 02, Issue 04, December 2026

A Blockchain-Enabled Decentralized Identity Management Framework Using Zero-Knowledge Proofs for Secure Digital Authentication

Irma Sánchez
Zaragoza, Spain
Sowjanya Addu
Computer Science & Engineering, Gokaraju Rangaraju Institute of Engineering Technology,
Hyderabad, Telangana, India
sowjanya1634@grietcollege.com

Abstract

Digital authentication remains dependent on centralized identity repositories, repeated disclosure of personal attributes, and platform-specific credentials. This paper proposes a decentralized identity management framework combining verifiable credentials, decentralized identifiers, a permissioned blockchain trust registry, and zero-knowledge proofs for selective, predicate-based authentication. Issuers sign credentials, holders maintain encrypted credentials locally, and relying parties verify fresh proofs without receiving underlying attributes. Blockchain records issuer keys, status commitments, governance decisions, and auditable events rather than personal data. A protocol is specified for enrollment, credential issuance, presentation, verification, revocation, and recovery, including challenge binding and domain separation. An analytical evaluation with explicitly illustrative values compares the proposed design against a centralized baseline and a signature-only decentralized baseline. The example shows the expected tradeoff: reduced attribute disclosure and improved resilience at the cost of additional proof computation and status-management complexity. These numerical values are scenario assumptions, not measured experimental findings. The paper contributes a reproducible test plan, security analysis, mathematical formulation, and implementation roadmap for future empirical validation.

Keywords: decentralized identity; blockchain; zero-knowledge proof; verifiable credentials; digital authentication; selective disclosure; privacy; revocation.

DOI: 10.66592/dfg5ws65

1. Introduction

Digital identity systems support access to banking, education, healthcare, public administration, and enterprise applications. The conventional arrangement commonly relies on an identity provider that registers individuals, stores identifiers, authenticates users, and distributes assertions to relying services. Central coordination can simplify support and policy enforcement, but it also creates a concentration of sensitive information and operational dependence on a small number of services. A compromised identity repository may expose many records simultaneously, while service interruptions may prevent legitimate users from accessing otherwise available resources.

Authentication and identity proofing are different tasks. Identity proofing establishes confidence in the relationship between a person and an asserted identity; authentication establishes that the claimant controls an appropriate authenticator during a session. An architecture that verifies a cryptographic proof does not automatically establish that the original credential was issued to the correct person. Consequently, decentralized identity must be designed as an end-to-end trust process that addresses enrollment, issuer assurance, wallet security, credential lifecycle, and relying-party policy.

Centralized single sign-on and federated identity reduce the number of passwords users manage, but federation still involves trusted providers, correlation opportunities, and infrastructure dependencies. A decentralized alternative gives the credential holder greater control over presentation. In an issuer–holder–verifier model, an authoritative issuer creates a signed claim, the holder stores it, and a verifier checks evidence of validity. This separation supports portability across organizational boundaries, provided that participants agree on cryptographic formats, trust lists, semantics, and revocation policies.

Blockchain can provide a shared, tamper-evident registry for issuer public keys, authorization roles, and credential-status commitments. It is not necessary to place names, birth dates, photographs, or credential contents on a chain. Such publication would be difficult to reverse and could undermine the very privacy the architecture seeks to protect. A permissioned ledger is therefore proposed as a coordination and accountability layer, while personally identifying data remain in encrypted holder wallets or protected issuer systems.

Zero-knowledge proofs offer a complementary privacy mechanism. A holder can demonstrate that a secret witness satisfies a publicly stated relation without revealing the witness itself. For example, an age credential can support a proof that an age threshold is met without exposing the date of birth. Practical deployments require careful choices of proving systems, trusted setup assumptions, cryptographic parameters, proof sizes, and verification costs. Privacy is not guaranteed merely by adding the term zero knowledge; metadata, reusable identifiers, issuer-specific formats, and verifier collusion may still permit linkage.

The W3C Verifiable Credentials Data Model v2.0 provides a standardized vocabulary for issuer, holder, verifier, and credential claims. W3C decentralized identifiers specify interoperable identifier and resolution concepts, although DIDs are optional for verifiable credentials. NIST SP 800-63-4 supplies an assurance-oriented framework for identity proofing, authentication, and federation. This study combines those ideas while distinguishing normative interoperability requirements from architectural design choices.

Important practical questions remain unresolved. How should a verifier discover that an issuer is trustworthy? How can a credential be revoked without creating a globally observable tracking identifier? How does a wallet recover from key loss without allowing an attacker to take over the identity? What happens when a blockchain node is unreachable or a validator quorum fails? How can applications balance verification latency against the privacy advantages of proof-based disclosure?

