Innovate 2025

Date: 22 March 2025

Team Num: T-9915

Team Name: Shwet

Title of Project / Idea	:	Zero-Knowledge Proofs for Privacy-Preserving Authentication.
Problem Theme	:	Cybersecurity
Institute Name	:	Thapar University
Team Members	:	Shwet Singh
Contact	:	ssingh21_be22@thapar.edu, +919915883998

Context & Problem

In today's digital world, authentication systems are crucial for securing access, but traditional methods like passwords and biometrics pose challenges by requiring users to share sensitive information with service providers. This creates several challenges as follows:

- Privacy Risks: Users' sensitive information stored on centralized servers is vulnerable to data breaches, insider threats, and misuse.
- Lack of User Control: Users have little control over how their data is stored, shared, or used by service providers.
- Trust Issues: Users must trust third-party service providers to handle their data securely, which is often not the case.

Problem Statement

The Need for Zero-Knowledge Proofs (ZKPs)

Zero-Knowledge Proofs (ZKPs) are a cryptographic technique that allows one party (the prover) to prove to another party (the verifier) that a statement is true without revealing any additional information. This makes ZKPs ideal for privacy-preserving authentication, as they enable users to prove their identity or credentials without disclosing sensitive data.

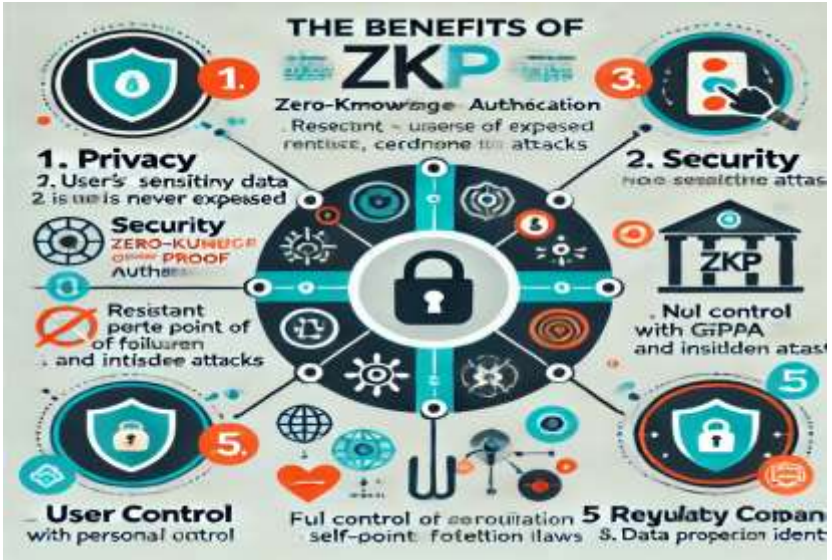
For example:

- A user can prove they know a password without revealing the password itself.
- A user can prove they are over 18 years old without revealing their exact date of birth.

ZKPs are particularly useful in decentralized systems, where trust is distributed, and users need to maintain control over their data.

Objectives

- Design a Privacy-Preserving Authentication System: Develop a system that allows users to prove their identity or credentials without revealing sensitive information.
- Enhance User Control: Empower users to control their own data and credentials, aligning with self-sovereign identity (SSI) principles.
- Achieve Regulatory Compliance: Ensure the system complies with privacy regulations like GDPR, HIPAA, and CCPA.
- Demonstrate Real-World Applicability: Showcase the system's use cases in industries like finance, healthcare, IoT, and government.



Proposed Solution

The solution involves designing and implementing a Zero-Knowledge Proof (ZKP)-based authentication system that allows users to prove their identity or credentials without revealing sensitive information.

- Zero-Knowledge Proof Protocol: Use zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) or zk-STARKs (Zero-Knowledge Scalable Transparent Arguments of Knowledge) to enable users to prove their identity or credentials without disclosing the underlying data.

- Decentralized Identity Management: Store user credentials and proofs on a blockchain or decentralized storage system. Use Decentralized Identifiers (DIDs) to uniquely identify users without relying on a central authority.

- User Authentication Flow:

1. Registration: Users register their credentials (e.g., password, biometric hash) with the system, which generates a cryptographic commitment and ZKP.