The proposed design addresses these questions through explicit roles, state transitions, proof verification, and status-checking rules. It specifies a privacy-preserving authentication protocol, not a replacement for all organizational identity processes. It assumes an issuer performs appropriate off-chain proofing and that the holder protects secret keys with suitable hardware or software controls. The verifier independently evaluates issuer trust, credential status, proof validity, audience binding, and freshness before making an access decision.

The study makes four contributions. First, it presents a layered architecture that minimizes on-chain personal information. Second, it defines a mathematical statement for selective predicate verification and a challenge-bound authentication workflow. Third, it develops an evaluation framework covering latency, throughput, disclosure, availability, and attack resistance. Fourth, it identifies operational limitations and a validation plan that can be implemented using established credential and cryptographic libraries.

The remaining sections examine the literature, describe the architecture and methodology, present transparent illustrative calculations, and discuss limitations. Because no software deployment or benchmark dataset was supplied, the numerical comparison is intentionally labeled as a scenario analysis rather than a claim of experimentally demonstrated performance. This distinction preserves the scientific interpretability of the manuscript and establishes a clear basis for subsequent implementation.

2. Literature Review

Identity management research has moved from isolated application accounts toward federated trust arrangements and user-held credentials. Federation enables cross-domain access but does not inherently prevent tracking by the identity provider or relying parties. Decentralized identity proposes a different distribution of responsibilities: issuers attest to facts, holders manage presentations, and verifiers evaluate cryptographic evidence and organizational trust. Its main challenge is therefore not only decentralization of storage but decentralization of trust decisions.

The W3C DID Core Recommendation (2022) describes identifiers that can resolve to DID documents containing verification methods and service information. DID methods vary in their governance, persistence, privacy, and resolution properties. The choice of DID method affects operational resilience and correlation risks; a public DID reused across services may function as a tracking handle. Pairwise identifiers and minimized resolution requests can reduce such risks.

The W3C Verifiable Credentials Data Model v2.0 (2025) defines a credential ecosystem and extensible claim structure. It separates credential content from securing mechanisms and does not require blockchain or DIDs in every deployment. This separation is important: interoperable credential data do not by themselves provide unlinkable presentation, and a blockchain-based registry does not guarantee that a credential is truthful. The issuer's authority and proofing processes remain foundational.

Cryptographic selective disclosure has been explored using anonymous credential systems, signature schemes supporting derived proofs, and succinct zero-knowledge protocols. Different constructions have different properties. Some can prove possession of a signature and selectively reveal fields, while others express richer arithmetic predicates using circuits. Their performance depends on circuit complexity, key generation, proof system, device capability, and whether proof verification can be performed without online issuer contact.

Zero-knowledge succinct non-interactive arguments of knowledge are attractive because they can compress complex statements into short verifiable proofs. However, some systems require a trusted setup, and many rely on assumptions that need explicit review. Transparent systems may avoid trusted setup but can have different proof-size or verification tradeoffs. Protocol selection should therefore be driven by security requirements and measured performance rather than a generic claim that all zero-knowledge proofs are lightweight.

Blockchain-backed identity proposals often use smart contracts to register issuers, publish credential commitments, and track status. The immutable ledger can provide a shared audit trail, but it may also reveal activity patterns or allow permanent correlation. Permissioned ledgers can constrain write access and provide predictable governance; nonetheless, validators may collude, censor updates, or become unavailable. Consensus choice and governance are thus part of the security model.

Revocation is particularly difficult for privacy-preserving credentials. A naive verifier query for a unique credential identifier can reveal which service a holder is using. Status lists, accumulators, and privacy-preserving non-revocation proofs are alternative approaches. Each introduces update frequency, synchronization, and witness-refresh costs. An effective design must define acceptable staleness, offline behavior, and fail-open versus fail-closed policies.

Wallet usability and key management are equally important. A mathematically secure proof protocol is ineffective when a device is compromised or a recovery mechanism permits unauthorized replacement of keys. Hardware-backed keys, local encryption, rate-limited recovery, guardian-based recovery, and revocation of old credentials are useful design options. Recovery must preserve accountability without restoring a centralized single point of failure.

NIST SP 800-63-4 (2025) emphasizes risk-based assurance across proofing, authenticators, and federation. The proposed framework adopts this separation: a proof of credential possession is not presented as equivalent to a high-assurance authenticator, and issuer trust is evaluated separately from the cryptographic proof. Phishing resistance, verifier impersonation, replay protection, and session binding require additional protocol controls.

The literature reveals a recurring tradeoff between privacy, operational complexity, and performance. Systems that maximize disclosure minimization may incur extra proving costs; systems that minimize online dependencies may have less current revocation information. Existing standards supply interoperable building blocks but leave substantial engineering choices open. The research gap addressed here is an integrated protocol specification that jointly considers issuer governance, proof semantics, challenge freshness, status privacy, wallet recovery, and measurable evaluation criteria.

3. Research Methodology

3.1 Research design and assumptions

This design-science study specifies an architecture and a reproducible evaluation plan. It does not claim a completed deployment. The threat model includes malicious relying parties, replay attackers, credential thieves, dishonest issuers, compromised wallet devices, and a minority of faulty validators. The model assumes standard cryptographic hardness, protected holder secrets, and a governance quorum that is not fully compromised. Network traffic metadata and compromised endpoints remain outside the zero-knowledge guarantee.

Journal of Contemporary Science and Technology Management
Volume 02, Issue 04, December 2026

3.2 Layered architecture

Identity proofing and issuer authorization

↓ **Credential issuance and encrypted holder wallet**

↓ **ZK circuit and challenge-bound proof generation**

↓ **Verifier trust, proof and status checks**

↓ **Permissioned ledger: issuer registry and status commitments**

↓ **Access decision, audit and lifecycle management**

The issuer signs a structured credential after an appropriate proofing procedure. A holder wallet stores encrypted credentials and private keys. The verifier publishes a policy, nonce, audience, and expiry window. The wallet constructs a proof of possession and a policy predicate. The ledger stores public governance state and batched status commitments, not personally identifying claims. The verifier checks the proof locally and reads a current authenticated status snapshot.

3.3 Roles and data objects

Actor	Responsibility	Protected data
Issuer	Proofing, signing, status updates	Issuance evidence and signing key
Holder	Wallet, witness storage, proof generation	Credentials and private key
Verifier	Policy, nonce, proof validation	Session decision and minimal logs
Validators	Registry and status consensus	Public keys and commitments
Governance body	Issuer admission and dispute process	Role approvals

3.4 Formal model and editable equations

Let C denote a credential, K the issuer verification key, w the holder witness, x the public statement, and R a polynomial-time relation. The proof system contains Setup, Prove and Verify algorithms. The following equations are editable Word math objects; symbols are deliberately written without subscript underscores or minus signs.

$$R(x, w) = 1$$

The relation equals one when the witness satisfies signature possession, the requested attribute predicate, a valid status witness, and binding to the verifier challenge. No witness fields are transmitted.

$$\pi = Prove(pk, x, w)$$

Here π is the generated proof and pk denotes public proving parameters. The holder computes π locally using the encrypted wallet after unlocking it.

$$Verify(vk, x, \pi) = 1$$

The verifier accepts a proof only if the verifying key vk corresponds to the approved circuit and the statement x matches the requested policy and session.

3.5 Credential issuance protocol

Holder generates a protected signing key

↓

Issuer completes identity proofing

↓

Issuer verifies holder binding and signs credential

↓

Holder verifies issuer signature and stores encrypted credential

↓

Issuer publishes batched status commitment

At enrollment, the issuer verifies real-world evidence using an assurance process proportionate to the application. The wallet creates a non-exportable key when hardware support is available. The issuer signs the credential including schema, validity interval, subject binding, and status reference. The wallet validates the signature before storage. The issuer updates a public registry commitment without publishing a subject identifier. The enrollment evidence is retained only according to applicable policy.

3.6 Authentication and zero-knowledge presentation

Verifier sends policy, random nonce and domain

↓

Wallet checks issuer and requested disclosure

↓

Wallet proves signature, predicate and status

↓

Verifier checks nonce, domain, expiry and proof

↓

Verifier grants or rejects scoped session

A fresh verifier-generated nonce and an application-specific audience are included in the public statement. This prevents a proof captured from one login from being accepted in another domain or session, assuming the verifier maintains nonce uniqueness and expiration. The wallet should display the requesting domain and exact requested predicate, reducing consent confusion and verifier impersonation risks.

$$x = H(policy \parallel nonce \parallel domain \parallel epoch \parallel root)$$

H is a collision-resistant hash, ‖ denotes unambiguous concatenation, epoch is the accepted status epoch, and root is the authenticated status commitment. The verifier recomputes x rather than trusting a wallet-supplied value.

$$Accept = ProofValid \wedge IssuerTrusted \wedge StatusValid \wedge Fresh \wedge AudienceValid$$

The final authorization decision is the logical conjunction of proof validity, issuer trust, status validity, freshness, and audience binding. A successful cryptographic proof alone does not authorize every resource.

3.7 Status, revocation, and unlinkability

The issuer periodically publishes a batched status root. The wallet obtains a status witness and proves that the credential belongs to an accepted set without disclosing a unique index. If the proof system cannot hide that index, the design must disclose this limitation and avoid claiming unlinkability. Verifiers reject snapshots beyond the application-defined maximum age. Revocation propagation latency is bounded by issuer publication, ledger finality, and wallet refresh intervals.

$$Trev \leq Tpublish + Tfinality + Trefresh$$

The equation is an operational upper-bound target under successful connectivity; it is not guaranteed during outages. Applications with high-risk access may require fresh status and fail closed. Lower-risk applications may accept a bounded offline grace period after a documented risk assessment.