2. Authentication: Users submit the ZKP to prove their identity without revealing the actual credentials.

3. Verification: The system verifies the ZKP using public parameters and grants access if the proof is valid.

Proposed Solution

- Access Control:

Implement smart contracts or policy-based access control mechanisms to enforce permissions based on ZKP verification.

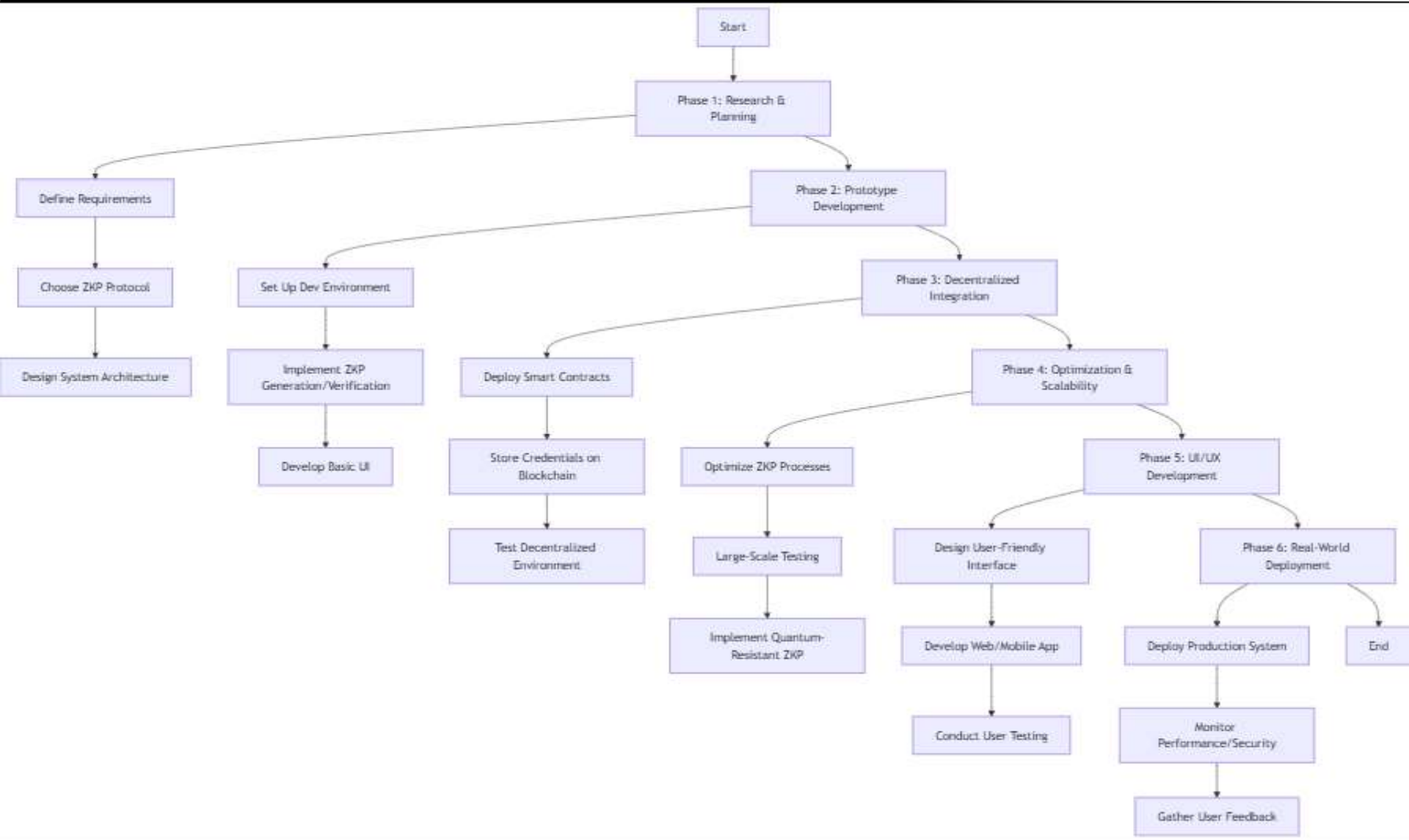
Example: A smart contract on Ethereum verifies the ZKP and grants access to a decentralized application (dApp).