3.8 Security and privacy analysis

Completeness means an honest holder with a valid credential and witness should pass verification. Soundness means an attacker without a valid witness should not produce an accepted proof except with negligible probability under the chosen assumptions. Zero knowledge means that a simulated proof transcript does not reveal the witness beyond the public statement. These are properties of the selected cryptographic construction and its implementation, not consequences of blockchain storage.

Journal of Contemporary Science and Technology Management
Volume 02, Issue 04, December 2026

$$Pr[Verify(vk, x, \pi) = 1 | R(x, w) = 0] \leq \varepsilon$$

The soundness error ε must be negligible at the chosen security parameter. Practical security additionally depends on approved issuer keys, correct circuit constraints, nonces, and protection against compromised endpoints.

$$DisclosureRatio = RevealedAttributes \div RequestedAttributes$$

Disclosure ratio quantifies attributes revealed to the verifier. For a predicate proof, the verifier learns the truth of the predicate, not the underlying date or numerical attribute. This metric does not capture metadata leakage and should be interpreted alongside linkability tests.

3.9 Governance, recovery, and failure handling

A governance contract maintains authorized issuer keys, key rotation records, schema approvals, and validator membership. Changes require multi-party approval and are logged. A holder who loses a device invokes a documented recovery process: re-proofing or threshold-assisted recovery, issuance of a replacement credential, and invalidation of old credentials. Recovery authorities cannot silently derive the original private key. Applications must implement timeout, retry, and status-staleness policies for ledger unavailability.

3.10 Experimental protocol for future implementation

Implement three comparable prototypes: centralized token authentication; decentralized signed-credential presentation without zero knowledge; and the proposed decentralized zero-knowledge presentation. Use identical verifier hardware, network topology, request traces, issuer populations, and authorization policies. Record at least 30 independent runs per workload after warm-up. Report median and 95th percentile latency, confidence intervals, proof generation time, verification time, bytes exchanged, ledger finality, peak throughput, and success rate under controlled faults.

$$Latency = Tnetwork + Tproof + Tverify + Tstatus$$

Latency is decomposed into network communication, local proof generation, verification, and status retrieval. Measurements must specify whether wallet unlock and issuer proofing are excluded from login latency.

$$Throughput = SuccessfulAuthentications \div ElapsedSeconds$$

Throughput is the number of completed successful authentication sessions divided by the elapsed benchmark duration. Concurrency should be increased until latency or error-rate thresholds are exceeded.

3.11 Statistical and reproducibility plan

Fix cryptographic libraries, curve and proof-system versions, hardware specifications, transaction batch size, validator count, and network latency distribution. Publish circuit source, configuration files, synthetic workload generator, and anonymized aggregated timings. Use bootstrap confidence intervals and paired comparisons across identical workload seeds. Measure verifier-side privacy separately from timing; a proof may disclose zero credential fields while still leaking network identifiers.

4. Results and Discussion

4.1 Nature of the numerical evidence

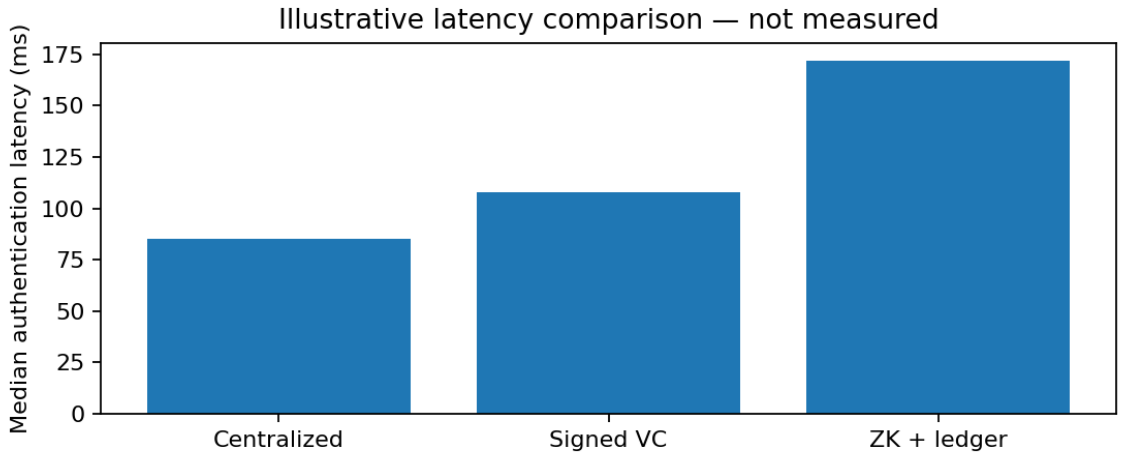
All values in this section are hypothetical scenario inputs constructed to demonstrate the reporting format and expected engineering tradeoffs. They are not measurements, benchmark outputs, empirical observations, or validated claims of superiority. The baseline figures and proposed-system figures must be replaced by results from the experimental protocol in Section 3 before submission as an empirical research article.

Architecture	Median login (ms)	P95 login (ms)	Proof generation (ms)	Attributes revealed	Peak throughput (auth/s)
Centralized identity	85	118	0	6	900
Signed decentralized credential	108	145	0	4	780

Journal of Contemporary Science and Technology Management
Volume 02, Issue 04, December 2026

Proposed ZK decentralized framework	172	230	74	1	610

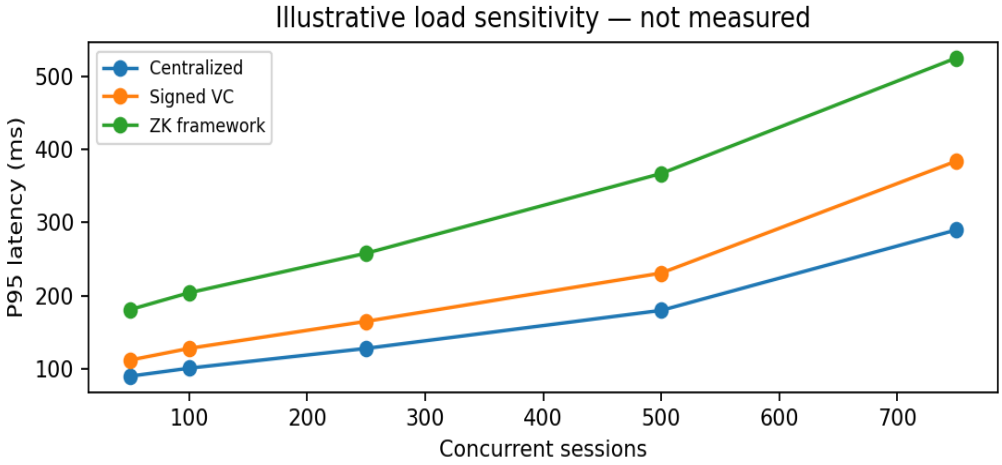
The illustrative proposed configuration reduces direct attribute exposure from six fields to one policy result, while increasing median latency from 85 to 172 ms relative to the centralized case. The 74 ms local proof-generation assumption accounts for much of this difference. A verifier must assess whether the privacy improvement justifies the latency cost for its service tier.



4.2 Illustrative load sensitivity

Concurrent sessions	Centralized P95 (ms)	Signed VC P95 (ms)	ZK framework P95 (ms)
50	90	112	181
100	101	128	204
250	128	165	258
500	180	231	367
750	290	384	525

This scenario assumes contention increases verification queues and status-cache misses as concurrency rises. The ZK approach remains computationally heavier in this model; caching authenticated status roots and parallelizing verification are plausible optimization strategies that require testing rather than assertion.



Journal of Contemporary Science and Technology Management
Volume 02, Issue 04, December 2026

4.3 Privacy and resilience scenario

Measure	Centralized	Signed VC	Proposed ZK
Direct fields disclosed per login	6	4	1 predicate result
Issuer contacted per login	Potentially	No	No
Credential data on ledger	Not applicable	None	None
Cross-service linkability	Possible	Possible	Mitigated, not eliminated
Dependence on registry availability	Identity provider	Issuer key cache	Registry and status cache

The privacy comparison is categorical and design-based, not an experimentally observed leakage rate. A predicate result can still be sensitive, and a relying party may correlate sessions using IP addresses, browser fingerprints, timing, or stable wallet identifiers. The design should use pairwise audience bindings, short-lived session tokens, and minimal logs to reduce avoidable correlation.

4.4 Threat-oriented assessment

Threat	Mitigation	Residual risk
Replay	Fresh nonce and short expiry	Nonce reuse or weak verifier state
Credential forgery	Issuer signature and sound proof	Compromised issuer key
Revoked credential	Authenticated status witness	Status lag and offline access
Verifier tracking	Predicate proof and pairwise identifiers	Metadata correlation
Wallet theft	Hardware key and unlock policy	Compromised device or recovery
Validator compromise	Quorum governance and audit	Majority collusion or censorship

The threat assessment distinguishes cryptographic prevention from operational mitigation. Zero knowledge protects secret witness information under the proof definition but cannot prevent an issuer from issuing a false claim, a verifier from retaining information voluntarily disclosed elsewhere, or malware from extracting data before proof generation. Ledger immutability does not remove governance and key-management risk.

4.5 Interpretation and limitations

The scenario highlights a plausible privacy–latency tradeoff rather than demonstrating a universal performance advantage. Real proof-generation times depend strongly on proof system, device class, circuit complexity, and cryptographic library. Verification and status-checking costs may differ substantially between permissioned and public blockchains. The analytical comparison omits deployment cost, energy use, user abandonment, issuer proofing time, accessibility, and recovery success rates. A complete evaluation must report these variables and provide uncertainty intervals from actual runs.