- User Interface:

Develop a user-friendly interface (web or mobile app) for seamless registration, authentication, and access control.

Example: A web app integrated with MetaMask for blockchain interactions.

Methodology



Phase 1: Setup and Planning:

- Set up development environment (Node.js, Python, Circom, SnarkJS, Hardhat/Truffle).
- Define use case (e.g., proving knowledge of a password without revealing it).
- Design ZKP circuit using Circom.

Phase 2: Build the Core System:

- Implement ZKP logic (Circom for circuits, SnarkJS for proofs).
- Develop smart contract in Solidity for ZKP verification.
- Use IPFS for decentralized storage of credentials.

Phase 3: Build the User Interface:

- Develop frontend using React.js/Vue.js.
- Include:

Registration form (input credentials).

Authentication form (submit ZKPs).

Dashboard (show successful authentication).

Integrate MetaMask for blockchain interactions.

Phase 4: Backend and API Development:

- Use Node.js/Python for backend logic.
- Create REST APIs for:

User registration.

Proof generation and verification.

Access control.

Phase 5: Testing and Debugging:

- Test ZKP circuit, smart contract, and end-to-end flow.

Phase 6: Deployment:

- Host frontend (Vercel/Netlify) and backend (AWS/Google Cloud/Heroku).

Novelty

- ELIMINATES THE NEED TO SHARE SENSITIVE DATA USING ZKPS.
- EMPOWERS USERS WITH DECENTRALIZED IDENTITY MANAGEMENT.
- COMPLIES WITH PRIVACY REGULATIONS LIKE GDPR, HIPAA, AND CCPA.
- DEMONSTRATES REAL-WORLD APPLICABILITY ACROSS MULTIPLE INDUSTRIES.
- PROVIDES A USER-FRIENDLY INTERFACE FOR SEAMLESS ADOPTION.
- USES SMART CONTRACTS FOR TRANSPARENT AND SECURE ACCESS CONTROL.
- COMBINES CUTTING-EDGE TECHNOLOGIES INTO A SINGLE, INNOVATIVE SOLUTION.

Budget Breakdown (INR)

Category	Low Estimate (₹)	High Estimate (₹)
ZKP Protocol Implementation	8,00,000	16,00,000
Smart Contract Development	4,00,000	8,00,000
Backend Development	6,40,000	12,00,000
Frontend Development	5,60,000	9,60,000
Cloud Hosting (per month)	40,000	80,000
Blockchain Gas Fees	80,000	2,40,000
Decentralized Storage	40,000	80,000
Security Audits	6,40,000	13,60,000
Project Management	40,000	80,000
Documentation & Training	2,40,000	5,60,000
Total	**34,40,000**	**69,60,000**

Outcome

1. Functional Outcomes:

- Privacy-Preserving Authentication – Users verify credentials without revealing data.
- Decentralized Identity Management – Blockchain-based credential storage with user control.
- Secure Access Control – Smart contracts verify ZKPs for authentication.
- User-Friendly Interface – Web/Mobile app with wallet integration.
- Zero-Knowledge Proofs (ZKP) – zk-SNARKs/zk-STARKs for proof generation & verification

2. Real-World Applications:

- Finance: DeFi, banking.
- Healthcare: Patient authentication, telemedicine.
- IoT: Smart homes, connected cars.
- Government: Voting, tax filing, social benefits.
- Corporate: Employee authentication.
- Education: Online learning, exam proctoring.
- E-Commerce: Age verification for restricted products.
- Travel: Booking systems, loyalty programs.
- Gaming: Age-restricted content access.
- Social Media: Secure logins, age verification.

3. Long-Term Impact:

- Enhanced Privacy & Security – No sensitive data exposure.
- Future-Proof – Resistant to emerging cyber threats.
- Regulatory Compliance – GDPR, HIPAA, CCPA aligned.
- User Empowerment – Full data ownership.

4. Security Outcomes:

- Cyber Threat Protection – Shields against phishing, credential theft, and insider attacks.
- Quantum Resistance – zk-STARKs ensure future-proof security.
- Auditability – All authentication events logged on blockchain.