5. Conclusion

This paper specifies a blockchain-enabled decentralized identity architecture that combines issuer-signed verifiable credentials with zero-knowledge predicate proofs and a governed ledger for issuer trust and status commitments. The protocol separates identity proofing from authentication, avoids storing personal attributes on-chain, binds proofs to fresh verifier challenges, and includes revocation and recovery mechanisms. The illustrative numerical analysis shows how privacy improvements can be accompanied by additional proof-generation latency; it must not be interpreted as experimental validation. Future work should implement the protocol with an audited proof system, compare cryptographic constructions on mobile and server devices, measure status-witness freshness and linkability, test recovery and validator faults, and assess conformance with applicable W3C and NIST guidance.

References

Rao, S. Madhusudhana, and Arpit Jain. "Advances in Malware Analysis and Detection in Cloud Computing Environments: A Review." International Journal of Safety & Security Engineering 14, no. 1 (2024).

Bhanushali, B. (2020). The industry needs a validated model to justify the immense cost of migrating from manual, periodic KYC (know your customer) to automated. International Journal of Multidisciplinary Research in Science, Engineering and Technology, 3(11), 1799–1805. https://doi.org/10.15680/IJMRSET.2020.0311009

Kumar, S., Sharma, K., Kumar, P. A., Jain, A., Bhagat, S. K., & Singh, P. (2024, September). An improved particle swarm approach for energy-aware location-aided routing in mobile ad-hoc network. In 2024 7th International Conference on Contemporary Computing and Informatics (IC3I) (Vol. 7, pp. 1119–1124). IEEE.

Bhanushali, B. (2023). Blockchain for trade finance: Enhancing cross-border payment transparency, reducing processing time, and mitigating fraud through distributed ledger systems. International Journal on Recent and Innovation Trends in Computing and Communication, 11(9), 5690–5697. https://www.ijritcc.org/index.php/ijritcc/article/view/11932

Kumar, S., Jain, A., & Moparthi, N. R. (2022). Enhanced method of object tracing using extended Kalman filter via binary search algorithm.

Rangappa, A. S. (2025). Agentic AI and cyber security: Autonomous threat hunting, intrusion detection, and adaptive defense mechanisms in a world of increasingly sophisticated cyber attacks. Journal of Digital Security and Forensics, 2(1), 128–138. https://doi.org/10.29121/digisecforensics.v2.i1.2025.86

Anchala, S. (2018). Adoption of infrastructure as code security validation in cloud-native environments for reducing configuration drifts and compliance violations through automated template scanning. International Journal of Innovative Research in Science, Engineering and Technology, 7(10), 8305–8313. https://doi.org/10.15680/IJIRSET.2018.0710087

Jain, A. (2024, December). Energy-Efficient Signal Processing Architectures for IoT Networks Design. In 2024 International Conference on Emerging Technologies and Innovation for Sustainability (EmergIN) (pp. 362–367). IEEE.

Chatrath, A. (2018). The impact of containerization on DevSecOps: Security considerations for Docker, Kubernetes, and microservices architectures in continuous deployment pipelines. International Journal of Advanced Research in Education and Technology, 5(6). https://doi.org/10.15680/IJARETY.2018.0506019

Devulapalli, N. (2024). Agentic AI in finance and economics: The role of autonomous agents in fraud detection, algorithmic trading, credit risk assessment, and ethical governance of automated financial systems. International Journal of Multidisciplinary and Scientific Emerging Research, 12(4), 1958–1964. https://doi.org/10.15662/IJMSERH.2024.1204057

Mahajan, R. A., Dey, R., Khan, M., Mohd Su'ud, M., Alam, M. M., & Jadhav, P. (2025). Enhancing personalization in IoT-based health monitoring via generative AI and transfer learning. Egyptian Informatics Journal, 32, Article 100788. https://doi.org/10.1016/j.eij.2025.100788

Bhanushali, B. (2022). Blockchain-driven carbon credit trading systems for improving environmental transparency, reducing double counting, and ensuring accountability in climate finance. International Journal on Recent and Innovation Trends in Computing and Communication, 10(11), 292–300. https://www.ijritcc.org/index.php/ijritcc/article/view/11927

Dwivedi, D. (2023). Integration of secure file upload mechanisms in user-facing applications for preventing malware injections and file-based exploits through content-type validation. International Journal on Recent and Innovation Trends in Computing and Communication, 11(6), 768–775. https://www.ijritcc.org/index.php/ijritcc/article/view/11940

Talasila, D. (2018). Integration of identity federation in multi-cloud access management for streamlining user authentication and reducing credential sprawl through centralized SSO protocols. International Journal of Multidisciplinary and Scientific Emerging Research, 6(4), 2017–2025. https://doi.org/10.15662/IJMSERH.2018.0604046

Jain, Jaishree, Arpit Jain, Saurabh Kumar Srivastava, Chaman Verma, Maria Simona Raboaca, and Zoltán Illés. "Improved security of e-healthcare images using hybridized robust zero-watermarking and hyper-chaotic system along with RSA." Mathematics 10, no. 7 (2022): 1071.

Gupta, A. (2020). Utilization of homomorphic encryption in healthcare data sharing for ensuring confidentiality and compliance through secure computation without data exposure. International Journal of Multidisciplinary Research in Science, Engineering and Technology, 3(10). https://doi.org/10.15680/IJMRSET.2020.0310011

Talasila, D. (2018). Utilization of runtime application self-protection (RASP) in production environments for real-time vulnerability mitigation and behavioral enforcement through embedded monitoring agents. International Journal of Advanced Research in Education and Technology, 5(3), 831–840. https://doi.org/10.15680/IJARETY.2018.0503017

Jain, Arpit, Nageswara Rao Moparthi, A. Swathi, Yogesh Kumar Sharma, Nitin Mittal, Ahmed Alhussen, Zamil S. Alzamil, and MohdAnul Haq. "Deep Learning-Based Mask Identification System Using ResNet Transfer Learning Architecture." Computer Systems Science & Engineering 48, no. 2 (2024): 341.

Devulapalli, N. (2025). Machine learning in database security: Leveraging anomaly detection, intrusion detection, and predictive analytics for proactive threat prevention. International Journal of Innovative Research in Science, Engineering and Technology, 14(2), 1685–1692. https://doi.org/10.15680/IJIRSET.2025.1402105

Talasila, D. (2019). DevSecOps in e-governance platforms: Safeguarding public data, ensuring transparency, and enhancing trust in cloud-based government applications. International Journal of Innovative Research in Science, Engineering and Technology, 8(2), 1561–1570. https://doi.org/10.15680/IJIRSET.2019.0802107

Rangappa, A. S. (2018). Deployment of immutable infrastructure in Dev Sec Ops practices for minimizing configuration drift and recovery time through ephemeral build environments. International Journal of Multidisciplinary and Scientific Emerging Research, 6(4), 2026–2034. https://doi.org/10.15662/IJMSERH.2018.0604047

Mahajan, R. A., Dey, R., Khan, M., Kanabar, K., & Gupta, S. K. (2025). Real-time vehicular-to-vehicular (V2V) communication and applications for pedestrian safety in AI-optimized VANETs for autonomous vehicles. In H. Maryam, M. M. Malik, I. U. Khan, & S. K. Gupta (Eds.), AI-driven transportation systems: Real-time applications and related technologies (pp. 151–167). Springer. https://doi.org/10.1007/978-3-031-98349-8_8

Anchala, S. (2019). Database security in e-governance systems: Protecting citizen records, ensuring transparency, and managing public trust in cloud-based platforms. International Journal of Multidisciplinary Research in Science, Engineering and Technology, 2(11), 2268–2276. https://doi.org/10.15680/IJMRSET.2019.0211014

Gupta, A. (2021). Application of cyber deception strategies in military networks for enhancing operational security and misleading attackers through adaptive honeypots and decoy systems. NeuroQuantology, 19(4), 317–333. https://doi.org/10.48047/nq.2021.19.4.NQ21068

Kumar, Ashok, and Arpit Jain. "Image smog restoration using oblique gradient profile prior and energy minimization." Frontiers of Computer Science 15, no. 6 (2021): 156706.

Harshitha, G., Kumar, S., Rani, S., & Jain, A. (2021, November). Cotton disease detection based on deep learning techniques. In 4th Smart Cities Symposium (SCS 2021) (Vol. 2021, pp. 496–501). IET.

Chatrath, A. (2020). DevSecOps for financial services: An in-depth study of secure CI/CD practices, regulatory compliance, and real-time threat monitoring in cloud-based FinTech applications. International Journal of Innovative Research in Science, Engineering and Technology, 9(11). https://doi.org/10.15680/IJIRSET.2020.0911098

Dwivedi, D. (2022). Application of AI-powered vulnerability scanners in legacy applications for identifying unpatched security flaws and weak dependencies through semantic code analysis. International Journal on Recent and Innovation Trends in Computing and Communication, 10(12), 617–623. https://www.ijritcc.org/index.php/ijritcc/article/view/11918

Talasila, D. (2018). Implementation of versioned configuration management in DevSecOps for preventing misconfigurations and ensuring environmental consistency through GitOps automation. International Journal of Innovative Research in Computer and Communication Engineering, 6(2), 1731–1739. https://doi.org/10.15680/IJIRCCE.2018.0602134

Devulapalli, N. (2024). The role of homomorphic encryption in database security: Enabling confidential query processing, secure data sharing, and privacy-preserving analytics. International Journal of Multidisciplinary Research in Science, Engineering and Technology, 7(8), 14319–14326. https://doi.org/10.15680/IJMRSET.2024.0708082

Khan, S. (2023). AI-driven speech and voice analytics in CRM platforms: A study on intelligent call center automation, emotion recognition, and service personalization. International Journal of Multidisciplinary Research in Science, Engineering and Technology, 6(2), 510–517. https://doi.org/10.15680/IJMRSET.2023.0602035

Jain, Arpit, Adesh Kumar, and Sanjeev Sharma. "Comparative design and analysis of mesh, torus and ring NoC." Procedia Computer Science 48 (2015): 330–337.

Chatrath, A. (2019). A multi-layered DevSecOps framework: Integrating secure coding, automated testing, vulnerability scanning, and incident response into end-to-end software development lifecycles. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 8(3). https://doi.org/10.15662/IJAREEIE.2019.0803102

Gupta, A. (2025). Agentic AI for national security and defense: An analytical study on autonomous agents in surveillance, strategic decision-making, and military operations with a focus on ethical and legal boundaries. Journal of Digital Security and Forensics, 2(1), 163–175. https://doi.org/10.29121/digisecforensics.v2.i1.2025.84

Gupta, Namit, Kunwar Singh Vaisla, Arpit Jain, Adesh Kumar, and Rajeev Kumar. "Performance Analysis of AODV Routing for Wireless Sensor Network in FPGA Hardware." Computer Systems Science & Engineering 40, no. 3 (2022).

Dwivedi, D. (2020). Adoption of secure software updates in application lifecycle for preventing supply chain tampering and maintaining trust through cryptographically signed patches. Journal of Critical Reviews, 7(10), 6974–6980. https://doi.org/10.69980/jcr.v7:i10.13406

Rangappa, A. S. (2020). Integration of behavioral analytics in developer workflows for detecting insider threats and malicious activities through activity monitoring and contextual profiling. International Journal of Innovative Research in Science, Engineering and Technology, 9(1), 13375–13383. https://doi.org/10.15680/IJIRSET.2020.0901064

M. M. Ghori, "Intelligent Package Recognition Using Deep Vision Networks," 2026 IEEE International Conference on Applied Intelligence Devices, Machine Learning, and Data Science (AIDML), Bilaspur, India, 2026, pp. 1-10, doi: 10.1109/AIDML70736.2026.11749289.

M. R. Marri, S. B. Kurada, S. Gupta, S. Dey, S. Kumar and R. Chauhan, "Graph-Based Deep Learning Model for Identifying Cyber Threats in Cloud Platforms," 2025 International Conference on Computational Intelligence, Security, and Artificial Intelligence (IntelliSecAI), Al-Khobar, Saudi Arabia, 2025, pp. 1-7, doi: 10.1109/IntelliSecAI66368.2025.11472831

V. Viswanathan, M. H. Mirza, D. S. Jatav, N. Mukhi, T. Gupta and S. B. Goyal, "Deep Reinforcement Learning Model to Enhance Accuracy of Forecasting in Supply chain Optimization," 2025 International Conference on Intelligent & Innovative Practices in Engineering & Management (IIPEM), Singapore, Singapore, 2025, pp. 1-6, doi: 10.1109/IIPEM65914.2025.11548310.

SinghJatav, D., Amin, M. M., Kodela, S., Nayan, V., Wannous, M., & Khalifa, G. S. (2025, November). Hybrid Reinforcement and Deep Learning Model for Payment Delay Optimization in Supply Chain Finance. In 2025 10th International Conference on Information Technology Trends (ITT) (pp. 170-175). IEEE.

M. Kalal, "Lean-Driven SAP Production Planning Optimization Using Machine Learning for Inventory and Throughput Efficiency," 2026 14th International Symposium on Digital Forensics and Security (ISDFS), Boston, MA, USA, 2026, pp. 1-6, doi: 10.1109/ISDFS69419.2026.11459029.

Kalal, M. (2024). Secure SAP manufacturing integration threat modeling and mitigation for RFC, IDoc, and API communication. International Journal of Communication Networks and Information Security, 16(4). https://doi.org/10.5281/zenodo.20088018

B. R. Rallabandi, D. Upadhyay, S. S. Mahadasu, D. S. Asudani, A. Jain and Y. K. Pandey, ""Investigation of Higher Order Sectorization Combined with Massive MIMO for Enhanced Spatial Diversity in 6G mmWave Environments,"" 2026 5th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0, Raigarh, India, 2026, pp. 1-6, doi: 10.1109/OTCON69332.2026.11629684.

S. S. Mahadasu, B. R. Rallabandi, N. Gudla, and N. Safavi, " An Energy-Efficient Localization Framework for Mobile Wireless Sensor Networks Using Neural Attention–Guided Stochastic Fractal Search," Internet Technology Letters 9, no. 6 (2026): e70361, https://doi.org/10.1002/itl2.70361.